Usable Security and Privacy

Vesna Krnjic

Usable Security and Privacy

Dissertation

at

Graz University of Technology

submitted by

DI Vesna Krnjic BSc

Institute for Software Technology,
Graz University of Technology

December 2019

Supervisor/Reviewer: Univ.-Prof. Dipl-Ing. Dr. techn. Wolfgang Slany

Abstract

Today's society is transforming into an information society in which mobile devices with wireless Internet connections are available to everyone. Thereby, the number of threats such as identity theft or theft of sensitive data, Social Media attacks, or mobile malware increases. End-users often struggle with security systems that are too difficult to use and not designed to satisfy their needs. Security and privacy have become increasingly complex to handle. Therefore, it is crucial to discuss the usability factors in information security. This thesis is a contribution to security and privacy technology and makes it more usable by understanding the end-user through usable security studies and by providing new concepts and designs that meet the users' needs. Throughout this thesis, we focused on usable security challenges around user authentication, electronic documents, and privacy.

This thesis consists of five parts. In the first part, we provide background on usability, security, privacy, and usable security and privacy.

In the second part, we propose ALAP, an Agile Authentication Provider. ALAP provides authentication factors from different categories and levels of assurance (LoA). ALAP allows service providers to define their security requirements for the user-authentication process through policies by claiming a global LoA dynamically and flexibly. Based on this policy, ALAP dynamically assembles an authentication process for the respective service to meet its security requirements. At the same time, the user can choose the preferred authentication factor. This work primarily addresses the usability of the system. We present the results of an investigation about the usability of the system in multi-stage usable security evaluations. Furthermore, we outline an extended architecture, the Convenient Agile Authentication (CALA), whereby usability plays a central role while selecting the authentication factors.

In the third part, papers are provided without modification from their original publications. All papers have been peer-reviewed, accepted, and presented at international conferences. The papers focus on the usability evaluation of e-Government solutions. Security and usability are critical requirements of e-Government applications. Security requirements are often met by reliance on approved cryptographic methods such as qualified electronic signatures. These methods usually rely on the integration of cryptographic hardware tokens such as smart cards. The inclusion of these tokens into e-Government applications introduces additional complexity and often affects the usability of these solutions. Usability issues raised by the integration of cryptographic hardware tokens into e-Government applications have not been considered in detail so far. We filled this gap by conducting a multi-step usability evaluation of three core

components of the Austrian e-Government infrastructure. We identify existing usability issues and derive possible improvements.

In the fourth part, we provide papers unchanged to their original publications. The contributions were reviewed by experts, accepted and presented at international conferences. The papers discuss the importance of security aspects, like the integrity and authenticity of public sector data. We present a concept to assure the integrity and authenticity of the provided data based on electronic signatures. Furthermore, we show that the idea can also be extended to data that needs to be anonymized, to meet privacy requirements, before provisioning by incorporating redactable signatures.

The last part of the thesis provides three papers about Pocket Code without modification from their original publication. Furthermore, we discuss the Catrobat Project and the necessity to investigate usable security and privacy, specifically for the target group of children and adolescents. We present the first concepts dealing with Pocket Code and security and privacy issues. Many sensitive smartphone system resources such as the camera, the microphone, or GPS sensors may be misused to violate the user's privacy. Furthermore, private data like photos or contacts may also be covertly passed on to third parties. These systems have been classified as critical by Google and Apple. Users have to give permission before installing an application. Nonetheless, some smartphone functions like Internet access are still not categorized as critical. We present a concept of how to solve the problem with Internet access in Pocket Code by warning the user that the currently used Catrobat project has Internet access. Furthermore, the user can view the used URL and decide whether to trust it or not. Moreover, it should be possible to manage the trusted domains in a whitelist.

This thesis closes with some final remarks and concluding thoughts.

Kurzfassung

Die heutige Gesellschaft entwickelt sich zunehmend zu einer Informationsgesellschaft, in der Mobiltelefone mit drahtlosem Internetzugang für jeden verfügbar sind. Dabei treten immer mehr Bedrohungen auf, wie zum Beispiel ein Identitätsdiebstahl oder Diebstahl sensibler Daten, Social Media-Angriffe oder mobile Maleware. Anforderungen an Sicherheit und Datenschutz überfordern Endbenutzer. Daher ist es wichtig, die Usability-Faktoren der Informationssicherheit zu untersuchen. Diese Arbeit zielt auf eine Verbesserung von Datenschutz und Sicherheitstechnologien, indem sie den Endbenutzer durch Usable-Security-Studien verstehbar macht und daraus neue Konzepte und Ansätze liefert. Im Mittelpunkt stehen Usable-Security-Herausforderungen rund um Benutzerauthentifizierung, elektronische Dokumente und Datenschutz.

Diese Dissertation besteht aus fünf Teilen. Im ersten Teil präsentieren wir Hintergrundinformationen zu Benutzerfreundlichkeit, Sicherheit, Datenschutz und Usable-Security.

Im zweiten Teil stellen wir ALAP vor - einen Agile Authentication Provider. ALAP bietet Authentifizierungsfaktoren aus verschiedenen Kategorien und Level of Assurance (LoA) an. Das System ermöglicht Service Providern, ihre Sicherheitsanforderungen an den Benutzerauthentifizierungsprozess durch Richtlinien zu definieren, indem sie dynamisch und flexibel einen globalen LoA in Anspruch nehmen. Basierend auf diesen Richtlinien stellt ALAP dynamisch einen Authentifizierungsprozess für den jeweiligen Dienst zusammen, um dessen Sicherheitsanforderungen zu erfüllen. Gleichzeitig kann der Benutzer den bevorzugten Authentifizierungsfaktor wählen. Dieser Teil der Arbeit konzentriert sich auf die Benutzerfreundlichkeit von ALAP. Wir untersuchen die Usability des Systems in einer mehrstufigen, Usable-Security-Evaluierung. Darüber hinaus stellen wir eine erweiterte Architektur, die Convenient Agile Authentication (CALA),vor bei der die Usability eine zentrale Rolle in der Auswahl der Authentifizierungsfaktoren spielt.

Im dritten Teil finden sich Artikel unverändert gegenüber den Originalpublikationen. Alle Texte wurden von Experten begutachtet, akzeptiert und auf internationalen Konferenzen präsentiert. Sie beschäftigen sich mit der Usability-Evaluierung von E-Government-Lösungen. Sicherheit und Usability sind kritische Anforderungen an E-Government-Anwendungen. Die Sicherheitsanforderungen werden oft durch den Einsatz bewährter kryptographischer Verfahren wie qualifizierter elektronischer Signaturen erfüllt. Diese Methoden basieren in der Regel auf der Integration von kryptographischen Hardware-Token wie Smartcards. Die Aufnahme dieser Token in E-Government-Anwendungen bringt zusätzliche Komplexität mit sich und beeinträchtigt oft die Usability der Lösun-

gen. Usability-Probleme, die sich aus der Integration von kryptographischen Hardware-Token in E-Government-Anwendungen ergeben, wurden bisher nicht im Detail untersucht. Wir haben diese Lücke geschlossen, indem wir Usability-Studien zu Kernkomponenten der österreichischen E-Government-Infrastruktur durchgeführt haben. Wir identifizieren bestehende Usability-Probleme und leiten Verbesserungen ab.

Im vierten Teil werden die Artikel unverändert zu ihren Originalpublikationen zur Verfügung gestellt. Alle Artikel wurden von Experten begutachtet, akzeptiert und auf internationalen Konferenzen abgedruckt. In diesen Artikeln diskutieren wir die Bedeutung von Sicherheitsaspekten, wie die Integrität und Authentizität von Daten des öffentlichen Sektors. Sie stellen ein Konzept zur Verfügung, um die Integrität und Authentizität der bereitgestellten Daten auf der Grundlage elektronischer Signaturen zu gewährleisten. Darüber hinaus zeigen wir, dass diese Methode auch auf Daten ausgedehnt werden kann, die anonymisiert werden müssen, um den Anforderungen des Datenschutzes gerecht zu werden, bevor sie durch die Einbindung redaktionell bearbeitbarer Signaturen bereitgestellt werden.

Der letzte Teil dieser Dissertation enthält drei Artikel über Pocket Code, unverändert gegenüber der Originalveröffentlichung.

Darüber hinaus diskutieren wir das Catrobat-Projekt und die Notwendigkeit, Usable Security and Privacy speziell für die Zielgruppe Kinder und Jugendliche zu untersuchen. Wir stellen die ersten Konzepte vor, die sich mit Pocket Codes Datenschutz- und Sicherheitsfragen befassen. Viele sensible Ressourcen des Smartphone-Systems wie Kamera, Mikrofon oder GPS-Sensoren können missbraucht werden, um die Privatsphäre des Benutzers zu verletzen. Darüber hinaus können auch private Daten wie Fotos oder Kontakte unbemerkt an Dritte weitergegeben werden. Diese Systeme wurden von Google und Apple als kritisch eingestuft. Die Berechtigung muss von den Benutzern vor der Nutzung erteilt werden. Andere Smartphone-Funktionen wie der Internetzugang werden noch nicht als kritisch eingestuft. Wir stellen ein Konzept vor, wie man das Problem des Internetzugangs in Pocket Code lösen kann, indem man den Benutzer warnt, dass das aktuell verwendete Projekt auf das Internet zugreift. Des weiteren kann der Benutzer die verwendete URL einsehen und entscheiden, ob er dieser vertrauen möchte oder nicht. Ergänzend sollte es möglich sein, die vertrauenswürdigen Domänen in einer Whitelist zu verwalten.

Die Dissertation schließt mit zusammenfassenden Anmerkungen und Schlussfolgerungen.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Date	Signature

Acknowledgements

First and foremost, I would like to thank my advisor Wolfgang Slany, for giving me the opportunity to complete my doctoral thesis and continue my research in the field of usable security and privacy. Thank you for your valuable feedback during the work on this thesis. In particular, I want to thank the entire Catrobat team for the great support I received.

I also want to thank my colleagues from the Institute for Applied Information Processing and Communications; to me, you're a group of great minds. Thank you, Arne Tauber and Herbert Leitold, you gave me the freedom to research in a topic I'm interested in, although my work was not at the core of the institute's research. My special thanks go to Thomas Lenz for lots of fruitful discussions and his valuable feedback during the years we worked together. Particular appreciation goes to my colleagues and co-authors of my papers from E-Government Innovation Center, Thomas Zefferer, Klaus Stranacher, Bernd Zwattendorfer, Tobias Kellner, Andreas Fitzek and Christian Maierhofer; it was a great time and a lot of fun working with you.

Furthermore, I would like to thank all test-users, who spent their time on user interviews, focus groups, and labor tests. Their committed participation and feedback were essential and indispensable for the successful conduction of numerous usability tests.

Last but not least, I would like to thank my parents, my siblings, and the whole family for supporting me during the past years. My extraordinary thanks go to Florian and my kids Oliver and Lea, for helping me in everything I do. Thank you for your patience and understanding. Without your support, this would not have been possible.

Vesna Krnjic
Graz, Austria, December 2019

List of Publications

[1] Harzl, A., Krnjic, V., Schreiner, F., and Slany, W. Comparing purely visual with hybrid visual/textual manipulation of complex formula on smartphones. In *DMS 2013* (2013), .

[2] Harzl, A., Krnjic, V., Schreiner, F., and Slany, W. Purely visual and hybrid visual/textual formula composition: A usability study plan. In *Proceedings of Programming for Mobile and Touch PRoMoTo 2013* (2013), .

[3] Krnjic, V., Stranacher, K., Kellner, T., and Fitzek, A. Modular architecture for adaptable signature- creation tools requirements, architecture, implementation and usability. In *EGOV - IFIP e-Government Conference* (2013), .

[4] Krnjic, V., Weber, P., Zefferer, T., and Zwattendorfer, B. Effizientes testen von e-government komponenten in der cloud. In *Arbeitskonferenz DACH Security* (2013), ., pp. 225–236.

[5] Lenz, T., and Krnjic, V. Agile smart-device based multi-factor authentication for modern identity management systems. In *Proceedings of the 14th International Conference on Web Information Systems and Technologies* (2018), vol. 1, pp. 113–124.

[6] Lenz, T., and Krnjic, V. Towards domain-specific and privacy-preserving qualified eid in a user-centric identity model. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (United States, 9 2018), IEEE Computer Society, pp. 1157–1163.

[7] Spieler, B., Krnjic, V., and Slany, W. Girls create games: Lessons learned. *CoRR abs/1907.05811* (2019).

[8] Stranacher, K., Krnjic, V., and Zefferer, T. Vertrauenswürdiges open government data. In *1.OGD D-A-CH-LI Konferenz* (2012), ., pp. 27–39.

[9] Stranacher, K., Krnjic, V., and Zefferer, T. Authentische und integritätsgesicherte verwaltungsdaten. *eGovernment review 11* (2013), 30–31.

[10] Stranacher, K., Krnjic, V., and Zefferer, T. Trust and reliability for public sector data. In *Proceedings of International Conference on e-Business and e-Government* (2013), vol. 73, ., pp. 124–132.

[11] Stranacher, K., Krnjic, V., Zwattendorfer, B., and Zefferer, T. Assessment of redactable signature schemes for trusted and reliable public sector data. In *Proceedings of the 13th European Conference on e-Government* (2013), ACPI, pp. 508–516.

[12] Stranacher, K., Krnjic, V., Zwattendorfer, B., and Zefferer, T. Evaluation and assessment of editable signatures for trusted and reliable public sector data. *Electronic Journal of e-Government 11*, 2 (2013), 360–372.

[13] Stranacher, K., Zwattendorfer, B., and Krnjic, V. Secure and efficient processing of electronic documents in the cloud. In *Proceedings of IAIDIS International Conference e-Society* (2013), ., pp. 217–224.

[14] Zefferer, T., and Krnjic, V. Towards user-friendly e-government solutions: Usability evaluation of austrian smart-card integration techniques. In *Advancing Democracy, Government and Governance* (2012), Lecture Notes in Computer Science, Springer, pp. 88–102.

[15] Zefferer, T., and Krnjic, V. Usability evaluation of electronic signature based e-government solutions. In *Proceedings of the IADIS International Conference WWW/INTERNET 2012* (2012), ., pp. 227–234.

[16] Zefferer, T., and Krnjic, V. Usability-evaluierung der österreichischen handy-signatur. In *D-A-CH Security 2012* (2012), ., pp. 365–376.

[17] Zefferer, T., Krnjic, V., Stranacher, K., and Zwattendorfer, B. *Measuring Usability to Improve the Efficiency of Electronic Signature-based E-Government Solutions*. Springer New York, 2014, pp. 45–74.

[18] Zefferer, T., Krnjic, V., and Zwattendorfer, B. Ein virtuelles testframework für e-government komponenten. In *D-A-CH Security 2011* (2011), ., pp. 492–503.

Contents

List of Figures

Background

On the Usability of Agile Authentication

Modular Architecture for Adaptable Signature-Creation Tools

Towards User-Friendly e-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques

Usability-Evaluierung der österreichischen Handy-Signatur

Usability Evaluation of Electronic Signature Based e-Government Solutions

Measuring Usability to Improve the Efficiency of Electronic Signature-Based e-Government Solutions

Ein virtuelles Testframework for E-Government Komponenten

Effizientes Testen von E-Government Komponenten in der Cloud

Trust and Reliability for Public Sector Data

Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data

Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data

Vertrauenswürdiges Open Government Data - Authentizität und Integritäet für öeffentliche Verwaltungsdaten

Authentische und integritätsgesicherte Verwaltungsdaten

Secure and Efficient Processing of Electronic Documents in the Cloud

Purely Visual and Hybrid Visual/Textual Formula Composition: A Usability Study Plan

Comparing Purely Visual with Hybrid Visual/Textual Manipulation of Complex Formula on Smartphones

Girls Create Games: Lessons Learned

Outlook - Usable Security for the Catrobat Project

List of Tables

Background

Convenient Agile Authentication

Towards User-Friendly e-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques

Measuring Usability to Improve the Efficiency of Electronic Signature-Based e-Government Solutions

Effizientes Testen von E-Government Komponenten in der Cloud

Trust and Reliability for Public Sector Data

Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data

Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data

Authentische und integritätsgesicherte Verwaltungsdaten

Comparing Purely Visual with Hybrid Visual/Textual Manipulation of Complex Formula on Smartphones

Part I

Introduction to Usable Security and Privacy

1 Introduction

1.1 Motivation

During the last decade, IT systems have become a significant part of people's daily lives. Our society is transforming into an information society, where mobile phones with wireless Internet access are available to everyone. The dependency on computer technologies involves serious threats like identity and data theft. Consequently, there is a strong need for security technology to protect information systems and data. Security requirements are typically met by approved cryptographic methods such as qualified electronic signatures. End users expect applications that are both secure and user-friendly; the security mechanisms used must not interfere with the use of the system. Usability is another vital quality attribute for most IT systems and services. The goal of security developers should be to design systems where the easiest way is also the most secure one. However, users should not have to work more than necessary when using security systems.

1.2 Agile Authentication

User authentication, as a crucial step for most electronic services, usually relies on multi-factor authentication to enhance their authentication processes. The technologies used are often not flexible enough to keep pace with the rapid progress of new implementations of authentication factors or ever-changing security requirements. Increasing the security of the authentication process often comes at the expense of significantly reduced usability – consequently, the compromise between security and usability leads to unsatisfying trade-offs that are neither secure nor usable. A primary example are password policies, allowing simple passwords makes them easy to use and remember but creates security problems. Simultaneously, secure passwords are not easy to remember and to recall, but meet specific security requirements [23].

To improve this situation, we propose ALAP, an Agile Authentication Provider. ALAP provides authentication factors from different categories (knowledge, possession, and inheritance) and Levels of Assurance (LoA). The system allows service providers to define their security requirements regarding the user-

authentication process through policies by claiming a global LoA dynamically and flexibly. Based on this policy, ALAP dynamically assembles an authentication process for the respective service to meet its security requirements; at the same time, the user can choose the preferred authentication factor. During the development of ALAP, great attention was paid to usability. In this thesis, we particularly investigated the usability of the system in multi-stage usable security evaluations, suggest usability improvements, and propose CALA, which places the user even more in the center of the authentication process.

1.3 Usability of e-Government Systems

The importance of usability in e-Government has been subject to ongoing research for many years. However, most work has focused on the usability of e-Government websites so far [1], [37], [63]. For instance, a quality inspection method for the evaluation of e-Government sites has been proposed by Garcia et al. [22]. Without a doubt, the usability of e-Government websites is an important topic. However, the integration of security-enhancing technologies such as smart cards into Web-based e-Government applications needs to be considered as well. Otherwise, usability evaluations of current e-Government solutions threaten to remain incomplete and to miss relevant aspects.

According to the Austrian e-Government strategy, security-enhancing technologies are integrated into e-Government applications using different Citizen Card Software (CCS) implementations. We assess the usability of MOCCA Local[1], MOCCA Online, and the Austrian Mobile Phone Signature[2], to identify persisting usability problems and to analyze user preferences. The evaluation is intended to contribute to further improving the efficiency of Austrian e-Government. The conducted usability evaluations delivered more in-depth insight into the usability of core components of the Austrian e-Government from the citizens' point of view. By collecting user feedback via various interviews and questionnaires, we were able to identify persisting weaknesses and further room for improvement. Valuable findings have also been obtained from an analysis of recorded user sessions. All results are incorporated into future releases of the CCS implementations. Thus, the conducted usability studies contribute to the security and usability of MOCCA Local, MOCCA Online, and the Mobile Phone Signature and hence, to more efficient e-Government services. Moreover, we introduced a testing framework for e-Government applications, based on cloud computing and virtual machines.

[1] `https://joinup.ec.europa.eu/solution/mocca`
[2] `a-trust.at/en/handy-signatur/`

1.4 Integrity and Authenticity of Public Sector Data (Privacy)

During the past few years, various developments in the IT sector have been significantly influenced by the "Open Movement". For instance, Open Source[3] has become a well-known term that describes the philosophy of making source code publicly available to everybody. Additionally, related concepts such as Open Access or Open Content have continuously increased popularity during the past years. Recently, especially the idea of Open Data has attracted attention. The general idea behind Open Data is that data should be freely available for everyone to be used and republished. The public sector holds large amounts of data on various areas such as social affairs, economy, or tourism. Numerous initiatives such as Open Government Data or the EU Directive on public sector information aim to make these data available for public and private service providers. In general, the term public sector data denotes all kinds of electronic data being produced, collected, provided, or processed by the public sector. In this thesis, we focus on public sector data used in the context of the Open Government Data initiative, and covers data being under control of governmental institutions. Requirements for the provision of public sector data are defined by legal and organizational frameworks but hardly cover security aspects such as integrity or authenticity. Security aspects such as data integrity and authenticity are essential factors that should also be considered by public sector data solutions. The use of forged data might lead to resource claims. In such cases, the supplier of data should be able to prove that initially provided data has been altered. Solutions based on public sector data usually do not support this feature. We discuss the importance of integrity and authenticity of public sector data and present a concept to assure the integrity and authenticity of provided data based on electronic signatures. We show that the given idea is suitable for the provisioning of unaltered data. Furthermore, we show that the concept can also be extended to data that needs to be anonymized before provisioning by incorporating redactable signatures to meet privacy requirements. The modification of data would break any electronic signature on these data. To overcome this issue, we extend our approach and replace the concept of electronic signatures by redactable signatures. Redactable signatures are a special kind of electronic signatures that allow for a limited modification of signed data without breaking the applied signature.

1.5 Usable Security for the Catrobat Project

Catrobat[4] is a nonprofit Free Open Source Software (FOSS) project that was initiated in 2010 in Austria at Graz University of Technology. A multidisciplinary team develops free coding apps for teenagers and programming novices

[3] `https://opensource.org/`
[4] `https://www.catrobat.org/`

intending to introduce them to the world of programming. Pocket Code, one of the apps developed by the Catrobat team, is a mobile visual coding environment designed for smartphones. This app allows children and adolescents to create games and programs directly on their smartphones in their language and thereby teaches them fundamental programming skills. Pocket Code is an integrated development environment (IDE) running on Android[5] and iOS[6]. In Pocket Code, users have the opportunity to share projects on the Catrobat community platform and interact with each other. Pocket Code can access all sensors supported by the smartphone and will soon provide a Web-Brick capable of accessing the Internet. Therefore, in-app security mechanisms to protect user data have to be implemented. Access to the Internet opens up some great opportunities for users but also brings new security threats such as disclosure of precise geographical location or exposure of any text a user has entered in the project.

Many sensitive smartphone system resources such as the camera, the microphone or GPS sensors may be misused to violate the user's privacy. Furthermore, private data like photos or contacts may also be covertly passed on to third parties. These systems have been classified as critical by Google and Apple. Before an app can access such a functionality of the smartphone, the user has to grant permission. Nonetheless, Internet access is still not categorized as critical permissions.

It is difficult to communicate the security and privacy aspects of a system or software to an adult, but it is even more difficult to explain this to younger children and adolescents. In this thesis we present the first steps in the direction of how to solve the problem with Internet access in the Pocket Code app by warning the user that the currently used project has Internet access. Furthermore, the user can view the used URL and decide whether to trust it or not. The user still has the possibility to trust a whole domain and add it to a whitelist.

1.6 Thesis Outline

This thesis is structured as follows.

- Part I – Introduction to Usable Security and Privacy
- Part II – Convenient Agile Authentication
- Part III – Publications – Usable Security Evaluations
- Part IV – Publications – Application of Electronic Signatures – Privacy
- Part V – Publications and Outlook – The Catrobat Project

[5] `https://play.google.com/store/apps/details?id=org.catrobat.catroid&hl=en`
[6] `https://apps.apple.com/at/app/pocket-code/id1117935892`

Part I - Introduction to Usable Security and Privacy – introduces the thesis and provides background information on usability, security and privacy, and how those research areas fit together. In detail, this part consists of the following chapters:

- Chapter 1 – Introduction
- Chapter 2 – Background

In Chapter 1 we give a brief introduction to usable security and the area usable security and privacy cover.

Part II – Convenient Agile Authentication - covers the design and architecture of an Agile Authentication Provider (ALAP) and describes usable security evaluations of the system.

- Chapter 3 – Convenient Agile Authentication

It is crucial to take into account usable security aspects even in the planning and development phase of security-related systems. Having that in mind, in Chapter 3 we presented multi-step usable security studies of ALAP in an early stage of development, with the aim to react to user feedback during ongoing architecture design. Our usability evaluations aim to find out people's usage approach, preferences, considerations, and concerns related to dynamic authentication systems like ALAP. Specifically, the objectives of this study are to answer the following questions:

- Is the workflow that needs to be followed when using ALAP clear to users?
- How well do users, with different domain knowledge, understand the underlying concepts, such as MFA, authentication factors, IDP?
- Is the information provided by the system to the users sufficient and clear for people with different knowledge background?

In Part III – Publications on usable security evaluations – papers without modification are provided from their original publications, therefore some of the papers are in German language. Thereby, this part comprises the following chapters:

- Chapter 4 – Modular Architecture for Adaptable Signature – Creation Tools Requirements - Architecture, Implementation and Usability
- Chapter 5 – Towards User-Friendly e-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques
- Chapter 6 – Usability - Evaluierung der öesterreichischen Handy-Signatur (German)
- Chapter 7 – Usability Evaluation of Electronic Signature-Based E-Government Solutions

- Chapter 8 – Measuring Usability to Improve the Efficiency of Electronic Signature-Based e-Government Solutions

- Chapter 9 – Ein virtuelles Testframework für E- Government Komponenten (German)

- Chapter 10 – Effizientes Testen von E-Government Komponenten in der Cloud (German)

Electronic documents are a crucial element for electronic communication like exchange or process of information or data. To assure the authenticity and integrity of electronic documents, electronic signatures are used. In particular, electronic signatures fulfilling certain security requirements are legally equivalent to handwritten signatures. Nevertheless, existing signature-creation tools have crucial drawbacks concerning usability and applicability. To solve these problems, we define appropriate requirements for signature-creation tools to be used in e-Government processes in Chapter 4. Taking care of usability and applicability we propose a modular architecture for adaptable signature-creation tools.

Chapter 5 presents a usability evaluation of MOCCA Local and MOCCA Online, the open source components which facilitate the integration of smart cards into national eGovernment applications.

The paper presented in Chapter 6 describes results from a usability evaluation of the Austrian mobile phone signature. The required degree of security in e-Government transactional processes could be met by using smart cards. This technique shows a series of disadvantages regarding usability. Particularly the necessity of a smart card reader turned out to be a problem in the past. To meet the usability requirements, the mobile phone signature was developed in Austria. The mobile phone signature allows Austrian citizens a secure authentication on e-Government applications and the creation of an electronic signature.

In Chapter 7 and Chapter 8, we present the results and lessons learned from a usability evaluation of a critical concept of the Austrian e-Government infrastructure, the Citizen Card. Security, usability, and efficiency have been defined as crucial requirements of e-Government solutions. It has been shown that there is a close correlation between efficiency and usability, while at the same time security and usability requirements are often contradictory.

Chapter 9 and Chapter 10 introduce a testing framework for e-Government applications, based on virtual machines. In the paper, we presented the advantages and disadvantages of the developed virtual testing framework and show that the framework can significantly facilitate the testing of e-Government components and thus contributes to their quality and security. In Chapter 10, we describe an advanced version of the test framework described in Chapter 9. The presented framework combines the advantages of established virtual test frameworks with the concepts of cloud computing.

Part IV deals with – Privacy – Publications on Redactable Signature Schemes for Public Sector Data.

- Chapter 11 – Trust and Reliability for Public Sector Data
- Chapter 12 – Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data
- Chapter 13 – Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data
- Chapter 14 – Authentizität und Integrität für öffentliche Verwaltungsdaten (German)
- Chapter 15 – Authentische und integritätsgesicherte Verwaltungsdaten (German)
- Chapter 16 – Secure and Efficient Processing of Electronic Documents in the Cloud

In Chapter 11, we discuss the importance of security aspects such as integrity or authenticity of public sector data and present a concept to assure the integrity and authenticity of provided data based on electronic signatures. Moreover, we show that our concept can also be extended to data that needs to be anonymized before provisioning by using redactable signatures. Chapter 12 identifies and discusses legal, organizational, and technical requirements that need to be met by redactable signature schemes when applied to public sector data to be published. Furthermore, different existing redactable signature schemes are examined and discussed in more detail. Based on the previously identified requirements, in Chapter 13, different editable signature schemes are discussed in detail. The conducted assessment reveals that blank digital signatures, which are a novel approach representing a subset of editable signature schemes, are uniquely suited to meet the predefined requirements. Chapter 14, a paper, and Chapter 15, published in the eGovernment Review journal, summarize the results from this section.

To ensure genuineness, usually, electronic signatures are applied to the documents. The validity of an electronic signature can be unambiguously determined by the receiver of a signed document using signature verification services. Besides, the automatic processing of electronic documents is essential for a cost-reducing, time-saving, and efficient public administration. The basis for automated processing is the availability of machine-readable data, i.e., structured electronic documents and appropriate metadata. Nevertheless, additional costs and time delays may arise if electronic documents or metadata are recognized as incomplete or wrong. Here, the need for prior data validation arises. Current solutions for the verification of electronic signatures usually support a subset of existing signature formats only. In Chapter 16, we introduce an approach for secure and efficient processing of electronic documents, mainly focusing on signature and metadata verification.

Part V – Publications and Outlook – The Catrobat Project

- Chapter 17 – Purely Visual and Hybrid Visual/Textual Formula Composition: A Usability Study Plan

- Chapter 18 – Comparing Purely Visual with Hybrid Visual/Textual Manipulation of Complex Formula on Smartphones
- Chapter 19 – Girls Create Games: Lessons Learned
- Chapter 20 – Usable Security for the Catrobat Project

Chapter 17 and 18 discuss different approaches for creating formulas with end-user programming languages, namely purely textual, purely visual, and hybrid strategies. In our paper, we introduce Pocket Code, an approach with visual programming and hybrid formula editing, which combines the easiness of visual programming with the effectiveness and clarity of textual formula displaying. Additionally, we present a proposal for an evaluation of the different approaches to formula manipulation in visual programming languages for smartphones.

Chapter 19 describes a girl-only intervention where girls were asked to create their games with the visual coding app Pocket Code. This "Girls' Coding Week" was designed as an off-school event and took place during summer 2018 with 13 girls between 11 to 14 years old. We collected qualitative and quantitative data through open interviews, as well as surveys that refer to motivational aspects. The findings show that gaming elements female teenagers tend to like, create, and play mostly follow stereotypical expectations.

Chapter 20 presents an overview of the Catrobat project and gives some insights toward the first steps we did to apply usable security and privacy to the the planed Web-Brick. With the introduction of Pocket Code's Web-Brick a Catrobat project will be able to access the Internet. Access to the Internet opens up some great opportunities for users but also brings new security threats such as disclosure of precise geographical location or exposure of any text a user has entered in the executed Catrobat project.

- Chapter 21 – Conclusion and Outlook

Finally, Chapter 21 closes with some final remarks and concluding thoughts.

2 | Background

In this chapter we provide background information on usability, security and privacy, and discuss the relationship between these research areas.

2.1 Usability

The ISO 9241-11 [61] defines usability as the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." Another definition of usability from Nielsen [43] is "a quality attribute that assesses how easy user interfaces are to use." Nielsen defines the following five quality components for usability:

Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?

Efficiency: Once users have learned the design, how quickly can they perform tasks?

Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?

Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

Satisfaction: How pleasant is it to use the design?

2.2 Usability Evaluation

Usability evaluation can be divided into two main categories, the expert evaluation such as heuristic inspection or cognitive walkthrough, and user studies such as interviews or user testing. Usability evaluations conducted by experts are usually cost-effective and identify more than 60 percent of usability problems. Nielsen and Molich [47] define Heuristic Evaluation as "a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge

its compliance with recognized usability principles (the "heuristics")." A small number of experts is needed to carry out a heuristic evaluation. They examine the interface and judge its compliance with pre-defined usability heuristics. After each evaluator has finished the inspection, the findings are combined. The heuristic evaluation can be conducted in an early stage of development; it is cheap and can be done without the involvement of the end-users.

In the year 1994, Wharton et al. [66] proposed cognitive walkthrough, a usability inspection method focussing on the explorative learnability of user interfaces. Although several variants of this method have evolved since then, the basic principle was preserved: the cognitive walkthrough focuses on the cognitive activities of users, especially on their goals and knowledge when performing a specific task. [38]

The following list briefly describes the 20 most common usability methods. Source: `https://www.nngroup.com/articles/which-ux-research-methods/` [55] Rohrer lists the following user research methods as the most common:

Usability-Lab Studies: Participants are brought into a lab, one-on-one with a researcher, and given a set of scenarios that lead to tasks and usage of specific interest within a product or service.

Ethnographic Field Studies: Researchers meet with and study participants in their natural environment, where they would most likely encounter the product or service in question.

Participatory Design: Participants are given design elements or creative materials in order to construct their ideal experience in a concrete way that expresses what matters to them most and why.

Focus Groups: Groups of 3 - 12 participants are lead through a discussion about a set of topics, giving verbal and written feedback through discussion and exercises.

Interviews: A researcher meets with participants one-on-one to discuss in depth what the participant thinks about the topic in question.

Eye-tracking: An eye-tracking device is configured to precisely measure where participants look as they perform tasks or interact naturally with websites, applications, physical products, or environments.

Usability Benchmarking: Tightly scripted usability studies are performed with several participants, using precise and predetermined measures of performance.

Moderated Remote Usability Studies: Usability studies conducted remotely with the use of tools such as screen-sharing software and remote control capabilities.

Unmoderated Remote Panel Studies: A panel of trained participants who have video recording and data collection software installed on their own

personal devices uses a website or product while thinking aloud, having their experience recorded for immediate playback and analysis by the researcher or company.

Concept Testing: A researcher shares an approximation of a product or service that captures the key essence (the value proposition) of a new concept or product in order to determine if it meets the needs of the target audience; it can be done one-on-one or with larger numbers of participants, and either in person or online.

Diary/Camera Studies: Participants are given a mechanism (diary or camera) to record and describe aspects of their lives that are relevant to a product or service, or simply core to the target audience; diary studies are typically longitudinal and can only be done for data that is easily recorded by participants.

Customer Feedback: Open-ended and/or close-ended information provided by a self-selected sample of users, often through a feedback link, button, form, or email.

Desirability Studies: Participants are offered different visual-design alternatives and are expected to associate each alternative with a set of attributes selected from a closed list; these studies can be both qualitative and quantitative.

Card Sorting: A quantitative or qualitative method that asks users to organize items into groups and assign categories to each group. This method helps to create or refine the information architecture of a site by exposing users' mental models.

Clickstream Analysis: Analyzing the record of screens or pages that users click on and see, as they use a site or software product; it requires the site to be instrumented properly or the application to have telemetry data collection enabled.

A/B Testing: A method of scientifically testing different designs on a site by randomly assigning groups of users to interact with each of the different designs and measuring the effect of these assignments on user behavior.

Unmoderated UX Studies: A quantitative or qualitative and automated method that uses a specialized research tool to capture participant behaviors (through software installed on participant computers/browsers) and attitudes (through embedded survey questions), usually by giving participants goals or scenarios to accomplish with a site or prototype.

True-Intent Studies: A method asking random site visitors what their goal or intention is upon entering the site, measuring their subsequent behavior, and asking whether they were successful in achieving their goal upon exiting the site.

Intercept Surveys: A survey that is triggered during the use of a site or application.

Email Surveys: A survey in which participants are recruited from an email message.

2.2.1 Tasks, User Groups and Mental Models

We can evaluate the usability of a product only if we know our users and their wishes and expectations from the product. For that reason, it is crucial to recruit representative test users and define typical tasks. The same design may be perfect for one user group and utterly inappropriate for another. Testing with wrong users results in false outcomes. That is why we need to specify the target audience and their goals precisely. We should ensure that we are not creating the wrong product for the specific user group. In the following section, we present two methods suitable for user and task research.

Personas

A persona, first introduced 1998 by Alan Cooper [14], is a representation of a user that includes the user's characteristics, experience, and goals. Personas characterize the target group of an application, giving a clear understanding of the system. They are based on the knowledge of real users collected through user research like interviews or observations and provide the project stakeholders insights into who the users are. Information collected from many users is combined to create a single persona. Each persona represents one particular user group. Generally, personas include the following key information [49][53]:

- Name
- Job title and job description
- Demographics like age, education and family status
- Goals and tasks they are trying to complete using the product
- Physical, social and technological environment
- Picture representing that user group
- A quote that summarizes what matters most to the person as it relates to the product

Figures 2.1, 2.2 and 2.3 show persona templates.

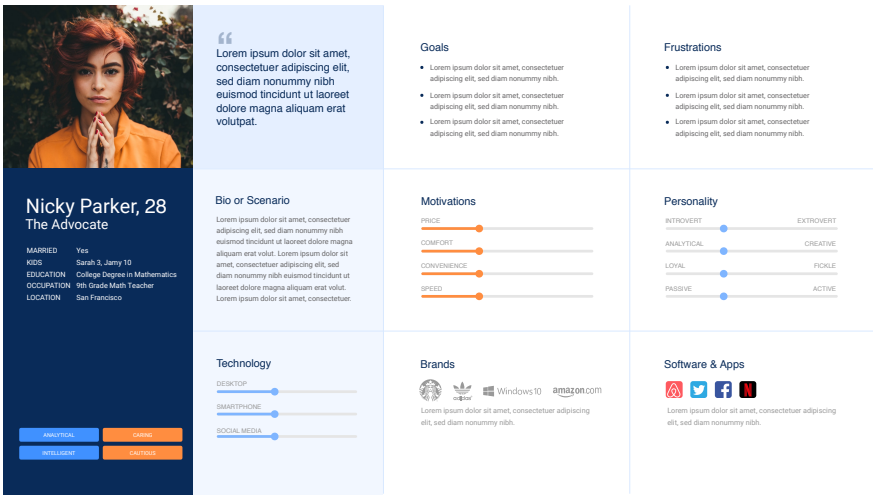


Figure 2.1: Typical persona template.
Source: https://pietalberts.com/download-persona-templates/

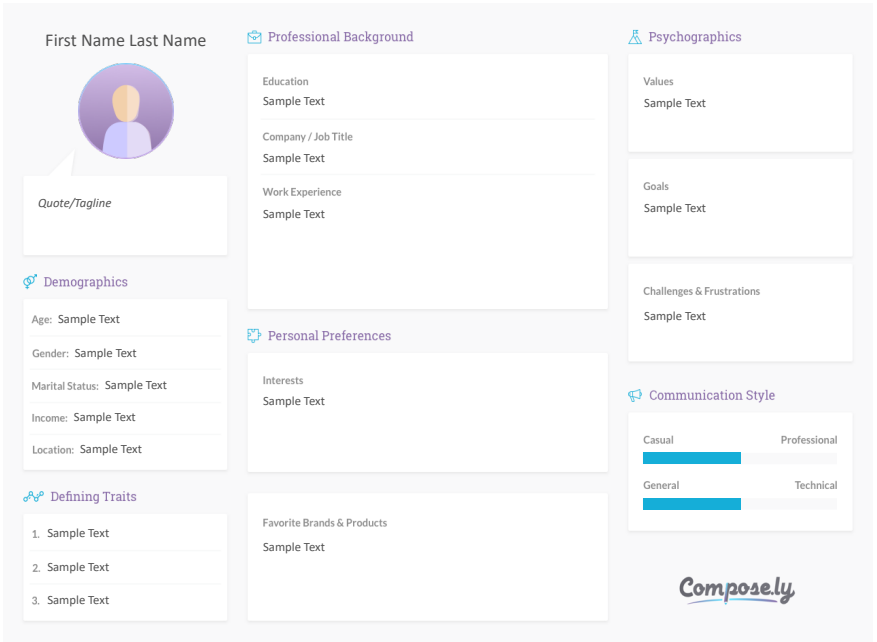


Figure 2.2: Personas. Source: https://compose.ly/strategy/user-persona-guide/

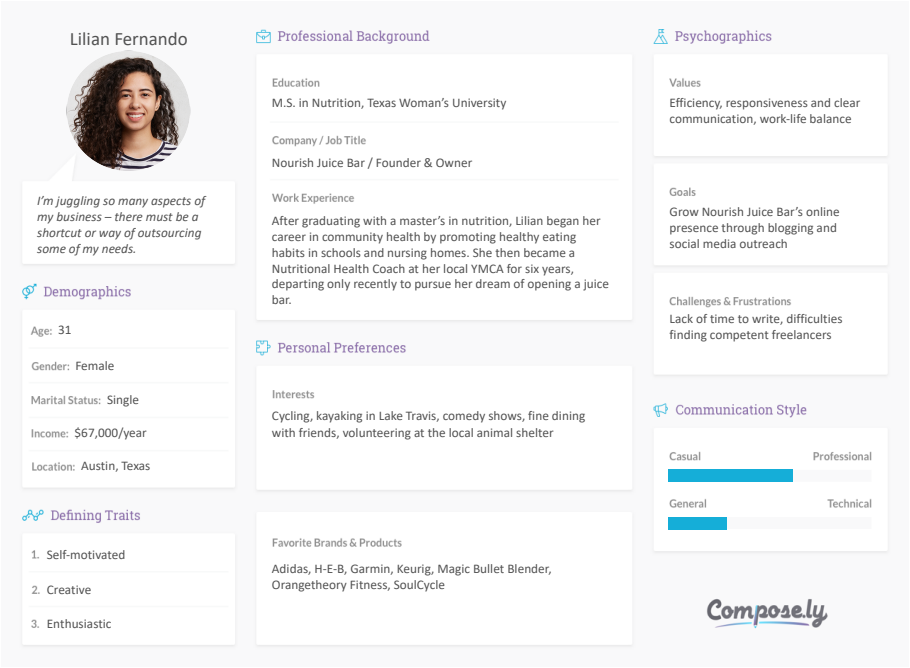


Figure 2.3: Persona with Data.
Source: https://compose.ly/strategy/user-persona-guide/

Personas are usually created at the beginning of a development process. With representative user groups, we can gather requirements and perform tasks around these groups.

Jobs-To-Be-Done

In the last few years a new technique has become famous in the user-centered design process: Jobs-To-Be-Done (JTBD) - a method focusing on customer needs. JTBD is defined by Laubheimer [35] as

> "a framework based on the idea that whenever users "hire" (i.e., use) a product, they do it for a specific "job" (i.e., to achieve a particular outcome). The set of "jobs" for the product amounts to a comprehensive list of user needs."

Christensen et all. [12] define the JTBD concept as follows:

> "Most companies segment their markets by customer demographics or product characteristics and differentiate their offerings by adding features and functions. But the consumer has a different view of the marketplace. He simply has a job to be done and is seeking to 'hire' the best product or service to do it."

The framework is a representation of user needs, identifying for which goals customers use or “hire” the product. Like already well-known methods such as task analysis or use cases, the JTBD framework focuses on the context, goals, and steps that arise during an interaction with a product. Still, it has much less prescriptive about what exactly the users’ task is and how they accomplish it.

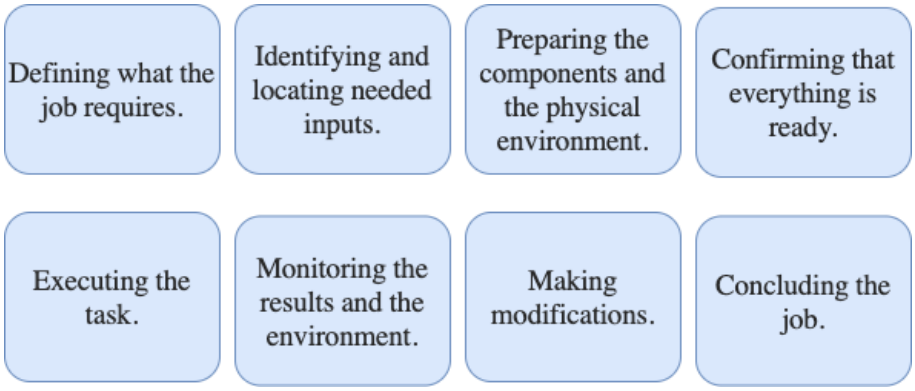


Figure 2.4: Universal structure of JTBD.

As described by Bettencourt et al. [34], all jobs have a universal structure (see Figure 2.4).

Mental Model

A mental model is a model regarding what users know about a system; it is about beliefs, not facts of a user. [44] System designers should design systems according to the shared mental model of end-users, instead of their mental model. Each user has a mental model because it is inside the user’s brain. Mental models could also change over time because they are only embedded in the brain of a user. Additional interaction with the system can change the model, or the users can update their models based on stimuli. The mental model is defined by Carola et all. as [11]

> "knowledge of the components of a system, their interconnection, and the process that change the components; knowledge that forms the basis for user being able to construct reasonable actions and explanations about what a set of actions is appropriate."

2.3 Security

IT security is the protection of computer systems from the damage of hardware, software, or data. Cryptography is the science to protect information. Mathematical concepts can be used to construct digital signatures or authentication protocols. Authentication methods are used to assure authenticity (assurance of

the identity) and integrity (nobody changes the data during the transmission). Two main authentication methods – electronic signatures and challenge-response authentication – exist. Whereas latter methods are mainly used in (low level) protocols, electronic signatures are used in various e-Business applications. Primarily the e-Government sector uses electronic signatures as a core technology enabling trusted services.

2.3.1 Authentication

The US American National Institute of Standards and Technology (NIST) defines authentication as "the process of establishing confidence in the identity of users" [48]. User authentication is crucial for any access control, for example, to a website, service, data, or information. Only if users are authenticated reliably, services can grant access to critical data and functionality to authorized users. One of the first and still most common forms of authentication is the simple provision of knowledge like username and password [9].

Password-based authentication is considered insecure because it is vulnerable to a large number of attacks such as brute-force attacks, malware attacks, dictionary attacks, or phishing. The latest work on password security shows that users and their passwords are still considered the weakest link. Although we know what safe passwords should be, we usually ignore this knowledge and use easy-to-remember passwords, because the fear of forgetting is stronger than the fear of being hacked.

Multi-Factor Authentication

Services with higher security requirements typically rely on multi-factor authentication (MFA) to strengthen their authentication processes by combining multiple authentications. The most popular version of MFA is Two Factor Authentication (2FA), where two different authentication factors from different categories are used. In general, four distinct authentication factors can be distinguished:

- Something the user knows (secrets such as password or PIN).
- Something the user is (biometric factors such as fingerprint or iris recognition).
- Something the user has (devices such as token or smart card).
- User location (GPS location, IP address).

Currently, MFA is mandatory for state-of-the-art implementation of sensitive Internet services like electronic banking or transactional e-Government services. However, Petsas et al. [52] analyzed more than 100,000 Google accounts and found that only about 6.4 percent of users had 2FA activated. According to twofactorauth.org [1], MFA is spreading more and more, a growing

[1] https://twofactorauth.org/

number of services support MFA. In the year 2016, SMS TAN, Software Token and Hardware Token, were the three most used implementations for the second authentication factor. To maintain a sufficient level of security, implemented authentication processes need to keep pace with the technological development and must react immediately to challenging threat scenarios and upcoming attack strategies such as attack threads for mobile phone-based 2FA as described by Konoth et al. [32] and Dmitrienko et al. [17]. This can be challenging in practice, as many technologies are often not flexible enough to keep pace with changing requirements.

Electronic Signatures

Electronic signatures assure the authenticity and integrity of digital data. Electronic signatures rely on public-key cryptography and represent the electronic equivalent to hand-written signatures. By applying a cryptographic method incorporating a private key to a set of data, the data is unambiguously linked to (i.e., signed by) the holder of the private key. The electronic signature can be verified using the corresponding public key. The verification process can only succeed, if the correct public key is used, and if the signed data is unaltered. Each modification of the signed data immediately breaks the electronic signature. This way, illegitimate alterations of signed data can be detected easily.

2.3.2 Mobile Security and Privacy

Personal Computers have been dominating the user market for many years. With the introduction of Apple iPhone with its touchscreen-based user interface, smartphones became more and more popular. Soon after the introduction of the iPhone, similar concepts were introduced. e.g., Android by Google in the year 2008. Nowadays, users are typically online and available 24/7 and also expect information and services to be available all the time. One-third of the Austrian population could no longer imagine everyday life without a smartphone. Currently, 77 percent of Austrian citizens are using smartphones. Among the under 30 year olds, the usage of smartphones amounts yet to 96 percent. Even the usage of smartphones depends on the educational achievements, the higher the educational level, the higher is the usage. This rate is over 85 percent by people with secondary school or a university degree [58]. In contrast, in 2008, only 10 percent of the US population was using smartphones in 2012, the smartphone penetration stood at 48 percent, in 2019 that grew to 77 percent, with even higher penetration in age groups under 34 years. The same trend has been seen worldwide, as in South Korea, for example, 94 percent of adults own a smartphone, the other 6 percent own mobile phones without smartphone functionality [16], [45]. Therefore, every adult in South Korea owns a mobile phone. Most of the time, users are browsing the Web, checking emails, using apps like YouTube or WhatsApp. With the extensive usage of smartphones and their ability to access the sensors of the phone and the personal information stored on it, an additional security and privacy challenge arises.

12 Most Abused Android App Permissions

		What it is for	How it can be abused	Apps that need this permission
1.	Network-based Location	It allows apps to retrieve an approximate location through network-based location sources like cell sites and Wi-Fi. App developers can use it to gain profit from location-based ads.	Malicious apps use it to launch location-based attacks or malware. For example, cybercriminals can direct Russia-based mobile users to malicious Russian language sites.	location apps, check-in apps
2.	GPS Location	It grants apps access to your exact location through the Global Positioning System (GPS) and other location sources like cell sites and Wi-Fi. Like network-based location, the GPS location can also be used by app developers to gain profit from location-based ads.	Malicious apps use it to load location-based attacks or malware.	location apps, check-in apps, social media apps

3.	View Network State	It allows apps to check for cellular network connections, including Wi-Fi. Apps require network connectivity to download updates or connect to a server or site.	Malicious apps use it to spot available network connections so they can perform other routines, like downloading other malware or sending text messages. Malicious apps can switch on these connections without your knowledge, draining your battery and adding to data charges.	location apps, check-in apps, social media apps
4.	View Wi-Fi State	It gives apps the right to access Wi-Fi network information, such as the list of configured networks and the current active Wi-Fi network.	Cybercriminals take advantage of device bugs to steal Wi-Fi passwords and hack into the networks you use.	browser apps, communication apps
5.	Retrieve Running Apps	It lets apps identify currently or recently running tasks and the processes running for each one.	Cybercriminals use this to steal information from other running apps. They can also check for and "kill" security apps.	task killer apps, battery monitoring apps, security apps
6.	Full Internet Access	This allows apps to connect to the Internet.	Malicious apps use the Internet to communicate with their command centers or download updates and additional malware.	browser apps, gaming apps, communication apps, productivity apps

7.	Read Phone State and Identity	It lets apps know if you are taking calls or are connected to a network. It also gives them access to information such as your phone number, International Mobile Equipment Identity (IMEI) number, and other identifying information. Apps often use this to identify users without requiring more sensitive information.	Information-stealing malicious apps often target device and phone information.	mobile payment apps, gaming apps, audio and video apps
8.	Start at Boot Automatically	Apps use this to tell the OS to run the application every time you start your device.	Malicious apps use this to automatically run at every boot.	task killer apps, battery monitoring apps, security apps
9.	Control Vibrator	This gives apps access to your device's vibrator function.	Malicious apps use it to stop vibrations, which would alert you of premium service notifications or verification text messages before the malicious app can intercept them.	communication apps, gaming apps
10.	Prevent From Sleeping	It keeps the processor from sleeping or the screen from dimming.	Malicious apps use this to prevent phones from going into sleep mode, so they can continuously run malicious routines in the background. This can also lead to battery drainage.	audio and video apps, gaming apps, browser apps

11.	Modify or Delete SD Card Contents	This lets apps write on external storage, like SD cards.	Cybercriminals use this to store copies of stolen information or save files onto your SD card before sending them to a command center. Malicious apps can also delete photos and other personal files on your SD card.	camera apps, audio and video apps, document apps
12.	Send SMS Messages	This allows apps to send text messages.	Premium service abusers use this to send messages to premium numbers. This leaves you with unexpected charges. Cybercriminals can also use it to communicate to command centers.	communication apps, social media apps

Table 2.1: 12 Most Abused Android App Permissions [40].

Table Source: http://about-threats.trendmicro.com/us/library/image-gallery/12-most-abused-android-app-permissions

2.4 Usable Security and Privacy

In the development of security-relevant applications, the primary design goal is the resistance against malicious attacks. However, the level of security that can be achieved is not only determined by the technical implementation. Of course, the algorithms applied, the technical components used, and the quality of the software implementation is highly crucial for security. Still, a high level of security can only be reached if a holistic view of security is applied. For instance, increasing the security of authentication processes, by adding more authentication factors, usually decreases usability. User-friendliness is often of secondary importance when it comes to security, if it has been considered at all. Security and data protection systems are often treated abstractly without paying special attention to the design of the user interface. As a result, security and data protection mechanisms are often not used as intended. Some mistakes made

by the user, using security and data protection systems cannot be easily reversed. For example, if a private data leak is detected, the user cannot be sure that the loss has not already been abused [67]. Therefore, usability problems caused by the inappropriate design of a system can lead to serious data protection or security issues. Very often security experts have to make a system secure after implementation, the same applies to usability experts. This "adding-on" behavior causes serious conflicts between security and usability [29].

In the past, security was only solved by focusing on technology, ignoring the human factor. Recently, the research area of usable security and privacy has evolved to take a different stance.

Usable security describes the interdisciplinary approach of designing security-enhancing techniques for digital products and services in such a way that users are optimally supported in their security-relevant goals and projects. This also enables non-technical users to fundamentally understand security elements and their necessity and to use the systems in the way initially intended. Usable privacy follows similar goals, focusing on technologies to promote privacy in digital systems and platforms.

Taking into account this holistic view of security, usable security evaluation is seen as an essential activity throughout the development process. Usable security evaluation looks at the users' interaction with the application and investigates how easy users can complete their tasks and achieve their goals by utilizing the application under evaluation. This is done to reveal flaws and aspects of the application that potentially lead to usability issues. In addition to looking at usability issues, usable security evaluation also investigates how easy users can unintentionally compromise the security of the system. The field of usable security and privacy uses many methods from the field of human-computer interaction; they are usually adapted to the given circumstances [69],[15],[42].

Security and usability cannot be considered separately as these concepts are closely linked. On the one hand the improvement of usability can impair security, on the other hand the increase of security can lead to serious usability problems. Usable Security Evaluation helps to apply an integrated approach to security application development that aims to optimize security without neglecting usability.

Although usable security and privacy is a young field of research, the importance of the consideration of security in connection with usability and the resulting challenges were already observed years ago. In the paper "The Protection of Information in Computer Systems", Saltzer et al.[57] identified already in 1975 "psychological acceptability" as a design principle stating that computer systems must be usable to be secure. The authors stated:

> "It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he

will make errors."

As described by [3] and [23] security is only a secondary concern for the end user. In order to achieve the primary goal, the security is often avoided or ignored. For example, end users use insecure passwords when requirements become too demanding and ignore security icons and warnings when they are motivated to achieve their goal [2].

In 1999, the authors of the paper "Why Johnny Can't Encrypt" Whitten et al. [67] triggered a broad rethink which has helped to establish a new field of research, which subsequently became the usable security and privacy community. In the paper, they describe the user problems that occur when encrypting emails. Asymmetric encryption was introduced in the 1970s and Pretty Good Privacy (PGP) in 1991, although these techniques are rarely used by users today because they are still too cumbersome.

In 2004 the authors of the paper "In search of usable security: Five lessons from the field" [7] pointed out that the research area of usable security had been very little investigated so far. In recent years, however, the importance has increased more and more. In the last decade, many researchers have focused on the topic of usable security and privacy. Therefore, they published numerous publications in this field:[3],[10],[23],[65],[15],[51],[25].

In the following, we briefly describe two areas of usable security that have a strong significance in this work, namely user authentication, and mobile security and privacy.

User Authentication

User authentication, such as passwords, is considered as main area for usable security. The goal is to improve the simple reliability and security of end-user authentication, such as improving text passwords, replacing text passwords with graphical passwords, or supplementing them with biometric or multi-level authentication [23]. According to Florencio and Herley, [20] passwords are still the primary way to authenticate websites.

Text Passwords Username and password are the most common technique for user authentication on the web today. Users tend to use short, easy-to-remember passwords and reuse them across multiple accounts [23]. Therefore, the main weakness of passwords is that an attacker can learn them so an unauthorized person can access an account. From the security point of view, long, more diverse, as well as distinct passwords for each system should be used. However, this causes a potentially large number of passwords the user has to remember.

Mobile Security and Privacy

The number of people using smartphones is increasing every year all over the world. New security and privacy challenges arise from the fact that smartphones

can access the phone's sensors and the personal data stored on them.

Privacy Most users ignore the privacy settings on their mobile devices. They are not aware that certain applications have been given access to data such as phone directories or photos, usually by the user himself. In the area of usable security it is necessary to invest a lot of work in the development of privacy settings. Most users do not find it important enough to read and adjust the privacy settings carefully. However, digital mechanisms for protecting sensitive data are only effective if they are as intuitive as possible and can be used correctly with little effort by the user.

Location Privacy Location privacy is a well researched domain in privacy. The user's location can be used to customize an application. Patil et al. [50] stated that users use location sharing for a number of different reasons like reporting approval or promotion of places or events, share and record travel and shaping appearance by indicating interesting activities. As described by Benisch et al. [8] users tend to share more information over the time not just location but also the times of day. The authors of the paper [4] found out that there is a need for improved visibility of the information collected by Social Web and to allow the user to better assess the implication of location sharing activities.

They concluded that:

> "it is highly feasible to infer rich personal information about users and their mobility users' spatiotemporal movement tracks and patterns.
>
> - Users' absence and presence in particular places.
> - Visiting frequencies and possible degree of association with specific places or place types.
> - Users' commuting habits.
> - Co-location patterns with other users and friends"

2.5 Usable Security and Privacy Guidelines and Methods

Similar to the regular usability evaluation, usable security and privacy uses two major methods, user studies and expert evaluation. Techniques widely used in Human Computer Interaction (HCI) are aimed at improving user effectiveness, efficiency or satisfaction, but they do not consider the requirements of the potential threats and vulnerabilities. Many usability studies have been conducted dealing with email encryption, authentication systems or secure devices paring [67],[24],[3],[33],[30],[64]. All these studies apply HCI methods that primarily aim to improve the ease of use. These methods have been developed for the overall usability evaluation of software and products. For the evaluation of security critical products it is necessary to consider both security and usability.

In this section we present guidelines and frameworks from literature that deal with the trade-off between usability and security and consider factors of both disciplines.

Researchers have developed guidelines and procedures to be followed to close the gaps between usability and security and privacy. The author of "Tradeoffs between Usability and Security" [56] presents a series of guidelines that consider the trade-offs between usability and security. Hof [27] presents a set of guidelines that should help software developers to improve end user usability of security-related mechanisms. The proposed guidelines are listed below.

G1 Understandability, open for all users.

G2 Empowered users.

G3 No jumping through hoops.

G4 Efficient use of user attention and memorization capability.

G5 Only informed decisions.

G6 Security as default.

G7 Fearless System.

G8 Security guidance, educating reaction on user errors.

G9 Consistency.

Another work [42] presents a quantification approach for assessing usable security in authentication mechanisms. The paper [29] proposes ten guidelines for secure interaction design. The guidelines are divided into three categories: General principles, Maintaining the actor–ability state und communicating with the user. The following list provides all guidelines defined by [68].

General principles

- Path of least resistance. The most natural way to do a task should also be the safest.
- Appropriate boundaries. The interface should draw distinctions among objects and actions along boundaries that matter to the user.

Maintaining the actor–ability state

- Explicit authorization. A user's authority should only be granted to another actor through an explicit user action understood to imply granting.
- Visibility. The interface should let the user easily review any active authority relationships that could affect security decisions.
- Revocability. The interface should let the user easily revoke authority that the user has granted, whenever revocation is possible.

- Expected ability. The interface should not give the user the impression of having authority that the user does not actually have.

Communicating with the user

- Trusted path. The user's communication channel to any entity that manipulates authority on the user's behalf must be unspoofable and free of corruption.
- Identifiability. The interface should ensure that identical objects or actions appear identical and that distinct objects or actions appear different.
- Expressiveness. The interface should provide enough expressive power to let users easily express security policies that fit their goals.
- Clarity. The effect of any authority-manipulating user action should be clearly apparent to the user before the action takes effect.

Kainda et al. [31] propose an HCISec security threat model (see Figure 2.5) that represents the critical factors that need to be considered when assessing usability and security. They argue in their work that it is not enough to conduct a conventional usability evaluation without addressing the security factors. In their model, the authors have defined four usability factors, five security factors, and another two factors affecting both areas. Moreover, they presented a matrix for measuring the security and usability elements and usage and threat scenarios shown in Figure 2.6.

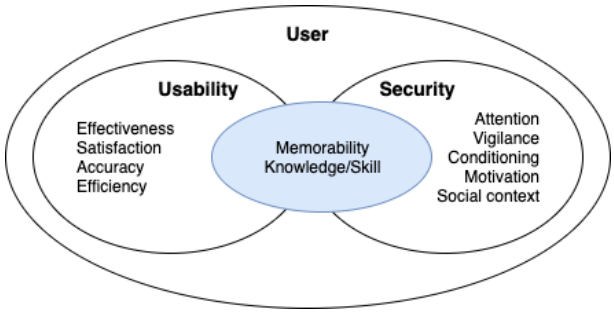


Figure 2.5: Security-usability threat model [31].

The assessment framework for usable-security (AFUS) aims to assess and integrate security, usability, and usable-security during the requirements engineering phase of the Software Development Life Cycle. The authors propose an Assessment Framework for Usable-Security based on two well-known techniques from the decision science field, OWASP Risk Rating Methodology[2]for security and the SALUTA Attribute Preference Table [21] for usability [26]. Mairiza et al. [39] describe a framework (sureCM) to manage security and usability

[2]https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

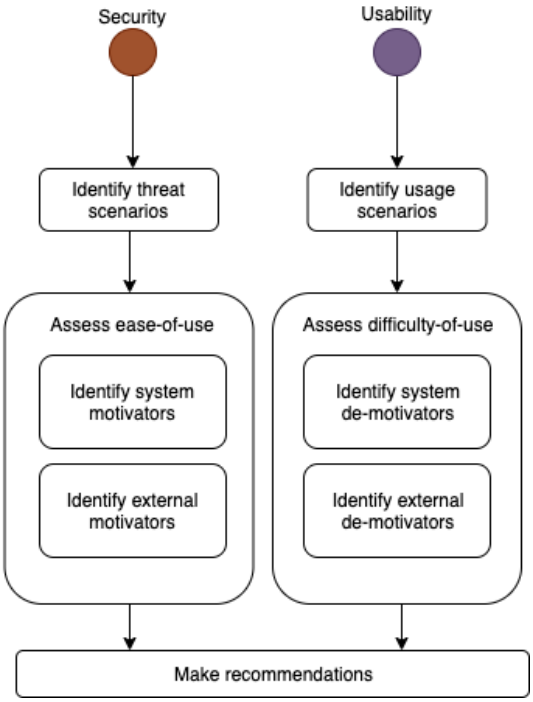


Figure 2.6: Process for security-usability analyses [31].

requirement conflicts. The framework consists of three major phases: the conception phase, the elaboration phase, and the definition phase. The proposed framework is shown in Figure 2.7 below.

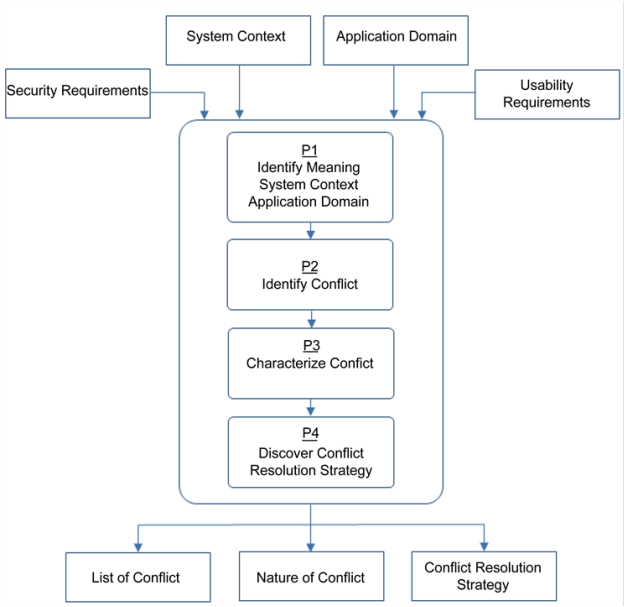


Figure 2.7: Security and Usability conflict requirement [39].

Part II

Convenient Agile Authentication

3 | Convenient Agile Authentication

Conference	NOT PUBLISHED YET
Authors	Vesna Krnjic, Andreas Fitzek, Michaela Kargl-Schrammel

3.1 On the Usability of Agile Authentication

User authentication, as a crucial step for most electronic services, typically relies on multi-factor authentication to strengthen their authentication processes. Used technologies are, however, not flexible enough to keep pace with the rapid progress of new authentication factor implementations or changed security properties. Increasing the security of the authentication process often comes at the cost of significantly decreased usability –consequently, the trade-off between security and usability results with unsatisfying compromises that are neither secure nor usable.

To improve this situation, we propose ALAP, an Agile Authentication Provider. ALAP provides authentication factors from different categories and levels of assurance (LoA). The system allows Service Providers to define their security requirements regarding the user-authentication process through policies by claiming a global LoA dynamically and flexibly. Based on this policy, ALAP dynamically assembles an authentication process for the respective service to fulfill its security requirements; at the same time, the user can choose the preferred authentication factor. In this paper, we demonstrated the feasibility of the proposed solution through a fully functional prototype and investigated the usability of the system in a multi-stage Usable Security Evaluation. In advance, an online survey with 200 participants was conducted to get a first impression of the attitudes and mental models of potential users of the ALAP system concerning two-factor-authentication. In the next step, an analytical Usable Security Inspection was conducted. Finlay, an empirical Usable Security Evaluation, was carried out. Based on the results, we identified usability and security issues and

derived possible improvements for dynamical agile authentication.

3.2 Introduction

Nowadays, the Internet is omnipresent and permeates many areas of everyday life. Many electronic services are consumed in a variety of domains such as e-Government, e-Banking, or Social Media. Most of them require authentication of users to ensure data security and privacy by restricting access for unauthorized persons. One of the first and still most common forms of authentication is the simple provision of knowledge like username and password [1]. This causes a potentially large number of passwords the user has to remember. Also, password-based authentication is considered insecure because it is vulnerable to a large number of attacks, such as brute-force attacks or phishing[3]. The latest work on password security [4] shows that users and their passwords are still considered the weakest link. Using additional authentication factors usually increases the security and reliability of user authentication. This concept is called multi-factor authentication (MFA). In MFA, two or more authentication factors are chosen from four different categories: something the user knows (e.g., password, PIN), something the user is (biometric factors), something the user typically does (behavior pattern) and something the user has (device signature, hardware device containing a credential, private key). More and more services offer their users access via MFA as an optional security feature. Currently, MFA is mandatory for state-of-the-art implementations of sensitive Internet services like eBanking or transactional eGovernment services. To maintain a sufficient level of security, implemented authentication processes need to be able to keep up with the technological developments and must react immediately to changing threat scenarios and upcoming attack strategies. This can become challenging in practice, as many technologies are often not flexible enough to keep pace with evolving requirements. In the design of security-relevant applications usually, resistance against malicious attacks is the primary design goal.

Moreover, the level of security that can be achieved is not only determined by the technical implementation. Of course, the algorithms applied, the technical components used, and the quality of the software implementation is indeed highly crucial for security, but a high level of security can only be reached if a holistic view of security is applied. Taking into account a comprehensive view of security, Usable Security and Privacy evaluation is seen as an essential activity throughout the development process. Usable Security and Privacy evaluation looks at the user's interaction with the application and investigates how easy users can complete their tasks and achieve their goals by utilizing the application under evaluation. The aim is to reveal flaws and aspects of the application that potentially lead to usability issues. In addition to looking at usability issues, Usable Security and Privacy evaluation also investigates how easy users can unintentionally compromise the security of the system. Security and usability cannot be assessed separately, as these concepts are highly interrelated: on the one hand improving usability may compromise security, and on the other hand,

implementation of measures that increase security may cause severe usability problems. Usable Security and Privacy evaluation helps to apply an integrated approach in the development of security applications, which strives to optimize security without neglecting usability.

There is a lack of modern authentication systems considering all those aspects. To close this gap, we developed an Agile Authentication Provider (ALAP), a secure and user-oriented prototype system providing a flexible solution for multi-factor authentication. ALAP provides an architecture that supports easy adding and compliant exchange of authentication factors. It enables the implementation of secure and reliable authentication solutions, which can react flexibly and fast on technological and strategic developments. Furthermore, it allows the user to choose the preferred authentication process by which the required LoA can be achieved. Through this mechanism, it is possible to consider both the needed security as well as a user-friendly authentication process. For these reasons, ALAP represents a modern, secure, and user-friendly authentication system that accommodates the required flexibility regarding rapid technological development and which can easily be extended. We conducted a Usable Security Evaluation of the ALAP system to investigate the usability of them and detect potential usability issues. In a preparatory step, closer insight into the knowledge, attitudes, and "mental models" of prospective users of the ALAP system concerning 2FA was gained by a survey. Based on these findings, a multi-stage Usable Security Evaluation was designed. First, an analytical Usable Security inspection, utilizing an adapted combination of the usability inspection methods Heuristic Evaluation and Cognitive Walkthrough, was conducted. The aim of this Usable Security inspection was to detect the most obvious usability issues. In the second stage, an empirical Usable Security Evaluation was carried out. This empirical Usable Security Evaluation included fourteen Thinking Aloud tests with users.

The contents of this paper are structured as follows. In Section 2, the existing authentication solutions for dynamic identity providers are discussed and assessed using these requirements. Section 3 defines requirements for an agile authentication system. Based on these requirements, an appropriate agile authentication architecture is proposed in Section 4. In Section 5, the usability evaluation of the presented prototype is discussed. Section 6 concludes and provides an outlook for possible future work. Finally, Section 7 provides an overview of the extension of the ALAP's authentication Policy. We introduce CALA – Convenient AgiLe Authentication – which is much more focused on the needs of users during an authentication process.

3.3 Related Work

In the past, service providers have mostly authenticated their users, which has increased implementation effort and costs. For economic reasons, often only one or just a few authentication methods are available, because establishing an infrastructure that can process different authentication factors is costly and

time-consuming. These are the main reasons why SP today often chooses to include third parties as identity providers (IDP) like Amazon, Facebook, or Google for outsourcing the authentication processes, which handles the identification and authentication of a user. In more detail, the IDP handles the authentication process of a user to provide an identity with a certain level of security to a Service Provider.

Consequently, the Service Provider does not necessarily have to handle user registration and authentication on its.[8]. To achieve a certain degree of system security, multiple authentication factors like the possession of a key, the knowledge of a password or biometric characteristics, should be combined [13]. The technical feasibility provided by the system often mainly determines the authentication factors that can be used. This aspect leads to static policies that enforce predefined combinations of specific authentication methods.

Nevertheless, it has to be considered that the bandwidth of possible methods is steadily growing and improving. To be able to react to the technology progress and use them for making the authentication more secure, the offered systems must be able to adapt or add authentication methods at any time. Otherwise, not only the security aspect but also the usability suffers from inflexibility. Multiple authentication factors should be replaced by more straightforward approaches if possible [9].

In modern approaches, users should be able to choose the authentication method they prefer [10]. Many different IDPs can be combined to be able to provide a broad selection of available methods. The proposed architecture in [6] represents an IDP proxy that offers secure authentication by combining multiple IDP authentication methods. The proxy selects specific IDP to authenticate the user based on the required LoA that must be reached and a pre-defined selection algorithm. However, this approach only considers the security aspect, whereby the usability is neglected. As stated in existing standards [11], overall user experience is critical to the success of any authentication methods. The prior experiences of users may influence their expectations. This is why the user experience should be as smooth as possible.

To mandate specific permissions, many existing systems use policies for authentication and authorization. Policies represent clear regulations by determining particular access requirements and their required types of authentication requests. They enable a mechanism by which Service Providers can establish their security requirements in which they determine the kind of accepted authentication methods to reach a specific LoA. Not only for Service Provider but also users, policies could be established for selecting authentication paths they want to go through to fulfill the requirements of an Service Provider. This can lead to increased user experience. The paths are mainly determined by their LoA and their related technical requirements. The LoA is a suggestion on how to secure sensitive data by describing the authentication process and by defining standards and requirements for each level, which are to be met.

However, we did not found any related work that has investigated the usability of a flexible authentication system like the one proposed by [6] or [7], which can change authentication factors on demand and enable Service Provider to

define risk-based authentication through a policy. To close this gap, we firstly define requirements derived from related work, provide a suitable architecture and implementation, and conduct a comprehensive Usable Security and Privacy evaluation. The objective of the Evaluation was to get first insights into the user's mental model, usage approach, preferences, considerations, and concerns related to the dynamic authentication system.

3.4 Requirements

As can be derived from Section 3.3, there are many requirements for a dynamic, secure, and at the same time usable authentication system. It requires a flexible authentication path, which in turn needs the ability to customize the authentication process according to the security requirements. To achieve this flexibility, the use of dynamic policies is recommended, and a modular structure is necessary to fulfill this required adaptability for entire authentication systems. We define four requirements for agile authentication in more detail as follows:

R1- Flexible authentication path:

An agile authentication system should propose a flexible authentication path to a user through the authentication process. The user should have the possibility to choose an authentication factor in each step of the authentication process.

R2- Flexible authentication policy:

An agile authentication system should allow the Service Provider to define an authentication policy in which it claims a global LoA on authentication. The agile authentication system is responsible for the execution of this policy and should be able to reflect the assurance levels of authentication factors. The policy should create decisions based on the attributes provided by the user and information defined in the authentication request. The flexible authentication path should also facilitate flexibility to a user through the authentication process.

R3 - Exchange of authentication factors:

An agile authentication system needs the ability to insert, delete or change the authentication factors in a quick and easy manner. Authentication factors should be easily and quickly exchangeable.

R4- Registration and Account Management:

An agile authentication system needs a registration process to be able to create an account on that system. Furthermore, an agile authentication system needs to provide the user with the ability to manage their active authentication factors. Each factor should have the ability to be added, changed, revoked or deleted.

Under consideration of these requirements, an agile authentication provider is proposed. The architecture and implementation are derived in the following section.

3.5 General Architecture

ALAP is an authentication provider that acts as an authentication proxy to combine different authentication factors dynamically. Each authentication factor is implemented as an independent Authentication Factor Provider (AFP) to fulfill the requirement of *exchange of authentication factors*. Figure 3.1 shows this general architecture of ALAP graphically.

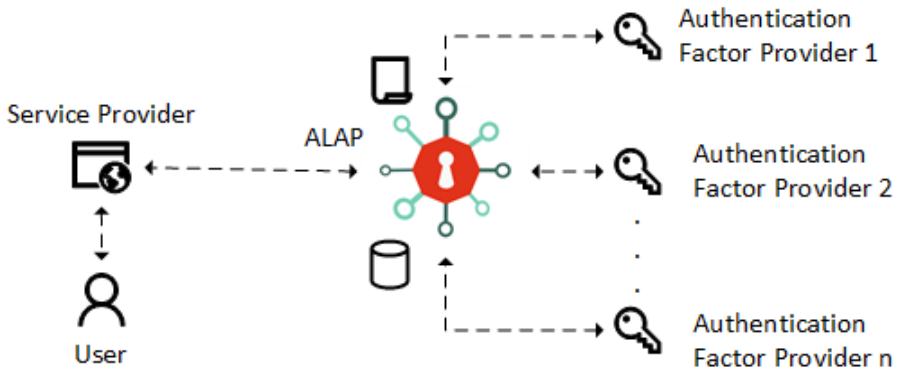


Figure 3.1: ALAP General Architecture

In more detail, ALAP provides authentication of users to a Service Provider, by combining different authentication factors. However, instead of verifying the credentials directly, ALAP dispatches an authentication request from a Service Provider to Authentication Factor Providers and awaits the authentication responses, so ALAP becomes an accumulating authentication system and acts in this process as an authentication proxy. The dispatching and accumulation of authentication factors are implemented as a policy that defines the required LoA and the available authentication factors. By using this policy-based approach, the ALAP system has two advantages. On the one hand, ALAP can exchange the authentication factors easily as they are not integrated into the system directly. On the other hand, ALAP allows each Service Provider to configure its authentication policy. This authentication policy is a JavaScript file that runs within a well-defined runtime environment in the ALAP system. The policy builds a set of authentication factors, which are selected based on the currently available environment attributes. Such environment attributes are the Service Provider metadata and settings, possible authentication factors, the information so far collected from the user, information from the original authentication request sent from Service Provider to ALAP, and information from already received assertions from Authentication Factor Providers.

Additionally, the policy allows the user to choose an authentication factor in each authentication factor selection step of the policy. So, ALAP meets the requirement *R1- Flexible authentication path* and the requirement *R2- Flexible*

authentication policy, by using the proposed policy-based approach. To facilitate this policy based authentication, ALAP provides two main functionalities to the users and Service Providers. The first functionality provides all required *Registration and Account Management* process to users, and the technical authentication of users.

3.5.1 Registration and Account Management

The registration and account-management functionality are implemented as an ALAP Web application that facilitates all functions to register new users or to manage already existing user accounts.

Registration.

According to NIST 800-63-3 [5], the registration process consists of two main parts, the identity proofing and the issuance of credentials. The ALAP system is an agile authentication provider. Therefore the main task of account management is to issue and manage credentials that should be used as authentication factors. So, identity proofing is out of the scope of the ALAP system and thus identify proofing is limited too issuing of unique identifiers only. The registration process, which can be initiated by any entity, only generates a new unique identifier that identifies the user in the context of ALAP. Afterward, ALAP executes the registration policy of the account management to calculate the set of possible authentication factors that have to be initialized during the registration process. This set of authentication factors represents the minimum set of user credentials that are necessary to use ALAP. Since ALAP is only an authentication proxy, the user is redirected to every AFP that implements an authentication factor from the available set. Therefore, each AFP has to provide a management page that implements all functions to manage a specific authentication factor. By using this page, the AFP issues the credential to the user and response to ALAP with a unique identifier that represents the user in the AFP context. After the successful issuing of the first authentication factor, ALAP continues with the registration of the next authentication factor.

Once the registration of all authentication factors is complete, all context-specific identifiers of AFPs are bound to the unique identifier of the ALAP system to identify the user at each AFP. After binding, the registration process is complete, and the user can use the ALAP system for multi-factor authentication.

Account Management.

Every registered user can use the account management application to manage her user account on the ALAP system. This management application allows the user to request new authentication factors to issue new credentials or to revoke already granted one. Access to the management page is protected by ALAP's account management access policy that includes the authentication factors from the registration policy. Consequently, a user can use its registered

credentials for a log-in on the account management application. After the log-in, the management application shows all already registered authentication factors and provides the functionality to register additional authentication factors that are currently supported by ALAP. The registered authentication factors can be easily managed on every AFP by using the AFP context-specific user identifier and a single sign-on approach between ALAP and the AFP. That allows users to renew or revoke already existing credentials on the AFP side. If a new authentication factor should be issued, the issuance procedure is equal to the registration procedure of an authentication factor describe in 3.5.1. Therefore, the proposed registration process, together with the account management functionality, fulfills the requirement *R4- Registration and Account Management.*

In the next section, we will show the whole authentication process of ALAP and demonstrate how the system meets the requirements.

3.5.2 Authentication

This subsection describes the authentication process of the proposed architecture, which is independent of the user authentication protocol between ALAP and the AFP. Figure 3.2 shows an abstracted version of an authentication process that illustrates the usage of two authentication factors. Since ALAP can combine any number of authentication factors, the following enumeration gives more details on the authentication process proposed in our architecture.

1. SP sends an authentication request to ALAP by using a standard authentication protocol.

2. ALAP detects the SP that originates the request and fetches its policy.

3. ALAP executes the first part of the policy and provides a set of authentication factors or failure response if a set can not be found.

    - If a set of authentication factors is returned, this set contains authentication factors that are selected based on the requested authentication assurance level defined in the policy minus the authentication factors which are already used during the current authentication process. The user chooses the preferred authentication factor that fulfills the policy.
    - If a failure is returned from the policy ALAP sends a negative authentication response to the SP.

4. Once, an authentication factor is selected, ALAP sends an authentication request to the selected AFP. The AFP performs the authentication with the user. After the user authentication has successfully performed on AFP side, the AFP response to ALAP with an assertion, containing the AFP specific identifier of the user.

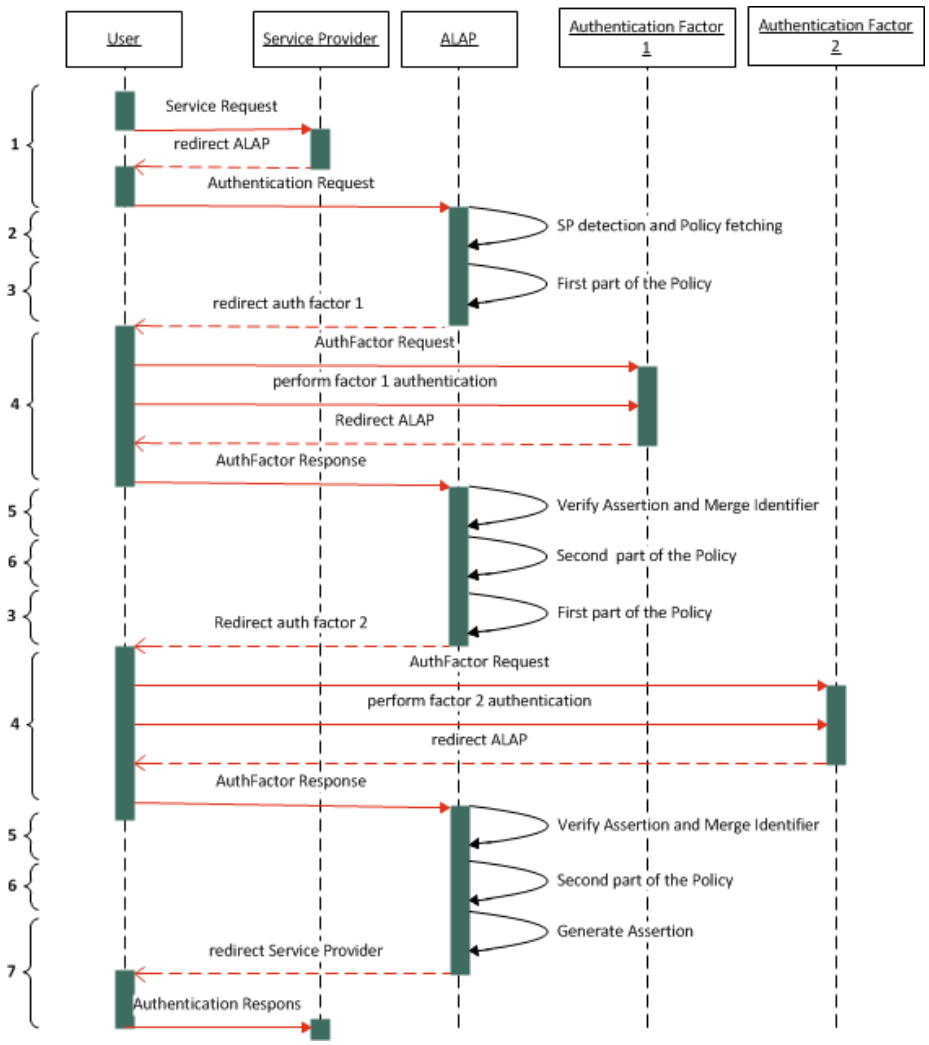


Figure 3.2: ALAP Authentication process

5. ALAP verifies the assertion and merges the user identifier provided by the AFP with the user information stored in the ALAP to determine the user profile.

    - If success was provided by the AFP in the assertion and the assertion is valid, ALAP match's the identifier. If the identifier provided by the currently used AFP matches to the user profile on ALAP and to identifiers provided by other already used AFPs than the matching was successful and the authentication factor is marked as valid used.
    - If an error response is returned from the authentication factor, the processing continues in step four.
    - If the assertions are not valid, or the identity provided by the AFP does not match to the identifiers bound to ALAP, ALAP sends a negative authentication response to the SP.

6. ALAP executes the policy again and checks if the set of validly used authentication factors fulfills the requested LoA defined in the policy.

    - If the set of validly used authentication factors does not fulfill the requested LoA, the process continues in step three and the selection of the next authentication factor.
    - If the set of validly used authentication factors fulfills the requested LoA, the authentication process is finished, resume in step 7.

7. ALAP generates an assertion that includes the unique user identifier on ALAP system which was determined by the executed authentication factors. ALAP stores all collected assertions from AFPs, a hash of the used policy, and the generated assertion sent to the SP, for potential incident revision. The generated assertion is sent as an authentication response to the SP.

3.6 Usable Security and Privacy Evaluation

It is crucial to take into account usable security aspects, even in the planning and development phase of security-related systems [14]. Having that in mind, we conducted a multi-step US study of ALAP in an early stage of development to be able to react on user feedback in further architecture development. This evaluation aimed to find out people's usage approach, preferences, considerations, and concerns related to dynamic authentication systems like ALAP. Specifically, the objectives of this study were to answer the following questions:

Q1 How well do users with different domain knowledge understand the underlying concepts, such as MFA, authentication factors, IDP?

Q2 Is the workflow that needs to be followed when using ALAP clear to users?

Q3 Is the information provided by the system to the users sufficient and clear for people with different knowledge background?

To answer the defined research questions we followed a well-defined method for the Usable Security and Privacy study, described in the next subsection.

3.6.1 Methodology

To answer the research questions a multi-step approach was taken for this evaluation. For designing a successful US evaluation, it is important to have a good knowledge of the relevant user group. As a preparatory step for the US study, a small-scale online survey with 200 participants was conducted, to get first impressions of the attitudes and mental models of potential users concerning 2FA. Based on the findings of the conducted survey, the design of the US study was elaborated as proposed by L. Cranor [15]. For the first evaluation step an analytical US inspection of ALAP was conducted. The aim of this inspection was to identify the most usability and security-related issues so that they could be fixed before asking test-users to evaluate ALAP. For the analytical US inspection a combination of the usability inspection method Heuristic Evaluation proposed by [20], [21] and an extended version of Cognitive Walkthrough [22], [21] was applied. After first improvements to the deployed prototype system had been made, a Thinking Aloud (TA) test was conducted with fourteen test users. TA test was described by Nielsen in 1989 [16] for qualitative usability studies and is one of the standard methods applied in usability evaluation since more than 20 years. While in HCI usability evaluation such test is conducted to detect usability issues, in US evaluation the purpose is not only to detect usability issues, but moreover to help to find security threats resulting from the users' interaction with the system. After completing each task of the TA test, the users were asked to assess how easy the completion was by rating the "Single Easy Question" on a 7-point Likert scale ranging from "very easy" to "extremely difficult". To obtain a metric for the test-users perception of the overall usability of the system, the test-user fill in the System Usability Scale (SUS) questionnaire after they had completed the TA test[17]. SUS consists of 10 alternating positively and negatively formulated questions, which must be rated by the user on a 5-point scale. After the last task, we have a wrap-up session where the evaluator interviewed the test-user and discussed the main issues found during the session. At that stage, the evaluator also asked why the user was behaving in a certain way during the test. The following subsection describes the evaluation setup used for the Usable Security Study. Figure 3.3 illustrates the outline of the evaluation structure in more detail.

Evaluation Setup.

The evaluation has been based on ALAP's prototype implementation accessible at https://demo.egiz.gv.at/aboutALAP. The deployed components have been arranged according to the evaluation setup shown in Figure 3.4. For the registration with ALAP, we used the Austrian eID solution, the Mobile Phone

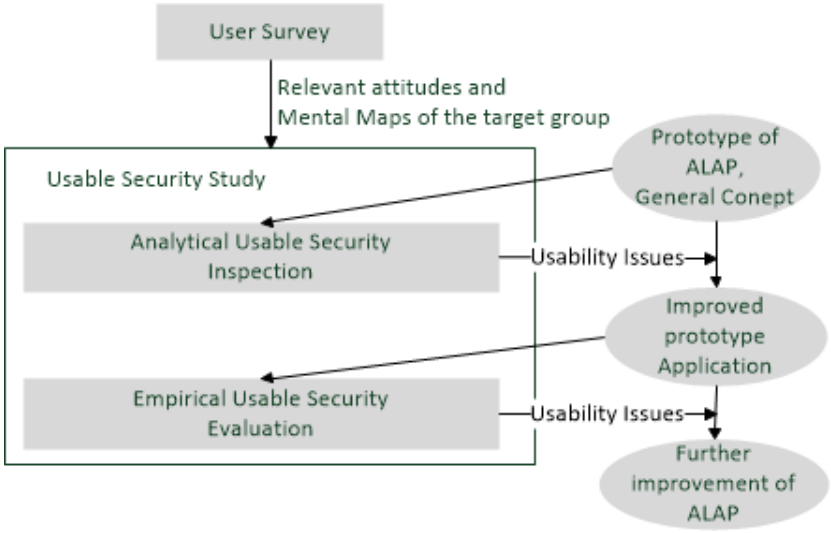


Figure 3.3: Outline of the structure of the evaluation

Signature. Five different authentication factors (One Time Password, Username Password, SMS Tan, QR Tan and Google login) were provided by ALAP. Furthermore, a SP PDF-Signature Online, which requires user authentication, has been implemented. PDF-Signature is also known concept from Austrian eGovernment. The evaluation setup reflects the general architecture as introduced in Section 3.5. With the evaluation setup, ALAP-based user-authentication processes as proposed in Section 3.5.2 could be carried out.

3.6.2 User Survey

An essential precondition for designing a successful Usable Security and Privacy study is a good knowledge of the relevant user groups and their mental models, their notions, and expectations regarding the functionality of the system as well as their approach and aims regarding the usage of the system [18], [19]. We tested ALAP in the area of Austrian eGovernment, therefor Austrian citizens are seen as a relevant user group for the system, so the online survey was conducted among them. Invitations to participate in the survey were spread via email and personal contacts among the private and business network of the authors. 38 % of the survey participants were younger than 35 years, whereby 40 % of them were women.

The questionnaire started with some introductory questions regarding gender, age, IT background, computer and mobile phone ownership and Internet usage habits of the participants. These introductory questions were followed by a group of questions regarding the Austrian Mobile Phone Signature, the Austrian eID solution which allows the use of Austrian transactional e-Government services as well as private sector services that require reliable authentication.

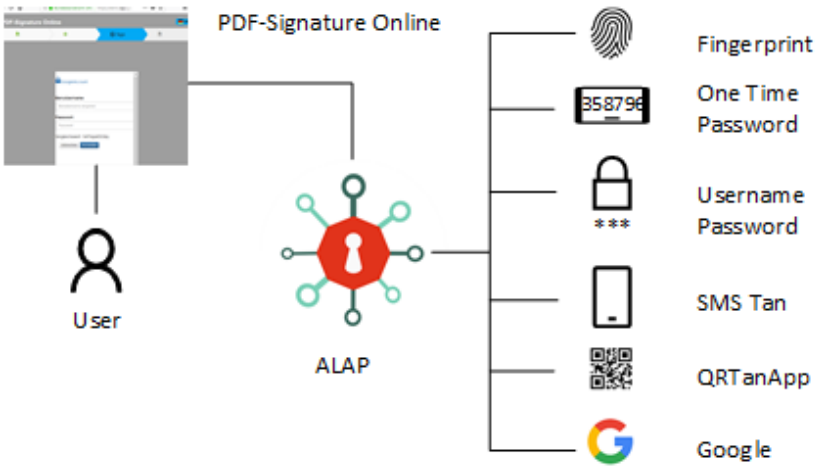


Figure 3.4: Evaluation Setup

The Austrian Mobile Phone Signature utilizes 2FA, whereby proof of possession of the user's phone is used as the second authentication factor. The second part of the questionnaire was about 2FA and included questions such as:

- Have you already used 2FA?
- Which kind of 2FA?
- For which services/websites have you used 2FA?
- Which factors would you prefer to use?
- Why do you use 2FA?
- Why don't you use 2FA voluntarily?

In the following section, we will describe lessons learned from the results of the survey among potential users of the ALAP prototype system, which are perceived to be important for the design and conduction of the Usable Security and Privacy Evaluation.

Lessons learned from Survey.

Survey results indicate that IT background might be the most important factor concerning user segregation in MFA because the questionnaire answers of participants with and without IT background were different. Figure 3.5 shows that authentication factors used by the participants are not always factors they would like to use. For example, only 5.3% of the participants from the "IT Group" use biometrics, but 23.0% of them would prefer such authentication

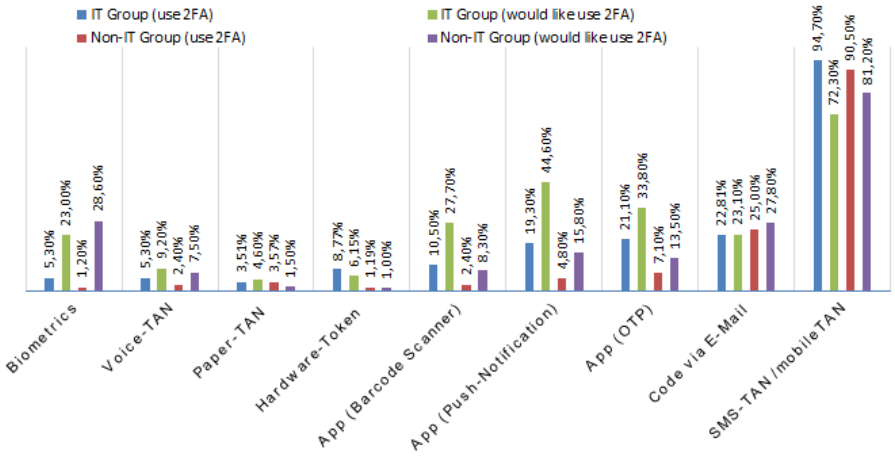


Figure 3.5: Authentication factors participants use versus factors they would like to use.

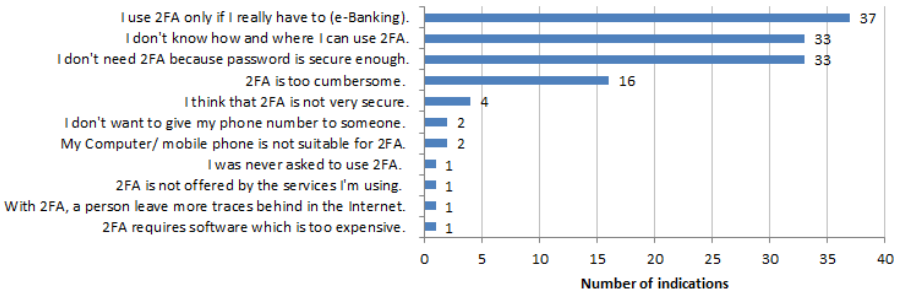


Figure 3.6: Why do not survey participants use 2FA voluntary?

method. Similar results can be observed for the Authentication App with push notification or barcode scanner.

The survey participants know 2FA primarily form eBanking. Mobile applications like Google Authenticator seem to be not known among the participants and seem to be not used by people without IT background. The majority of the survey participants preferred SMS-TAN as an authentication factor. As shown in Figure 3.6 participants do not use 2FA for various reasons, for example, they think that authentication via password is sufficiently secure or they do not know, where and how to use 2FA. Another reason is that they think that usage of 2FA is cumbersome. The survey shows that there are big differences between participants with IT background and the one without regarding 2FA and security considerations in general. Participants without IT background do not use 2FA voluntary; they use it only if they have to. So they are only familiar with the technology they rely must use for certain services like eBanking. In Austria, most banks use SMS-TAN as the second factor so most participants are familiar with it. Authentication applications like Google Authenticator are not common especially by the participants without IT background.

3.6.3 Analytical Usable Security Inspection

The analytical US inspection of ALAP basically involved the following four steps:

- Creation of Sequence Models.
- Conduction of the first US inspection round following the sequences of steps as listed in the sequence model and applying the adapted Heuristics and the adapted Cognitive Walkthrough questions.
- Conduction of second US inspection round testing corner cases by leaving the correct path of steps as described in sequence model and by simulating user errors, to check failure behavior of the system.
- Rating the relevant usability issues.

Main Findings:

During the analytical US Inspection 66 potential usability issues were revealed by three evaluators. Almost all of the identified issues were primarily related to problems like typing errors, inconsistent wording, usage of technical terms without explanation, lack of information, obscure or confusing layout, missing feedback of the system or unclear error messages. The evaluators strongly recommended a redesign of ALAP's account management, as the process flow for factor authentication is not always understandable. Furthermore, users should get more help from the system with the activation of the authentication factors. For example, the system should suggest the user which factor should be activated next and provide more feedback about already activated factors. All results were used for the further development of ALAP. Minor improvements

could be implemented before the empirical US evaluation. Some findings could not be solved immediately because a complete redesign of the underlying architecture or the user interface would be necessary to address that issues.

3.6.4 Empirical Usable Security Inspection

The purpose of the empirical US evaluation was to find out which aspects of the system hinder easy and secure usage of ALAP and the deployed environment. Firstly, we define three research questions listed before in the section. Subsequently, we clarify Use Cases that should answer that research questions.

Use Case I: Registration to ALAP and activation of the authentication factors.

Use Case II: Using the SP "PDF-Signature Online", authentication with ALAP.

We conducted a TA test with fourteen participants, whereby two of the test participants were pre-test users. From the previous analysis of potential ALAP user, we already know that IT background is the most important factor regarding 2FA, whereby age and gender are no suitable user segregation factors as no significant differences were observed between the answers of survey participants of different age groups and between different gender. So, two groups of test-users were recruited, which differed mainly in their IT background. Six test users from the "IT Group" were students of computer science at the Graz University of Technology, the six test- users of the "Non-IT Group" did not have any IT related education or work. Each tests sessions took one and a half hour and were conducted in the lab of the Graz University of Technology. At the beginning of the session, each test-user was asked to answer a short introductory questionnaire, which included some demographical questions as well as some questions regarding the level of IT experience and usage of 2FA.

The TA test was split into two blocks related to Use Case I, the registration to ALAP and account management, and Use Case II, using the PDF-Signature Online for signing a pdf-document. In the first session the test-users were asked to complete the following four tasks:

1. Inform yourself about ALAP on https://demo.egiz.gv.at/aboutALA .
2. Register to ALAP.
3. Configure ALAP so that you can use it to authenticate yourself on a SP.
4. Activate also the remaining authentication factors provided by ALAP.

In the second part of the test, which was related to Use Case II, was about using PDF-Signature Online, where the user authentication is done by ALAP.

1. Inform yourself on www.buergerkarte.at about PDF-Signature.
2. Use the service on https://demo.egiz.gv.at/g2f-pdfas and sign the "test-document" file electronically.

Main Findings:

In total, 182 different usability issues were revealed by the twelve TA tests which severity was rated according to their potential influence on the task completion. For this severity rating, four categories were defined: 1 - cosmetic problem, 2 - medium usability issues, 3 - severe usability issues, 4 - security-related usability issues. Most of the issues (65,4%) were from category 1 and 2, which means that users were still able to complete the task. 30.2% of the problems found were from category 3, so they are potentially hindering some users from completing their task.

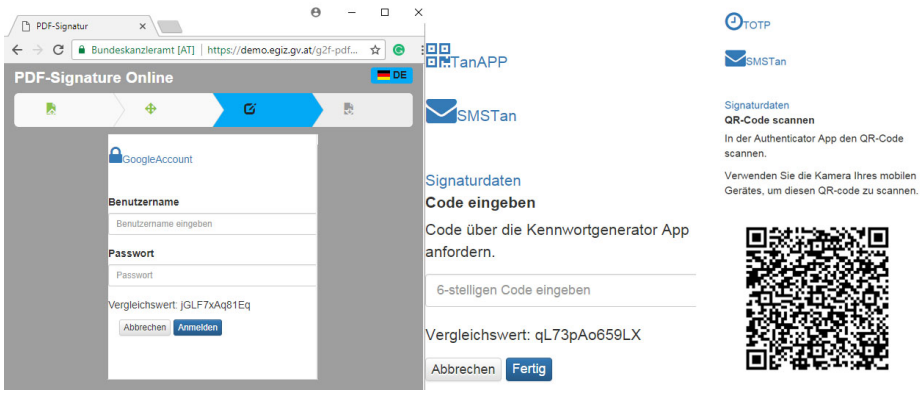


Figure 3.7: Authentication process of PDF-Signature Online

In addition to the classical usability problems, we were able to identify 4.4% persisting security-related weaknesses caused by the user interface from category 4. One example of such an issue was the graphical presentation of the authentication window of PDF-Signature Online, as demonstrated in Figure 3.7. As the first authentication factor, users could choose between login with Google account or Username/Password. All test-users interpreted the "Google Account" link as the heading of that form, and thus filled-in their Google account credentials into this form. This is a security-related usability issue that is caused by a mismatch of the mental model of the developers and the users. On the contrary, the selection of the second factor did not lead to such problems. For more detail see Table 3.1.

Figure 3.8 illustrates the actual task completion rate, which varied from 50% for Task I.3 to 100% for Task I.2. 50% of the test-users did not know which factors they should activate to authenticate them by "PDF-Signature Online". Four test-users needed some help from the moderator to complete this task, and two could not complete this task even with help and give up. Some test-users believed that ALAP was ready to use as soon as they had activated the "Username/Password" factor. Other test-users believed that they had finished the task as soon as they hat installed the Authenticator Application for QR-Code and OTP on their Smartphone but did not provide the activation in

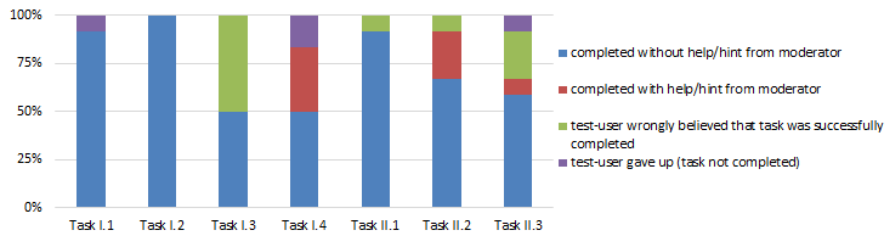


Figure 3.8: Task Completion

ALAP. After completion of each test-task, the test-users were asked to assess how easy the task completion was 7-point Likert scale. For the "IT Group" all test-tasks were easier than for the test-users of "Non-IT Group" (see Figure 3.9 and **??**). From Use Case I, the activation of authentication factors, especially the activation of "QRTanApp" and the "TOTP" factor, were perceived to be the most difficult task by both groups. From the test-task of Use Case II, the authentication on "PDF Signature Online" was perceived as the most difficult task by both test-user groups. However, this task showed the largest differences in the assessments from the two user groups. While this task was assessed as only slightly difficult by the "IT Group", it was perceived as very difficult by the "Non-IT Group".

The test-users fill in the System Usability Scale (SUS) questionnaire after they had completed the TA test to obtain a metric for the test-users perception of the overall usability of ALAP. It can be seen from Table 3.2 that the average of the SUS scores assigned by the test-users of the "IT Group", with a score of 49,2, was significantly higher than the average of the SUS score assigned by the "Non-IT Group" which was 27.8.

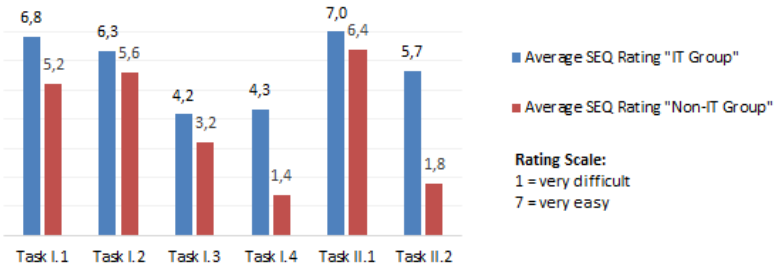


Figure 3.9: Perceived Ease of Test-Task

Table 3.1: Number of different issues revealed per severity category

Severity Category and description	**Issues revealed**	**Issues in %**
1: Only cosmetic problem, not affecting users' ability to complete their task.	13	7.2%
2: Medium usability issues, makes it more complicated for some users to complete their task.	106	58.2%
3: Severe usability issue, potentially hinders some users to complete their task.	55	30.2%
4: Security related usability issues, rated as severe since it potentially causes security problems.	8	4.4%

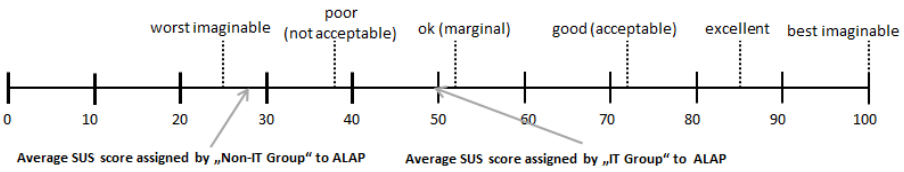


Figure 3.10: Average SUS score of ALAP concerning the adjective rating scale

Table 3.2: SUS score assigned to ALAP Prototype by the Test-Users.

Test-User	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
SUS score calculated from the questionnaires	62.0	55.0	25.5	65.5	53.5	33.5	33	23.0	49.5	23.5	10	38.5
Average SUS Score per Test-User Group	49.2						27.8					

3.6.5 System Usability Scale

To interpret the SUS score, we use the seven-point adjective-anchored Likert scale introduced by Bangor et al. [23]. Therefore the SUS score of ALAP can be translated to "marginal" for the "IT Group" and "worst imaginable" for the "Non-IT Group". For more detail, see Figure 3.10.

Observation and Results from Interviews. Valuable feedback for the future development of ALAP comes just from observing the users by using ALAP as well as from the wrap-up interviews after the usability tests. We get insights into users' acceptations and observe how they use the system. From the observations, we get an idea of how we could redesign the account management page to make the activation process of the authentication factors more usable. Once the factors were activated, users did not have difficulties to use them, especially the choice of the second factor did not lead to problems. All test-users positively mentioned the possibility of choosing their authentication factor during the authentication process. Some of the users would like to have the option to select preferred factors in the account management of ALAP. From the interviews, it has become clear that test-users would prefer to have more explanations and descriptions throw the whole registration, activation, and authorization process. They were not familiar with the used language and authentication therms. They mentioned that most of the problems would not occur if ALAP would provide more information.

3.6.6 Discussion

The conducted Usable Security Evaluation yielded many interesting results. In this section, we interpret these results to answer the predefined research questions. From the conducted user survey as well as from the TA test, we can infer that users' background knowledge is the most critical factor regarding ALAP. To answer research question Q1, we can reveal that users with IT education or work understand the purpose of MFA and also the overall workflow of ALAP. On the contrary, people without IT background had difficulties to follow all necessary steps needed for the usage of the ALAP, so the overall purpose of the system was not clear for them. For example, they do not know the difference between ALAP registration and activation of the authentication factors. Some of them did not even know why they must activate the authentication factors they want to use in ALAP. Despite these, all users mentioned in the interview that they would like to use such a dynamic authentication system like ALAP in practice, as they want to choose between all possible factors. We can answer the research question Q2 by concluding that the workflow of ALAP is not clear to all users and should be improved in the way that all people can use it regardless of their education or work.

Regarding research question Q3, the Usable Security Evaluation revealed that the test-users from the "Non-IT Group" were not familiar with the language used in the authentication process in general. All test-users from the "Non-IT Group" state that the information provided by ALAP is insufficient, or they did

not understand used therms like Authentication Application, Youbikey, or Password generator. So, they need much more information about the authentication provider's purpose as well as a detailed description of each authentication factor offered by the system.

3.7 Conclusion and Future Work

We presented the architecture and implementation of a dynamic, Agile Authentication Provider. ALAP provides authentication factors from different categories and allows SP to define its security requirements through policy by claiming a global LoA. Our solution dynamically assembles an authentication process for a service to fulfill its security requirements, whereby the users can choose the preferred authentication factor. With the implementation of a fully functional solution, we demonstrated the feasibility of the system. To evaluate the system from the users' perspective, we conducted a multi-step Usable Security study. The evaluation of ALAP in a real-world scenario has lead to valuable findings. The obtained results show that users with IT background would like to use systems like ALAP as they could choose their preferred authentication method. All participants from "IT Group" were able to register to ALAP successfully, activate all authentication factors, and authenticate them with ALAP to an SP. In contrast, test-users from "Non-IT Group needed assistance from the evaluator to accomplish the whole process. By observing users' interaction with ALAP and collect user feedback through different questionnaires, we were able to identify persisting weaknesses and a further room for improvement.

3.8 Authentication Policy - Convenient AgiLe Authentication (CALA)

The dynamic IdP-Proxy algorithm from related work and also our ALAP implementation of a dynamic authentication provider are developed from the security point of view. There is a need for a multi-factor IdP-Proxy that is, of course, secure but also user-oriented and convenient to use. From our first conducted pilot studies, we know that end-users are mostly overwhelmed with the Agile Authentication system. Most of them do not know much about multi-factor authentication and do not want to waste time on sophisticated authentication. Therefore, we propose CALA- the Convenient AgiLe Authentication.

3.8.1 Level of Assurance (LoA):

The strength of an authentication process is indicated by a measurement known as the Authentication Assurance Level (AAL). The National Institute of Standards and Technologies (NIST) defines three different AAL's.

AAL1 provides some assurance that the claimant controls the authenticator, it requires at least single-factor authentication.

AAL2 provides high confidence that the claimant controls authenticators. Two different authentication factors and approved cryptographic techniques are required.

AAL3 provides very high confidence that the claimant controls the authenticator. Authentication is based on proof of possession of a key through a cryptographic protocol, it requires a hard cryptographic authenticator. For the selection of the authentication factors in ALAP only security requirements are considered: The global LoA claimed by the Service Provider.

Many other attributes like user's location or preferred authentication factors from the user's view could be used for the selection which would improve the usability of the system.

In this section we propose CALA the Convenient Agile Authentication. In CALA, we calculate the global AAL based on security under consideration of usability. Firstly, we calculate the AAL based on the probability of false acceptance as proposed by [6]. Accordingly, the assurance level indicates the probability that the claimant is someone else. Afterwards, we categorize all available authentication factors by means of their usability. The CALA authentication selection algorithm will calculate the most usable authentication factor set that meets the security requirements defined by the Service Provider policy. CALA will be able to ensure the claimed security and at the same time suggest the most usable authentication factor available at each step of authentication with the help of machine learning.

First ideas for the selection algorithm of the authentication factors: Each authentication factor will be presented as a node of a directed graph. The nodes will be weighted with different weights: security, usability, privacy. The Authentication Assurance Level (security) is calculated from the probability of false acceptance. All nodes will be weighted with a usability weight. The selection algorithm proposed in the literature do only take the security requirement into account, we will try to develop an algorithm with the possibility to consider different attributes while selecting the most usable authentication factor for a specific user and at the same time fulfill the security requirements of the Service Provider.

Algorithm

Users Profile:

- ID - Identifier of the user
- List of all available IdPs the claimant is authenticated.
- IdP Usability Level (IdP- UL): Usability level of the authentication factor provided by an IdP. We assume that the authentication factors are divided in three usability categories. The usability level is derived from claimant

ratings on authentication factors and known usability considerations of the authenticator types.

- List of used IdP's
- List of unused IdP's
- C-AAL: current authentication assurance level

Service Provider:

- ID of the application server
- R-AAL: AAL is an authentication assurance level that a relying party (RP) requests to authentication for providing an application service.

Authentication Factor Profile:

- ID- Identifier of the Authenticator
- IdP-AAL: Authentication Assurance Level agreed by an IdP.
- Authentication Method

1. Read C-AAL from user profile
2. Read R-AAL from IdP profile
3. Is CAAL $>=$ RAAL
4. Success if CAAL $\geq$ RAAL
5. If CAAL $<$ RAAL then select authentication factor with greatest IdP-UL
    - Must be from the List of unused IdP's
    - Selected authentication factor must be from different category as already used IdP's
6. Sending Authentication Request to selected Authenticator
7. The selected IdP performs the authentication with the claimant
8. Updated the claimant profile if authentication

Account Management: The IdP-Proxy account management page is a very important part of the architecture. The user is using the account management page to register and to organize the authentication factors (activate, change, delete and revoke the factors). So the user is not only using the account management page ones, moreover it is the most relevant part of the system for the end-user. Based on the findings from the Usable Security Evaluations propose a new solution for the registration and account management process for CALA should be designed with strong involvement of end-users. In order to

reach that goal we need to use usability techniques from HCI and other fields but also modify and develop new methods for the usability evaluation of agile IdP-Proxies.

Open research question: which methods can be used in order to evaluate a dynamic Agile Authentication system? Is there a methodology that we can follow for our evaluation or should we modify and design our own methods for the usable security evaluation? The results from the redesign should contribute to the body of knowledge in the context of usability of dynamic authentication systems and development of user-oriented IdP proxies so that the researchers in the same field can benefit from it.

Bibliography

[1] N. D'Lima and J. Mittal: Password authentication using Keystroke Biometrics, 2015 International Conference on Communication, Information Computing Technology (ICCICT),1-6, (2015)

[2] Robert H. Morris Sr. and Ken Thompson: Password Security - A Case History, Communications of the ACM, 22, 11,594–597, (1979)

[3] Florencio, Dinei and Herley, Cormac and Van Oorschot, Paul C: An administrator's guide to internet password research, 28th Large Installation System Administration Conference (LISA14), 44–61, (2014)

[4] V. Taneski and M. Heričko and B. Brumen: Password security — No change in 35 years?, Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on, 1360–1365, (2014)

[5] P.l A. Grassi, Michael E. Garcia, J. L. Fenton: Digital Identity Guidelines, NIST Special Publication 800-63-3, (2017)

[6] T. Kaji, F. Fujishiro and S. Irube: IdP proxy for combined authentication based on multiple IdPs, IEEE 4th International Symposium on Advanced Networks and Telecommunication Systems, (2010)

[7] author=Y. Shah and V. Choyi and L. Subramanian: Multi-factor Authentication as a Service, 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, p. 144-150,(2015)

[8] A. A. Malik and H. Anwar and M. A. Shibli: Federated Identity Management (FIM): Challenges and opportunities, 2015 Conference on Information Assurance and Cyber Security (CIACS), 75-82, (2015)

[9] Han, Weili and Sun, Chen and Shen, Chenguang and Lei, Chang and Shen, Sean: Dynamic combination of authentication factors based on quantified risk and benefit, Security and Communication Networks,7,385–396,(2014)

[10] W. Anani and A. Ouda: The importance of human dynamics in the future user authentication, 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 1-5, (2017)

[11] https://www.iso.org/standard/52938.html

[12] https://patents.google.com/patent/US8978100B2/en

[13] Martin, Jason; Rajan, Anand; Steigerwald, Bob: Authenticate once and be done: User-centric authentication through rich device capabilities, Intel Technology Journal, Vol. 18 Issue 4, p8-28. 21p. (2014)

[14] Smetters, D. K. and Grinter, R. E.: Moving from the Design of Usable Security Technologies to the Design of Useful Secure Applications, Proceedings of the 2002 Workshop on New Security Paradigms, NSPW '02 ,82–89 ,8,ACM, (2002)

[15] L. F. Cranor: Conducting Usable Security Studies: It's Complicated, USENIX Association, Washington, D.C.,(2015)

[16] Nielsen, Jakob, Usability Engineering at a Discount, Proceedings of the Third International Conference on Human-computer Interaction on Designing and Using Human-computer Interfaces and Knowledge Based Systems (2Nd Ed.), Boston, Massachusetts, USA ,394–401, 8, Elsevier Science Inc., New York, NY, USA, (1989)

[17] J.Brooke, "SUS: A Retrospective, " Journal of Usability Studies (JUS), Volume 8, Issue2, pp.29-40, (2013)

[18] L. Cranor: A Framework for Reasoning About the Human in the Loop, in Usability, Psychology ans Security 2008 UPSEC', (2008)

[19] J. Nielsen: Mental Models,20010 Available online at: https://www.nngroup.com/articles/mental-models/. Last seen at November 2019

[20] M.Hertzum, N.C. Juul, N.Jorgensen and M. Norgaard: Usable Security and E-Banking: Ease of Use vis-à-vis Security - Data collection in an evaluation of six Danish web-based electronic banking systems. Technical Report, Denmark, (2004)

[21] J.Hung and P.Zablosky: "fluid - Heuristic Evaluation, Cognitive Walkthrough, Online, 2009, Available at: `https://wiki.fluidproject.org/display/fluid/UX+Walkthrough+Protocols+and+Checklists` (Last seen: November, 2019)

[22] S. Rick: The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00, The Hague, The Netherlands, p.353–359,7,ACM, New York, NY, USA, (2000)

[23] A. Bangor, P. Kortum and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale, J. Usability Studies, Usability Professionals' Association, v 4, 3, p114–123, 10, (2009)

Part III

Publications
Usable Security Evaluations

4 Modular Architecture for Adaptable Signature-Creation Tools Requirements, Architecture, Implementation and Usability

Conference	EGOV - IFIP e-Government
Language	English
Title	Modular Architecture for Adaptable Signature - Creation Tools Requirements,Architecture, Implementation and Usability
Authors	Vesna Krnjic, Klaus Stranacher, Tobias Kellner, Andreas Fitzek
Publisher	Springer, Berlin, Heidelberg
Booktitle	EGOV - IFIP e-Government Conference

Modular Architecture for Adaptable Signature-Creation Tools

Requirements, Architecture, Implementation and Usability

Vesna Krnjic, Klaus Stranacher, Tobias Kellner, and Andreas Fitzek

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Graz, Austria
{vesna.krnjic,klaus.stranacher,tobias.kellner,
andreas.fitzek}@iaik.tugraz.at

Abstract. Electronic signatures play an important role in e-Business and e-Government applications. In particular, electronic signatures fulfilling certain security requirements are legally equivalent to handwritten signatures. Nevertheless, existing signature-creation tools have crucial drawbacks with regard to usability and applicability. To solve these problems, we define appropriate requirements for signature-creation tools to be used in e-Government processes. Taking care of these requirements we propose a modular architecture for adaptable signature-creation tools. Following a user-centered design process we present a concrete implementation of the architecture based upon the Austrian Citizen Card. This implementation has been used to prove the applicability of the architecture in real life. Our tool has been successfully tested and has been assessed as usable and intuitive. It has already been officially released and is widely used in productive environments.

Keywords: Electronic Signatures, Qualified Signature, Signature-Creation, Usability, User-Centered Design.

1 Introduction

Electronic services have gained importance in the last years. Compared to conventional services they allow cost reduction and more efficient procedures. An increasing number of electronic services are being provided in all e-Business domains. For security and privacy sensitive services such as e-Government, electronic signatures guarantee authenticity and integrity.

Especially in the e-Government sector the legal aspects of electronic signatures play a major role. In 1999, the European Commission published the EU Signature Directive [1]. The Directive had to be implemented by national laws and defines equivalence between a handwritten signature and an electronic signature fulfilling certain security requirements (‘qualified signature’).

The European Commission Decision 2011/130/EU [2] defines standard signature formats for advanced electronic signatures. In addition, the Digital Agenda for Europe

M.A. Wimmer, M. Janssen, and H.J. Scholl (Eds.): EGOV 2013, LNCS 8074, pp. 274–285, 2013.

[4] and the e-Government action plan [5] aims to create a single digital market for Europe. Obviously, these activities demand for appropriate signature tools.

Currently a variety of signature-creation tools and applications are on the market. Unfortunately most of them lack usability or applicability. Either they do not support 'qualified signatures' or all standard formats, or they are available as online tools only. Nevertheless, many citizens and companies want or have to use an offline tool due to security and privacy obligations. Therefore there is a need for an offline tool creating 'qualified signatures'. In addition, current signature-creation tools do not allow to freely position a visual representation of the signature in the document. To fill this gap our paper presents a modular and adaptable architecture for signature-creation tools. In addition – to validate the applicability of our proposed architecture – we present a concrete and user-oriented implementation of the architecture based on the Austrian Citizen Card. The main reasons for choosing the Austrian Citizen Card as a basis are: (a) electronic signatures are widely used in Austria and thus we expect a high volume of users and (b) the Austrian official signature as introduced by Leitold et al. [12] defines a visual representation of the signature and therefore an adequate positioning of this representation is needed.

The remainder of this paper is structured as follows: Section 2 gives an overview of the legal and technical framework our solution is based on. In Section 3 we elaborate on requirements for adaptable and secure signature-creation tools and applications. Section 4 presents our modular architecture for signature-creation tools. In addition, details about the implementation of this architecture are given. Section 5 describes the user-centered design method we followed to achieve a high grade of usability of our solution. Finally, we draw conclusions and discuss future work.

2 Legal and Technical Framework

2.1 Legal Regulations

The Digital Agenda for Europe aims to "*develop a digital single market in order to generate smart, sustainable and inclusive growth in Europe*" [6]. To achieve this objective, (cross-border) electronic services are one of the key enabling factors. This has been refined in the e-Government action plan for the period 2011-2105 [5]. The action plan objective is to create a new generation of administrative services. However, electronic signatures are necessary to provide secure and reliable electronic services.

Electronic signatures have been discerned as a key factor for successful e-Government early on. Already in 1999, the European Commission published the Directive on a Community framework for electronic signatures[1] [1]. The Directive defines a basis for legal recognition of electronic signatures. It includes a definition of different characteristics of electronic signatures and defines their legal effect. In

[1] Better known as the EU Signature Directive.

particular, it defines that an advanced electronic signature must meet the following requirements:

> *"(a) it is uniquely linked to the signatory;*
> *(b) it is capable of identifying the signatory;*
> *(c) it is created using means that the signatory can maintain under his sole control; and*
> *(d) it is linked to the data to which it relates in such a manner that any subsequent change of the data is detectable;"* [1]

In addition, Article 5 of the Directive defines that *"advanced electronic signatures which are based on a qualified certificate and which are created by a secure-signature-creation device […] satisfy the legal requirements of a signature in relation to data in electronic form in the same manner as a handwritten signature satisfies those requirements in relation to paper-based data"*[2] [1]. This means that such 'qualified signatures'[3] are legally equivalent to handwritten signatures which is a common precondition for e-Government processes.

2.2 Technical Background

From a technical perspective we concentrate on the Austrian Citizen Card concept [10] as the implementation of our solution is based on it. This concept defines the Citizen Card as a technology neutral instrument that enables to create and verify electronic signatures according to the Austrian e-Government act [8] and e-Signature law [9][4]. That means different forms of Citizen Card tokens can exist. Currently, smart card-based as well as mobile phone-based Citizen Card implementations are available.

To integrate these various tokens a middleware is used. This *Citizen Card Software (CCS)* implements a high level interface[5] that provides diverse functionality such as the creation and verification of electronic signatures. Different types of this citizen card software exist:

- *Online-based CCS*: This smart card-based CCS runs on the server side and provides the desired functionality via a Java applet to the user. Actually, the only available online based CCS is MOCCA Online[6].

[2] The terms 'qualified certificate' and 'secure signature creation' and their requirements are defined in Article 2 of the Signature Directive.
[3] The term 'qualified signature' is not explicitly defined in the Signature Directive. However, this term is usually used in literature.
[4] The Citizen Card offers additional functionality, such as identification of citizens and data encryption. However, these are not needed for our use cases.
[5] The so-called 'Security Layer '
[6] MOCCA Online: `http://joinup.ec.europa.eu/software/mocca/description`

- *Local/Client-based CCS*: This CCS is also smart card-based and has to be installed locally on the client machine. Here, different implementation exists, e.g. MOCCA Local[7], a.sign Client[8] or TrustDesk[9].
- *Mobile phone signature-based CCS*: This CCS which uses a simple mobile phone is available at https://www.handy-signatur.at/. It is based upon a two factor authentication using a password and a TAN (sent via SMS to the mobile phone).

Concerning signature formats, the European Commission, in their Decision 2011/130/EU [2] from 2011, published a set of standard signature formats which must be processable by all competent authorities acting under the EU Services Directive [3]. Namely these formats are the advanced electronic signatures CAdES, XAdES, and PAdES. However, Austria has rolled out a proprietary PDF-based signature format (PDF-AS) several years ago [11,12]. This format is going to be replaced by PAdES, but currently it is still widely used. Therefore, we have chosen this signature format to implement in our signature tool (see Section 4 for details).

3 Requirements

The secure and reliable signature-creation of electronic documents plays a central role in most e-government solutions. Signature-creation tools must meet several requirements to satisfy legal regulations as well as the needs of all user groups. On the one hand, the signature-creation tools must fulfill the requirements for the public sector and organizations. On the other hand, the tools should be intuitive and convenient to use for every single citizen. Considering the needs of all user groups, reliability, usability, adaptability, and modularity are identified as core requirements for signature-creation tools. These requirements are refined as follows:

- **Reliability and Privacy**

Signature-creation tools typically process sensitive personal and business data. Misuse of this data may seriously compromise citizens and businesses. Hence, reliability and trustworthiness of this data is an essential requirement. In addition, the public administration needs certainty about the identity of the citizens or businesses. The same applies for the identity of the public administration. So, reliability of the affected parties must be achieved. Finally, citizens, businesses, and public administration need assurance that the data processing satisfies legal and privacy regulations.

- **Usability**

Usability is another major requirement for signature-creation tools. Signature-creation tools are using cryptographic techniques like public key infrastructure (PKI) or secure signature creation devices (SSCD) such as smart cards as required for generating 'qualified signatures'. Most likely, users do not have the necessary background

[7] MOCCA Local: http://joinup.ec.europa.eu/software/mocca/description
[8] http://www.a-trust.at/
[9] http://www.itsolution.at/digitale-signatur-produkte/desktop/trustDesk.html

knowledge about complex cryptographic concepts and legal regulations. Plenty of security-sensitive tools are simply too complex for most users. In general, users are not interested in technical details. To improve the usability of signature-creation tools, this complexity must be hidden from the user. Instead, the focus has to be on presenting important information to users. To ensure usability, identified user groups must be involved in the design and development process of such tools.

- **Comprehensive Format Support**

In the next years a significant increase of electronic signature enabled cross-border services is expected (see Digital Agenda for Europe [6] or EU Services Directive [3] for instance). Although the European Commission Decision 2011/130/EU [2] has defined standard signature formats, various other (partly proprietary and nation-wide) formats are still in use. This implies that the support of these signature formats is still required. Hence, the ability to enhance signature-creation tools to support additional signature formats is crucial. Obviously these enhancements should be possible with minimal effort.

- **Cross-Platform Applicability**

Usually, e-Government applications and services must not be limited to specific hardware or software components. Services provided by public authorities must be accessible for all citizens without any restrictions and irrespective of the used environment. Thus, the availability of cross-platform applications is an essential requirement for signature-creation tools.

- **Offline Availability**

In many cases electronic documents contain personal or sensitive data. Therefore document owners are interested in keeping this data undisclosed, either due to privacy regulations, business policies or because of other privacy reasons. Server-based approaches are problematic in this context, because users do not want to upload sensible data to a remote server. Therefore, signature creation tools should offer a client-based implementation for creating electronic signatures.

4 Architectural Design

In this section we elaborate on a modular architecture and design for signature-creation tools satisfying the identified requirements. To verify the applicability of this architecture we have implemented a signature-creation tool for use cases of the Austrian e-Government. Due to the widespread usage of the Austrian signature format PDF-AS we have given this format priority. The following subsections describe the proposed architecture and give details on the implementation.

4.1 Architecture

Fig. 1 illustrates our proposal for a modular and adaptable architecture for signature-creation tools. The architecture supports various document formats, allows for the

creation of different signature formats and makes use of different signature-creation devices. This modular approach is achieved by defining a generic signature-creation process. Depending on the current state of the process, specific implementations of the various components are used to create a signature for the current document. All those generic components are adaptable and open for further implementations and extensions to support new document types, signature formats, or signature-creation devices. Subsequently we describe our architecture and the involved components or modules (see Fig. 1):

- **Input**

The input module reads a given document and determines the MIME type[10] for further processing. It generates a document dependent state which is used during the whole signature-creation process. This module can support local files, network files or even streams, and presents this input data in a common form to the other modules. When the input module has finished its task, the state is handed over to the viewer module.

- **Viewer**

The viewer module enables presentation of the document to be signed. It uses document-specific implementations for the presentation. These may be e.g. PDF renderer, MS-Word renderer, XML renderer, HTML renderer, and so on.
Depending on the used signature format, a visual signature representation and a customized signature positioning can be supported. In this case the viewer module provides a Positioning component which presents an overlay to allow the user to position the visual signature representation. The chosen position is then stored in the state of the signature process.

- **Signer**

The signer module is responsible for the delegation between the signature component adapter and the signature creation device component. Depending on the state, the signer component chooses an appropriate signature component for the given document, or uses a preconfigured component for the given document class. It provides the chosen signature component adapter with a specific instance of a matching signature creation device, which again is either chosen on the fly or may be preconfigured.

- **Signature Component Adapter**

This adapter is used to provide a common interface to e.g. a signature library. The signature format implementation generates the signature data and uses an abstract signature-creation device to obtain a valid signature for this signature data. Given the signature and the signer certificate the concrete signature component is able to create a valid digitally signed document. This signed document is again stored in the process state.

[10] The MIME type defines the document format.

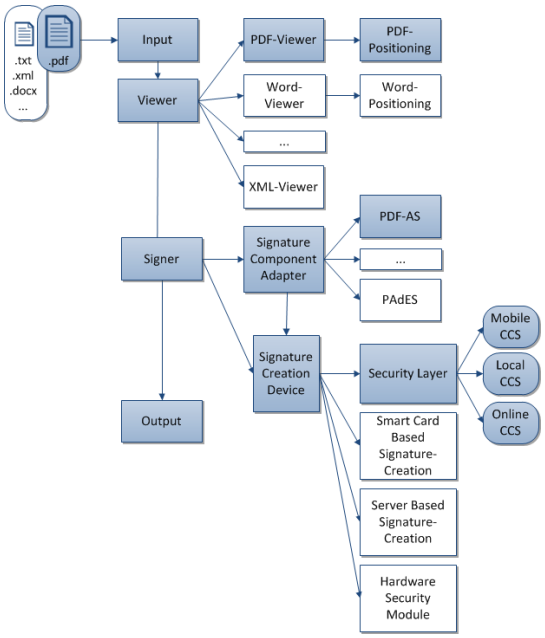


Fig. 1. Modular and adaptable architecture for signature-creation tools

- **Signature Creation Device**

The signature creation device is an abstraction layer for signature-creation. It translates the given requests to implementation specific commands to create a signature. The specific implementations can support any kind of signature-creation device, e.g. smart card [11] -based or server-based signature-creation devices, or hardware security modules. In addition, it supports Austrian citizen card software, which is integrated via the standardized interface 'Security Layer'. Thus, all citizen card software implementations (online, local and mobile phone-based) are supported.

- **Output**

When a signed document is available within the process state, the output module allows the user to save the signed document, to open it with the default application or to view it again with the appropriate viewer module.

4.2 Implementation

To put the proposed architecture into practice and to verify its applicability, a well-defined subset of this architecture has been implemented: signing of PDF documents with the Austrian PDF-based signature format PDF-AS was chosen. Our implementation is based on Java, thus achieving platform independence. Fig. 1

[11] Using the PC/SC (Personal Computer/Smart Card) interface.

highlights the modules that have been implemented in our application. Namely these main modules are:

- PDF-Viewer module including positioning of the visual signature representation
- Signature Component Adapter PDF-AS
- Signature-Creation Devices based on the Austrian Citizen Card via 'Security Layer'

The process flow starts with the input component, which allows the user to select a PDF document to sign, either via drag and drop, or via an operating system file selection dialog. The viewer displays the PDF document and enables the user to position the visual signature representation. This step can be skipped if the user configured the application for automatic signature positioning. The signer component receives the document to be signed and the desired position of the signature block. With this information, a signature request for the citizen card software is built by the signature component. Here, the user chooses the concrete implementation of the signature creation software (online, local or mobile phone-based implementation). Subsequently, the signature request is signed using the selected citizen card software. Finally, this signature is incorporated into the PDF document by the PDF-AS signature component and the thus signed document is sent to the output component. Within the output component the user is able to save and open the signed PDF document.

The user interface is based on this linear process flow and guides the user through the necessary steps. Fig. 4 shows a screenshot of this interface. Depending on the configuration of the tool, certain process steps can be shortened or entirely skipped for daily use by advanced users. For instance, the document to be signed can be selected by dropping it on the program icon, the signature block can be positioned automatically, the citizen card software can be preselected, or the output filename or folder can be set in advance.

Our tool called *PDF-Over* has been officially launched in Austria[12] and is already widely used[13]. As we followed a user-centered design method for the implementation, the tool has been assessed as easily understandable and usable as well as intuitive. The following section gives detailed insights into this design methodology as applied to PDF-Over.

5 User-Centered Design Method

To fulfill the usability requirements of signature-creation tools discerned above, we followed the user-centered design (UCD) principles [14] in order to implement a security-sensitive application that is effective and usable in practice. UCD is a design methodology that at each stage of the design process focuses on user's needs, goals,

[12] PDF-Over, Version 4.0.0, 15.1.2013, `http://www.buergerkarte.at/pdf-signatur.de.php`

[13] Since the official launch about 2.000 users per month are gained.

preferences, and limitations. It is an iterative design process that requires continuous user feedback and tests. As shown in Fig. 2 the methodology consists of four design stages: analysis, design, implementation and validation. The method enables a complete remodel and rethinking of the design by early testing of conceptual models and design ideas. For the development of PDF-Over we have defined to repeat the entire design process three times[14] before launching an official release. The different stages in the creation of PDF-Over were:

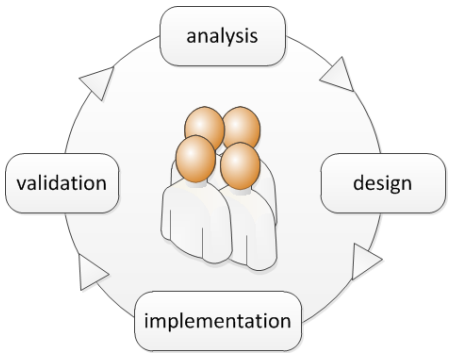


Fig. 2. Four design phases of User-Centered Design Process

- **Analysis**

At the beginning of the process we identified the end-users of PDF-Over. It turned out that the user groups of the signature-creation tool are citizens and authorities. In both groups users can again be divided into standard users and advanced users. After identifying those user groups we posed the question what each user group's main tasks and goals are and what functions are needed to accomplish those. The use case for citizens as standard users is to electronically sign a PDF document. They expect a simple und useable interface without any complexity. The authorities as standard users are interested in applying official signatures. To fulfill the Austrian official signature as introduced by Leitold et al. [12] certain criteria must be met, such as the placement of the visual signature representation. Additionally, advanced users need the possibility to e.g. pre-selected citizen card software, or enable automatic positioning of the visual signature representation. We also analyzed user's need of previous knowledge. In our case the end-user must know what the Austrian Citizen Card is and how to use it.

- **Design**

The second step in the iteration process is the design process. First, paper-based prototypes (see Fig. 3) and the initial architectural design were created. The focus on

[14] This is a common approach for most developments as indicated in http://www.nngroup.com/articles/iterative-design/

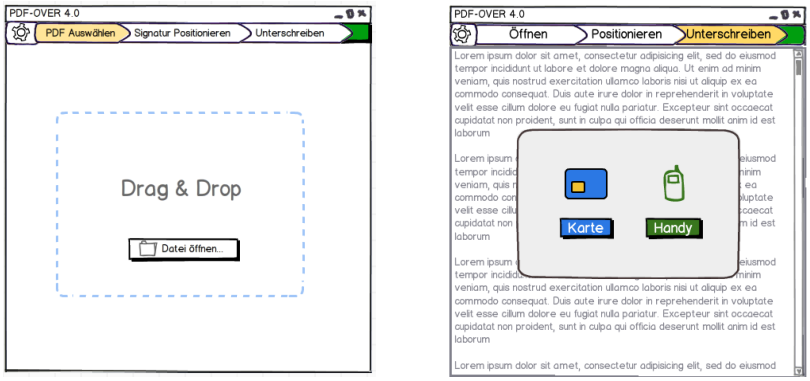


Fig. 3. Design prototypes

the end-users is very important in the early phase of the design. In order to get feedback from the users before writing code or beginning with the development, we performed usability tests with paper mockups.

- **Implementation**

In the implementation stage the detailed design and specifications were implemented and first source code was written. This stage builds upon the results of all prior stages. End-users were not directly involved during the implementation. Fig. 4 illustrates a first implementation of the tool, showing the positioning of the visual signature representation.

- **Validation**

After the implementation phase two approved usability methods have been applied to evaluate PDF-Over. First of all, an expert review was conducted. Here, an evaluator used the tool and assessed its usability against a set of usability principles, the so-called heuristics[15]. The heuristic evaluation provided quick and inexpensive feedback to the design. In the following implementation iteration the results from the heuristic evaluation were implemented.

In the last iteration, we performed a thinking-aloud test with five representative end-users. As indicated by Nielsen [13], five test users are sufficient to find almost all usability problems one would find using many more test participants. Test users have been asked to do representative tasks, while observers, including the developers, watched the test and took notes. The obtained results were analyzed and implemented in the last iteration. With the conducted usability analysis we improved the acceptability and usability of PDF-Over.

[15] http://www.nngroup.com/articles/ten-usability-heuristics/

Fig. 4. PDF-Over free positioning of the visual signature representation

6 Conclusions

Signature-creation is essential for many e-Government processes. Especially the creation of 'qualified signatures' is of high importance. In this paper we have presented a modular architecture for adaptable signature-creation tools. To prove the practical applicability and flexibility, we have given a concrete implementation of this architecture. To achieve a high impact our solution is based on the Austrian Citizen Card concept. We have followed a user-centered design to achieve a high usability of our tool. This tool has been successfully tested and is ready to accept current and upcoming challenges. The tool has already been officially launched in Austria and is licensed under the European Public Licence EUPL [7]. The current number of downloads amounts to about 2000 per month which confirms the high acceptance and usability of our solution.

Currently, we are integrating additional signature formats. Based upon the European Commission Decision 2011/130/EU we are implementing a PAdES signature component adapter to support PDF advanced electronic signatures. In addition, we are working on the support of batch signatures to allow signing of several documents in one step.

References

1. European Union, Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. Official Journal L 13, 12–20 (January 19, 2000)
2. European Commission Decision, Establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market, notified under document C(2011) 1081, 2011/130/EU (February 25, 2011)
3. European Union, Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market. Official Journal L 376, 36–68 (December 27, 2006)
4. European Commission, Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions A Digital Agenda for Europe, COM/2010/0245 (May 19, 2010)
5. European Commission, The European eGovernment Action Plan 2011-2015 Harnessing ICT to promote smart, sustainable & innovative Government, COM/2010/743 (December 15, 2010)
6. European Union, Digital Agenda for Europe, Summaries of EU Legislation, http://europa.eu/legislation_summaries/information_society/strategies/si0016_en.htm
7. European Community, European Union Public Licence, EUPL v.1.1 (2007), http://joinup.ec.europa.eu/software/page/eupl/licence-eupl
8. The Austrian E-Government Act: Federal Act on Provisions Facilitating Electronic Communications with Public Bodies. Austrian Federal Law Gazette, part I, Nr. 10/2004; last amended part I, Nr. 111/2010
9. The Austrian Signature Law: Federal Electronic Signature Law. Austrian Federal Law Gazette, part I, Nr. 190/1999; last amended part I, Nr. 75/2010
10. Leitold, H., Hollosi, A., Posch, R.: Security Architecture of the Austrian Citizen Card Concept. In: Proceedings of ACSAC 2002, pp. 391–400 (2002)
11. Leitold, H., Posch, R., Rössler, T.: Reconstruction of electronic signatures from eDocument printouts. Computers and Security 29(5), 523–532 (2010)
12. Leitold, H., Posch, R., Rössler, T.: Media-Break Resistant eSignatures in eGovernment: An Austrian Experience. In: Gritzalis, D., Lopez, J. (eds.) SEC 2009. IFIP AICT, vol. 297, pp. 109–118. Springer, Heidelberg (2009)
13. Hinderer, S.D., Nielsen, J.: How to Recruit Participants for Usability Studies. Nielsen Norman Group (2003), http://www.nngroup.com/reports/tips/recruiting
14. International Organization for Standardization, ISO 9241-210:2010, Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems

5 Towards User-Friendly e-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques

Conference	EGOVIS/EDEM
Language	English
Title	Towards User-Friendly e-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques
Authors	Thomas Zefferer, Vesna Krnjic
Publisher	Springer-Verlag Berlin Heidelberg
Booktitle	Advancing Democracy, Government and Governance

Towards User-Friendly e-Government Solutions: Usability Evaluation of Austrian Smart-Card Integration Techniques

Thomas Zefferer and Vesna Krnjic

e-Government Innovation Center Austria,
Inffeldgasse 16a, 8010 Graz, Austria
{thomas.zefferer,vesna.krnjic}@egiz.gv.at
http://www.egiz.gv.at

Abstract. Security and usability are key requirements of e-Government solutions. Security requirements are often met by reliance on smart card technology. In Austria, the open source components *MOCCA Local* and *MOCCA Online* facilitate the integration of smart cards into national e-Government applications. While MOCCA Local and MOCCA Online guarantee security, their capability to meet given usability requirements has not been assessed so far.

To bridge this gap, the usability of MOCCA Local and MOCCA Online has been evaluated by means of a usability test. The obtained results show that MOCCA Local and MOCCA Online basically meet given usability requirements. Still, some minor issues have been identified that threaten to reduce the usability of these components.

In this paper we introduce the basic architecture of MOCCA Local and MOCCA Online and present results of the conducted usability test. Based on these results we identify existing usability issues and derive possible improvements.

Keywords: E-Government, Smart cards, Usability, MOCCA.

1 Introduction

Secure authentication of citizens and creation of electronic signatures are common requirements of e-Government applications. These requirements are perfectly met by smart cards as they support the secure storage of authentication information and can be used as *Secure Signature Creation Devices (SSCD)*. This way, smart cards fulfill the requirements of qualified electronic signatures according to the Directive 1999/93/EC of the European Parliament and of the Council on a Community framework for electronic signatures [1].

Besides their security enhancing features, their wide acceptance is another key advantage of smart cards. Bank institutes have recognized the potential of smart card technology early and nowadays provide customers with multifunctional bank account cards. In various countries, smart cards are also used in the health sector. For instance, in Austria citizens are supplied with health

A. Kő et al. (Eds.): EGOVIS/EDEM 2012, LNCS 7452, pp. 88–102, 2012.

insurance cards that facilitate use and charging of public health services. Due to their various fields of application, smart cards can nowadays be regarded as well accepted and approved technology.

Its wide acceptance and its capability to fulfill given security requirements make smart card technology perfectly suitable for e-Government solutions. It is thus less astonishing that e-Government strategies of various European countries foresee the use of smart cards. In Austria, smart cards are used in e-Government applications as they are able to meet given legal requirements such as the Austrian Signature Act [2] and the Austrian e-Government Act [3]. Also in numerous other European countries smart cards are an integral part of current e-Government solutions. A comprehensive overview of national eID solutions is given in [13].

Unfortunately, the use of smart cards in e-Government applications also raises various challenges for citizens, governments, and application developers. For governments, the implementation of appropriate card roll-out and personalization processes is a serious challenge as this usually involves significant organizational and financial efforts. For citizens, the need for appropriate card reader devices often represents a serious barrier as off-the-shelf PCs and laptops do usually not support this functionality by default. The integration of smart card technology into e-Government applications also raises several technological challenges for application developers. For instance, in[4], Orthacker et al. have discussed accessibility challenges that arise with the use of smart cards in e-Government applications. So far, less attention has actually been paid to usability aspects of smart card based solutions. Although smart card vendors often advertise the usability of their products, there is still a lack of scientific research on this topic.

Nevertheless, usability is a crucial factor that heavily influences user acceptance. Since e-Government applications allow for more efficient administrative and governmental proceedings, a high degree of user acceptance is desirable to improve efficiency and to save costs. The requirement for user acceptance directly leads to the demand for an appropriate level of usability in e-Government solutions. Significant effort has already been made to optimize the usability of Web based e-Government solutions. Related work has been discussed in [5] and [6]. At the same time, less attention has been paid to the usability of different approaches to integrate smart cards into these applications. We filled this gap by conducting a comprehensive usability analysis of established smart card integration methods of the Austrian e-Government infrastructure. In this paper we present results of the conducted usability analysis and propose several enhancements that can help to improve the usability of existing smart card integration approaches.

This paper is structured as follows. In Section 2 we discuss relevant concepts of the Austrian e-Government infrastructure and introduce MOCCA Local and MOCCA Online in more detail. We provide details of the methodology that has been followed to evaluate the usability of these components in Section 3. Results of the conducted usability test are presented in Section 4 and discussed in Section 5. Finally, conclusions are drawn.

2 Smart-Card Integration: The Austrian Approach

The integration of smart cards is a serious technological challenge for developers of e-Government applications. In this section we introduce different approaches that are followed in Austria to overcome this challenge. Usability aspects of these approaches will be discussed later in this paper.

Austrian e-Government solutions are based on the so called *Citizen Card* concept. The Citizen Card represents a token that allows citizens to authenticate at remote services and to create qualified electronic signatures according to the EU Signature Directive. Although the term Citizen Card might suggest the use of smart cards, the Citizen Card concept is actually technology-independent and can also be applied to e.g. mobile phones[1]. Despite of its technology-neutral nature, smart cards still play an important role in the Austrian Citizen Card concept. Currently, Citizen Card implementations that rely on health insurance cards, bank account cards, or special signature cards are available in Austria.

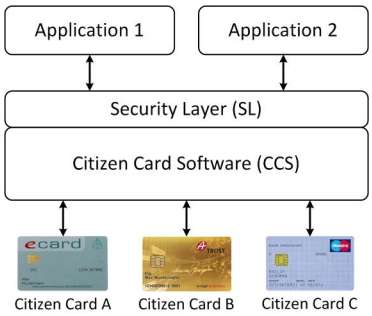


Fig. 1. The Security Layer provides e-Government applications a common interface to different Citizen Card implementations

The support of different cards is beneficial for citizens as they can use their preferred card type. However, this flexibility significantly complicates the integration of Citizen Card functionality into e-Government applications, as support for each card type has to be implemented separately. With a growing number of Citizen Card implementations (i.e. smart card types), development of new and maintenance of existing applications increase complexity and cause additional costs.

To overcome this problem, the Austrian Citizen Card concept follows a middleware approach and defines the so called *Security Layer (SL)* interface as shown in Fig. 1. The Security Layer has been introduced in [7] and represents a common XML based interface between e-Government applications and different Citizen Card implementations. The Security Layer interface is implemented by the so called *Citizen Card Software (CCS)*. This software provides access to all Citizen Card implementations and makes their functionality available to e-Government

[1] A Citizen Card implementation using mobile phones has been introduced in [8].

applications through the common SL interface. This way, application developers are released from integrating and maintaining different smart card types, as this task is outsourced to and implemented by the CCS.

The middleware approach shown in Fig. 2 raises the question about possible implementation variants for the CCS. Following the most obvious approach, the CCS is often implemented as software running on the user's local system (cf. Fig. 2). This way, the CCS has access to locally connected smart cards through the system's PC/SC interface. Following this approach, the SL interface is provided by the CCS through a local network port. This way, also Web based e-Government applications can easily access the SL interface through the user's Web browser. Since all specifications of the SL interface are open and publicly available, there are already various CCS implementations from different vendors available on the market. Some of these solutions such as the A-Trust a-sign client [9] are for free, others charge a license fee. The only open source CCS available in Austria so far is called *MOCCA Local* and has been introduced in [10]. MOCCA Local represents one of the main outcomes of the MOCCA (Modular Open Citizen Card Architecture) project [11] that aims to provide Austrian citizens with Java based open source CCS implementations. Fig. 3 shows MOCCA Local's GUI that allows users to review data to be signed and to confirm it by pressing the *Sign* button.

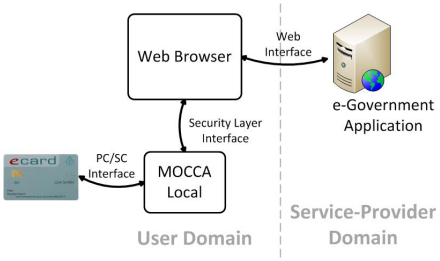


Fig. 2. MOCCA Local is a software that runs on the user's local system and acts as intermediary between the Web browser and locally connected smart cards

Fig. 3. The GUI of MOCCA Local allows users to review data to be signed and to start the signature creation process

All solutions mentioned above including MOCCA Local follow the approach shown in Fig. 2 and make use of software running on the user's local system. While this approach works fine from a functional point of view, it raises several problems for citizens. Especially inexperienced users sometimes have problems to carry out software installations on their own. In some cases, users do not even have the privileges to install software on the computer they are currently using. To overcome these issues, a minimal footprint CCS implementation has been developed in the course of the MOCCA project. According to this minimal footprint approach, users are not required to install software on their local

system. To underline the installation-free nature of this approach, this solution has been named *MOCCA Online*. Similar to MOCCA Local, MOCCA Online is based on the Java framework. Java has been chosen as underlying technology for all MOCCA components in order to achieve platform independence. Reliance on Java requires a current Java Runtime Environment (JRE) to be installed on the user's local system.

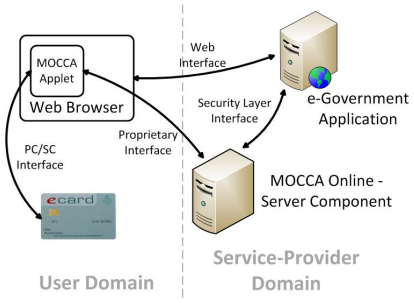


Fig. 4. MOCCA Online follows a distributed approach and consists of a server component and a Java Applet

Fig. 5. The GUI provided by MOCCA Online's Java Applet allows users to review data to be signed and to start the signature creation process

MOCCA Online has been introduced and discussed by Centner et al. in [10]. The basic architecture of MOCCA Online is shown in Fig. 4. The main idea behind MOCCA Online is to split the entire CCS functionality into two core components. The first component runs on a server and implements the SL interface. E-Government applications communicate with this server component to access smart card functionality. A Java Applet represents the second core component of MOCCA Online. The Java Applet runs in the user's Web browser and implements smart card communication based on the PC/SC protocol. Furthermore, the Applet provides a GUI through which users can review and confirm the data to be signed. The Applet's GUI is shown in Fig. 5 and has been designed similar to the GUI provided by MOCCA Local (cf. Fig. 3). This way, a similar user experience is achieved irrespective of the used MOCCA variant. Typically, the Applet is integrated into Web based e-Government applications by means of a HTML IFRAME tag. The two components of MOCCA Online, i.e. the server component and the Java Applet, communicate with each other over a proprietary interface.

Currently, MOCCA Local and MOCCA Online are among the predominating CCS implementations in Austria. During the past few years, much effort has been invested to assure and improve the security of these components. Less attention has been paid to usability aspects so far. To bridge this gap, we have conducted an extensive usability analysis of MOCCA Local and MOCCA Online in order to identify usability problems and to further improve the user acceptance of these components. In the following section we discuss the applied methodology of the conducted usability test.

3 Methodology

Approved usability evaluation methods have been applied to analyze the usability of MOCCA Local and MOCCA Online. This section defines a set of research questions and discusses the followed methodology to answer them by means of the conducted usability test.

3.1 Research Questions

The following research questions have been defined in order to cover all relevant usability aspects of the evaluated components.

Q1 Does reliance on Java technology cause additional usability problems?
Q2 What are the main usability problems that have been encountered during the installation of MOCCA Local and which user groups are especially affected?
Q3 Once MOCCA Local is correctly installed, what are the main usability problems that have been encountered during the usage of MOCCA Local and which user groups are especially affected?
Q4 What are the main usability problems that have been encountered during the usage of MOCCA Online and which user groups are especially affected?
Q5 Which MOCCA variant appears more secure and trustworthy to users and are there significant differences between different user groups?

3.2 Test Method and Setup

To find answers to the predefined research questions, a thinking-aloud test has been conducted. A Thinking-aloud test is an approved method to evaluate the usability of software products or websites and has been discussed by Nielsen in [12]. In a thinking-aloud test, test users are asked to carry out a set of well-defined tasks with the software to be evaluated and to articulate their thoughts during the test run. This way, users' interactions with the software under test can be observed and valuable user feedback can be collected. Since the users' emotional state can also be of interest, test users are usually recorded with a camera during the test.

Thinking-aloud tests typically consist of two phases. During the *test phase*, test users are asked to carry out predefined tasks. In the subsequent *analysis phase*, data recorded and collected during the test phase is analyzed in order to identify common usability problems and to find answers to predefined research questions.

We have used special software during both phases. During the test phase, the used software assisted in recording and collecting relevant data by tracking test users' mouse movements and keyboard inputs. Furthermore, the used software has automatically related additionally recorded video and sound data

to the tracked user inputs. During the test phase, additional user feedback has been collected by means of different questionnaires and a conclusive interview. Recorded user sessions and collected user feedback have been analyzed in the subsequent analysis phase. Again, the used software has facilitated an efficient analysis of the collected data sets and the application of statistical methods.

All tests have been carried out on an off-the-shelf PC with Microsoft Windows 7 operating system. Test users were asked to use the Microsoft Internet Explorer 8 Web browser. We have chosen this system configuration as it represents a common OS/Web browser combination. To facilitate a systematic analysis of the collected data and to ease comparisons between different test users, we did not give test users the opportunity to choose their preferred operating system or Web browser configuration.

3.3 Test Users and User Groups

In total, 20 test users participated in the conducted usability test. In order to achieve convincing and sound results, test users have been chosen in a way that they approximately form a representative sample of the Austrian society. All test users have been asked to carry out the following three tasks using their personal smart card based Citizen Card.

- **T1** Install MOCCA Local using a provided Java Web Start based installing routine.
- **T2** Use MOCCA Local to carry out a given e-Government procedure including identification and signature creation.
- **T3** Use MOCCA Online to carry out a given e-Government procedure including identification and signature creation.

To avoid the influencing of results by learning effects, test users were split into two groups. Group A was asked to carry out the tasks in the order given above and hence started with installation and usage of MOCCA Local. Contrary, Group B was asked to carry out task T3 first, followed by task T1 and task T2. Thus, test users of Group B started with an evaluation of MOCCA Online first.

Both MOCCA Local and MOCCA Online require a current Java Runtime Environment (JRE) to be installed on the client system. As we were also interested in the usability of the Java installation process and its integration in the evaluated MOCCA components, the test system was provided without an installed JRE. Test users were requested to install the required JRE during the test. Depending on their assigned group, Java had to be installed either during the installation of MOCCA Local or during the first usage of MOCCA Online. This way, we were able to compare the usability of the Java installation processes of MOCCA Local and MOCCA Online.

The assignment of test users to Group A and Group B was completely random. Additionally, all test users have been subdivided into different user groups according to different characteristics. This way, we were able to assess the impact

of given usability flaws on different user groups. The following table summarizes user groups that have been analyzed separately. Details of the obtained results are discussed in the next section.

Table 1. Test users have been classified according to four different characteristics

ID	Description	Users
Group ALL	This group comprises all test users.	20
Group A	Test users of this group started with the evaluation of MOCCA Local.	10
Group B	Test users of this group started with the evaluation of MOCCA Online.	10
Group 30+	Test users of this group are more than 30 years old.	8
Group 30-	Test users of this group are 30 or less years old.	12
Group U	Test users of this group have a university degree.	12
Group NU	Test users of this group have no university degree.	8
Group T	Test users of this group are technicians.	7
Group NT	Test users of this group are no technicians.	13

4 Results

All results provided in this section have been obtained by analyzing data collected during the usability test. The obtained results are presented in the following subsections.

4.1 Installation of the Java Runtime Environment

Both MOCCA Local and MOCCA Online represent Java based solutions. To answer research question Q1, we assessed whether the given dependency on Java raises additional usability issues. Both MOCCA Online and MOCCA Local automatically check for an installed JRE upon start-up. If no JRE can be detected, MOCCA Local automatically redirects the user to the Java installation page provided by Oracle[2]. Similarly, MOCCA Online provides users an appropriate error message containing a link to the Java installation page. Users have to follow this link manually.

Surprisingly, it turned out that without exception all test users were basically aware of Java. When being requested by MOCCA Local or MOCCA Online to install a JRE, 95% of the test users started the Java installation process using the provided link or button. Only 5% did not know what to do in this situation.

[2] `http://www.java.com/en/download/`

After starting the Java installation process, 20% of all test users had problems to successfully complete it. These users mostly successfully downloaded Java but did not realize that the downloaded installer file had to be executed afterwards. User group specific results are illustrated in Fig. 6 and show that the Java installation process was especially problematic for older and non-graduate users. As expected, also test users without technical background were more prone to problems during the Java installation process. Interestingly, users of Group B, i.e. users who had to install Java in the course of using MOCCA Online, had more problems with the Java installation compared to users of Group A.

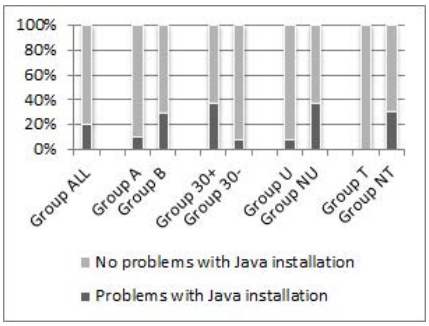


Fig. 6. The installation of Java was problematic especially for users of Group 30+, Group NU, and Group NT

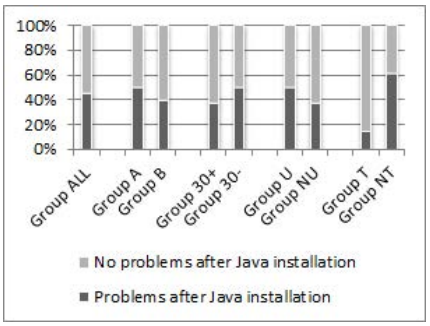


Fig. 7. Especially users of Group NT had problems to proceed after the Java installation process

After completion of the Java installation process, users had to manually restart the Java Webstart based installation process of MOCCA Local or to manually reload the Java Applet of MOCCA Online. It turned out that this was problematic for 45% of all test users. The group specific results illustrated in Figure 7 show that especially technically inexperienced users had problems in this situation. This time, users of Group A were slightly more prone to usability problems. Obviously, after completion of the Java installation process it was more intuitive for users to manually reload the website containing the Java Applet of MOCCA Online than to manually restart the installation procedure of MOCCA Local.

4.2 Installation of MOCCA Local

The installation of MOCCA Local is based on Java Webstart technology. Hence, test users simply had to click a provided button on a website to start the installation process. This was intuitive for 95% of all test users. After completion of the Java Webstart based installation process, a new Web-browser tab was opened automatically. The website shown in this tab asked test users to install a certificate into their Web browser[3]. The certificate to be installed was provided

[3] This certificate is required to establish an appropriate trust status between the Web browser and MOCCA Local.

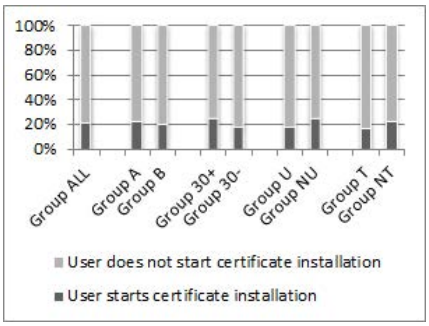


Fig. 8. No significant differences between different user groups could be observed regarding the installation of certificates

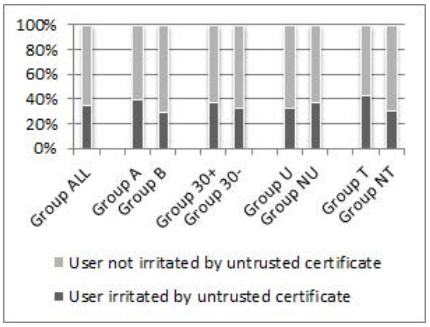


Fig. 9. Users of all user groups were irritated by untrusted certificates

via a link. 20% of all test users just ignored this message and didn't install the certificate at all. Fig. 8 shows that this affected all user groups. 15% of all test users downloaded the certificate but did not install it. Another 10% were not able to complete the certificate installation process on their own.

The certificate to be installed was not recognized as trusted by the used Web browser. Hence, a security warning was shown during the installation process. 35% of all test users felt irritated by this security warning and were not sure whether to proceed with the installation process or not. Fig. 9 shows that again there were only marginal differences between different user groups.

4.3 Card Reader Interaction

MOCCA Local and MOCCA Online support both smart card reader devices with and without integrated PIN pads. Experience has shown that usually devices with integrated PIN pad cause more usability problems. Thus, we have used a Reiner SCT card reader device with integrated PIN pad during the conducted usability test.

It turned out that 25% of all test users had problems to enter the PIN through the integrated PIN pad correctly. In most cases, users didn't realize that entered PINs had to be confirmed using a green OK button. Fig. 10 shows that especially graduated users had problems to enter the PIN correctly. Interestingly, test users starting with the evaluation of MOCCA Local (Group A) were also more prone to problems during the entering of PINs.

A significant learning effect could be observed. Test users, who ran into an timeout because of not confirming the PIN entry by pressing the OK button, did not make the same mistake twice. Already the second PIN entry could be completed successfully by all test users.

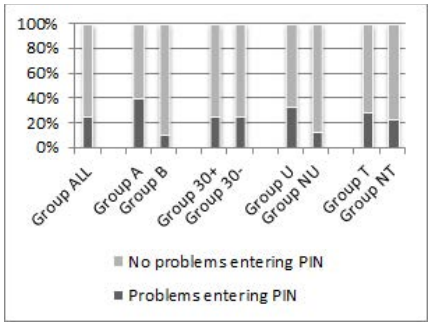


Fig. 10. Users of Group A had significantly more problems to enter the PIN correctly

4.4 Usage of MOCCA Local

In order to test the usability of MOCCA Local, test users were asked to carry out a typical e-Government procedure using their Citizen Card and MOCCA Local. This procedure comprised the reading of identification data from the user's smart card and the electronic signing of an application form.

Usually, when MOCCA Local is requested to access the locally connected smart card, there is a short delay until MOCCA Local starts up its GUI. However, it turned out that only 5% of all test users were irritated by this delay.

The GUI basically serves two purposes. First, it allows users to enter secret PINs if no card reader with integrated PIN pad is used. Furthermore, it allows users to review the data that is about to be signed during a signature creation process. Users can follow a link labeled "Signature Data" in order to open a separate window that finally contains the data to be signed. Interestingly, the conducted usability test revealed that only 40% of all test users were interested in the provided signature data and followed the shown link to inspect them. All other test users just completed the signature process without reviewing the data to be signed. Fig. 11 shows that this affected all user groups. Interestingly, test users with technical background showed most interest in the provided signature data.

In general, all test users were able to carry out the e-Government procedure using MOCCA Local. Severe usability issues did not arise. The usability, security, and trustworthiness of MOCCA Local has also been attested by the test users. Most of them perceived MOCCA Local as secure and trustworthy. Fig. 12 illustrates group specific results.

4.5 Usage of MOCCA Online

Similar to MOCCA Local, the usability of MOCCA Online has been evaluated by requesting test users to carry out a typical e-Government procedure. Again, this procedure comprised the reading of identity data from the user's smart card and the electronic signing of an application form.

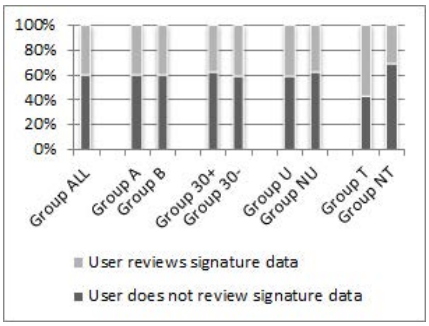


Fig. 11. The majority of all test users was not interested in the data to be signed

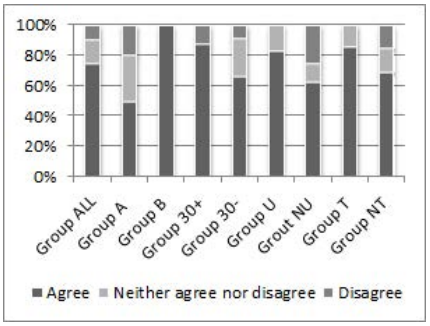


Fig. 12. Most users perceived MOCCA Local as secure and trustworthy

On the client system, a Java Applet represents the key component of MOCCA Online. The Applet implements access to the locally connected smart card and offers the user a GUI. Again, this GUI can be used to enter PINs (if a smart card reader without integrated PIN pad is used) and to access and review data to be signed. Compared to MOCCA Local, the Java Applet usually takes more time to load and to provide the user with the GUI. In total, 20% of all test users were irritated by the delay caused by the Applet loading process.

Since the used Java Applet requires access to local resources (i.e. the smart card), the Applet needs to be signed. For the conducted usability test we used a test instance of MOCCA Online that was signed with a test certificate only. This certificate was not recognized to be trusted by the used Web browser. Hence, during the loading of the Java Applet a security warning was shown. 35% of all test users were irritated by this security warning and considered to cancel the loading process. Fig. 13 illustrates group specific results and shows that especially non-graduated users were irritated by the displayed security warning.

Similar to MOCCA Local, only a small percentage of all test users showed interest in the provided signature data. 80% completed the electronic signing process without verifying the data to be signed. Fig. 14 shows that escpecially non-graduated test users were not interested in the data to be signed.

As shown in Fig. 15, the majority of all test users perceived MOCCA Online as secure and trustworthy. Especially older and well-educated test users rated the security and trustworthiness of MOCCA Online positively.

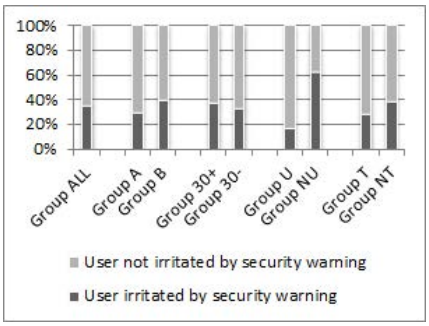


Fig. 13. Especially users of Group NU were irritated by the shown security warning

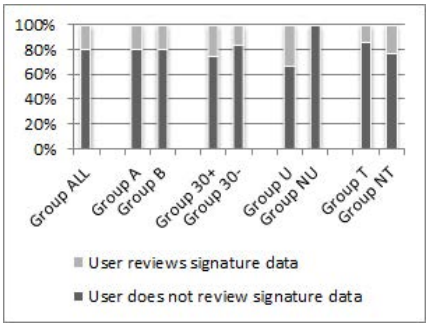


Fig. 14. The majority of all users was not interested in the data to be signed

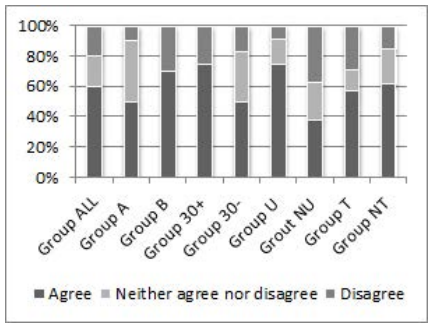


Fig. 15. Most users perceived MOCCA Online as secure and trustworthy

5 Discussion

The conducted usability test yielded various interesting results. In this section we interpret these results to answer the five predefined research questions. Furthermore, we sum up the most relevant lessons learned and derive possible improvements for the evaluated CCS implementations.

To answer research question Q1, we can conclude that reliance on Java technology does not automatically lead to usability problems. All test users were aware of Java and most of them were able to complete the Java installation process successfully. Still, usability problems could be identified regarding the integration of the Java installation process. After completion of the Java installation process, some users did not know how to proceed. This especially applied to technically inexperienced users. In order to overcome this problem and to improve the integration of the Java installation process, users should be provided with more information and guidance during the installation process.

Regarding research question Q2, the conducted usability test revealed that the Java Webstart based installation process of MOCCA Local does not cause severe usability problems. Most users were able to install MOCCA Local without

assistance. However, several users had problems with the subsequent certificate installation that had to be carried out in the used Web browser. Again, this problem can be overcome by providing users with more information and guidance during the installation process. Additionally, used certificates should always be chosen such that their trust status is recognized by common Web browsers. Otherwise, displayed security warnings might irritate users and lead to an abort of the certificate installation process.

The use of MOCCA Local turned out to be unproblematic for users. Minor problems occurred only during the first PIN entry, when users did not know that entered PINs had to be confirmed using the green OK button on the card reader device. To avoid possible errors already during the first use of MOCCA Local, users should be informed appropriately if a PIN confirmation is required. It also turned out that most users did not verify provided signature data before electronically signing them. To solve this issue, the link that has to be followed in order to display the signature data should be placed more prominently in the shown GUI window (cf. Fig.3). Despite these minor issues, we can answer research question Q3 by concluding that MOCCA Local is usable for most users without problems.

Similar results have been obtained for research question Q4. Evaluation of MOCCA Online has shown that additional information about an expected confirmation of PIN entries could improve usability. Similar to MOCCA Local, signature data to be signed was hardly ever reviewed by test users. A more prominent placement and design of the shown link that leads to the signature data (cf. Fig. 5) thus seems reasonable. The conducted usability test has also shown that users are often irritated by displayed security warnings. Hence, it should be guaranteed that the trust status of used signing certificate of the MOCCA Applet is recognized by common Web browsers.

To answer research question Q5, we can conclude that both security and trustworthiness of MOCCA Local and MOCCA Online have been rated positively by most test users. A direct comparison of the results obtained for MOCCA Local and MOCCA Online shows that MOCCA Local has been rated slightly better than MOCCA Online. Interestingly, older and graduated users rated both evaluated CSS implementations better than younger and non-graduated users. According to the obtained results, technicians rated the security and trustworthiness of MOCCA Local higher. For users without technical background MOCCA Online appeared to be more secure and trustworthy. For both CCS implementations, it turned out that the use of untrusted certificates significantly reduces the perceived security and trustworthiness. Hence, it is crucial that CCS implementations rely on certificates that are recognized as trusted by common Web browsers.

6 Conclusions

The conducted usability evaluation of MOCCA Local and MOCCA Online has led to valuable findings. The obtained results show that both MOCCA Local and MOCCA Online basically fulfill given usability requirements. Most test users

were able to successfully install and use the evaluated components without assistance.

Still, some minor usability problems could be identified. Provision of more detailed information and improved guidance through installation routines will probably solve most of the identified issues. Additionally, reliance on certificate being recognized as trusted by common Web browsers is crucial for the perceived security and trustworthiness of MOCCA Local and MOCCA Online.

All obtained results and findings will be incorporated in future releases of the evaluated CCS implementations. This way, the conducted usability test will contribute to the usability of MOCCA Local and MOCCA Online and will help to improve the user acceptance of e-Government applications that rely on these components.

References

1. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. Offcial Journal of the European Communities L 013, 12–20 (2000)
2. Austrian Federal Act on Electronic Signatures. Federal Law Gazette, part I, Nr. 137/2000, last amended by Nr. 59/2008 (2000)
3. Austrian Federal Act on Provisions Facilitating Electronic Communications with Public Bodies. Federal Law Gazette, part I, Nr. 10/2004 (2004)
4. Orthacker, C., Zefferer, T.: Accessibility Challenges in e-Government: an Austrian Experience. In: Proceedings of the Forth International Conference on Internet Technologies and Applications (ITA 2011), pp. 221–228 (2011)
5. Garcia, A.C.B., Maciel, C., Pinto, F.B.: A Quality Inspection Method to Evaluate E-Government Sites. In: Wimmer, M.A., Traunmüller, R., Grönlund, Å., Andersen, K.V. (eds.) EGOV 2005. LNCS, vol. 3591, pp. 198–209. Springer, Heidelberg (2005)
6. Sørum, H.: An Empirical Investigation of User Involvement, Website Quality and Perceived User Satisfaction in eGovernment Environments. In: Andersen, K.N., Francesconi, E., Grönlund, Å., van Engers, T.M. (eds.) EGOVIS 2011. LNCS, vol. 6866, pp. 122–134. Springer, Heidelberg (2011)
7. Leitold, H., Hollosi, A., Posch, R.: Security Architecture of the Austrian Citizen Card Concept. In: Proceedings of the 18th Annual Computer Security Applications Conference, pp. 391–400. IEEE Computer Society, Washington, DC (2002)
8. Orthacker, C., Centner, M., Kittl, C.: Qualified Mobile Server Signature. In: Proceedings of the 25th TC 2011 International Information Security Conference, SEC 2010 (2010)
9. a.sign client (2012), `https://www.a-trust.at/default.aspx?lang=GE&ch=1&node=765`
10. Orthacker, C., Centner, M.: Minimal-Footprint Middleware to Leverage Qualified Electronic Signatures. In: Filipe, J., Cordeiro, J. (eds.) WEBIST 2010. LNBIP, vol. 75, pp. 60–68. Springer, Heidelberg (2011)
11. Modular Open Citizen Card Architecture (2012), `http://mocca.egovlabs.gv.at/`
12. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers (1993)
13. Siddhartha, A.: National e-ID card schemes: A European overview. Information Security Technical Report 13, 46–53 (2008)

6 Usability-Evaluierung der österreichischen Handy-Signatur

Conference	D-A-CH Security 2012
Language	German
Title	Usability-Evaluierung der österreichischen Handy-Signatur
Authors	Thomas Zefferer, Vesna Krnjic
Publisher	—
Booktitle	D-A-CH Security 2012

Usability-Evaluierung der österreichischen Handy-Signatur

Thomas Zefferer · Vesna Krnjic

IAIK – Technische Universität Graz

{thomas.zefferer | vesna.krnjic}@iaik.tugraz.at

Zusammenfassung

Sicherheit ist eine zentrale Anforderung vieler E-Government-Lösungen. Das nötige Maß an Sicherheit wird dabei in der Regel durch den Einsatz von Chipkarten, die an Bürgerinnen und Bürger ausgegeben werden, erfüllt. Diese weisen jedoch eine Reihe von Nachteilen in Bezug auf Benutzerfreundlichkeit auf. Vor allem die Notwendigkeit eines Chipkartenlesegeräts stellte sich in der Vergangenheit oft als Problem heraus. In Österreich wurde aus diesem Grund vor einigen Jahren die Handy-Signatur entwickelt, die eine Alternative zu chipkartenbasierten Ansätzen darstellt. Die Handy-Signatur erlaubt österreichischen Bürgerinnen und Bürgern eine sichere Authentifizierung an E-Government-Anwendungen und die Erstellung qualifizierter elektronischer Signaturen.
Eine Erhöhung der Benutzerfreundlichkeit und in weiterer Folge eine Steigerung der Akzeptanz österreichischer E-Government-Lösungen war ein zentrales Motiv für die Entwicklung der Handy-Signatur. Im Rahmen einer Usability-Studie wurde überprüft, ob dieses Ziel erreicht werden konnte. Zwanzig Testpersonen wurden dazu gebeten, einige typische Abläufe sowohl mit der Handy-Signatur, als auch mit chipkartenbasierten Lösungen durchzuführen. Die erhaltenen Ergebnisse zeigen, dass die Handy-Signatur in Bezug auf Benutzerfreundlichkeit und Akzeptanz chipkartenbasierten Ansätzen durchwegs überlegen ist und von einem Großteil der Testpersonen positiv bewertet wurde. Die durchgeführte Usability-Analyse zeigte damit, dass die österreichische Handy-Signatur sowohl gegebene Sicherheitsanforderungen erfüllt, als auch ein entsprechendes Maß an Benutzerfreundlichkeit und Akzeptanz gewährleistet.

1 Einleitung

Sicherheit spielt in E-Government-Anwendungen eine zentrale Rolle. Der Schutz persönlicher Daten, die im Rahmen von E-Government-Verfahren übertragen und verarbeitet werden, muss jederzeit höchste Priorität haben. Zur geeigneten Absicherung von E-Government-Verfahren kommen daher in der Regel erprobte kryptographische Methoden zur Anwendung. Um Bürgerinnen und Bürger geeignet über das Internet authentifizieren zu können – passwortbasierte Verfahren bieten bekanntenmaßen im Allgemeinen kein ausreichendes Sicherheitsniveau – kommen häufig Hardware-Token, die die Implementierung von Zwei-Faktor-Authentifizierungen ermöglichen, zum Einsatz. In den meisten europäischen Ländern sind für diesen Zweck Chipkarten das Mittel der Wahl.

Die Integration dieser Hardware-Token in zumeist webbasierte E-Government-Anwendungen stellt Diensteanbieter immer wieder vor Herausforderungen. Vor allem wenn zusätzliche Anforderungen wie Plattform- oder Browserunabhängigkeit die Implementierungsalternativen einschränken, müssen zur Integration von Hardware-Token in E-Government-Anwendungen

N. N · N. N (Hrsg.) · DACH Security 2012 · syssec (2012) pp-pp.

mitunter komplexe Lösungswege beschritten werden. In vielen Fällen werden Hardware-Token über entsprechende Middleware-Lösungen in bestehende E-Government-Verfahren integriert. Dieser Ansatz wird unter anderem in Österreich verfolgt und konnte sich aus rein funktionaler Sicht bereits seit vielen Jahren bewähren.

Die österreichische E-Government-Strategie sieht eine größtmögliche Flexibilität in Bezug auf die zu verwendenden Technologien vor. Als gemeinsame Basis dient die Bürgerkarten-Spezifikation [BÜRG12], die abstrakte Anforderungen an zu verwendende Hardware-Token definiert, jedoch keine Einschränkungen bezüglich derer Umsetzung macht. Durch diese Flexibilität entstand in den letzten Jahren in Österreich ein Ökosystem an unterschiedlichen Lösungen. So können österreichische Bürgerinnen und Bürger neben verschiedenen Chipkarten auch ihre Mobiltelefone verwenden, um sich sicher an E-Government-Anwendungen zu authentifizieren und um qualifizierte elektronische Signaturen gemäß EU Signaturrichtlinie zu erstellen.

Bei der Entwicklung all dieser Lösungen hatte verständlicherweise die Erfüllung gegebener Sicherheitsanforderungen oberste Priorität. Für den Erfolg und die Akzeptanz von IT-Lösungen im Allgemeinen und E-Government-Lösungen im Speziellen spielt neben der Sicherheit jedoch auch die Benutzerfreundlichkeit eine entscheidende Rolle. Diese wurde in Vergangenheit jedoch oft nur am Rande beachtet und im Design- und Entwicklungsprozess verschiedenster E-Government-Lösungen nicht geeignet berücksichtigt.

Vor allem die Verwendung von Mobiltelefonen anstelle von Chipkarten als Hardware-Token scheint jedoch ein vielversprechender Schritt in Richtung einer verbesserten Benutzerfreundlichkeit zu sein, entfällt bei der Verwendung von Mobiltelefonen doch die Notwendigkeit eines Chipkartenlesegeräts. Eine genaue Untersuchung der Benutzerfreundlichkeit und Akzeptanz der auf Mobiltelefonen beruhenden Lösung im Vergleich zu etablierten chipkartenbasierten Lösungen wurde bisher jedoch in Österreich noch nicht durchgeführt.

Um diese Lücke zu schließen, wurde im Rahmen einer Usability-Analyse die Benutzerfreundlichkeit und Akzeptanz chipkartenbasierter und mobiltelefonbasierter E-Government-Lösungen in Österreich verglichen. Dieser Artikel stellt die evaluierten Komponenten vor, beschreibt Design und Durchführung des zur Evaluierung angewendeten Usability-Tests und diskutiert schließlich im Detail die erhaltenen Resultate.

2 Bürgerkarten-Implementierungen

Die *Bürgerkarte* ist die zentrale Grundlage aller E-Government-Lösungen in Österreich und erfüllt prinzipiell zwei Aufgaben. Mit der Bürgerkarte können sich österreichische Bürgerinnen und Bürger einerseits sicher über das Internet an E-Government-Applikationen authentifizieren und andererseits qualifizierte elektronische Signaturen erstellen. Die Bürgerkarte fungiert damit als sichere Signaturerstellungseinheit und ist in der Lage, sämtliche Anforderungen der Signaturrichtlinie der Europäischen Union für die Erstellung qualifizierter Signaturen [EuPa99] zu erfüllen.

Obwohl der Begriff Bürgerkarte die Verwendung von Chipkarten zu suggerieren scheint, ist das Konzept der österreichischen Bürgerkarte tatsächlich weitgehend technologieneutral und erlaubt grundsätzlich verschiedene Implementierungsvarianten. Aktuell können in Österreich sowohl Chipkarten als auch Mobiltelefone als Bürgerkarte verwendet werden. Die Unterstützung verschiedener Implementierungen ist aus Sicht der Bürgerinnen und Bürger durchaus vorteilhaft, können diese doch die von ihnen präferierte Implementierungsvariante wählen.

Aus Sicht von Dienstanbietern kann sich daraus jedoch ein beträchtlicher Mehraufwand ergeben, da mehrere verschiedene Bürgerkarten-Implementierungen (Chipkarten, Mobiltelefone, etc.) unterstützt werden müssen.

Um die Komplexität in Grenzen zu halten, wurde für das österreichische Bürgerkartenkonzept ein middlewarebasierter Ansatz vorgesehen. Zentrales Element der gewählten Architektur ist der sogenannte *Security Layer*, der in [LeHP02] vorgestellt und diskutiert wurde. Der Security Layer ist eine abstrakte XML-basierte Schnittstelle zwischen E-Government-Applikationen und verschiedenen Bürgerkarten-Implementierungen. Applikationen können über diese abstrakte Schnittstelle auf Bürgerkartenfunktionalität (Authentifizierung, Signatur, etc.) zugreifen, ohne über Implementierungsdetails der verwendeten Bürgerkarte Bescheid wissen zu müssen.

Da in der Regel weder Chipkarten noch Mobiltelefone in der Lage sind, eine XML-basierte Schnittstelle zu implementieren, bedarf es einer Middlewarekomponente, die diese Aufgabe übernimmt. Gemäß der Nomenklatur des österreichischen E-Governments wird diese Middlewarekomponente *Bürgerkartenumgebung (BKU)* genannt. Abb. 1 zeigt das prinzipielle Zusammenspiel von Security Layer und Bürgerkartenumgebung. Durch die offene Spezifikation der Security Layer Schnittstelle gibt es in Österreich aktuell mehrere verschiedene Bürgerkartenumgebungen (d.h. Middleware-Implementierungen) [MOWS12][TrDB12]. Auch hier können Bürgerinnen und Bürger die von ihnen präferierte Lösung wählen. Jene BKUs, deren Benutzerfreundlichkeit in weiterer Folge näher untersucht werden soll, werden in den folgenden Unterabschnitten ausführlicher vorgestellt.

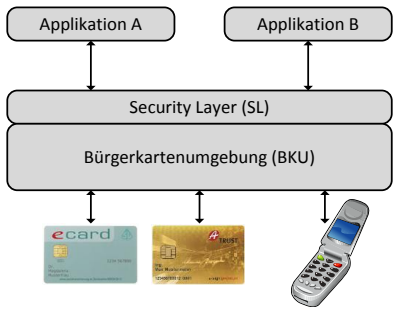


Abb. 1. Die Bürgerkartenumgebung implementiert die Security Layer Schnittstelle und fungiert als Middleware zwischen Applikationen und Bürgerkarten-Implementierungen.

2.1 MOCCA Lokal

Die Integration von Chipkarten in (zumeist webbasierte) E-Government-Applikationen ist eine nicht-triviale Aufgabe. Nahezu alle verfügbaren Bürgerkartenumgebungen verfolgen zur Lösung dieses Problems einen sogenannten *lokalen* Ansatz. Bei diesem muss am lokalen System der Bürgerin bzw. des Bürgers eine Software installiert werden, die in der Lage ist, mit der lokal am System der Benutzerin bzw. des Benutzers vorhandenen Chipkarte (z.B. über das PC/SC-Protokoll) zu kommunizieren. Die Software stellt außerdem das Security Layer Interface über einen lokalen Netzwerk-Port zur Verfügung. Webbasierte E-Government Applikationen können damit auf dieses Interface und in weiterer Folge auf die Bürgerkarte einfach über den Web-Browser der Benutzerin bzw. des Benutzers zugreifen. Abb. 2 zeigt den prinzipiellen Aufbau von BKUs, die dem lokalen Ansatz folgen.

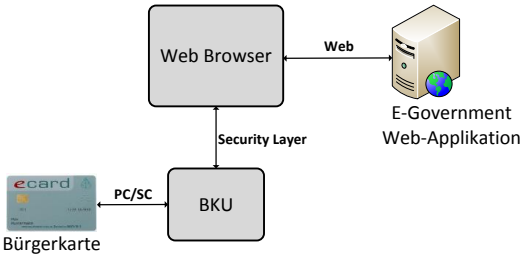


Abb. 2. Der lokale Ansatz beruht auf Software, die am lokalen System installiert werden muss.

Aktuell ist in Österreich eine Reihe von BKUs, die dem lokalen Ansatz folgen, verfügbar (z.B. [TrDB12]). *MOCCA Lokal* [MOWS12] ist jedoch derzeit die einzige Lösung, die als Open Source zur Verfügung steht. MOCCA Lokal wurde im Rahmen des MOCCA Projekts [MOCC12] entwickelt und stellt derzeit eine der meistverwendeten BKUs für den Zugriff auf chipkartenbasierte Bürgerkarten-Implementierungen dar.

2.2 MOCCA Online

Während der im vorherigen Abschnitt beschriebene lokale Ansatz durchaus zuverlässig funktioniert, ergeben sich für diesen diverse Mängel in Bezug auf die Benutzerfreundlichkeit. Vor allem die Notwendigkeit der Installation einer lokalen Softwarekomponente erwies sich in der Praxis in vielen Fällen vor allem für wenig erfahrene Computerbenutzerinnen und Benutzer als problematisch. Um diesem Problem entgegenzuwirken, wurde im Rahmen des oben erwähnten MOCCA-Projekts eine installationsfreie BKU-Alternative namens *MOCCA Online* entwickelt.

MOCCA Online wurde in [CeOB10] bereits ausführlich vorgestellt und diskutiert. Der prinzipielle Aufbau von MOCCA Online ist in Abb. 3 dargestellt. MOCCA Online besteht aus einer zentralen Server-Komponente und einem lokal im Web-Browser der Bürgerin bzw. des Bürgers gestarteten Java Applets. Die zentrale MOCCA-Komponente implementiert das Security Layer Interface. Das am lokalen System laufende Java Applet ist hingegen für den Zugriff auf die Chipkarte am lokalen System verantwortlich. Die beiden Komponenten kommunizieren über ein internes, proprietäres Interface.

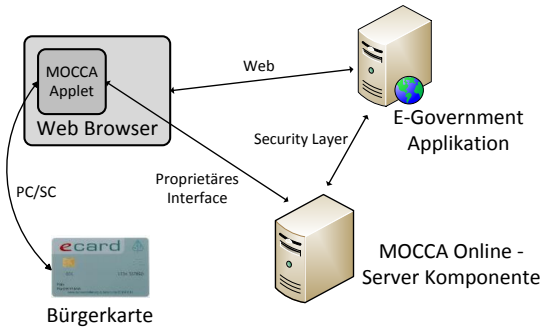


Abb. 3. MOCCA Online beruht auf einer verteilten Architektur bestehend aus einer zentralen Server-Komponente und einem lokalen Java Applet.

Durch die Auslagerung eines Teils der BKU-Funktionalität auf eine zentrale Komponente und die Verwendung eines Java Applets ist bei diesem Ansatz auf dem lokalen Systems der Bürgerin bzw. des Bürgers keinerlei Softwareinstallation nötig. Die einzige Anforderung an das lokale System ist eine entsprechend aktuelle Java Laufzeitumgebung zur Ausführung des Java Applets.

2.3 Österreichische Handy-Signatur

Für chipkartenbasierte Bürgerkarten-Implementierungen konnte durch MOCCA Online bereits eine signifikante Verbesserung der Usability erreicht werden. Generell bergen chipkartenbasierte Ansätze jedoch das Problem, dass Bürgerinnen und Bürger über ein entsprechendes Chipkartenlesegerät verfügen müssen. Die Erfahrung zeigte, dass dies in der Praxis oft eine ernstzunehmende Hürde darstellte.

Um dieses Problem zu umgehen, wurde in Österreich die *Handy-Signatur* als Alternative zu chipkartenbasierten Bürgerkarten-Implementierungen entwickelt. Die Handy-Signatur wurde in [OrCK10] vorgestellt und diskutiert. Der prinzipielle Aufbau der Handy-Signatur ist in Abb. 4 dargestellt. Kernkomponente der Handy-Signatur ist das zentrale Handy-Signatur Service. Dieses stellt ähnlich wie MOCCA Online die Security Layer-Schnittstelle zur Verfügung, über die E-Government-Applikationen Zugriff auf die Bürgerkartenfunktionalität erlangen können. Im Gegensatz zu MOCCA Online werden bei der Handy-Signatur alle Schlüssel und Zertifikate zentral vom Handy-Signatur Service sicher in einem HSM gehalten. Für einen Zugriff auf diese Daten bedarf es einer entsprechenden Autorisierung der Bürgerin bzw. des Bürgers. Dazu muss in einem ersten Schritt ein geheimes Passwort zusammen mit der Mobiltelefonnummer über ein abgesichertes Web-Formular an das Handy-Signatur Service übermittelt werden. Konnte dieses Passwort positiv verifiziert werden, wird ein zeitlich begrenzt gültiges Einmalpasswort (TAN) an das Mobiltelefon der Bürgerin bzw. des Bürgers gesendet. Zur endgültigen Autorisierung des Zugriffs auf die Bürgerkartenfunktionalität muss dieses Einmalpasswort über das Web-Formular ebenfalls an das Handy-Signatur Service übermittelt werden.

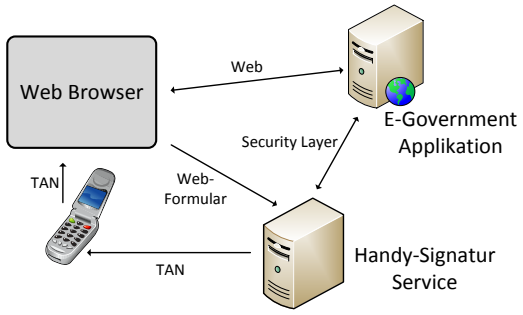


Abb. 4. Die Handy-Signatur sieht eine zentrale Signaturerstellung in einem serverseitigen HSM vor.

Wie bei chipkartenbasierten Lösungen implementiert auch die Handy-Signatur eine Zwei-Faktor-Authentifizierung. Die Sicherheit dieses Verfahrens ist damit vergleichbar zu jener von chipkartenbasierten Ansätzen. Eine ausführliche Diskussion der Sicherheit der österreichischen Handy-Signatur und ein Vergleich zu anderen mobilen Signaturerstellungsvarianten kann unter anderem in [ZeTL11] nachgelesen werden.

Die Handy-Signatur funktioniert unabhängig von chipkartenbasierten Lösungen. Österreichische Bürgerinnen und Bürger können daher die Handy-Signatur und alternative chipkartenbasierte Lösungen parallel verwenden. Personen, die bereits über eine gültige chipkartenbasierte Bürgerkarte verfügen, können diese zudem verwenden, um die Bürgerkartenfunktionalität auf ihrem Mobiltelefon über ein entsprechendes Web-Portal freischalten zu lassen. Das Aufsuchen einer Registrierungsstelle ist in diesem Fall nicht mehr nötig.

3 Test-Design

Alle drei in Abschnitt 2 vorgestellten Bürgerkarten-Implementierungen (bzw. BKU-Implementierungen) kommen in Österreich regelmäßig zur Anwendung. Während der Fokus im Rahmen der Entwicklung dieser Komponenten hauptsächlich auf Funktionalität und Sicherheit lag, wurde Aspekten der Benutzerfreundlichkeit oft zu wenig Aufmerksamkeit zuteil. Im Rahmen einer umfangreichen Usability-Analyse wurde diesem Umstand Rechnung getragen und die Benutzerfreundlichkeit der in Abschnitt 2 beschriebenen Komponenten (MOCCA Lokal, MOCCA Online, Handy-Signatur) evaluiert. Dazu wurde ein Thinking-Aloud-Test [BoRa99] mit insgesamt zwanzig Testbenutzern durchgeführt. Durch diesen Test sollte vor allem festgestellt werden, ob die Handy-Signatur gegenüber chipkartenbasierten Lösungen aus Sicht von Benutzerinnen und Benutzern signifikante Vorteile aufweist. Das gewählte Test-Design, sowie Details zu dessen Umsetzung werden in diesem Abschnitt diskutiert.

3.1 Fragestellungen

Zur Sicherstellung einer strukturierten Vorgehensweise wurden in einem ersten Schritt spezifische Fragestellungen definiert. Ziel der durchgeführten Usability-Analyse war es, geeignete Antworten auf die im Folgenden angeführten Fragestellungen zu finden.

- Wie wird die Benutzerfreundlichkeit der Aktivierung der Handy-Signatur aus Sicht der Benutzerinnen und Benutzer wahrgenommen?
- Wie wird die Benutzerfreundlichkeit der Verwendung der Handy-Signatur aus Sicht der Benutzerinnen und Benutzer wahrgenommen?
- Welche Bürgerkarten-Implementierung wird von Benutzerinnen und Benutzern generell bevorzugt?
- Wie wird die Sicherheit und Vertrauenswürdigkeit der einzelnen Bürgerkarten-Implementierungen durch Benutzerinnen und Benutzer bewertet?

Die durch die durchgeführte Usability-Evaluierung erhaltenen Antworten zu diesen Fragestellungen werden in Abschnitt 4 dieses Artikels ausführlich diskutiert.

3.2 Testmethode und Setup

Die Benutzerfreundlichkeit der evaluierten Komponenten wurde im Rahmen eines Thinking-Aloud-Tests untersucht. Bei einem Thinking-Aloud-Test werden Benutzerinnen und Benutzer gebeten, eine Reihe von praktischen Aufgaben durchzuführen. Die Tätigkeiten der Testpersonen werden dabei beobachtet und mit technischen Hilfsmitteln aufgezeichnet. Dadurch können in einem folgenden Analyseschritt auffällige Verhaltensweisen im Umgang mit den zu evaluierenden Komponenten analysiert und das Verhalten verschiedener Testpersonen verglichen werden.

Im Rahmen des durchgeführten Usability-Tests kam eine spezielle Software zum Einsatz[1]. Diese erlaubte die Aufzeichnung sämtlicher Benutzereingaben und ermöglichte zudem die Aufnahme der von den Testpersonen während des Tests getätigten Äußerungen. Zusätzlich wurde das Gesicht der Testpersonen aufgezeichnet, um etwaige Stimmungsschwankungen über deren Mimik ablesen und analysieren zu können. Die eingesetzte Software ermöglichte eine strukturierte Aufzeichnung und Speicherung der gesammelten Daten und bot zudem Unterstützung bei deren nachfolgenden Auswertung.

Die Aufzeichnung der Benutzerinteraktion ermöglichte eine objektive Analyse verschiedener Aspekte der evaluierten Komponenten. Darüber hinaus war jedoch auch der subjektive Eindruck der Testpersonen, den diese von den untersuchten Komponenten hatten, von Interesse. Um diesen festhalten zu können, wurden alle Testpersonen gebeten, nach Durchführung der einzelnen ihnen gestellten Aufgaben die verwendeten Komponenten über einen Fragebogen zu bewerten. Zusätzlich wurden alle Testpersonen im Rahmen eines abschließenden Interviews gebeten, persönliche Eindrücke und Präferenzen zu artikulieren.

Nach Abschluss des Thinking-Aloud-Tests wurden die gesammelten Daten analysiert und entsprechend statistisch aufbereitet. Dadurch war es schlussendlich möglich die zu Beginn der Usability-Überprüfung definierten Fragestellungen zu beantworten und somit wertvolle Erkenntnisse in Bezug auf die Benutzerfreundlichkeit der evaluierten Komponenten zu gewinnen.

3.3 Aufgabenstellungen

Um bestmögliche Antworten auf die definierten Fragen zu erhalten, wurden die im Rahmen des Thinking-Aloud-Tests durchzuführenden Aufgaben auf die gegebenen Fragestellungen zugeschnitten. Aus den gegebenen Fragestellungen wurden daher folgende Aufgaben generiert und den Testpersonen zur Ausführung vorgelegt.

- Führen Sie ein gegebenes E-Government-Verfahren unter Verwendung ihrer chipkartenbasierten Bürgerkarte und MOCCA Lokal durch!
- Führen Sie ein gegebenes E-Government-Verfahren unter Verwendung ihrer chipkartenbasierten Bürgerkarte und MOCCA Online durch!
- Aktivieren Sie die Handy-Signatur mit Hilfe Ihrer chipkartenbasierten Bürgerkarte!
- Führen Sie ein gegebenes E-Government-Verfahren unter Verwendung der Handy-Signatur durch!

Um etwaige Verfälschungen der Resultate, die sich durch Lerneffekte ergeben hätten können, zu verhindern, wurden die oben genannten Aufgaben teilweise in unterschiedlicher Reihenfolge ausgeführt. Dadurch konnte die Auswirkung von Lerneffekten größtenteils ausgeglichen und die Gültigkeit und Aussagekraft der erhaltenen Resultate verbessert werden.

3.4 Testpersonen

Zur Evaluierung der Benutzerfreundlichkeit der verschiedenen österreichischen Bürgerkarten-Implementierungen wurde der Thinking-Aloud-Test mit insgesamt zwanzig Testpersonen

[1] Für die Durchführung der Tests wurden die Softwareprodukte Morae® Recorder, Morae® Observer und Morae® Manager der Firma TechSmith® verwendet. Weitere Informationen zu diesen Softwarelösungen finden sich unter http://www.techsmith.com/morae.html.

durchgeführt. Bei der Auswahl der Testkandidatinnen und Testkandidaten wurde darauf geachtet, die österreichische Bevölkerung möglichst genau abzubilden. Es wurden daher Personen unterschiedlichen Alters und mit verschiedenem Bildungsniveau zur Teilnahme eingeladen. Ebenso wurde darauf geachtet, dass auch technisch wenig versierte Bürgerinnen und Bürger an diesem Usability-Test teilnahmen.

Die folgenden Abbildungen zeigen die Verteilung der an dem Usability-Test teilnehmenden Testpersonen in Bezug auf Alter, Ausbildungsniveau und technischem Vorwissen. Die personenbezogenen Daten wurden für jede Testkandidatin bzw. jeden Testkandidaten vor Durchführung des Tests im Rahmen eines Einführungsgesprächs erhoben. Wie aus den Abbildungen ersichtlich ist, konnte für jede Kategorie eine entsprechend ausgewogene Verteilung erreicht werden.

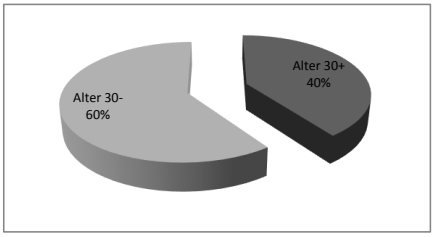


Abb. 5: Altersverteilung der teilnehmenden Testpersonen.

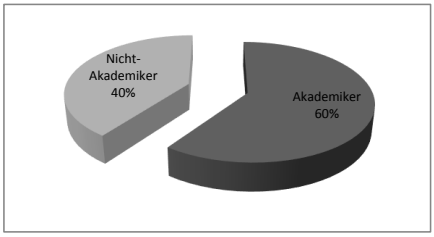


Abb. 6: Ausbildungsniveau der teilnehmenden Testpersonen.

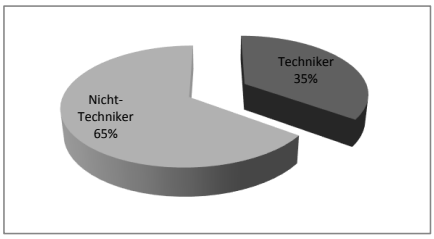


Abb. 7: Technisches Vorwissen der teilnehmenden Testpersonen.

4 Resultate

Ziel des durchgeführten Usability-Tests war die Evaluierung der Benutzerfreundlichkeit und Akzeptanz der österreichischen E-Government-Komponenten MOCCA Lokal, MOCCA Online und Handy-Signatur. Dadurch sollte überprüft werden, ob durch die chipkartenlose mobile Alternative die Akzeptanz von E-Government-Applikationen generell entscheidend erhöht werden kann. In diesem Abschnitt werden die erhaltenen Resultate des durchgeführten Usability-Tests präsentiert und diskutiert. Die folgenden vier Unterabschnitte geben dabei Antworten auf die vier zu Beginn des Usability-Tests festgelegten Fragestellungen.

4.1 Aktivierung der Handy-Signatur

Bevor die österreichische Handy-Signatur als Alternative zu chipkartenbasierten Lösungen verwendet werden kann, muss diese für das Mobiltelefon der Bürgerin bzw. des Bürgers aktiviert werden. Die Aktivierung kann in eigens dafür vorgesehenen Registrierungsstellen oder unter Verwendung einer chipkartenbasierten Bürgerkarte von der Bürgerin bzw. dem Bürger selbst vorgenommen werden. Die Benutzerfreundlichkeit letzterer Variante wurde im Rahmen des Usability-Tests überprüft.

Abb. 8 zeigt, dass die Aktivierung der Handy-Signatur von den Testpersonen durchwegs positiv bewertet wurde. Der Aktivierungsvorgang konnte in allen abgefragten Kategorien gute bis sehr gute Ergebnisse erreichen. Insgesamt kann festgehalten werden, dass die Aktivierung der Handy-Signatur von beinahe allen Testpersonen selbstständig und problemlos durchgeführt werden konnte und diesbezüglich keine groben Mängel festgestellt werden konnten.

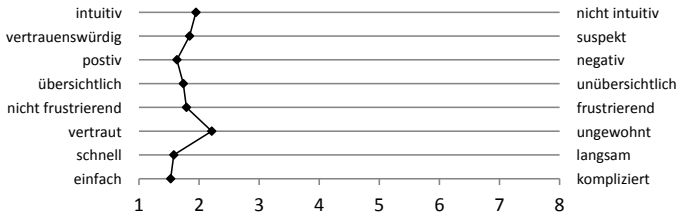


Abb. 8: Die Aktivierung der Handy-Signatur wurde von allen Testpersonen positiv bewertet.

4.2 Verwendung der Handy-Signatur

Die wohl interessanteste Frage, die im Rahmen des Usability-Tests untersucht wurde, betraf die Verwendung der Handy-Signatur und deren Abschneiden im Vergleich zu chipkartenbasierten Bürgerkarten-Implementierungen. Die Testpersonen wurden aufgefordert, ein typisches E-Government-Verfahren sowohl mit einer chipkartenbasierten Bürgerkarte (unter Verwendung von MOCCA Lokal und MOCCA Online), als auch mit der Handy-Signatur durchzuführen und danach verschiedene Aspekte der Benutzerfreundlichkeit zu bewerten. Abb. 9 zeigt die erhaltenen Resultate.

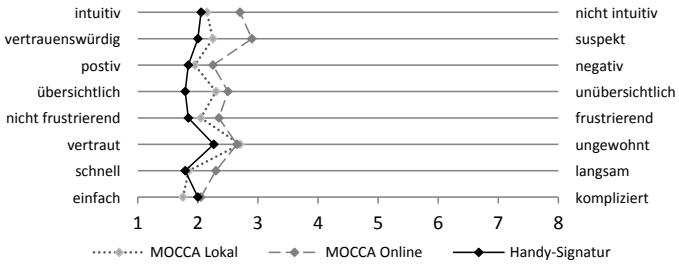


Abb. 9. Die Handy-Signatur wurde in fast allen Aspekten positiver bewertet als chipkartenbasierte Lösungen.

Die Handy-Signatur wurde im Vergleich zu chipkartenbasierten Lösungen in fast allen Kategorien positiver bewertet. Diese Beobachtung gilt unabhängig von der verwendeten Bürgerkartenumgebung bei der Verwendung von Chipkarten. Einzig in der Kategorie Einfachheit konnte sich MOCCA Lokal knapp gegenüber der Handy-Signatur durchsetzen.

4.3 Bevorzugte Implementierung

Nach Abschluss der Tests wurden alle Testpersonen gefragt, welche Bürgerkarten-Implementierung (Chipkarte oder Handy-Signatur) bzw. welche BKU (MOCCA Lokal,

MOCCA Online, etc.) sie im Rahmen einer zukünftigen privaten Nutzung der Bürgerkarte bevorzugen würden. Auch hier ging die Handy-Signatur als klarer Sieger hervor. Abb. 10 illustriert die erhaltenen Resultate im Detail.

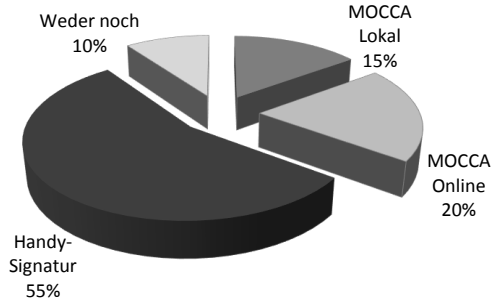


Abb. 10. Mehr als die Hälfte aller Testpersonen würde die Handy-Signatur im Rahmen einer zukünftigen privaten Verwendung der Bürgerkarte bevorzugen.

Wie erwartet erreichte MOCCA Online etwas besserer Werte als MOCCA Lokal. Hier wurde von den Testpersonen unter anderem der Umstand gewürdigt, dass MOCCA Online im Gegensatz zu MOCCA Lokal keiner Softwareinstallation bedarf. Sowohl MOCCA Online als auch MOCCA Lokal liegen jedoch deutlich hinter der Handy-Signatur zurück, die insgesamt 55% der Testbenutzerinnen und Benutzer überzeugen konnte.

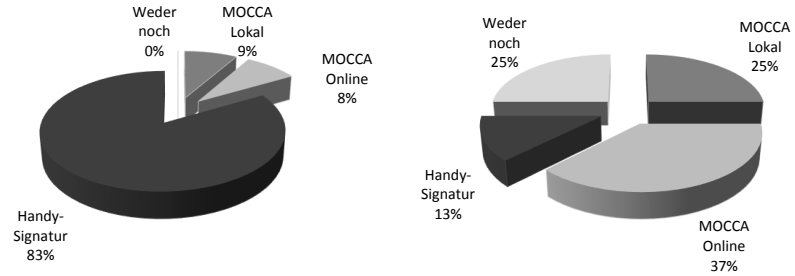


Abb. 11. Linkes Diagramm: Bei unter 30-jährigen Testpersonen konnte sich die Handy-Signatur klar durchsetzen. Rechtes Diagramm: Bei älteren Testpersonen wurden chipkartenbasierte Lösungen klar bevorzugt (rechts).

Eine getrennte Betrachtung der Resultate älterer und jüngerer Testpersonen führte zu überraschend deutlichen Ergebnissen. Abb. 11 zeigt, dass die Handy-Signatur offenbar vor allem bei Benutzerinnen und Benutzern, die jünger als 30 Jahre sind, außerordentlich beliebt ist. Bei älteren Testpersonen konnten sich in dieser Frage hingegen chipkartenbasierte Ansätze (MOCCA Lokal und MOCCA Online) durchsetzen. Auch in dieser Gruppe behielt MOCCA Online klar die Oberhand gegenüber MOCCA Lokal.

4.4 Sicherheit und Vertrauenswürdigkeit

Aus Sicht von Dienstanbietern und Applikationsbetreibern ist auch das Maß an Sicherheit und Vertrauenswürdigkeit, das Benutzerinnen und Benutzer den verwendeten Lösungen attestieren, von Interesse. Alle Testpersonen wurden daher nach Absolvierung der ihnen gestellten

Aufgaben gefragt, ob sie die verwendeten Bürgerkarten-Implementierungen und Softwarekomponenten als sicher und vertrauenswürdig einstufen würden.

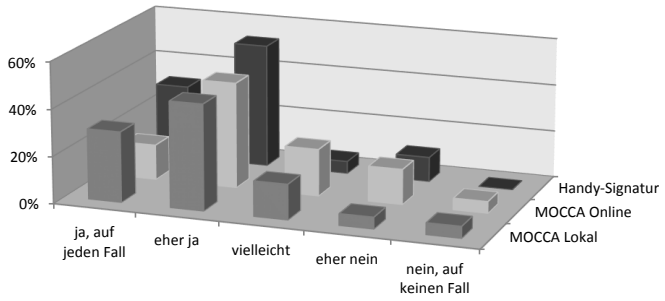


Abb. 12. Testbenutzerinnen und Testbenutzer hatten das Gefühl, dass alle Versionen der Bürgerkartenumgebungen eher vertrauenswürdig sind.

Die in Abb. 12 dargestellten Ergebnisse zeigen, dass generell allen evaluierten Implementierungen ein entsprechendes Maß an Sicherheit und Vertrauenswürdigkeit attestiert wurde. Auch bei dieser Frage konnte sich die Handy-Signatur schlussendlich gegenüber chipkartenbasierten Lösungen durchsetzen. Die Unterschiede waren hier jedoch nicht so signifikant wie bei den zuvor diskutierten Fragestellungen.

5 Fazit

Seit der Vorstellung des Konzepts der Bürgerkarte in Österreich wurden beträchtliche Ressourcen in die Entwicklung entsprechender Implementierungen investiert. Erste Varianten lokaler Bürgerkartenumgebungen für chipkartenbasierte Bürgerkarten-Implementierungen konnten funktionale Anforderungen erfüllen, waren jedoch oft nicht in der Lage, ein geeignetes Maß an Benutzerfreundlichkeit zu bieten. Die nachfolgende Entwicklung von MOCCA Online konnte einige dieser Nachteile beseitigen, beruhte jedoch ebenfalls noch auf der Chipkartentechnologie. Die österreichische Handy-Signatur repräsentiert die aktuelle Stufe der Entwicklung, in der erstmals eine vollständige Alternative zu chipkartenbasierten Lösungen geboten wird.

Die durchgeführte Usability-Analyse zeigte, dass der eingeschlagene Weg stimmt. In den meisten evaluierten Punkten konnte sich MOCCA Online gegenüber der lokalen BKU-Implementierung durchsetzen. Die Abkehr vom lokalen Ansatz und die Entwicklung einer installationsfreien BKU-Alternative für chipkartenbasierte Bürgerkarten-Implementierungen kann im Nachhinein daher als richtige Entscheidung bewertet werden.

Noch deutlicher fiel das Ergebnis der Evaluierung der Benutzerfreundlichkeit in Bezug auf die Handy-Signatur aus. Diese konnte in allen untersuchten Punkten die ebenfalls evaluierten chipkartenbasierten Lösungen klar übertreffen. Sowohl in puncto Benutzerfreundlichkeit, als auch in Bezug auf die Akzeptanz ging die Handy-Signatur jeweils als klarer Sieger hervor. Die Entscheidung, eine mobile Alternative zu chipkartenbasierten Bürgerkarten-Implementierungen zu entwickeln stellte sich damit im Nachhinein als richtig heraus.

Die erhaltenen Resultate der durchgeführten Usability-Analyse decken sich größtenteils mit den Erfahrungen des Alltags. Diese zeigen, dass die Handy-Signatur in der österreichischen

Bevölkerung im Vergleich zu chipkartenbasierten Lösungen eine breitere Akzeptanz findet. Die österreichische Handy-Signatur scheint damit in der Lage zu sein, den Spagat zwischen strikten Sicherheitsanforderungen und einem adäquaten Level an Benutzerfreundlichkeit zur Zufriedenheit der der österreichischen Bevölkerung zu meistern.

Literatur

[EuPa99] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. Official Journal of the European Communities L 013, (2000) 12-20.

[LeHP02] Leitold, H., Hollosi, A., Posch, R.: Security Architecture of the Austrian Citizen Card Concept. Proceedings of the 18th Annual Computer Security Applications Con- ference, Washington, DC, USA, IEEE Computer Society, (2002) 391–400.

[CeOB10] Centner, M., Orthacker C., Bauer, W.: Minimal-Footprint Middleware for the Creation of Qualified Signatures. Proceedings of the 6th International Conference on Web Information Systems and Technologies, Portugal, (2010) 64–69.

[MOCC12] Modular Open Citizen Card Architecture, http://mocca.egovlabs.gv.at/, (2012).

[BoRa99] M. Ted Boren, J. Ramey: Thinking Aloud: Reconciling Theory and Practice. In: IEEE Transactions on Professional Communication Vol. 43, No. 3 (2000) 261-278.

[BÜRG12] Bürgerkarte – Spezifikationen, http://alt.buergerkarte.at/de/spezifikation/index.html, (2012).

[MOWS12] Bürgerkartenumgebung MOCCA Webstart, http://webstart.buergerkarte.at/mocca/, (2012).

[TrDB12] ITSolution: trustDesk basic, http://www.itsolution.at/trustDesk-basic.html, (2012).

[OrCK10] Orthacker, C., Centner, M., Kittl, C.: Qualified Mobile Server Signature. Proceedings of the 25th TC 11 International Information Security Conference, SEC 2010, (2010).

[ZeTL11] Zefferer, T., Teufl, P., Leitold, H.: Mobile qualifizierte Signaturen in Europa. Datenschutz und Datensicherheit 11|2011, (2011).

7 | Usability Evaluation of Electronic Signature Based E-Government Solutions

Conference	IADIS International Conference WWW/INTERNET 2012
Language	English
Title	Usability Evaluation of Electronic Signature Based E-Government Solutions
Authors	Thomas Zefferer, Vesna Krnjic
Publisher	Academic Conferences Limited
Booktitle/Journal	Proceedings of the IADIS International Conference WWW/INTERNET 2012

USABILITY EVALUATION OF ELECTRONIC SIGNATURE BASED E-GOVERNMENT SOLUTIONS

Thomas Zefferer
E-Government Innovation Center (EGIZ)
Inffeldgasse 16a, 8010 Graz, Austria

Vesna Krnjic
E-Government Innovation Center (EGIZ)
Inffeldgasse 16a, 8010 Graz, Austria

ABSTRACT

Usability and security are crucial requirements of e-Government applications. Security requirements are typically met by approved cryptographic methods such as qualified electronic signatures. These methods usually rely on integration of cryptographic hardware tokens such as smart cards or mobile phones. Integration of these tokens into e-Government applications introduces additional complexity and often affects the usability of these solutions. To date, research on usability in e-Government has primarily focused on the evaluation of e-Government websites. Usability issues raised by the integration of cryptographic hardware tokens into e-Government applications have not been considered in detail so far. We filled this gap by conducting a usability analysis of three core components of the Austrian e-Government infrastructure. The evaluated components act as middleware and facilitate integration of cryptographic hardware tokens into e-Government applications. We have tested the usability and perceived security of these middleware components by means of a thinking-aloud test. This paper introduces the evaluated components, discusses the followed methodology of the conducted usability test, and presents obtained results.

KEYWORDS

Usability, E-Government, Security, Austrian Citizen Card, MOCCA, Mobile Phone Signature.

1. INTRODUCTION

In many countries, governments and public administrations make use of information and communication technologies (ICT) to improve the efficiency of administrative procedures and to ease interaction with citizens. These attempts have become commonly known under the term electronic government (e-Government). Current e-Government applications range from simple informational services (e.g. publication of relevant information on governmental websites) to complex transactional applications (e.g. filing tax documents and payments over the Internet). Transactional e-Government applications potentially comprise the transmission and processing of security and privacy sensitive data. Hence, these applications typically have to fulfill increased security requirements. To meet these requirements, approved cryptographic methods such as strong user authentication schemes and electronic signatures are employed.

Electronic signatures play an important role especially in the European Union, where qualified electronic signatures are legally equivalent to handwritten signatures according to the Directive 1999/93/EC of the European Parliament and of the Council on a Community framework for electronic signatures (EU Parliament and Council, 2000). To meet the requirements of qualified electronic signatures as defined by this directive, a secure signature creation device (SSCD) has to be used for the signature creation process. Since SSCDs typically rely on a secure hardware token, implementation alternatives are actually limited. Following the current state of the art, e-Government solutions usually require citizens to use personalized smart cards to create legally binding electronic signatures. Smart card based e-Government solutions have been deployed successfully in Austria, Belgium, Estonia, Portugal, Spain, and various other European countries. A couple of

European countries such as Austria or Estonia additionally provide citizens mobile signature solutions that employ mobile phones as hardware tokens instead of smart cards.

Regardless of the nature of the used hardware token, the question arises how these tokens can be used and accessed by e-Government applications. Currently, most countries rely on some kind of middleware that acts as intermediary between cryptographic hardware tokens and e-Government applications. This approach is also followed in Austria where several middleware solutions have been developed during the past decade. These solutions allow for a smooth integration of qualified electronic signatures and assure the security of e-Government applications.

Besides security, usability is another key success and acceptance factor of e-Government solutions. Schultz et al. have shown that the demand for usability often conflicts with given security requirements (Schultz et al., 2001). While an appropriate level of security requires the application of complex cryptographic methods and protocols, the increased complexity often significantly affects usability. It is thus hardly surprising that usability is often neglected in e-Government applications with high security requirements. This is problematic as it potentially leads to a scenario, in which e-Government applications are virtually restricted to expert users. To make e-Government solutions usable for all social and educational classes, usability has to be recognized as important requirement for e-Government applications and solutions.

The importance of usability in e-Government has been subject to ongoing research. However, most related work has focused on the usability of rather simple e-Government websites so far. For instance, a quality inspection method for the evaluation of e-Government sites has been proposed by Garcia et al. (Garcia et al., 2005). The usability of different e-Government websites in the UK has been evaluated by Ma et al. (Ma et al., 2003). Recently, also the usability of Norwegian e-Government websites has been discussed (Sorum, 2011). Without doubt, the usability of e-Government websites is an important topic. However, techniques to integrate qualified electronic signatures into Web based e-Government applications definitely need to be considered as well. Otherwise, usability evaluations of current e-Government solutions threaten to remain incomplete and to miss relevant aspects.

In Austria, electronic signatures are integrated into e-Government applications by means of different middleware solutions. We have evaluated the usability of these core components of the Austrian e-Government infrastructure by means of a usability test. The basic goal of this test was to compare the usability and user acceptance of different middleware implementations and to identify persisting weaknesses. In this paper we briefly introduce the evaluated components, discuss the followed methodology of the conducted usability test, and present obtained results.

The paper is structured as follows. In Section 2 we discuss core concepts of the Austrian e-Government and introduce the evaluated components in detail. The methodology of the conducted usability test is explained in Section 3. Subsequently, obtained results are discussed in Section 4. Conclusions are finally drawn in Section 5.

2. EVALUATED E-GOVERNMENT COMPONENTS

The key concept of the Austrian e-Government infrastructure is called Citizen Card (CC). The Citizen Card is an abstract definition of a cryptographic token that allows citizens to securely authenticate at e-Government services and to create qualified electronic signatures. The Citizen Card concept complies with the EU Signature Directive and fulfills all requirements of a secure signature creation device. This way, the Citizen Card represents an important enabler of secure e-Government applications in Austria.

Although the term Citizen Card might suggest the use of smart cards, the Citizen Card specifications (Hollosi, 2008) are actually rather abstract and not limited to a certain technology. This flexibility has led to the development of different Citizen Card implementations. Currently, both smart card based and mobile phone based Citizen Card implementations are available in Austria.

Irrespective of the underlying technology, all Citizen Card implementations facilitate secure user authentication and creation of qualified electronic signatures. Due to the technology neutral approach, citizens can individually choose their preferred implementation. Unfortunately, this flexibility significantly increases the complexity of application development processes. In order to integrate Citizen Card functionality, e-Government applications need to support all available Citizen Card implementations. To

overcome this problem, the Austrian e-Government strategy follows the middleware based approach illustrated in Figure 1.

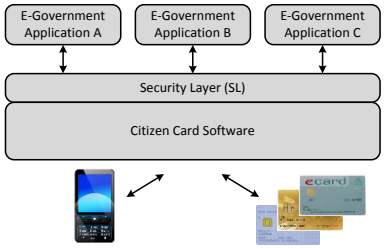


Figure 1. Access to Citizen Card implementations is provided by the Citizen Card Software.

The core element of this approach is the so called Security Layer (SL) interface that has been introduced and discussed by Leitold et al. (Leitold et al., 2002). The Security Layer is an abstract XML based interface that can be used by e-Government applications to access Citizen Card functionality. This way, applications do not need to integrate specific Citizen Card implementations, since all implementations can be accessed through a common interface. All implementation specific functionality is outsourced to the so called Citizen Card Software (CCS). The CCS implements access to specific Citizen Card implementations (e.g. smart cards) and provides their functionality through the common SL interface. Acting as middleware between e-Government applications and Citizen Card implementations, the Citizen Card Software plays a central role in the Austrian e-Government infrastructure. Since the SL specifications are open, different CCS implementations have emerged during the past years. The following figures show the basic concepts of the three most popular CCS implementations that are currently available in Austria.

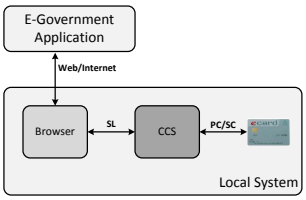


Figure 2. MOCCA Local.

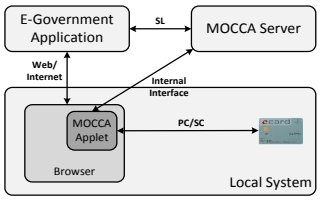


Figure 3. MOCCA Online.

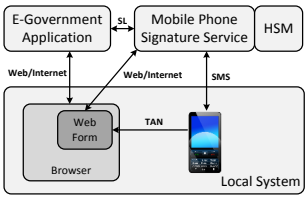


Figure 4. Mobile Phone Signature.

Figure 2 shows the basic architecture of the CCS *MOCCA Local*. The MOCCA (Modular Open Citizen Card Architecture) project[1] has been started in 2008 and aims to provide open source CCS solutions for Austrian citizens. MOCCA Local is one outcome of this project and implements a CCS by means of software being installed and running on the user's local system. MOCCA Local typically runs in the background and features a minimalistic user interface. If access to a locally connected smart card is requested by an e-Government application, a GUI window pops up. Through this window, users are provided with relevant information (e.g. the data that is about to be signed) and required user input (e.g. secure PIN to authorize the signature creation on the smart card) is collected.

From a usability point of view, the main drawback of MOCCA Local is the need to install software on the local computer, which can be problematic especially for technically inexperienced users. To overcome this problem, the MOCCA project has also investigated possibilities to implement an installation-free alternative. These efforts finally led to the development of *MOCCA Online*.

The basic architecture of MOCCA Online has been discussed by Centner et al. (Centner et al., 2010) and is shown in Figure 3. MOCCA Online follows a server based approach. The SL interface is not implemented by locally installed software, but by the central MOCCA Server component. E-Government applications contact the MOCCA Server in order to access citizens' smart cards. Physical access to the locally connected smart card is implemented by a Java Applet running on the citizen's local computer. MOCCA Applet and MOCCA Server together represent the CCS and exchange data through an internal interface. The MOCCA

[1] http://joinup.ec.europa.eu/software/mocca/home

Applet acts as user interface for the provision of relevant information (e.g. the data to be signed) and the collection of required user input (e.g. PINs).

MOCCA Online renders the need for local software installations unnecessary but still requires users to buy and use appropriate smart card reader devices. The goal to render smart cards completely unnecessary has been the main driver behind the development of mobile CCS solutions. In Austria, the so called *Mobile Phone Signature* (Orthacker et al., 2010) represents a mobile alternative to established smart card based approaches. The general architecture of the Mobile Phone Signature is shown in Figure 4.

Similar to MOCCA Online, a central service (Mobile Phone Signature Service) implements the SL interface. A hardware security module (HSM) that is attached to this central service acts a SSCD. The HSM is capable of creating qualified electronic signatures on behalf of users. To access Citizen Card functionality, e-Government applications send an appropriate request to the Mobile Phone Signature Service. Provision of the requested functionality (e.g. signature creation) has to be authorized by the citizen. Therefore, the Mobile Phone Signature Service requests the citizen to enter the phone number and a secret password through a Web form. The password is defined by the user during the personalization and activation process. If the provided credentials can be verified correctly, an SMS message is sent to the citizen's mobile phone containing a one-time password (Transaction Authentication Number - TAN). This TAN has to be entered in the Mobile Phone Signature Service's Web form in order to authorize execution of the e-Government application's request. The main advantage of this mobile approach is the central HSM, which renders smart cards unnecessary. By relying on a strong two-factor authentication scheme that makes use of two separated communication channels (i.e. Web and SMS), an adequate level of security is assured.

All three CCS implementations – MOCCA Local, MOCCA Online, and Mobile Phone Signature – meet given security requirements. To check whether these components are also able to fulfill given usability requirements, a usability test has been conducted. The followed methodology of this test is discussed in the next section.

3. METHODOLOGY

To evaluate the usability of MOCCA Local, MOCCA Online, and Mobile Phone Signature, the following four research questions have been defined beforehand.

- **Q1:** Do required software installations on the local system represent a barrier and reduce usability?
- **Q2:** How do users rate the overall usability of MOCCA Local, MOCCA Online, and Mobile Phone Signature?
- **Q3:** How do users rate the security and trustworthiness of MOCCA Local, MOCCA Online, and Mobile Phone Signature?
- **Q4:** Which CCS implementation do users prefer in general?

Answers to these questions have been obtained by the conducted usability test. We have applied a thinking-aloud test with 20 test users in order to evaluate the usability of the three different Austrian CCS implementations. The selected set of test users represented different social classes of the Austrian society. A well balanced distribution has been achieved regarding test users' ages, educational levels, and technical background.

The basic test run was identical for all test users and consisted of the following four phases.

- **P1 - Welcome:** Test users have been welcomed, have been provided with relevant information about the usability test, and have been asked to sign a non-disclosure agreement.
- **P2 - Background questionnaire:** At the beginning of the usability test, relevant information about the participating test user has been collected using a prepared questionnaire.
- **P3 - Execution of tasks:** In this phase, test users have been asked to carry out a sequence of predefined tasks using the three CCS implementations to be evaluated. After each task, test users have been asked to fill out a prepared questionnaire and to rate the tested component (post-task rating).
- **P4 - Conclusive interview:** After completion of all tasks, a conclusive interview has been conducted with all test users. After the interview, test users have been asked to fill out a final questionnaire (post-study rating) covering some general questions.

During Phase P3, test users have been asked to carry out predefined tasks using an off-the-shelf desktop PC. Representing a common configuration, all tests have been carried out using the Microsoft Windows 7 operating system and Microsoft Internet Explorer 8 Web browser. The desktop PC was equipped with a Reiner SCT card reader device. Test users were not allowed to use other system configurations (e.g. a different Web browser) as this would have rendered direct comparisons between test users difficult. An extension of the conducted usability test to other test system environments (e.g. alternative operating systems and Web browsers) is regarded as future work.

The used test system was equipped with Morae® Recorder software. This software allows the tracking and recording of user sessions including all user activities such as mouse movements and keyboard inputs. Additionally, comments and facial expressions of test users have been recorded with a web cam and stored together with the recorded user session for later analysis. Additionally, we have used a standard camera to record user comments during Phase P2 and Phase P4.

The filled questionnaires have represented an important data source for later analysis. To obtain as much valuable feedback as possible, we relied on semantic differentials. The method of semantic differentials (Boslaugh et al., 2008) is frequently used in social sciences and user experience research. In general, semantic differentials are used to measure the connotative meaning of an object and to further derive the attitude towards this object. We used semantic differentials to allow users to assign weighted properties to the evaluated software components.

Besides the filled questionnaires, also the recorded user sessions and user comments have been incorporated in the analysis process. These data has turned out to be extremely helpful in order to understand the collected user feedback and to identify reasons for negative ratings. Obtained results of the evaluation process will be presented in Section 4.

Most relevant information has been collected during Phase P3 of the usability test, i.e. during the execution of predefined tasks by test users. We have defined these tasks such that answers to the predefined Research Questions Q1-Q4 could be derived easily from the collected data. All test users have been asked to carry out the following five tasks.

- **T1:** Install the Citizen Card Software MOCCA Local on the local system.
- **T2:** Use MOCCA Local to file a demo e-Government application.
- **T3:** Use MOCCA Online to file a demo e-Government application.
- **T4:** Activate the Mobile Phone Signature for your mobile phone.
- **T5:** Use the Mobile Phone Signature to file a demo e-Government application.

A valid smart card based Citizen Card was the only prerequisite for test users. The first three tasks covered the evaluation of the smart card based CCS implementations MOCCA Local and MOCCA Online. The last two tasks covered the evaluation of the Austrian Mobile Phone Signature. In order to cancel out learning effects that might have biased the obtained results, we split the group of test users randomly into two subgroups. Group A started with Task T1 and carried out all tasks in the order shown above. In contrast, Group B was asked to start with Task T3 followed by T1, T2, T4, and T5 instead. This way, half of the test users started with evaluating MOCCA Local, while the other half started with testing MOCCA Online. Since the use of MOCCA Local or MOCCA Online was required to carry out Task T4 and T5, these two tasks have been carried out at the very end by both user groups. As the Mobile Phone Signature follows a completely different approach than the two smart card based solutions MOCCA Local and MOCCA Online, learning effects could be neglected.

4. RESULTS

Following the methodology discussed in Section 3, the usability test has been carried out with 20 test users in total. Obtained results of the conducted usability test and answers to the predefined research questions are presented in the following subsections.

4.1 Usability of installation-based CCS

In order to answer Research Question Q1, we evaluated whether the required installation process of MOCCA Local represents a barrier for users and hence reduces usability. To install MOCCA Local using Java

Webstart technology, test users had to navigate to a given website and click a launch button. After that, test users were asked to manually install a certificate into the used Web browser. Figure 5 shows that most test users rated the usability of the installation process positively. This corresponds to the observations that have been made during the test runs. Most users were able to complete the installation on their own.

An analysis of the recorded user sessions revealed that for some user the required certificate installation was problematic. To answer Research Question Q1, we can conclude that the required software installation process of MOCCA Local does not raise severe usability issues. Still, installation routines for certificates should be simplified in order to make this a feasible task also for inexperienced users.

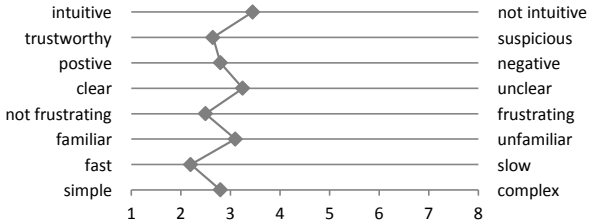


Figure 5. Evaluation results of the installation process of MOCCA Local.

4.2 Usability of different CCS implementations

According to Research Question Q2, we analyzed how the use of MOCCA Local, MOCCA Online, and Mobile Phone Signature had been rated by the test users in terms of usability. All test users have been asked to file a demo e-Government application using their Citizen Card and each of the three evaluated CCS implementations as defined by Tasks T2, T3, and T5.

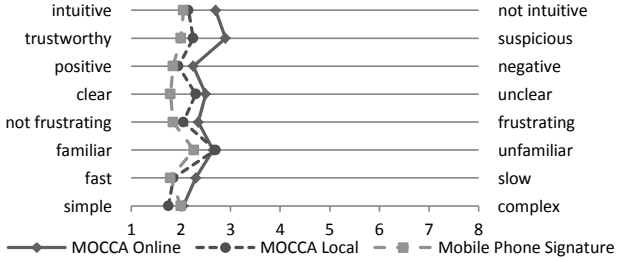


Figure 6. Perceived usability of different CCS implementations.

Figure 6 illustrates the results that have been obtained from analysis of the collected user. In general, all tested CCS implementations have been rated positively. Direct comparison of the obtained results shows that users rated the Mobile Phone Signature's usability best in most categories, followed by MOCCA Local and MOCCA Online.

4.3 Security and trustworthiness

Besides usability, the security and trustworthiness of used components is crucial for the acceptance of e-Government solutions. According to Research Question Q3, we have analyzed whether the three evaluated CCS implementations appear secure and trustworthy for users. To answer this question, test users have been asked to rate the perceived level of security and trustworthiness for all three CCS implementations. Ratings have again been collected by means of a questionnaire.

Figure 7 illustrates the obtained results for the three evaluated CCS implementations. Again, the Mobile Phone Signature achieved the best results. 84% of all test users rated the Mobile Phone Signature as secure and trustworthy. MOCCA Local obtained only slightly worse results. 74% of all test users perceived MOCCA Local as secure and trustworthy. Analysis of the recorded user sessions and of information extracted from the conducted interviews revealed main reasons for potential suspiciousness. During the

installation process of MOCCA Local, users were asked to install a certificate in the used Web browser. This is necessary in order to establish an appropriate trust relationship between the Web browser and MOCCA Local. Unfortunately, the trust status of the used certificate was not accepted by default by the used Web browser. Hence, test users were faced with a security warning during the installation of this certificate. While most users simply ignored it, some test users were unsettled by the shown security warning.

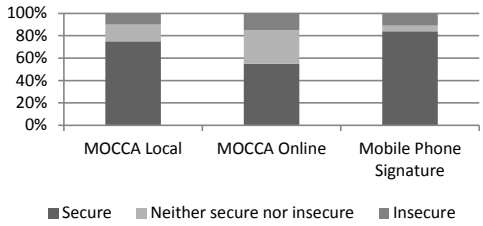


Figure 7. Perceived security and trustworthiness of the evaluated CCS implementations.

MOCCA Online obtained the worst ratings regarding security and trustworthiness. Still, 55% of all test users assumed MOCCA Online to be secure and trustworthy. Similar to MOCCA Local, suspiciousness was mainly caused by shown security warnings. Since the Java Applet of MOCCA Online accesses local resources (i.e. the user's smart card), the Applet needs to be signed. Again, the trust status of the signing certificate was not accepted by the used Web browser. Hence, a security warning was shown during the loading of the Applet.

To answer Research Question Q3 we can conclude that users basically attested all three CCS implementations an appropriate level of security and trustworthiness. Still, there is some room for improvement especially for smart card based solutions, which definitely need to improve their handling of SSL certificates. A direct comparison of the three CCS implementations shows that the Mobile Phone Signature appears to be the most secure and trustworthy solution, followed by MOCCA Local and MOCCA Online.

4.4 Personal preferences

Personal preferences of individual test users have been identified in the course of conclusive interviews. All test users have been asked whether they will continue to use their Citizen Card and which of the three tested CCS they prefer. Most test users have been convinced of the Citizen Card and stated to use it in the future for e-Government procedures. Regarding the preferred CCS, the Mobile Phone Signature has turned out to be the favored alternative. Figure 8 illustrates the obtained results. The Mobile Phone Signature has been selected by more than 50% of all test users to be the favored CCS. 20% of the test users stated that MOCCA Online is their preferred solution. For approximately 15%, MOCCA Local is the favored implementation alternative. In order to answer Research Question Q4, we can conclude that the Mobile Phone Signature is definitely the favored CCS implementation for citizens.

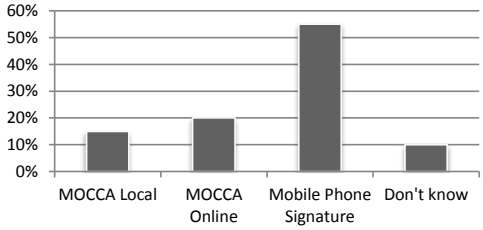


Figure 8. Preferred CCS implementation.

5. CONCLUSIONS

The goal of this work was to evaluate the usability of three core components of the Austrian e-Government infrastructure. Four research questions have been defined to cover relevant usability aspects. In order to answer these questions, a thinking-aloud test has been conducted. By analyzing the data that has been collected during these tests we were able to find appropriate answers to all previously defined research questions. In general, the conducted usability test revealed the following basic findings:

- The need for local software installation represents no serious barrier for users. However, the provided routine for the installation of certificates should be improved.
- All evaluated CCS implementations could be used without major problems and have been rated positively in terms of usability. The Mobile Phone Signature is the clear winner and appears to be the most usable solution for most test users.
- All evaluated CCS implementations have been rated positively regarding security and trustworthiness. Unsettledness has only been caused by the use of certificate with missing trust status. The Mobile Phone Signature has obtained the best ratings regarding security and trustworthiness.
- In general, the Mobile Phone Signature is the preferred CCS implementation for most test users.

While the two smart card based solutions MOCCA Local and MOCCA Online obtained comparable ratings in most categories, the Mobile Phone Signature turned out to be the clear winner in terms of popularity, security, trustworthiness, and usability. Hence, we can conclude that reliance on mobile solutions seems to be a good strategy also for future developments.

The conducted usability test delivered deeper insights into the usability of core components of the Austrian e-Government from the citizen point of view. By observing users' interactions with these components and collecting user feedback by means of different questionnaires we were able to identify persisting weaknesses and further room for improvement. Obtained results will be incorporated into future releases of the evaluated CCS implementations and help to further improve the usability of these solutions.

REFERENCES

Altameem, T. et al, 2006. Critical success factors of e-government: A proposed model for e-government implementation. Innovations in Information Technology 2006, pp. 1-5.

Boslaugh, S. and Watters, P.A., 2008. Statistics in a Nutshell, vol. 54. O'Reilly.

Centner, M. et al, 2010. Minimal-footprint middleware for the creation of qualified signatures. *Proceedings of the 6th International Conference on Web Information Systems and Technologies*, pp. 64-69. INSTICC, Portugal.

Garcia, A. C. B. C. et al, 2005. A quality inspection method to evaluate e-government sites. *Electronic Government Fourth International Conference EGOV 2005*, pp. 198-209.

Gil-Garcia J. R. and Helbig, N., 2006. Exploring E-Government Benefits and Success Factors, vol. 1, pp. 803-811. Idea Group Inc.

Hollosi, A. et al., 2008. The Austrian citizen card. http://www.buergerkarte.at

Leitold, H. et al, 2002. Security architecture of the Austrian citizen card concept. *Proceedings of the 18th Annual Computer Security Applications Conference, ACSAC '02*, pp. 391-, IEEE Computer Society, Washington, DC, USA.

Ma, T. H. Y. and Zaphiris, P., 2003. The Usability and Content Accessibility of the E-government in the UK, *Human Computer Interaction*. vol. 2007, pp. 760-764.

Orthacker, C. et al, 2010. Qualified Mobile Server Signature. *Proceedings of the 25th TC 11 International Information Security Conference SEC 2010.*

EU Parliament and Council, 2000. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. *Official Journal of the European*

Communities, L 013:12-20.

Schultz, E. E. et al, 2001. Usability and security an appraisal of usability issues in information security methods. *Computers Security*, vol. 20(7) pp. 620-634.

Sorum, H., 2011. An empirical investigation of user involvement, website quality and perceived user satisfaction in egovernment environments. *Proceedings of the Second international conference on Electronic government and the information systems perspective, EGOVIS'11*, pp. 122-134, Springer-Verlag, Berlin, Heidelberg.

8 Measuring Usability to Improve the Efficiency of Electronic Signature-Based e-Government Solutions

	Bookchapter
Language	English
Title	Measuring Usability to Improve the Efficiency of Electronic Signature-Based e-Government Solutions
Authors	Thomas Zefferer, Vesna Krnjic, Klaus Stranacher, Bernd Zwattendorfer
Publisher	Springer Science+Business Media New York
Booktitle	Measuring E-government efficiency. The opinions of Public Administrators and other Stakeholders

Chapter 4
Measuring Usability to Improve the Efficiency of Electronic Signature-Based e-Government Solutions

Thomas Zefferer, Vesna Krnjic, Klaus Stranacher, and Bernd Zwattendorfer

Abstract Usability and security are crucial requirements of efficient e-Government services and applications. Given security requirements are mostly met by integration of approved cryptographic methods such as two-factor authentication and qualified electronic signatures. Integration of these technologies into e-Government applications usually introduces additional complexity and often affects the usability of these solutions. So far, research on usability as efficiency-measuring instrument in e-Government has primarily focused on the evaluation of e-Government Web sites only. Usability issues raised by the integration of security-enhancing technologies into e-Government applications have not been considered in detail yet. We filled this gap by conducting a usability analysis of three core components of the Austrian e-Government infrastructure to improve efficiency in this domain. The evaluated components act as middleware and facilitate integration of e-ID and e-Signature tokens such as smart cards and mobile phones into e-Government applications. We have assessed the usability and perceived security of these middleware components by means of a thinking-aloud test with 20 test users. This chapter introduces the evaluated components, discusses the followed methodology, and presents obtained results of the conducted usability test.

4.1 Introduction

During the past years, e-Government solutions have evolved towards complex systems involving a broad spectrum of different players and stakeholders. In this context, particularly citizenry represents an important stakeholder. Citizens play

T. Zefferer (✉) • V. Krnjic • K. Stranacher • B. Zwattendorfer
Institute for Applied Information Processing and Communications,
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
e-mail: thomas.zefferer@iaik.tugraz.at; vensa.krnjic@iaik.tugraz.at;
klaus.stranacher@iaik.tugraz.at; bernd.zwattendorfer@iaik.tugraz.at

M.P. Rodríguez-Bolívar (ed.), *Measuring E-government Efficiency*,
Public Administration and Information Technology 5, DOI 10.1007/978-1-4614-9982-4_4,

a central role in e-Government solutions mainly for two reasons. First, citizens indirectly finance the development and maintenance of e-Government solutions through taxes. Second, citizens represent one of the main beneficiaries of e-Government solutions. Therefore, citizens have a strong interest in successful, high-qualitative, and efficient e-Government solutions.

While efficiency apparently represents a global requirement for e-Government solutions, the term efficiency itself can actually have diverging meanings and implications for different stakeholders. Regarding the special importance of the stakeholder citizenry, citizens' interpretation of the term efficiency needs to be taken into account when evaluating e-Government solutions.

For citizens, e-Government solutions are typically efficient if they help to reduce efforts and save costs. Hence, from citizens' point of view, e-Government solutions should mainly be fast, cheap, and convenient to use. In other words, e-Government solutions should provide an appropriate level of usability. Hence, there is obviously a close correlation between the usability of an e-Government solution and its efficiency. This obvious correlation between the aspects efficiency and usability has been discussed in detail by Frokjaer et al. (2000). From a citizen-centric view on the term efficiency, usability can hence be derived as key requirement and success factor of efficient e-Government solutions that aim to satisfy citizens' needs (Gil-García and Pardo 2005).

While usability definitely represents a key requirement for efficient e-Government solutions especially from the citizens' point of view, also other aspects need to be taken into account. As citizens have a strong interest that private data being processed in e-Government processes are appropriately protected, suitable security measures usually need to be integrated into nowadays e-Government solutions.

Security is of special importance for complex transactional applications that potentially comprise transmission and processing of security-critical and privacy-sensitive data. This has been discussed by Zavareh et al. (2012) in detail. The findings obtained by Zavareh et al. are consistent with several other studies, such as those from Geetha and Malarvizhi (2010), Howcroft et al. (2002), and White and Nteli (2004). To meet given security requirements, usually approved cryptographic methods such as strong user authentication schemes and electronic signatures are employed. In this context, electronic signatures play an important role especially in the European Union, where qualified electronic signatures are legally equivalent to handwritten signatures according to the Directive 1999/93/EC of the European Parliament and of the Council on a Community framework for electronic signatures (EU Parliament and Council 2000).[1] To meet the requirements of qualified signatures as defined in this directive, a secure signature-creation device (SSCD) has to be used to securely store cryptographic keys and to compute electronic signatures.

Since SSCDs typically rely on a secure hardware token, implementation alternatives are limited. Most e-Government solutions require citizens to use personalized

[1] At the present time, this Directive is still the basis for electronic signatures across Europe. However, the European Commission is currently working on a new proposal for a regulation (EU Parliament and Council 2012).

smart cards in order to create legally binding electronic signatures. Smart card-based solutions are already in productive operation in Austria, Belgium, Estonia, Portugal, Spain, and various other European countries (Siddhartha 2008). Some European countries such as Austria or Estonia additionally provide citizens mobile signature-creation solutions. These solutions use mobile phones as hardware tokens instead of smart cards.

Regardless of the type of the used hardware token, the question arises how these tokens can be used and accessed by e-Government applications, e.g., to securely authenticate citizens or to create electronic signatures. Currently, most countries rely on some kind of middleware, which acts as intermediary between hardware tokens and e-Government applications. This approach is also followed in Austria, where several different middleware implementations have been developed during the past decade. These implementations allow Austrian citizens to securely authenticate at remote services and to create qualified electronic signatures by using either personalized smart cards or their personal mobile phones.

Representing core components of the Austrian e-Government infrastructure, the different middleware implementations being currently in use in Austria have oftentimes proven to be secure and to be able to meet given functional requirements. Unfortunately, in general there is a well-known trade-off between the security and the usability of IT solutions. Highly secure solutions usually tend to be less usable and vice versa (Schultz et al. 2001). Hence, it still can be observed that citizens often hesitate to actively use provided e-Government services due to lacking usability. We tried to find out the main reasons for this lack of user acceptance by conducting a usability analysis. The basic goal of this analysis was to measure and compare the usability and hence the efficiency of different Austrian middleware implementations in order to identify persisting weaknesses and to find out user preferences. Given the close correlation between usability and efficiency, the conducted usability evaluation has revealed interesting insights on the usability and efficiency of the assessed e-Government components. In this chapter we introduce the evaluated components of the Austrian e-Government infrastructure, discuss the followed methodology of the conducted usability test, and present obtained results.

The chapter is structured as follows. Section 4.2 discusses basic requirements of current e-Government applications and emphasizes existing trade-offs between security and usability. Subsequently, Sect. 4.3 introduces relevant concepts and components of the Austrian e-Government infrastructure. The methodology that has been followed to assess the usability of these core components is discussed in Sect. 4.6. Results of the conducted usability analysis are presented in Sect. 4.8. Finally, conclusions are drawn.

4.2 Requirements of E-Government Applications

Development and implementation of successful e-Government solutions are non-trivial tasks. Identification of critical success factors have been discussed for instance by Gil-Garcia (2007) and Altameem et al. (2006). If we focus on the development of

transactional e-Government applications that require remote interaction with citizens, then security, usability, and efficiency turn out to be key for success. We will discuss these three requirements, their relation to each other, and possible implications in the following in more detail.

4.2.1 Security

Security is crucial for most governmental and administrative procedures. If citizens go to a public office, e.g., to file an application, they usually have to prove their identity first by showing a valid identification document. Furthermore, citizens have to sign their applications in order to confirm that all data are correct and to preclude later repudiation. Reliable identification of citizens and handwritten signatures have been key concepts of governmental procedures for many years.

When such procedures are mapped to the digital world, the basic requirements remain the same. However, when using e-Government applications, citizens are not required to go to public offices any longer. Instead, they make use of their PC, laptop, or mobile device to carry out administrative procedures. Still, citizens need to be identified reliably and need to sign their applications to meet given security requirements.

The requirement for security in e-Government applications hence directly leads to the requirement for secure and reliable authentication mechanisms and to the requirement for a secure electronic pendant to handwritten signatures. Both requirements can be met by applying approved cryptographic methods such as two-factor authentication[2] schemes and qualified electronic signatures. We will discuss later how these methods are used in Austrian e-Government solutions to provide an appropriate level of security.

4.2.2 Usability

Usability is another key success factor of e-Government solutions that heavily influences user acceptance and that is closely related to efficiency (Frokjaer et al. 2000). Unfortunately, the demand for usability often conflicts with the demand for security. This problem has been discussed by Schultz et al. (2001). While an appropriate level of security requires the application of complex cryptographic methods and protocols, the increased complexity often significantly affects usability.

This dilemma is comparable to the problem regarding accessibility in e-Government applications that has been discussed by Orthacker and Zefferer (2011). The authors conclude that accessibility is crucial for e-Government

[2] Two-factor authentication defines an authentication approach requiring the presentation of two different authentication factors, e.g., something the user possesses (e.g., smart card) and something the user knows (e.g., password).

applications but is often difficult to achieve in practice due to limited implementation alternatives caused by given security requirements.

It is thus less surprising that usability is often neglected in current e-Government applications. This is problematic and potentially leads to scenarios in which e-Government applications can be used by expert and technical-affine users only. This phenomenon has become commonly known under the term *digital divide* (Norris 2003). To counter digital divide and to make e-Government solutions usable and efficient for all social and educational classes, usability has to be recognized as important requirement for efficient e-Government applications and solutions.

4.2.3 Efficiency

The basic goal of e-Government initiatives and solutions is to speed up governmental procedures to save time and costs. In this context, efficiency is of course an important aspect, since the efficiency of an e-Government solution is directly proportional to its potential to save money. However, the term efficiency can actually have different meanings for different stakeholders of e-Government solutions. Especially for citizens, usability is a key aspect of efficient e-Government solutions, as usable solutions provide more potential to save time when doing governmental procedures online. Since usable solutions are more likely to be frequently used by citizens, usability is also an important issue for governments providing e-Government solutions, which are willing to tap the full potential of their electronic services. Hence, usability, which has already been defined as key requirement above, is actually closely related to the requirement for efficiency (Chircu and Hae-Dong Lee 2005). Hence, when designing and implementing efficient e-Government solutions, usability definitely needs to be taken into account.

Considering the well-known trade-off between usability and security and taking into account the close relation between usability and efficiency, the integration of security-enhancing technologies into e-Government solutions can be a serious challenge in practice, which needs to be tackled.

4.3 e-Government in Austria: Concepts and Core Components

Security, usability, and efficiency have been defined as crucial requirements of e-Government solutions. It has been shown that there is a close correlation between efficiency and usability, while at the same time security and usability requirements are often contradictory. This section discusses the Austrian approach to cope with this situation and to meet all given requirements. For this purpose, basic concepts and core components of the Austrian e-Government infrastructure are briefly sketched in the following subsections.

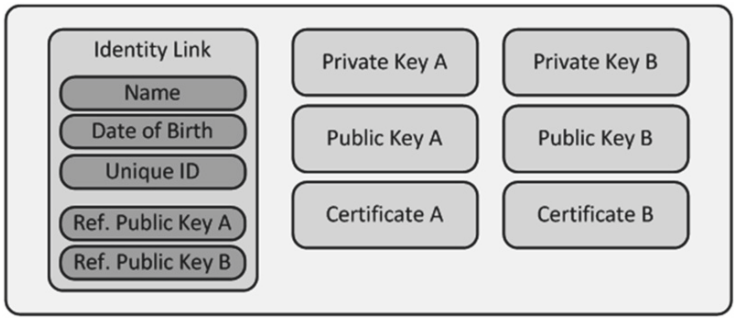


Fig. 4.1 Basic architecture of the Austrian Citizen Card

4.3.1 The Austrian Citizen Card Concept

The key concept of the Austrian e-Government infrastructure is called *Citizen Card* (*CC*). The CC is an abstract definition of a cryptographic token that allows citizens to securely authenticate at e-Government services and to create qualified electronic signatures. According to its specification (Hollosi et al. 2008), a CC securely stores cryptographic keys, which allow citizens to create qualified electronic signatures. Furthermore, a CC contains an XML-based data structure called *Identity Link*. The Identity Link itself contains—among others—the citizen's name, her unique ID, and references to the citizen's cryptographic public keys. This way, the Identity Link unambiguously links the citizen's identity with her personal cryptographic keys. Figure 4.1 summarizes the relevant components of the Austrian Citizen Card.

The CC concept perfectly meets the predefined security requirements. The identifier stored on the CC allows citizens to be unambiguously identified and authenticated at e-Government services. Since Austria is a member state of the European Union, the CC concept complies with the EU Signature Directive and fulfills all requirements of a SSCD. Thus, the CC allows citizens to create qualified electronic signatures that are legally equivalent to handwritten signatures.

Although the term Citizen Card might suggest the use of smart cards, the CC specifications are abstract and not limited to a certain technology. This flexibility has led to the development of different CC implementations during the past decade. These implementations can be classified into two categories. In *smart card-based approaches*, the CC is implemented by a smart card. For instance, Austrian citizens can use their health insurance card as CC. Alternatively, also bank account cards or smart card-based ID documents can be used as CC after an appropriate activation and personalization process.

Mobile approaches represent the second category of CC implementations. Mobile approaches render the use of smart cards unnecessary and make use of the citizen's mobile phone to achieve an adequate level of security. The Austrian Mobile Phone Signature, which is based on a concept that has been introduced by Orthacker et al. (2010), is the main representative of this category and currently in productive operation in Austria.

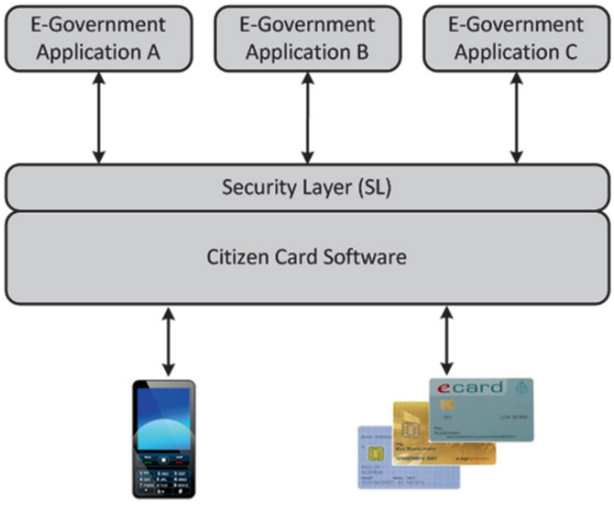


Fig. 4.2 Architecture of the Security Layer

4.3.2 Application Integration

Irrespective of the underlying technology, all CC implementations facilitate secure user authentication and creation of qualified electronic signatures in e-Government processes. The technology-neutral concept guarantees that each citizen can individually choose her preferred implementation.

Of course, this flexibility increases the complexity of application-development processes. In order to integrate CC functionality without preferring a particular solution, e-Government applications would need to support all available CC implementations. Also, the introduction of new CC implementations would cause significant maintenance costs for already deployed applications. In order to overcome this problem, the Austrian e-Government strategy follows a middleware-based approach.[3]

Figure 4.2 illustrates the basic architecture of this middleware approach. Central element is the so-called *Security Layer* (*SL*) interface, which has been introduced and discussed by Leitold et al. (2002). The Security Layer is an abstract XML-based interface that can be used by e-Government applications to easily access Citizen Card functionality. This way, applications do not need to integrate different and special CC implementations. Actually, applications do not even need to be aware of the used implementation, since all implementations can be accessed through a common interface. All implementation-specific functionality is outsourced to the

[3] In this context, a middleware constitutes an intermediary layer between the application and the underlying CC implementation. The middleware thereby hides CC-implementation specifics and provides easy access to CC functionality for the application.

so-called *Citizen Card Software* (*CCS*). The CCS implements access to specific CC implementations (e.g., smart cards) and provides their functionality through the common SL interface.

Acting as middleware between e-Government applications and CC implementations, the CCS plays a significant role in the Austrian e-Government infrastructure. This raises the question how the CCS can be implemented efficiently in practice. This question will be answered in the following subsections for smart card-based CC implementation approaches as well as for mobile CC implementation approaches.

4.4 CCS for Smart Card-Based Approaches

After introduction of the CC concept in Austria in 2002, smart card-based approaches have soon been available for citizens. First, smart cards following the CC specification have already been issued in 2002. Today, citizens can use their health insurance cards as CC for free, making this a popular smart card-based alternative.

The efficient implementation of CCS acting as middleware between smart cards and e-Government application is no trivial task and still subject to ongoing research. The most obvious approach is the use of software that has to be installed by citizens on their local computers. This approach is illustrated in Fig. 4.3. The locally installed software (i.e., the CCS) communicates with locally connected smart cards over the PC/SC[4] protocol and provides their functionality to e-Government applications through the standardized SL interface. Applications can access this interface through the citizen's Web browser and a local network socket that is opened by the CCS.

For many years, this approach has been the only available alternative. Still, local CCS solutions are offered by different vendors such as the Austrian certification

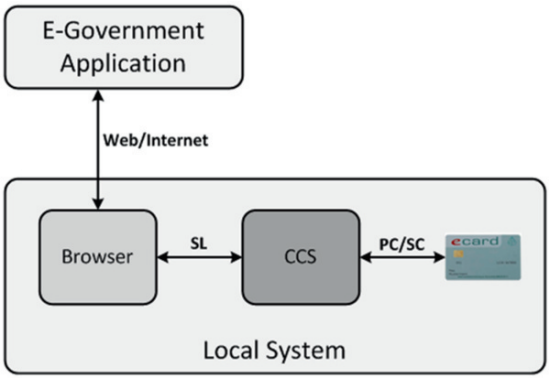


Fig. 4.3 Local Citizen Card Software (CCS) implementations

[4] http://www.pcscworkgroup.com

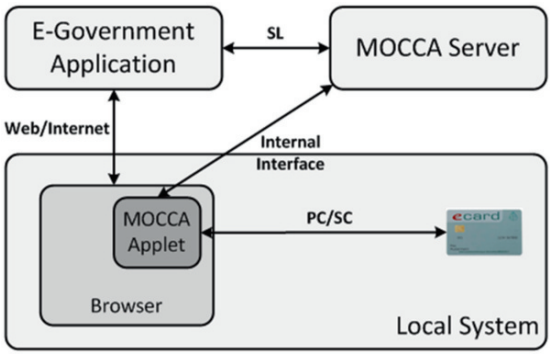


Fig. 4.4 General architecture of MOCCA Online

authority A-Trust[5] or the Austrian software company IT Solution.[6] The only open-source solution following the local approach is called *MOCCA Local*. The MOCCA (Modular Open Citizen Card Architecture) project[7] has been started in 2008 with the goal to provide open-source CCS solutions for Austrian citizens.

MOCCA Local features a minimalistic user interface and typically runs in the background. If access to a locally connected smart card is requested by an e-Government application, a small window pops up. Through this window, users are provided with relevant information (e.g., the data to be signed) and required user input (e.g., secure PIN to authorize the signature-creation process) is collected.

The local approach, which is shown in Fig. 4.3, works fine from a functional perspective. However, several years of field experience have revealed several drawbacks of this solution (Kubicek 2011). The main problem of this approach is the need to install the CCS on the local system. It turned out that this can be a severe problem especially for inexperienced users. Also, the need for a local software installation renders this approach infeasible in situations in which citizens do not have the required privileges to install software on the used system.

To overcome these problems, the MOCCA project has also investigated technical capabilities of an installation-free alternative. These efforts finally led to the development of *MOCCA Online*, an installation-free CCS. The basic architecture of MOCCA Online has been discussed in Centner et al. (2010) and is shown in Fig. 4.4. MOCCA Online follows a server-based approach. The SL interface is not implemented by locally installed software, but by the central MOCCA Server component. e-Government applications contact the MOCCA Server in order to access citizens' smart cards. Physical access to the locally connected smart card is implemented by a Java Applet running on the citizen's local system. MOCCA Applet and MOCCA Server together represent the CCS and communicate with each other through an internal interface. The MOCCA Applet is usually integrated in the Web front-end of

[5] http://www.a-trust.at/info.aspx?ch=2&lang=GE&node=733

[6] http://www.itsolution.at/trustDesk-basic.html

[7] https://joinup.ec.europa.eu/software/mocca/home

e-Government applications by means of an HTML IFRAME element. This way, the used Web browser can act as user interface for the provision of relevant information (e.g., the data to be signed) and the collection of required user input (e.g., PINs).

Since all required communication steps with locally connected smart cards are implemented by an automatically deployed Java Applet, no manual software installation is needed. The only requirement for the client system is availability of a current Java Runtime Environment (JRE).

Compared to local CCS approaches such as MOCCA Local, MOCCA Online is easier to use as it does not require any software installation. Unfortunately, local software installations are not the only barrier that can be identified for smart card-based solutions. For many citizens, the use of smart cards itself is already problematic as this requires appropriate reader devices.

4.5 CCS for Mobile Approaches

The goal to render smart cards completely unnecessary has been the main driver behind the development of mobile CCS solutions. In Austria, the so-called *Mobile Phone Signature*, which is based on a concept that has been discussed in 2010 by Orthacker et al. (2010), represents a mobile alternative to established smart card-based approaches. The general architecture of the Mobile Phone Signature is shown in Fig. 4.5.

Similar to MOCCA Online, a central service (Mobile Phone Signature Service) implements the SL interface. Instead of a smart card, a hardware security module (HSM) that is attached to this central service acts an SSCD. The HSM is capable of creating qualified electronic signatures on behalf of the citizen. To access CC functionality, e-Government applications send an appropriate request to the Mobile Phone Signature Service. Provision of the requested functionality (e.g., signature creation) or data (e.g., Identity Link) has to be authorized by the citizen. Therefore, the Mobile Phone Signature Service requests the citizen to enter her phone number and a secret password through a Web form. This Web form is usually integrated into e-Government applications by means of an HTML IFRAME element.

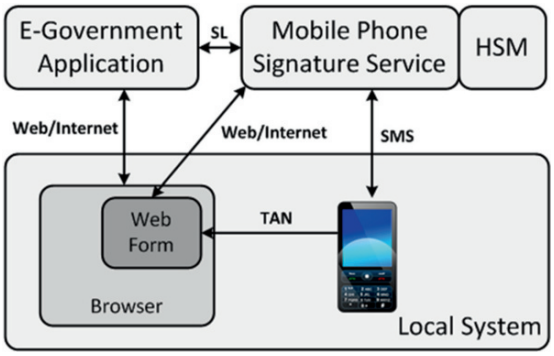


Fig. 4.5 General architecture of the Austrian Mobile Phone Signature

If the provided credentials can be verified correctly, an SMS message is sent to the citizen's mobile phone containing a one-time password[8] (TAN). This TAN has to be entered in the Mobile Phone Signature Service's Web form to authorize execution of the e-Government application's request.

The main advantage of this mobile approach is the central HSM, which renders smart cards unnecessary. By relying on a strong two-factor authentication scheme that makes use of two separated communication channels (i.e., Web and SMS), a sufficient level of security is assured.

4.5.1 Usability Considerations

The importance of usability—also as criterion for measuring efficiency—in e-Government has been subject to ongoing research for many years. However, most work has focused on the usability of e-Government Web sites so far. For instance, a quality inspection method for the evaluation of e-Government sites has been proposed by Garcia et al. (2005). In Ma and Zaphiris (2003), the authors have evaluated the usability of different e-Government Web sites in the UK. Recently, also the usability of Norwegian e-Government Web sites has been discussed by Sørum (2011).

Without doubt, usability of e-Government Web sites is an important topic. However, integration of security-enhancing technologies such as smart cards into Web-based e-Government applications definitely needs to be considered as well. Otherwise, usability evaluations of current e-Government solutions threaten to remain incomplete and to miss relevant aspects.

According to the Austrian e-Government strategy, security-enhancing technologies are integrated into e-Government applications by means of different CCS implementations. Currently, MOCCA Local, MOCCA Online, and the Austrian Mobile Phone Signature represent frequently used implementations of Citizen Card Software. We have assessed the usability of these three components to identify persisting usability problems and to analyze user preferences in order to further improve the efficiency of the Austrian e-Government. Details of the conducted usability test are provided in the next section.

4.6 Methodology

A usability test has been conducted to assess the usability and to measure the efficiency of the three CCS implementations that have been introduced in the previous section. In this section, relevant aspects of the methodology that has been followed

[8] A one-time password constitutes a password which is valid for one transaction or one login only.

for the conducted usability test are discussed. We define research questions to define the scope of the conducted usability test first. In the following, we present the applied test method and introduce the used test setup. Finally, we discuss the concrete design of the conducted test and discuss details regarding the selection and classification of test users.

4.6.1 Research Questions

The general goal of this empirical study was to find out which CCS implementation model is favored by Austrian citizens to further improve usability and efficiency in e-Government services. This investigation was based on a thorough usability analysis of the individual CCS implementations. The usability of MOCCA Local, MOCCA Online, and Mobile Phone Signature has been assessed by means of the following research questions. By answering these research questions, we attempted to find out whether the different CCS implementations satisfy usability requirements and are able to achieve an appropriate level of user acceptance:

Q1. Do required software installations represent a barrier and reduce usability?
Q2. Does reliance on Java-based solutions cause any additional usability issues?
Q3. How do users rate the overall usability of MOCCA Local, MOCCA Online, and the Mobile Phone Signature?
Q4. How do users rate the security and trustworthiness of MOCCA Local, MOCCA Online, and the Mobile Phone Signature?
Q5. Which CCS implementation variant do users prefer in general?

To answer these questions, a usability test has been conducted. Details of the applied test method and the used test setup are provided in the following section.

4.6.2 Test Method and Setup

We have applied a thinking-aloud test with 20 test users to evaluate the usability of different Austrian CCS implementations. We have chosen this number of test users, as this is a sufficiently large number to produce reliable and meaningful results (Nielsen 2013). The basic test run was identical for all 20 test users and consisted of the following four phases.

P1. Welcome: Test users have been welcomed, have been provided with relevant information about the usability test, and have been asked to sign a nondisclosure agreement.
P2. Background questionnaire: At the beginning of the usability test, relevant information about the participating test user has been collected using a prepared questionnaire.

P3. Execution of tasks: In this phase, test users have been asked to carry out a sequence of well-defined tasks using the three CCS implementations to be evaluated. After each task, the test user has been asked to fill out a prepared questionnaire and to rate the tested component (post-task rating).
P4. Conclusive interview: After completion of all tasks, a conclusive interview has been conducted with the test users. After the interview, test users have been asked to fill out a final questionnaire (post-study rating) covering some general questions.

During Phase P3, test users have been asked to carry out predefined tasks using an off-the-shelf desktop PC. In order to use a common configuration, all tests have been carried out using the *Microsoft Windows 7* operating system and *Microsoft Internet Explorer* 8 Web browser. The desktop PC was equipped with a *Reiner* SCT card reader device. Test users were not allowed to use other system configurations (e.g., a different Web browser) as this would have rendered direct comparisons between test users difficult.

The used test system was equipped with Morae Recorder software.[9] The use of this software allowed tracking and recording of user sessions including all user activities such as mouse movements and keyboard inputs. Additionally, comments and facial expressions of test users have been recorded with a Web cam and stored together with the recorded user session for later analysis. To be able to record all user comments during Phases P2 and P4, we have additionally used a standard camera to record the entire test.

The filled questionnaires have represented the most important data sources for later analysis. To obtain as much valuable feedback as possible, we relied on semantic differentials. The method of semantic differentials has been discussed by Boslaugh and Watters (2008) and is frequently used in social sciences and user-experience research. In general, semantic differentials are used to measure the connotative meaning of an object and to further derive the attitude towards this object. We used semantic differentials to allow users to assign weighted properties to the evaluated software components.

Besides the filled questionnaires, also the recorded user sessions and user comments have been incorporated in the analysis process. These data have turned out to be extremely helpful in order to understand the collected user feedback and to identify reasons for negative ratings. Obtained results of the evaluation process will be presented in Sect. 4.8.

4.6.3 Tasks

Most relevant information has been collected during Phase P3 of the usability test, i.e., during the execution of predefined tasks. We have defined these tasks such

[9] http://www.techsmith.com/morae.html

that answers to the predefined research questions could be derived easily from the collected data. All test users have been asked to carry out five tasks. For these tasks we have set our focus on the typical standard use case within the Austrian e-Government. This includes the installation process of the required software for the Citizen Card Software MOCCA Local and MOCCA Online, the activation of the Mobile Phone Signature, and using a typical demo e-Government application[10] with these three CCS implementations. A valid smart card-based Citizen Card was the only prerequisite for test users. In the following we elaborate on the tasks related to the smart card-based and mobile phone-based CCS implementations.

Tasks related to the smart card-based CCS implementations MOCCA Local and MOCCA Online:

T1. Install the Citizen Card Software MOCCA Local on the local system.
T2. Use MOCCA Local to file a demo e-Government application.
T3. Use MOCCA Online to file a demo e-Government application.

MOCCA Local and MOCCA Online are Java-based solutions. To cover all possible real-life scenarios, the used test system has been provided without a JRE. The JRE had to be installed by the test user during the test run. Both MOCCA Local and MOCCA Online automatically check for an installed JRE upon start-up and guide users through the Java installation process if no JRE is found on the local system. In order to be able to evaluate the usability of this functionality for both MOCCA variants, we split the test users randomly into two groups. Group A started with Task T1 as shown above. Hence, this group had to install Java during the installation process of MOCCA Local. In contrast, Group B was asked to start with Task T3 followed by T1 and T2. This way, users of this group had to install Java during the first use of MOCCA Online.

By splitting test users into two groups, we were able to directly compare the integration of the Java installation process into MOCCA Local and into MOCCA Online. Furthermore, we were able to cancel out learning effects that would otherwise have biased obtained results.

After completing T1–T3, the users had to execute the following tasks related to the Mobile Phone Signature:

T4. Use your smart card-based Citizen Card to activate the Mobile Phone Signature for your mobile phone.
T5. Use the Mobile Phone Signature to file a demo e-Government application.

The activation of the Mobile Phone Signature can be done by users themselves, using an existing smart card-based Citizen Card. Hence, the only prerequisite was to have a valid smart card-based Citizen Card and having a smart card-based CCS implementations installed.[11]

[10] This typical demo e-Government application consists of filling out a form and signing it (using a CCS) afterwards.

[11] Therefore, the test users have been requested to execute Tasks T1, T2, and T3 beforehand.

4.7 Test Users and User Group

The usability test has been conducted with 20 test users in total. According to Nielsen (2013), this is a sufficient number of test users to obtain reliable results. In order to obtain meaningful results, we have selected a representative sample of the Austrian population for our test.

As explained above, test users have been randomly assigned to two different user groups. Depending on the assigned group, test users have been asked to execute the predefined tasks in a different order. While the assignment of users to Group A and Group B was completely random, users have additionally been assigned to different user groups according to several personal characteristics. This way, we have split test users according to their age, education, and technical experience. Table 4.1 lists all predefined user groups.

4.8 Results

The goal of the conducted usability test was to answer the five research questions defined in the previous section. Results of the conducted usability test and answers to these research questions are presented and discussed in this section.

4.8.1 Usability of Installation-Based CCS

To answer Research Question Q1, we assessed whether the required installation process of MOCCA Local represents a barrier for users and hence reduces the usability of local CCS implementations. To install MOCCA Local using Java Webstart[12] technology, test users had to navigate to a given Web site and click a "Launch" button. After that, test users were asked to manually install a certificate into the used Web browser.

Table 4.1 User groups

Group ID	Description	Users
Group ALL	This group comprises all test users	20
Group A	Users of this group started with Task T1	10
Group B	Users of this group started with Task T3	10
Group 30+	Users of this group were more than 30 years	8
Group 30−	Users of this group were 30 or less years old	12
Group U	Users of this group had a university degree	12
Group NU	Users of this group had no university degree	8
Group T	Users of this group had a technical education	7
Group NT	Users of this group had no technical education	13

[12] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136112.html

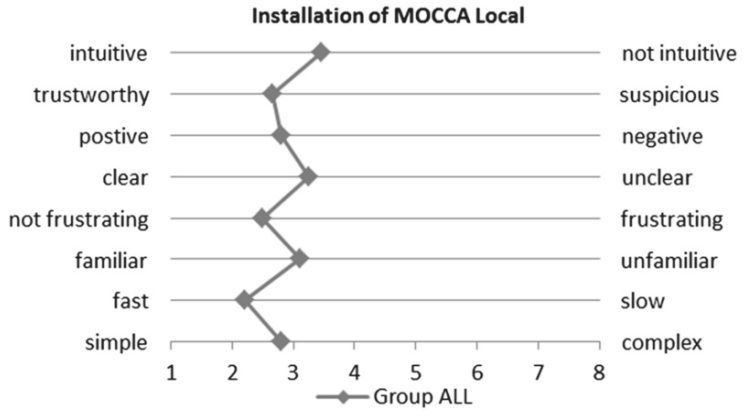


Fig. 4.6 Evaluation results of the installation process of MOCCA Local

Figure 4.6 shows that in general most test users rated the installation process positively. This corresponds to the observations that have been made during the test runs. Most users were able to successfully complete the installation on their own.

A user group-specific analysis yielded several interesting results. For instance, it turned out that users of Group B rated the installation process of MOCCA Local better than users of Group A. This is probably due to the fact that users of Group B already had the chance to gain experience with another CCS implementation, i.e., MOCCA Online, before.

Also the educational level of users has influenced the rating of the installation process. University graduates rated all aspects of the installation process more positively than nongraduate users. An analysis of the recorded user sessions revealed that especially the required certificate installation was problematic for those users. The reason is probably that the use of digital certificates is not well known to technically inexperienced users. However, this step is not directly related to the CCS implementation.

Interestingly, neither the age nor the technical background of users has influenced the obtained results significantly. Details of group-specific results are illustrated in Fig. 4.7. To answer Research Question Q1, we can conclude that a required software installation process does not raise severe usability issues. Still, installation routines should be simple and intuitive in order to make this a feasible task also for inexperienced users.

4.8.2 Usability Issues of Java-Based Approaches

In order to answer Research Question Q2, we tried to find out whether reliance on Java-based approaches raised any usability issues. Since both MOCCA Local and MOCCA Online represent Java-based solutions, test users had to install Java either

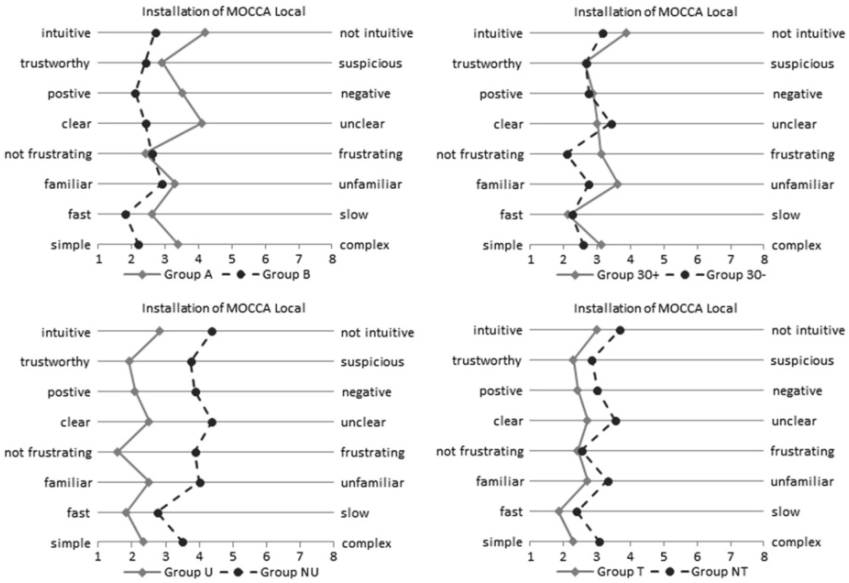


Fig. 4.7 Group-specific evaluations of the installation process of MOCCA Local

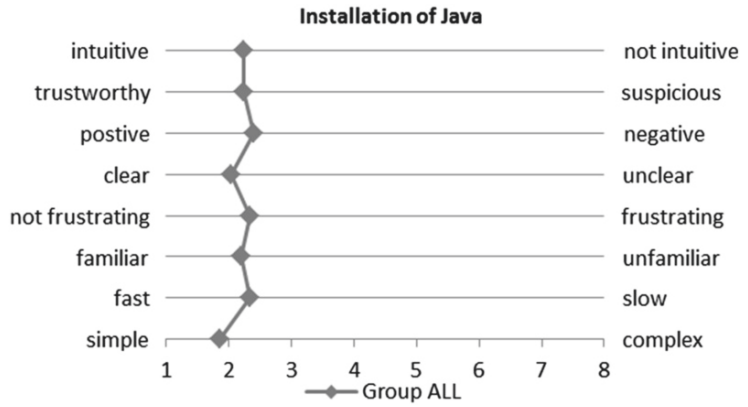


Fig. 4.8 Evaluation results of the Java installation process

during the installation of MOCCA Local (Task T1) or during the first usage of MOCCA Online (Task T3), depending on the assigned user group. The conducted usability analysis revealed that hardly any user had problems with the Java installation process. Therefore, the Java installation process and its integration into MOCCA Local and MOCCA Online have been rated positively by most users. Figure 4.8 illustrates these results.

The user group-specific analysis again yielded interesting results. We expected users without technical background and with higher age to have more problems

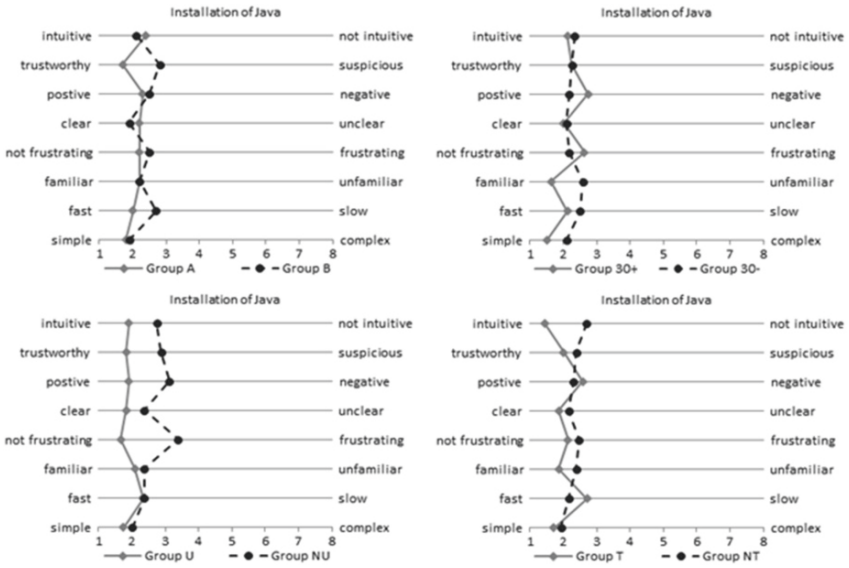


Fig. 4.9 Group-specific evaluation of the installation process of Java

with the Java installation process. Hence, we expected worse ratings from these user groups. However, the obtained results have shown that there are only marginal differences between these user groups. Again, users with a university degree rated the Java installation process more positively than nongraduate users. Details of the obtained user group-specific results are illustrated in Fig. 4.9.

To answer Research Question Q2, we can conclude that reliance on Java does obviously not raise significant usability issues. Most users had no problems to start and complete the Java installation process on their own. Surprisingly, Java was well known to virtually all test users. Analysis of the recorded user sessions revealed that most problems were encountered right after the installation process. Users were left on the Java Web site after the installation process and not automatically redirected back to MOCCA Local or MOCCA Online, respectively. This caused confusion with some test users. Although this has not significantly affected the overall rating of the Java installation process, this issue should be addressed.

4.8.3 Usability of Different CCS Implementations

According to Research Question Q3, we have analyzed how MOCCA Local, MOCCA Online, and the Mobile Phone Signature have been rated by the test users. Test users have been asked to file a demo e-Government application using their Citizen Card and each of the three evaluated CCS implementations as defined by Tasks T2, T3, and T5. We discuss obtained results for the three CCS in the following subsections.

4.9 Evaluation of MOCCA Local

Figure 4.10 shows that in general the use of MOCCA Local has been rated positively by all test users. This corresponds to the observations that have been made during the test runs. Most test users were able to complete the assigned task using MOCCA Local without any problems.

Comparison of the results obtained for Group A and Group B yielded interesting results. The use of MOCCA Local has been rated more positively by test users of Group B. This is probably due to learning effects. Test users of Group B started with the evaluation of MOCCA Online. Hence, these users were already more familiar with the handling of their Citizen Card than test users of Group A, who started directly with the evaluation of MOCCA Local.

Significant differences could again be identified between graduate and nongraduate test users. Again, ratings from nongraduate users were more negative than ratings from graduate users. The user's technical background had a similar impact on the user ratings. However, differences between users with technical background and users without technical background were not as significant as between graduates and nongraduates. Contrary, the user's age had no significant impact on the rating of MOCCA Local. Details of group-specific results are provided in Fig. 4.11.

4.10 Evaluation of MOCCA Online

Similar to MOCCA Local, also the use of MOCCA Online has been rated predominantly positive. Figure 4.12 illustrates the obtained results. Again, most users were able to complete the assigned task and to file a demo e-Government application using MOCCA Online on their own.

The group-specific analysis revealed that this time users of Group A provided more positive ratings than users of Group B. This result supports the theory that

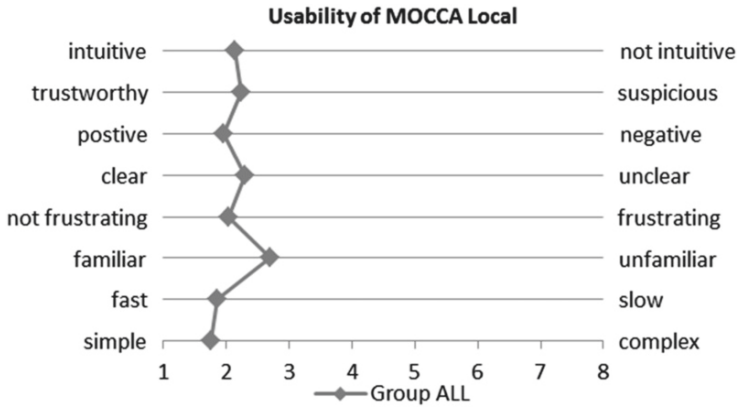


Fig. 4.10 Usability evaluation results of MOCCA Local

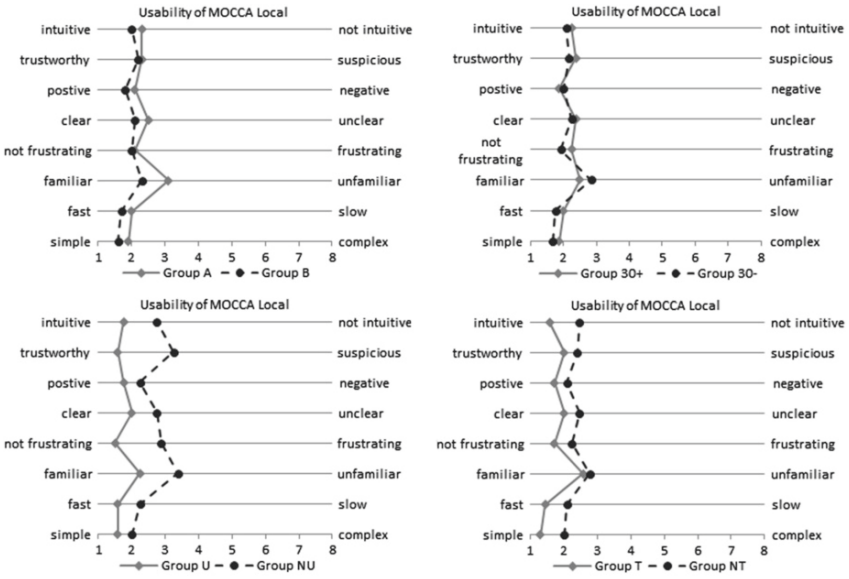


Fig. 4.11 Group-specific evaluation of MOCCA Local user experience

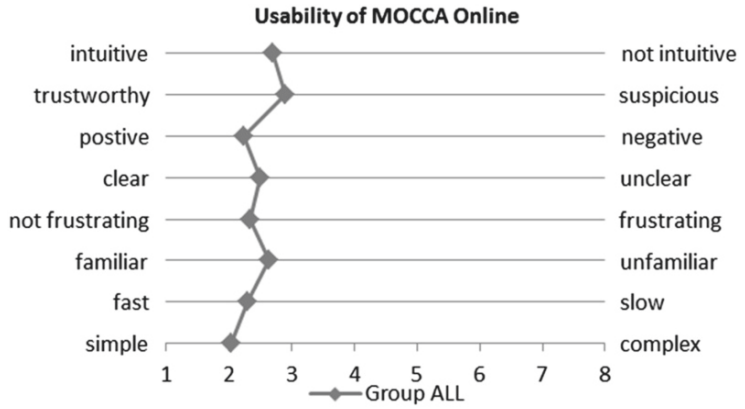


Fig. 4.12 Usability evaluation results of MOCCA Online

gained experience and learning effects influence user ratings. Again, users, who had already evaluated MOCCA Local before, rated MOCCA Online better.

Analysis of other group-specific results yielded interesting results. Surprisingly, users with technical background rated the use of MOCCA Online less positively than non-technicians. Significant differences in the obtained results could also be observed between graduate and nongraduate users. Again, nongraduates rated MOCCA Online less positively in most aspects. Details of obtained group-specific results are provided in Fig. 4.13.

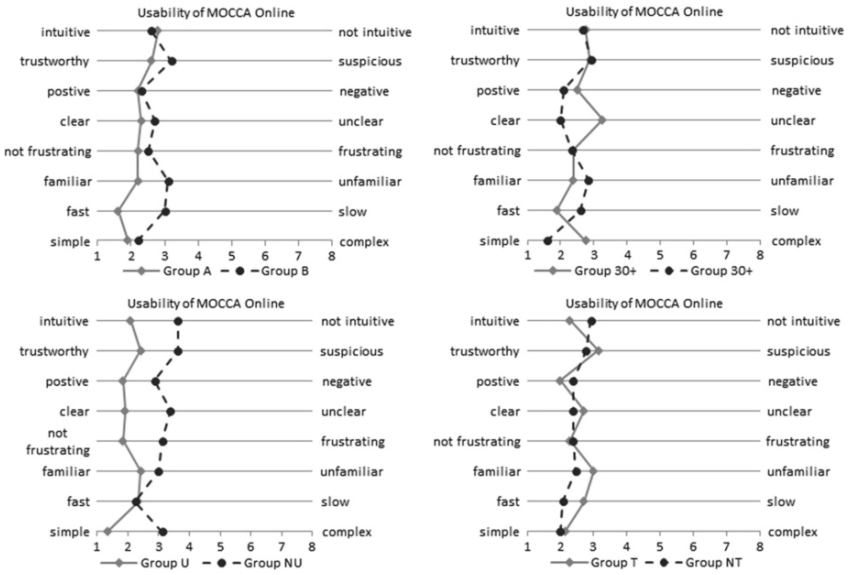


Fig. 4.13 Group-specific evaluation of MOCCA Online use

4.11 Evaluation of the Mobile Phone Signature

Test users were also asked to file a demo e-Government application using their mobile phone and the Austrian Mobile Phone Signature and to rate the usability of this approach. Figure 4.14 shows that also the use of the Mobile Phone Signature has been rated mainly positively.

Comparison of group-specific results shows that again graduate users rated the usability more positively that nongraduates. This time, no significant differences could be observed between the results of Group A and Group B. This is comprehensible, since both groups have evaluated the Mobile Phone Signature after using MOCCA Local and MOCCA Online. Thus, users of both groups had the same level of experience before testing the Mobile Phone Signature.[13] Group-specific results are provided in Fig. 4.15.

[13] Note that we were forced to schedule the evaluation of the Mobile Phone Signature after evaluation of the two smart card-based CCS implementations. This was due to the fact that the activation process of the Mobile Phone Signature was part of the usability test (Task T4). Since the activation process required a Citizen Card-based user authentication, either MOCCA Local or MOCCA Online was required to activate the Mobile Phone Signature.

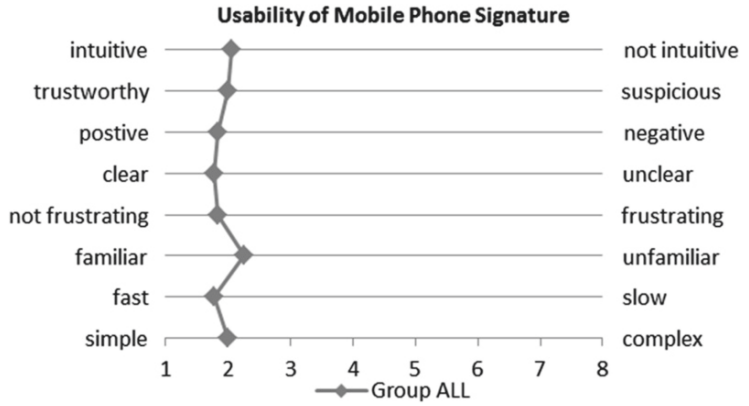


Fig. 4.14 Usability evaluation results of Mobile Phone Signature

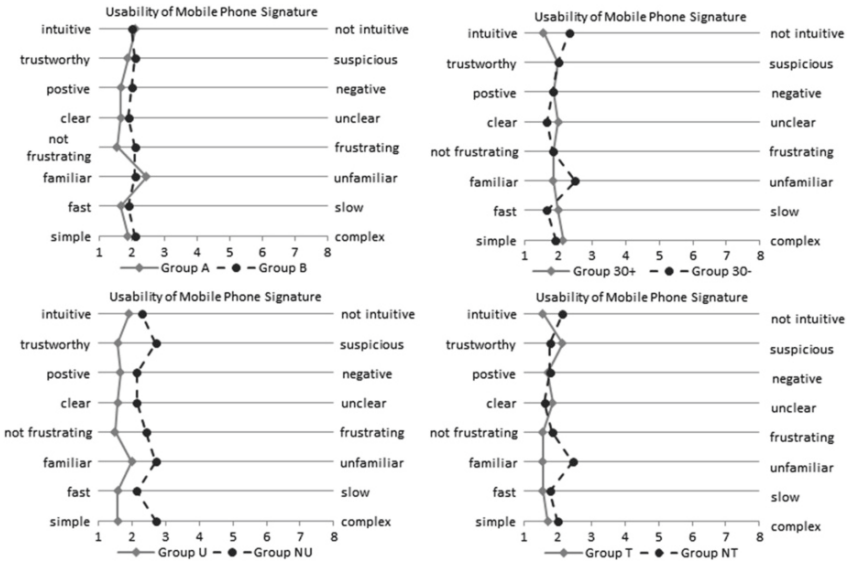


Fig. 4.15 Group-specific evaluation of Mobile Phone Signature

4.12 Comparison of Different CCS Implementations

A direct comparison of the obtained ratings for all three CCS implementations is shown in Fig. 4.16. In total, the usability of the Mobile Phone Signature has been rated best, followed by MOCCA Local.

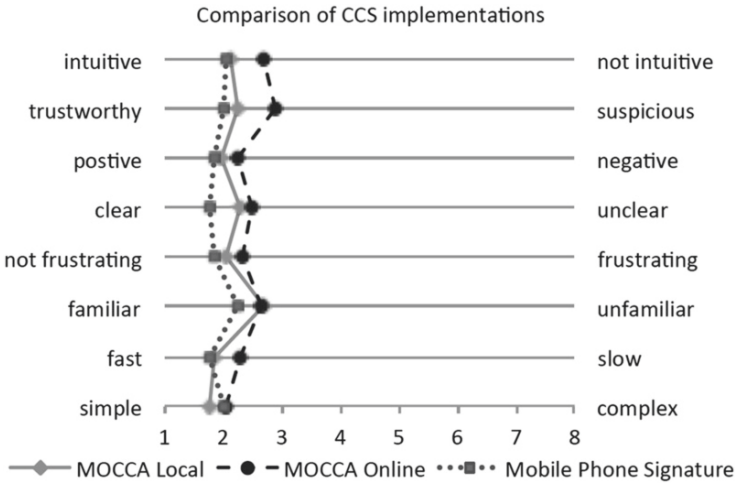


Fig. 4.16 Comparison of different CCS implementations

In order to answer Research Question Q3, we can conclude that the usability of all three CCS implementations has been rated positively. Significant differences can be mainly identified between graduate users and nongraduates. The latter have rated all three CCS implementation less positively. In total, the Austrian Mobile Phone Signature has achieved the best results and thus seems to be the favored solution for Austrian citizens.

4.12.1 Security and Trustworthiness

Besides usability, the security and trustworthiness of used components is crucial for the acceptance of e-Government solutions. According to Research Question Q4, we have analyzed whether the three evaluated CCS implementations appear secure and trustworthy for users. To answer this question, test users have been asked to rate the perceived level of security and trustworthiness for all three CCS implementations. The ratings have been collected by means of a questionnaire.

Figure 4.17 illustrates the obtained results for MOCCA Local. In general, the majority of users rated MOCCA Local to be secure and trustworthy. Only few test users assumed this CCS to be insecure and not trustworthy at all. Users of Group B rated the security and trustworthiness of MOCCA Local more positively than users of Group A. Younger test users regarded MOCCA Local with more suspicion than older users. Also nongraduate users turned out to be slightly more skeptical than graduates. Similar differences could be observed between technicians and users without technical background. The latter regarded MOCCA Local with more suspicion than technically experienced users.

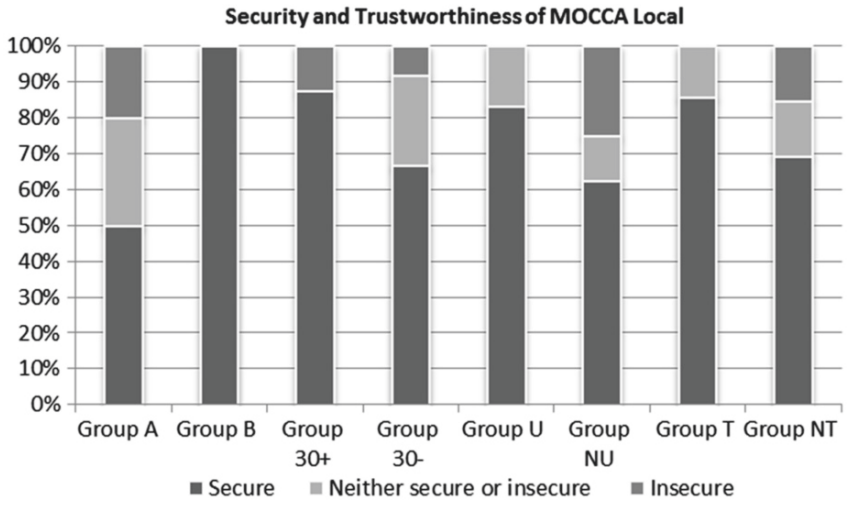


Fig. 4.17 Perceived security and trustworthiness of MOCCA Local

Analysis of the recorded user sessions and of information extracted from the conducted interviews revealed the main reasons for potential suspiciousness. As explained above, users were asked to install a certificate in the used Web browser during the installation process of MOCCA Local. This is necessary in order to establish an appropriate trust relationship between the Web browser and MOCCA Local. Unfortunately, the trust status of the used certificate was not accepted by the used Web browser. Hence, test users were faced with a security warning during the installation of this certificate. While most users simply ignored it, some test users were unsettled by the shown security warning.

Compared to MOCCA Local, MOCCA Online received worse ratings regarding security and trustworthiness. Obtained results are illustrated in Fig. 4.18. This time, similar results could be obtained for Group A and Group B. Again, older test users rated the security and trustworthiness of MOCCA Online more positively than younger users. Worst ratings have actually been obtained from nongraduate users. Less than 40 % of nongraduates rated MOCCA Online to be secure and trustworthy. No significant differences could be observed between technicians and users without technical background.

Similar to MOCCA Local, suspiciousness was mainly caused by shown security warnings. Since the Java Applet of MOCCA Online accesses local resources (i.e., the user's smart card), the Applet needs to be signed. Again, the trust status of the signing certificate was not accepted by the used Web browser.[14]

[14] This was due to the fact that a test instance of MOCCA Online has been used during the tests. The Java Applet of this test instance was signed with a test certificate only.

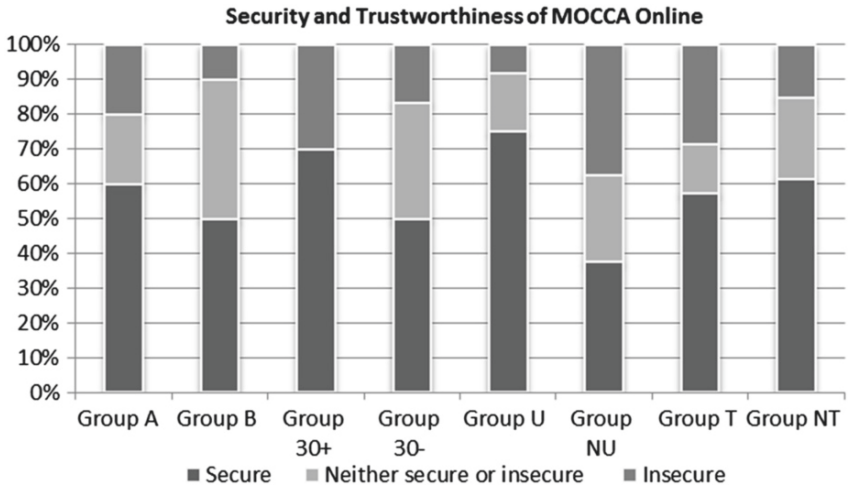


Fig. 4.18 Perceived security and trustworthiness of MOCCA Online

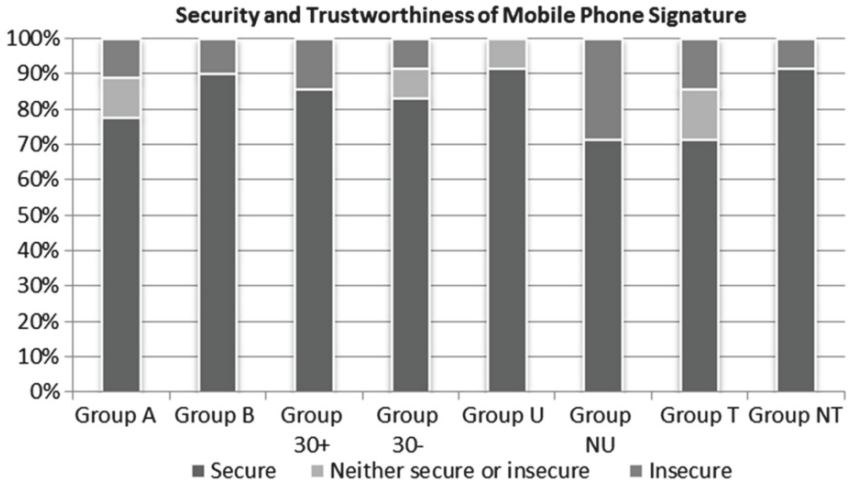


Fig. 4.19 Perceived security and trustworthiness of Mobile Phone Signature

Hence, a security warning was shown during the loading of the Applet. Some users were unsettled by this security warning.

In comparison to the two smart card-based CCS implementations MOCCA Local and MOCCA Online, the Mobile Phone Signature obtained significantly better ratings. Results are illustrated in Fig. 4.19. In all user groups, more than 70 % of the test

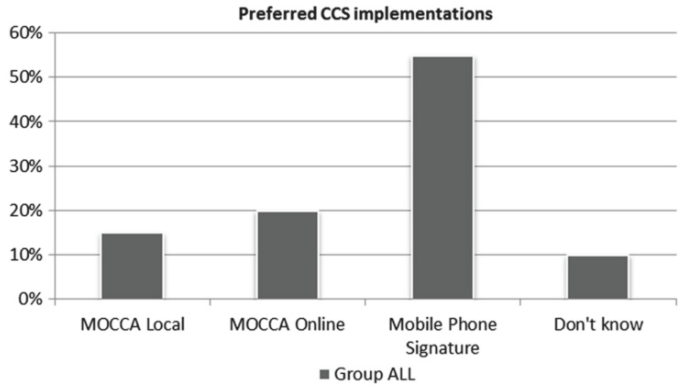


Fig. 4.20 Preferred CCS implementation

users rated the Mobile Phone Signature to be secure and trustworthy. Most significant differences in group-specific results could again be observed between graduates and nongraduate users. Also users without technical background rated the security and trustworthiness of the Mobile Phone Signature significantly better than technicians.

To answer Research Question Q4, we can conclude that users attested all three CCS implementations an appropriate level of security and trustworthiness. Still, there is some room for improvement especially for smart card-based approaches. A direct comparison of the three CCS implementations shows that the Mobile Phone Signature appears to be the most secure and trustworthy solution, followed by MOCCA Local and MOCCA Online.

4.12.2 Personal Preferences

Personal preferences of the individual test users have been identified in the course of the conducted conclusive interviews. All test users have been asked whether they will continue to use their Citizen Card for private affairs and which of the three tested CCS they will use.

Obviously, most test users have been convinced of the Citizen Card and stated to use it in the future for e-Government procedures. Regarding the preferred CCS, the Mobile Phone Signature has turned out to be the favored alternative. Figure 4.20 illustrates the obtained results. The Mobile Phone Signature has been selected by more than 50 % of all test users to be the favored alternative. 20 % of the test users stated that MOCCA Online is their preferred CCS. For approximately 15 % of all test users, MOCCA Local is the favored implementation alternative.

Again, interesting results can be obtained by comparing different user groups. While no major differences could be identified between users of Group A and Group B,

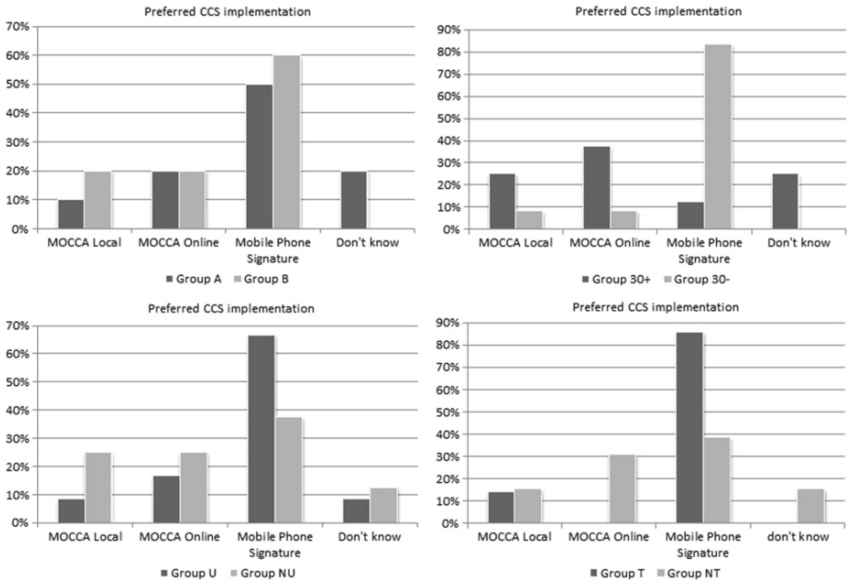


Fig. 4.21 Group-specific preferred version of CCS

the user's age obviously influenced the choice of the preferred CCS significantly. In the group of users being 30 or more years old, only about 10 % selected the Mobile Phone Signature as favored CCS alternative. In this group, most users preferred MOCCA Online. Even MOCCA Local achieved a higher acceptance than the Mobile Phone Signature in this group. In contrast to that, in the group of users being younger than 30, more than 80 % of all test users favored the Mobile Phone Signature, while the two smart card-based CCS implementations were favored by less than 10 % only.

Significant differences could also be observed between graduate test users and nongraduates. In both groups, the Mobile Phone Signature was the favored choice. However, while in the graduate group the Mobile Phone Signature was the clear winner, results were less unambiguous in the group of nongraduate test users. Similar results could be observed between technicians and users without technical background. Technicians clearly preferred the Mobile Phone Signature. This was also the favored choice of users without technical background. However, results were less unambiguous in this user group. All group-specific results are illustrated in Fig. 4.21.

In order to answer Research Question Q5, we can conclude that the Mobile Phone Signature is definitely the favored CCS implementation for citizens. This especially applies to young and well-educated people. Also users with a technical background clearly prefer the mobile CCS implementation variant.

4.13 Conclusions

The goal of this work was to evaluate the usability of several core components of the Austrian e-Government infrastructure, namely, different CCS implementations, in order to measure their efficiency. In total, five research questions have been defined to cover relevant usability aspects and to clearly define the scope of this work. To find answers to these questions, a thinking-aloud test has been conducted with 20 test users in total.[15] By analyzing the data that had been collected during these tests, we were able to find appropriate answers to all previously defined research questions.

Obtained results show that most recent developments have positively influenced the usability of Austrian e-Government processes. For instance, results show that reliance on Java technology does not raise severe usability problems. Hence, we can conclude that it was the right decision to base most Austrian e-Government components on Java. While it allows for platform-independent solutions, Java technology does not cause any severe usability problems for the evaluated solutions.

As shown in Figs. 4.17 and 4.18, the usability, security, and trustworthiness of MOCCA Online has been rated slightly worse compared to MOCCA Local. However, more users finally stated to personally prefer MOCCA Online to MOCCA Local. Obviously, the required software installation process of MOCCA Local is the decisive argument for users to rely on the installation-free alternative provided by MOCCA Online.

While MOCCA Local and MOCCA Online obtained comparable ratings in most categories, the Mobile Phone Signature turned out to be the clear winner in terms of popularity, security, trustworthiness, and usability. As depicted in Fig. 4.21, especially young and well-educated users clearly preferred the Mobile Phone Signature over smart card-based approaches. Hence, we can conclude that reliance on mobile technologies and solutions was the right decision and that this strategy appears to be promising also for future developments. This conclusion is consistent with the findings of Hung et al. (2013).

The conducted usability test delivered deeper insight into the usability of core components of the Austrian e-Government from the citizens' point of view. By collecting user feedback via various questionnaires, we were able to identify persisting weaknesses and further room for improvement. Valuable findings have also been obtained from an analysis of recorded user sessions. All results will be incorporated into future releases of the three evaluated CCS implementations. Thus, the conducted usability test contributes to the future security and usability of MOCCA Local, MOCCA Online, and the Mobile Phone Signature and hence to more efficient e-Government services.

[15] According to Nielsen (2013), a group of five people is fully sufficient for such tests (this number has been also reached for each subgroup of the test users).

References

Altameem T, Zairi M, Alshawi S (2006) Critical success factors of e-Government: a proposed model for e-Government implementation. Innovations in Information Technology, Dubai, pp 1–5. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4085489&tag=1

Boslaugh S, Watters PA (2008) Statistics in a nutshell, vol 54. O'Reilly, Sebastopol

Centner M, Orthacker C, Bauer W (2010) Minimal-footprint middleware for the creation of qualified signatures. In: Institute for Systems and Technologies of Information, Control and Communication (ed) Proceedings of the 6th international conference on web information systems and technologies. INSTICC—Institute for Systems and Technologies of Information, Control and Communication, Portugal, pp 64–69

Chircu AM, Hae-Dong Lee D (2005) E-government: key success factors for value discovery and realisation. Int J Electron Govern 2(1):11–25

EU Parliament and Council (2000) Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a community framework for electronic signatures. Off J Eur Commun L 013:12–20

EU Parliament and Council (2012) Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on electronic identification and trust services for electronic transactions in the internal market/*COM/2012/0238 final—2012/0146 (COD)

Frokjaer E, Hertzum M, Hornbaek K (2000) Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? Proceedings of the SIGCHI conference on human factors in computing systems, The Hague, The Netherlands, pp 345–352

Garcia ACB, Maciel C, Pinto FB (2005) A quality inspection method to evaluate e-Government sites. Electronic government fourth international conference, EGOV, vol 3591, pp 198–209

Geetha KT, Malarvizhi V (2010) Acceptance of e-Banking among customers (an empirical investigation in India). J Manag Sci 2(1). ISSN: 2249–1260

Gil-Garcia JR (2007) Exploring e-government benefits and success factors, encyclopedia of digital government. IGI Global, Hershey, pp 803–811

Gil-García JR, Pardo TA (2005) E-government success factors: mapping practical tools to theoretical foundations. Govern Inform Q 22(2):187–216

Hollosi A, Karlinger G, Rössler T, Centner M (2008) The Austrian citizen card. http://www.buergerkarte.at/konzept/securitylayer/spezifikation/20080220/

Howcroft B, Hamilton R, Heder P (2002) Consumer attitude and the usage and adoption of home-based banking in the United Kingdom. Int J Bank Market 20(3):111–121

Hung S-Y, Chang C-M, Kuo S-R (2013) User acceptance of mobile e-government services: an empirical study. Govern Inform Q 30(1):33–44. ISSN: 0740-624X, http://dx.doi.org/10.1016/j.giq.2012.07.008

Kubicek H (2011) Akzeptanzprobleme sicherer elektronischer Identitäten: Ergebnisse einer vergleichenden Untersuchung in acht Ländern, pp 43–47

Leitold H, Hollosi A, Posch R. (2002) Security architecture of the Austrian Citizen Card concept. In: Proceedings of the 18th annual computer security applications conference, ACSAC '02, IEEE Computer Society, Washington, DC, pp 391–400

Ma H-YT, Zaphiris P (2003) The usability and content accessibility of the e-government in the UK. In: Stephanidis C (ed) Universal access in HCI, vol 2007. Lawrence Erlbaum, Mahwah, NJ, USA, pp 760–764

Nielsen J (2013) Why you only need to test with 5 users. http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

Norris P (2003) Digital divide: civic engagement, information poverty, and the Internet worldwide. Prometheus 21(3):1–320

Orthacker C, Zefferer T (2011) Accessibility challenges in e-Government: an Austrian experience. In: Houlden N, Oram D, Picking R, Cunningham S, Grout V (eds) Proceedings of the fourth international conference on internet technologies and applications (ITA 11), Centre for Applied Internet Research (CAIR), Glyndŵr University, Wrexham, UK, pp 221–228

Orthacker C, Centner M, Kittl C (2010) Qualified mobile server signature. Proceedings of the 25th TC 11 international information security conference, SEC 2010

Schultz EE, Proctor RW, Lien MC, Salvendy G (2001) Usability and security an appraisal of usability issues in information security methods. Comput Secur 20(7):620–634

Siddhartha A (2008) National e-ID card schemes: a European overview. Inf Secur Tech Rep 13(2):46–53

Sørum H (2011) An empirical investigation of user involvement, website quality and perceived user satisfaction in eGovernment environments. Proceedings of the second international conference on electronic government and the information systems perspective, EGOVIS '11. Springer-Verlag, Berlin, Heidelberg, pp 122–134

White H, Nteli F (2004) Internet banking in the UK: why are there not more customers. J Financ Serv Market 9:49–57

Zavareh FB, Ariff MSM, Jusoh A, Zakuan N, Bahari AZ, Ashourian M (2012) E-service quality dimensions and their effects on E-customer satisfaction in internet banking services. Procedia Soc Behav Sci 40:441–445, ISSN: 1877–0428

9 Ein virtuelles Testframework für E- Government Komponenten

Conference	D-A-CH Security 2011
Language	German
Title	Ein virtuelles Testframework für E- Government Komponenten
Authors	Thomas Zefferer, Vesna Krnjic, Bernd Zwattendorfer
Booktitle	D-A-CH Security 2011

Ein virtuelles Testframework für E-Government Komponenten

Thomas Zefferer · Vesna Krnjic · Bernd Zwattendorfer

Institut für Angewandte Informationsverarbeitung und
Kommunikationstechnologie (IAIK) – Technische Universität Graz
{thomas.zefferer, vesna.krnjic, bernd.zwattendorfer}@iaik.tugraz.at

Zusammenfassung

Um ein adäquates Level an Qualität und Sicherheit zu gewährleisten sind im Rahmen der Entwicklung von E-Government Komponenten umfangreiche Tests von besonderer Wichtigkeit. E-Government Komponenten dürfen im Rahmen von Tests jedoch nicht nur isoliert betrachtet, sondern müssen auch im Zusammenspiel mit der Umgebung, in der sie betrieben werden, überprüft werden. Für clientseitige Applikationen ergibt sich dadurch ein beträchtlicher Aufwand, da aufgrund der Pluralität an verfügbaren Betriebssystemen, Browsern oder auch Kartenlesegeräten eine Vielzahl an möglichen Konfigurationen der Umgebung berücksichtigt werden müssen.
Gängige Testtools und Teststrategien sind für E-Government Komponenten oft nur bedingt anwendbar. Dieses Paper skizziert einen alternativen Ansatz basierend auf virtuellen Maschinen und stellt eine konkrete Umsetzung dieses Ansatzes vor. Wir diskutieren Vor- und Nachteile des entwickelten virtuellen Testframeworks und zeigen, dass dieses in der Lage ist, das Testen von E-Government Komponenten signifikant zu erleichtern und damit zu deren Qualität und Sicherheit beizutragen.

1 Einleitung

Im Zuge der professionellen Entwicklung von Software sind speziell für sicherheitskritische Komponenten geeignete Teststrategien zur Gewährleistung vordefinierter Qualitätsstandards unumgänglich. Im Besonderen gilt dies auch für Softwarekomponenten aus dem Bereich des E-Governments. Da von diesen Komponenten sicherheitskritische Funktionen wie beispielsweise elektronische Signatur oder sichere Benutzerauthentifizierung zur Verfügung gestellt werden, ist deren Sicherheit und Integrität von besonderer Relevanz. Daher sind auch ausführliche und umfassende Tests einzelner E-Government Kernkomponenten für den Betrieb einer sicheren und vertrauenswürdigen E-Government Infrastruktur notwendig.

In Österreich stellen beispielsweise die Softwaremodule MOCCA (Modular Open Citizen Card Architecture) [CeOB10] für den Java Applet basierten Zugriff auf Chipkarten (Bürgerkarten), PDF-AS (PDF Amtssignatur) [LePR09] zur Erstellung von bürgerkartenbasierten PDF-Signaturen, oder MOA-ID [LeHP02] für die bürgerkartenbasierte Benutzerauthentifizierung derartige sicherheitskritische Kernkomponenten dar. Aufgrund derer Bedeutung für das österreichische E-Government werden Releases dieser Komponenten in einigen Fällen zusätzlichen externen Sicherheitsüberprüfungen unterzogen um so einen hohen Level an Qualität und damit einen höchstmöglichen Grad an Sicherheit zu gewährleisten.

Während Qualitätsansprüche zentraler und serverseitiger Softwaremodule wie Web-Services relativ effizient erfüllt werden können, stellt sich die Situation für clientseitige Softwarekom-

P. Schartner · J. Taeger (Hrsg.) · DACH Security 2011 · syssec (2011) pp-pp.

ponenten meist ungleich komplexer dar. Hier zeigte die Erfahrung, dass unabhängig von der Softwarekomponente selbst auch die Umgebung, in der diese Komponente ausgeführt wird, einen negativen Einfluss auf deren Funktionalität haben kann. Beispielsweise verursachte vor wenigen Monaten ein Update des Java Runtime Environments (JRE) Probleme im Graphical User Interface (GUI) des MOCCA Applets und führte dazu, dass Labels in der GUI unter gewissen Umständen nicht mehr korrekt dargestellt werden konnten. Benutzer waren dadurch nicht mehr in der Lage im Rahmen der Erstellung elektronischer Signaturen die zu signierenden Daten einzusehen. Dadurch stand eine wichtige Funktion im Rahmen der Signaturerstellung nicht mehr zur Verfügung, was negative Auswirkungen auf die Sicherheit des Signaturerstellungsprozesses hatte. Weitere konkrete Beispiele, in denen die Umgebung, in der sicherheitskritische E-Government Komponenten betrieben werden, negative Auswirkungen auf die Sicherheit der eigentlichen Komponente haben kann, werden im Verlauf dieses Papers noch ausführlicher diskutiert.

Die Problematik negativer, unvorhersehbarer Auswirkungen der Umgebung auf sicherheitskritische Softwarekomponenten ist vor allem für clientseitige Software gegeben. Während zentrale, serverseitige Komponenten in einer wohldefinierten und getesteten Umgebung betrieben werden können, sind Annahmen über Clientumgebungen aufgrund der Vielzahl an Betriebssystemen, JREs und Browsern nur schwer zu treffen. Die Pluralität an möglichen Clientsystemen und Konfigurationen wird durch diverse Verbreitungsstatistiken für Betriebssysteme und Web Browser verdeutlicht [NMS11] [BS11]. Verschärft wird diese Problematik zusätzlich durch den Umstand, dass die einzelnen Komponenten der Umgebung (Betriebssystem, Browser, etc.) in der Regel relativ kurzen Updatezyklen unterworfen sind, wodurch sich die Sammlung möglicher Umgebungen und Konfigurationen ständig ändert und erweitert. Darüber hinaus bedingen Änderungen in den Umgebungen unter Umständen Änderungen an den E-Government Komponenten selbst, wodurch sich auch für diese verkürzte Updatezyklen – und somit häufigere Testdurchläufe – ergeben.

Durch die Dynamik und Pluralität möglicher Zielumgebungen gestaltet sich die Durchführung vollständiger Tests clientseitiger Softwarekomponenten komplex und aufwändig. Ein systematischer Ansatz zur Durchführung von Tests in unterschiedlichen Zielumgebungen ist daher unumgänglich. Die Erfahrung zeigte, dass herkömmliche Testframeworks in diesem Zusammenhang nur bedingt einsetzbar sind. Vor allem die für E-Government Komponenten typische Integration der Chipkartenkommunikation stellt in der Praxis oft eine ernstzunehmende Hürde dar und erschwert die Durchführung systematischer Testdurchläufe.

Dieses Paper beschreibt, wie durch den Einsatz eines virtuellen Testframeworks den typischen Problemen, die beim Testen clientseitiger E-Government Softwarekomponenten auftreten können, begegnet werden kann. Das in diesem Paper beschriebene Framework ermöglicht flexible, effiziente und nachvollziehbare Tests von Softwarekomponenten mit unterschiedlichsten virtuellen Systemen und Umgebungen und trägt somit entscheidend zur Qualitätssicherung und damit auch zur Gewährleistung der Sicherheit von E-Government Kernkomponenten bei.

Dieses Paper ist wie folgt aufgebaut. Abschnitt 2 diskutiert die Wichtigkeit von umfassenden Tests von E-Government Komponenten in verschiedenen Umgebungen und definiert Anforderungen für ein entsprechendes Testframework. Abschnitt 3 gibt einen Überblick über existierende Testtools und Strategien und zeigt, dass diese die in Abschnitt 2 definierten Anforderungen meist nur bedingt erfüllen können. In Abschnitt 4 wird schließlich ein alternativer Ansatz vorgestellt, dessen konkrete Umsetzung in Form eines virtuellen Testframeworks gezeigt

und gesammelte Erfahrungen diskutiert. Abschnitt 5 fasst schließlich die wichtigsten Punkte dieses Papers zusammen.

2 Anforderungen von E-Government Komponenten

Elektronische Behördenwege spielen im Zeitalter des Internets eine zunehmend wichtige Rolle. Einerseits erlauben sie Bürgern oder Unternehmen eine raschere Abwicklung von Anträgen bzw. einen „Rund um die Uhr"-Zugang zu Behörden, andererseits erleichtert E-Government auch den Behörden selbst die vereinfachte Durchführung gewisser Aktivitäten wie z.B. die Bearbeitung von elektronischen Akten [BKA01] im Back-Office Bereich.

Häufig werden bei behördlichen Verfahren sensible und schützenswerte Daten von Bürgern oder Unternehmen verarbeitet. Im Rahmen von E-Government Anwendungen muss Bürgern und Unternehmen daher die gleiche Sicherheit, Vertrauenswürdigkeit oder Transparenz wie bei traditionellen behördlichen Prozessen geboten werden. Um diesen Anforderungen gerecht zu werden, müssen eingesetzte E-Government Komponenten nicht nur sorgfältig entworfen und entwickelt, sondern nach deren Fertigstellung auch ausgiebig auf ihre korrekte Funktionalität und ihr Verhalten getestet werden. Während solche Tests für serverseitige Komponenten zumeist relativ unkompliziert und automatisiert bewerkstelligbar sind, stellt sich die Situation bei clientseitigen Komponenten ungleich komplexer dar. Zu heterogen ist die Betriebssystem- bzw. Browserlandschaft, die auf den lokalen Rechnern von Benutzern installiert ist, was auch durch diverse Statistiken belegt wird [NMS11] [BS11].

Wesentliche Forderungen im E-Government sind oftmals Technologieunabhängigkeit bzw. die Möglichkeit der Nutzung für alle Bürger [DigÖ08]. In Österreich sind diese Forderungen sogar gesetzlich verankert [EGovG04]. Für E-Government Komponenten soll dabei die Unabhängigkeit von speziellen Hardware- und Softwarekomponenten als Ziel verfolgt werden, um sich auch in Zukunft in keine Abhängigkeit von einzelnen externen Komponenten begeben zu müssen und Technologieneutralität gewährleisten zu können.

Für die Entwicklung von E-Government Komponenten bedeutet dies, dass nicht speziell nur auf eine Hardware oder bestimmte Umgebung gesetzt werden kann. Die Forderung nach Plattformunabhängigkeit und Technologieneutralität schlägt sich auch auf Test-Prozeduren von E-Government Komponenten nieder. So darf für Tests nicht nur die E-Government Software isoliert betrachtet werden, sondern ebenfalls ihr Verhalten in unterschiedlichen Umgebungen, da diese unter Umständen die Funktionalität der Software beeinflussen können.

Die Erfahrung zeigte, dass die Umgebung, in der eine E-Government Komponente betrieben wird, tatsächlich schwerwiegende negative Auswirkungen auf Funktionalität und Sicherheit der eigentlichen E-Government Komponente haben kann. Im Folgenden wird anhand einiger Fallbeispiele gezeigt, inwiefern unterschiedliche Umgebungen die Funktionalität und Sicherheit von E-Government Komponenten negativ beeinflussen können.

- **Integration von Chipkarten (Kartenleser):** Chipkarten spielen im Bereich des E-Governments in vielen Ländern eine zentrale Rolle. Einerseits werden sie häufig zur eindeutigen elektronischen Identifikation und Authentifizierung verwendet, anderseits finden sie Einsatz als *Secure Signature Creation Device (SSCD)* zur Erstellung von elektronischen Signaturen gemäß der Signaturrichtlinie [EP99]. Identifikation oder elektronische Signaturen bilden wesentliche Bestandteile eines sicheren und zuverlässigen E-Governments. Sollte diese Funktionalität nach einem Software-Update in der Umgebung des Benutzers nicht mehr gegeben sein, so erfolgt eine gravierende Einschränkung in der

Nutzung von E-Government Anwendungen. Im Rahmen von Tests österreichischer E-Government Komponenten konnte festgestellt werden, dass Treiber-Updates im Betriebssystem dazu führen können, dass einzelne Kartenleser nicht mehr ordnungsgemäß funktionieren und so die Funktionalität der E-Government Anwendung nicht mehr gegeben ist. Diese Problematik erstreckt sich nicht nur auf ein Betriebssystem, sondern konnte auf mehreren Systemen (darunter MS Windows, Linux und Apple Mac OS) beobachtet werden:

- **Microsoft Windows:** Hier zeigte sich unter anderem das Verhalten, dass die mit MS Windows ausgelieferten Default-Treiber für Kartenlesegeräte nicht immer korrekt funktionierten. Außerdem konnte ein unterschiedliches Verhalten bei verschiedenen Versionen der von den Herstellern bereitgestellten Treiber festgestellt werden.
- **Linux:** Die Unterstützung für Kartenleser-Treiber ist unter Linux oft mangelhaft. In einigen Distributionen fehlen entsprechend fertige Treiber-Pakate, sodass Benutzer angehalten werden, selbst Treiber-Binaries zu erstellen und zu installieren. Des Weiteren ist die Funktionalität von Pinpad-Kartenleser unter Linux nicht immer verfügbar.
- **Apple Mac OS:** Auch unter Apple Mac OS konnte der Effekt festgestellt werden, dass Pinpad-Kartenleser nicht funktionieren, da der in MAC OS enthaltene PCSC Daemon die Pinpad Funktionalität nach dem PCSCv2 Standard nicht unterstützt. Probleme traten nach Betriebssystemupdates unter anderem auch mit der PCSC-Schnittstelle, welche für die Chipkartenkommunikation zuständig ist, auf.

- **Abhängigkeit von Java:** Im Rahmen des österreichischen E-Governments wird vermehrt auf Java gesetzt, um die gewünschte Plattformunabhängigkeit zu erreichen. So ist beispielsweise die österreichische Bürgerkartenumgebung MOCCA, welche eine Client-Middleware für die Kommunikation mit der österreichischen Bürgerkarte darstellt, als Java Applet implementiert. Aufgrund dessen ist die korrekte Funktionalität von Java eine wichtige Vorrausetzung für ein korrektes Verhalten der Bürgerkartenumgebung. Vergangene Tests haben gezeigt, dass z.B. ein Bug in einem neuen Java-Update unter bestimmten Umständen (wenn das Applet aus dem Browser-Cache geladen wurde) das Nichtanzeigen von Labels in Java-Applets zur Folge hatte. Die Auswirkung auf die Bürgerkartenumgebung war, dass der zu signierende Text für den Bürger nicht mehr dargestellt wurde und somit der Bürger die zu signierenden Daten nicht mehr einsehen konnte. Dies ist ein sicherheitskritisches Beispiel dafür, wie ein Update einer Umgebungskomponente eine E-Government-Komponente negativ beeinflussen kann. Da Java selbst regelmäßigen Update-Zyklen ausgesetzt ist, ist es daher notwendig, sämtliche auf Java basierende kritische E-Government Komponenten nach jedem Java-Update auf deren korrekte Funktionalität hin zu überprüfen.
- **Web-basierte Services (Browser):** Web-basierte Services bzw. Web Browser spielen eine wesentliche Rolle bei der Verwendung von E-Government Anwendungen. Obwohl einheitliche Spezifikation für z.B. HTML oder das HTTP-Protokoll existieren, so unterscheiden sich trotzdem meist die Implementierungen dieser Spezifikationen. Diese (oft ungewollte) unterschiedliche Auffassung der Spezifikationen durch Entwickler kann dazu führen, dass sich bei Verwendung von E-Government Komponenten in verschiedenen Umgebungen ebenfalls ein unterschiedliches Verhalten zeigt. Ein konkretes Beispiel ist die Implementierung der Same-Origin-Policy (SOP), die in vielen Browsern leicht unterschiedlich interpretiert und umgesetzt wird und bereits öfters zu Problemen bei der Verwendung diverser E-Government Komponenten geführt hat. Aus diesem Grund ist es un-

erlässlich, die Funktionalität von E-Government Komponenten in unterschiedlichen Umgebungen auf Korrektheit zu überprüfen.

- **Änderungen in Betriebssystemen:** Gravierenden Änderungen an Betriebssystemen können negative Auswirkungen auf die Funktionalität einzelner Softwarekomponenten haben. Dies gilt natürlich auch für Komponenten, die im Rahmen von E-Government Anwendungen eingesetzt werden. Als konkretes Beispiel kann hier die Einführung virtueller Verzeichnisse in Microsoft Windows Versionen ab Vista genannt werden. Hier konnten einige Probleme beobachtet werden, die zwar nicht unmittelbar die Sicherheit der E-Government Anwendung, sehr wohl jedoch deren Funktionalität in Mitleidenschaft zogen. Es ist jedoch grundsätzlich nicht auszuschließen, dass sich Änderungen an Betriebssystemen unter Umständen auch negativ auf das Sicherheitsniveau von E-Government Komponenten auswirken können.

Wie die beschriebenen Beispiele bzw. Probleme zeigen, ist es im Rahmen der Qualitätssicherung von E-Government Komponenten wichtig, dass diese Komponenten nicht nur isoliert, sondern auch im Kontext mit unterschiedlichen Umgebungen auf korrekte Funktionalität überprüft werden. Um der Heterogenität der Systeme der einzelnen Benutzer-Clients gerecht zu werden, ist es notwendig, die entwickelte Software auf unterschiedlichen Systemkonfigurationen zu testen, um auf etwaige Einflüsse der Benutzerumgebung rechtzeitig und entsprechend reagieren zu können. Im Wesentlichen können die Anforderungen, welche ein für diese Zwecke geeignetes Testframework erfüllen muss, folgendermaßen zusammengefasst werden:

- **Zentrale Verfügbarkeit:** Entwicklern bzw. Testingenieuren soll mittels eines geeigneten Testsystems die Arbeit zur Qualitätssicherung abgenommen bzw. erleichtert werden. Das Testsystem mit unterschiedlichen Umgebungen (z.B. Betriebssystemen oder Browserversionen) sollte deshalb zentral verwaltet werden, um so auch mehreren Personen gleichzeitig ein einheitliches System zur Verfügung stellen zu können.
- **Dynamische Adaptierbarkeit:** Die Update-Zyklen von Softwareprodukten sind üblicherweise sehr dynamisch und finden nicht immer zu fixen Zeitpunkten statt. Um daher auf Änderungen wie z.B. ein neues Browser-Update rasch reagieren zu können, bedarf es eines Testsystems, welches rasch angepasst oder erweitert werden kann.
- **Effiziente Verwaltung:** Der Mix aus unterschiedlichen Betriebssystemen, Browsern oder Java-Versionen kann schnell zu einer unübersichtlichen Test-Matrix führen. Aus diesem Grund ist eine effiziente Verwaltung des Testframeworks eine weitere zentrale Anforderung.
- **Nachvollziehbarkeit:** Das Testen von Komponenten in einer bestimmten Umgebung ist kein einmaliger Vorgang. Sollte es beispielsweise trotz positiver Testergebnisse zu Problemen kommen, so sollte die verwendete Testumgebung leicht wiederherstellbar sein, um die durchgeführten Tests nochmals nachvollziehen zu können.

Die Erfüllung dieser Anforderungen ist für die Durchführung von effizienten und umfangreichen Tests von E-Government Komponenten unerlässlich. Im folgenden Abschnitt werden bestehende Teststrategien und Frameworks diskutiert und deren Tauglichkeit zur Erfüllung der oben genannten Anforderungen evaluiert.

3 Teststrategien und Frameworks

Aufgrund der Wichtigkeit von systematischen Tests zur Qualitätssicherung in der Softwareentwicklung existieren bereits eine Vielzahl an bewährten Strategien und Werkzeugen. In die-

sem Abschnitt werden einige dieser Ansätze diskutiert und auf deren Tauglichkeit für E-Government Komponenten hin überprüft.

Sicherheitskritische Softwarekomponenten werden häufig gemäß dem *Common Criteria for Information Technology Security Evaluation (Common Criteria, CC)* Standard [CCMB09] formell evaluiert und zertifiziert. Dieser international abgestimmte Standard zur Informationssicherheit enthält allgemeine Kriterien für die Prüfung und Bewertung der Sicherheit von Informationstechnologien. Vor einer Common Criteria Evaluierung wird das sogenannte *Target of Evaluation (TOE)*, das meistens das zu evaluierende System darstellt, genau definiert und von anderen Komponenten, die der Umgebung zugeordnet werden, abgegrenzt. Die Kriterien werden nur auf dieses TOE angewendet, während die Umgebung nicht näher betrachtet wird.

Während sich dieser Ansatz zur Überprüfung und Evaluierung einzelner isolierter Komponenten bewährt hat, reicht die Common Criteria Evaluierung einzelner Komponenten für komplexe E-Government Lösungen oft nicht aus. Wie die Erfahrung zeigte, kann eine an sich sichere Komponente durch diverse Gegebenheiten der Umgebung in Funktion und Sicherheit signifikant beeinträchtigt werden. Komplexe E-Government Softwarelösungen können es sich daher nicht leisten, nur einen bestimmten und definierten Teilbereich einer intensiven Qualitäts- und Sicherheitsüberprüfung zu unterziehen. Vielmehr muss die Anwendungen als Einheit mit ihrer Umgebung, mit der diese in Wechselwirkung steht, gesehen werden. Dementsprechend ist auch die Teststrategie auszurichten.

Durch die Miteinbeziehung der Umgebung in die Teststrategie gestalten sich die Anforderungen an ein Testframework für clientseitige oder webbasierte E-Government Softwarekomponenten umfangreich und komplex, da eine Vielzahl an Faktoren bei der Durchführung von Softwaretests berücksichtigt werden müssen. Eine besondere Rolle spielt hier die lokale Umgebung des Benutzers. Durch die unterschiedlichen Betriebssysteme und die Fülle an gängigen Browsern, die diversen Java Runtime Environments (JRE) Versionen, und der Vielzahl an unterschiedlichen Kartenlesern und Chipkarten ergibt sich eine komplexe Testmatrix mit einer erheblichen Anzahl an abzudeckenden Testszenarien.

Das Testen webbasierter Anwendungen auf unterschiedlichen Clientsystemen ist keine auf den E-Government Bereich beschränkte Herausforderung. Dementsprechend existieren bereits eine Vielzahl an Tools und Frameworks, die systematische und teilweise automatisierte Tests webbasierter Anwendungen erlauben.

- Selenium [HoKe06] ist ein auf HTML und JavaScript basierendes Testframework für Web Anwendungen. Durch die Aufzeichnung der Benutzerinteraktion mit der Web Anwendung können Tests automatisch wiederholt werden, wodurch Testdurchläufe schneller und zuverlässiger durchgeführt werden können. Selenium wird in den vier Ausführungen Selenium Core, Selenium IDE , Selenium Remote Control (RC) und Selenium Grid angeboten. Das Framework ist betriebssystemunabhängig und kann mit allen Browsern verwendet werden. Selenium unterstützt zwar die wiederholte Durchführung von Testdurchläufen, bietet jedoch keine Möglichkeiten um Tests auf verschiedenen Systemkonfigurationen systematisch durchzuführen.
- Die HP Functional Testing Software [HP11] und IBM Rational Functional Tester [IBM11] sind gängige Testwerkzeuge, welche Aufnahmen der Benutzerinteraktion mit einer Softwarebenutzeroberfläche erlauben und damit ein automatisiertes Testen ermöglichen. Diese Tools unterstützen die Erzeugung von Funktions- und Regressionstests. Die Testaufnahmen können einfach über einen Rekorder erstellt werden. Alternativ kann für die Erzeugung der automatischen Testfälle auch eine Skriptsprache verwendet werden.

Dadurch können Aufnahmen verändert und parametrisiert werden. Ähnlich wie Selenium benötigen auch diese Test-Frameworks eine Testumgebung auf der sie eingesetzt werden können. Da diese Tools nicht plattformunabhängig sind, können sie zum Testen von E-Government Anwendungen jedoch nur bedingt eingesetzt werden.

- Im Gegensatz zu den bereits genannten Tools verfolgen AdobeBrowserLab [Adobe11] und CrossBrowserTesting [CBT11] einen anderen Ansatz. Diese Online-Tools ermöglichen die Evaluierung einer Webseite mit unterschiedlichen Betriebssystemen und Browsern. Durch Angabe einer URL wird die unter dieser URL erreichbare Seite automatisch überprüft. Über diese Tools sind prinzipiell Tests auf verschiedenen Plattformen möglich. Die für E-Government Anwendungen relevanten Aspekte wie Kompatibilität zu Kartenlesegeräten oder verschiedenen JREs können mit diesen Tools jedoch ebenfalls nicht abgedeckt werden.

Obwohl bereits eine Vielzahl an Ansätzen und unterstützenden Tools für automatisiertes Testen existiert, sind diese für komplexe E-Government Anwendungen nur bedingt einsetzbar. Vor allem die für E-Government Anwendungen spezifischen Aspekte wie Chipkartenkommunikation oder die Abhängigkeit von Java Laufzeitumgebungen machen einen effizienten Einsatz bestehender Tools und Frameworks oft unmöglich.

Aus diesem Grund wurde für Elemente des österreichischen E-Governments ein alternativer Ansatz entwickelt und in Form eines virtuellen Testframeworks, welches im nächsten Abschnitt näher beschrieben wird, umgesetzt.

4 Virtuelles Testsystem

Durch die besonderen Anforderungen, die umfangreiche Tests von E-Government Komponenten mit sich bringen, bietet die Verwendung herkömmlicher Testframeworks oft keine zufriedenstellende Lösung. Um der Dynamik sich ständig verändernder Umgebungen begegnen zu können, wurde von den Autoren ein alternativer Ansatz verfolgt. Ziel war die Entwicklung eines zentralen virtuellen Testframeworks, welches flexibel auf sich ändernde Anforderungen angepasst und einfach für effiziente Tests verschiedener E-Government Komponenten verwendet werden kann.

In diesem Abschnitt werden Aufbau und Funktionen des entwickelten Frameworks näher vorgestellt. Anhand praktischer Erfahrungen, die mit diesem Framework bisher gesammelt werden konnten, werden Vorteile dieses Ansatzes diskutiert und weitere Verbesserungsmöglichkeiten skizziert.

4.1 Aufbau

Es wurde ein Ansatz virtualisierter Systeme verfolgt. Das beschriebene Framework verwendet Komponenten der Firma VMware, die zur besseren Veranschaulichung für den Leser im Folgenden auch mit Produktnamen genannt sind. Das grundlegende Konzept ist aber auch auf andere Virtualisierungslösungen übertragbar.

Das erstellte Testframework bietet Benutzern ein umfangreiches Repository an unterschiedlichen Testumgebungen und Systemkonfigurationen in Form von virtuellen Maschinen. Zur Verwaltung der virtuellen Maschinen kommen im Speziellen die Komponenten VMware ESX Server [VmS11], VMware vCenter Server [VmC11] und VMware LabManager Server [VmL11] zum Einsatz. Abb. 1 zeigt den prinzipiellen Aufbau des Testframeworks.

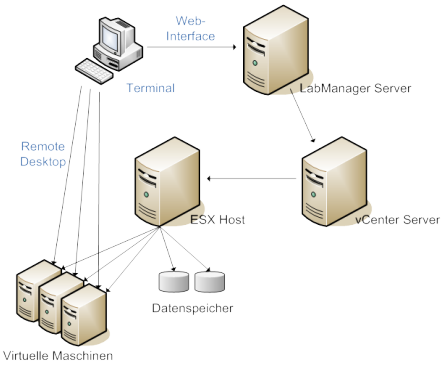


Abb. 1: Allgemeiner Aufbau des virtuellen Testframeworks

Ein zentraler VMware ESX Server, welcher über ausreichend Rechenleistung, sowie Arbeits- und Datenspeicher verfügt, dient als zentrale Komponente und stellt eine Reihe definierbarer virtueller Maschinen für Testzwecke zur Verfügung. Die Verwaltung dieser virtuellen Maschinen kann über ein Web-Interface vorgenommen werden. Dieses wird vom zentralen VMware LabManager Server zur Verfügung gestellt, welcher seinerseits über einen VMware vCenter Server auf den zentralen ESX Host Zugriff hat. Aus Sicht der Benutzer sind sowohl ESX Server als auch vCenter Server völlig transparent.

Der Zugriff auf die einzelnen virtuellen Maschinen kann entweder über den VMware LabManager Server und eine VMware Konsole in Form eines Browser Plug-ins, oder auch direkt über Remote-Desktop Protokolle erfolgen.

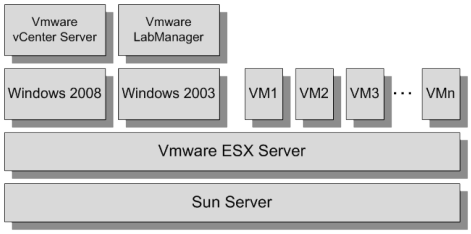


Abb. 2: Verwendete Infrastruktur

Der in Abb. 1 dargestellte prinzipielle Aufbau wurde gemäß dem in Abb. 2 dargestellten Schema umgesetzt. Als einzige physikalische zentrale Komponente kam dazu ein Sun Server zum Einsatz. Sämtliche anderen Komponenten wurden basierend auf dieser physikalischen Komponente rein virtuell ausgeführt. Der VMware ESX Server wurde direkt auf dem Sun Server installiert und dient als Basis für alle weiteren virtuellen Komponenten. Auf dem ESX Server wurden zwei virtuelle Microsoft Windows Rechner aufgesetzt, auf welchen die beiden VMware Produkte vCenter Server und LabManager Server installiert wurden[1]. Der ESX Server bildet des Weiteren die Grundlage für alle anderen virtuellen Maschinen des Testframeworks.

[1] Ein Betrieb dieser beiden VMware Komponenten auf einem einzigen System war aufgrund unterschiedlicher Anforderungen an das Betriebssystem nicht möglich.

4.2 Funktionen

Die verwendeten VMware Komponenten bieten Benutzern prinzipiell eine Fülle an Möglichkeiten und Funktionen. Für Tests von E-Government Kernkomponenten ist folgende Verwendung des Testframeworks vorgesehen.

1. Der Benutzer verbindet sich mit dem VMware Lab Manager Server und öffnet das webbasierte Interface zur Verwaltung von Konfigurationen[2] und virtuellen Maschinen.
2. Der Benutzer wählt aus einer vorhandenen Sammlung von Konfiguration die für den jeweiligen Test passenden virtuellen Maschinen aus und klont daraus eine eigene Arbeitskopie.
3. Der Benutzer startet die in der Arbeitskopie enthaltenen geklonten virtuellen Maschinen.
4. Der Benutzer verbindet sich zu den gestarteten virtuellen Maschinen über die webbasierte VMware Konsole oder direkt über Remote Desktop Protokolle.
5. Der Benutzer führt die vorgesehenen Tests aus.
6. Nach Beendigung der Tests fährt der Benutzer die verwende Arbeitskopie herunter.
7. Die Arbeitskopie wird ja nach Einstellung nach einer bestimmten Zeit gelöscht.

Neben dieser Grundfunktionalität bietet das Testframework zahlreiche zusätzliche Funktionen, sodass sämtliche in Abschnitt 2 definierte Anforderungen erfüllt werden können. Im Speziellen bieten sich durch den Einsatz dieses Systems für Benutzer unter anderem die im Folgenden näher erläuterten Möglichkeiten und Vorteile.

4.2.1 Zentrale Verfügbarkeit

Durch den Einsatz zentraler Komponenten und den Betrieb des Testframeworks auf einem allgemein zugänglichen Server kann das Framework von mehreren Personen (auch gleichzeitig) genutzt werden. Der zentrale Ansatz erleichtert zudem die Wartung der verschiedenen virtuellen Maschinen und ermöglicht eine effiziente Nutzung von Ressourcen.

4.2.2 Dynamische Adaptierbarkeit

Das Testframework verfügt über ein Set an vordefinierten virtuellen Maschinen mit unterschiedlichen Konfigurationen. Diese reichen von Systemen mit neu installierten Betriebssystemen bis hin zu Systemen mit diversen vorinstallierten Softwarekomponenten wie JRE, Browser und Bürgerkartenumgebungen. Somit ist ein Großteil aller gängigen Systemkonfigurationen für die Durchführung von Tests sofort verfügbar.

Die vordefinierten Systeme können von Benutzern jederzeit geladen, geklont und für eigene Testzwecke beliebig verwendet werden. Nach Abschluss der Tests können diese Klone entweder gelöscht, oder aber dem Set an vordefinierten Systemen hinzugefügt und so für eine spätere Wiederverwendung (auch durch andere Benutzer) aufbewahrt werden. Der Zeitpunkt des Löschens, kann bei der Erstellung der Konfiguration bestimmt und jederzeit dem Testverlauf angepasst werden. Durch das Klonen werden die Basiskonfigurationen erhalten wodurch der Urzustand des jeweiligen Systems für den nächsten Testdurchlauf wieder verwendet werden kann. Sollte keine der im Testframework vordefinierten virtuellen Maschinen für den

[2] Unter einer Konfiguration versteht man eine Sammlung virtueller Maschinen.

durchzuführenden Test geeignet sein, können Benutzer Betriebssysteme, die sich unter den VM Vorlagen befinden, zu einem eigenen Set hinzufügen und verwenden.

4.2.3 Effiziente Verwaltung

Eine effiziente Verwaltung des Testframeworks wird prinzipiell durch den zentralen Ansatz begünstigt. Der verwendete VMware LabManager Server erlaubt neben dem raschen Hinzufügen, Klonen und Entfernen von virtuellen Maschinen auch eine flexible Gruppierung und Archivierung von virtuellen Maschinen. Durch ein flexibles Benutzerverwaltungs- und Rechtemanagementsystem können zudem Berechtigungen für Benutzer dynamisch angepasst und auf gegebene Anforderungen abgestimmt werden.

4.2.4 Nachvollziehbarkeit

Der VMware LabManager Server verfügt über eine umfangreiche Archivierungsfunktion. Diese kann dazu genutzt werden um bestimmte Systemkonfigurationen zu konservieren und für eine spätere erneute Verwendung verfügbar zu halten. Damit kann die Nachvollziehbarkeit von Testdurchläufen auf bestimmten Systemkonfigurationen jederzeit gewährleistet werden.

4.3 Evaluierung

Im praktischen Einsatz zeigt sich sehr rasch, dass durch die Verwendung des virtuellen Testframeworks die Effizienz und der Umfang von Testdurchläufen gesteigert werden konnte. Vor allem ermöglicht das Testframework eine erhöhte Frequenz von Tests sicherheitsrelevanter Komponenten, wodurch etwaige durch Änderungen der jeweiligen Umgebung hervorgerufene Probleme früh erkannt werden können. Daneben offenbarten sich während der Verwendung des Testframeworks auch einige Probleme, die jedoch im Zuge einer weiteren Optimierung des Frameworks lösbar sein sollten.

Ein klarer Vorteil des bei diesem Framework verfolgten Ansatzes ist dessen zentraler Ansatz. Dadurch stehen Entwicklern und Supportmitarbeitern jederzeit beliebige Testumgebungen zur Verfügung. Vor allem im Zuge der Bearbeitung von Supportanfragen erwies sich das Framework als hilfreich, da dieses ein rasches Rekonstruieren bestimmter Umgebungen und Durchführen entsprechender Tests erlaubt. Von der raschen Verfügbarkeit beliebiger Konfigurationen profitieren auch Entwickler von E-Government Komponenten, da diese bereits während des Entwicklungsprozesses einzelne Komponenten auf deren Kompatibilität zu verschiedenen Umgebungen hin überprüfen können. Schließlich konnte durch das virtuelle Testframework auch die Durchführung abschließender, systematischer Tests signifikant verbessert werden.

Der intensive Einsatz des virtuellen Testframeworks offenbarte auch diverse Verbesserungspotentiale. Im Zusammenhang mit E-Government Komponenten erwies sich vor allem die bei manchen Betriebssystemen fehlende Unterstützung für eine Weiterleitung der USB Schnittstelle an das virtuelle Testsystem als störend. Dies betrifft jedoch hauptsächlich Linux basierte Testumgebungen und sollte durch Verwendung alternativer Remote Desktop Programme lösbar sein. Als relativ aufwändig erwies sich auch die Wartung der unterschiedlichen virtuellen Maschinen des Testframeworks. Durch den Einsatz entsprechender Skripts (z.B. zur automatisierten Durchführung von Betriebssystemupdates) konnte jedoch auch dieser Herausforderung begegnet werden. Probleme ergaben sich auch aufgrund lizenzrechtlicher Rahmenbedingungen, die beispielsweise eine Virtualisierung von Mac OS Betriebssystemen verhindert.

Diese Limitierung lässt sich durch ein Ausweichen auf physikalische Systeme umgehen, führt jedoch zu erhöhtem Aufwand und geringeren Variationsmöglichkeiten.

Insgesamt führte der praktische Einsatz der virtuellen Testumgebung zu einer deutlichen Erhöhung der Effizienz in der Durchführung von Applikationstests und trug so zur Gewährleistung adäquater Qualitäts- und Sicherheitsstandards bei.

5 Fazit

Umfangreiche Tests sind für die Qualitätssicherung in der Softwareentwicklung im Allgemeinen und für E-Government Komponenten im Speziellen von besonderer Bedeutung. Vor allem im E-Government Bereich, in dem häufig mit sensiblen und sicherheitskritischen Daten gearbeitet wird, ist eine korrekte Funktionalität der eingesetzten Software wichtig, um einen adäquaten Schutz dieser Daten zu gewährleisten. Die Erfahrung zeigte, dass übliche Test- und Evaluierungsansätze wie z.B. die Common Criteria Evaluierung oft zu kurz greifen, da diese zwar eine definierte Softwarekomponente detailliert betrachten, die Umgebung, in der diese Komponente eingesetzt wird, jedoch oft außer Acht lassen. Vor allem in clientseitigen E-Government Anwendungen spielt die Umgebung (Betriebssystem, Browser, JRE, etc.) jedoch oft eine entscheidende Rolle und kann Funktionalität und Sicherheit der eigentlichen E-Government Komponente kompromittieren. Aus diesem Grund ist eine Miteinbeziehung der Umgebung in die Durchführung von Tests unerlässlich.

Aufgrund der großen Anzahl an gängigen Betriebssystemen, Browsern und anderen Komponenten der Umgebung stellt die Durchführung vollständiger Tests eine ernstzunehmende Herausforderung dar. Erschwert wird die Situation zusätzlich durch häufige Updates der Komponenten der Umgebung (Browser-Updates, Betriebssystem-Updates, etc.), die stets neue Testdurchläufe erforderlich machen. Um dieser Komplexität und Dynamik Herr zu werden, sind systematische Ansätze notwendig. Eine Analyse bestehender Tools und Frameworks zeigte, dass diese vorhandenen Testwerkzeuge meist nicht in der Lage sind die Anforderungen für Tests von E-Government Komponenten zu erfüllen. Hier stellte sich besonders die Integration von Chipkarten als schwer zu automatisierendes Problem dar.

Aufgrund der Unzulänglichkeiten existierender Ansätze und Frameworks wurde ein alternativer Ansatz entwickelt und in diesem Paper vorgestellt. Basierend auf Produkten der Firma VMware Inc. wurde ein virtuelles Testframework erstellt. Diese besteht aus einem einfach zu verwaltenden Repository virtueller Maschinen, die unterschiedliche Umgebungen für Tests zur Verfügung stellen. Durch den Einsatz dieses virtuellen Testframeworks konnten der Umfang und die Effizienz der Tests österreichischer E-Government Komponenten signifikant verbessert werden. Die dadurch erreichbare Steigerung der Qualität führte auch zu einer Erhöhung der Sicherheit der getesteten Komponenten. Das in diesem Paper vorgestellte Testframework trägt somit wesentlich zur Qualität und Sicherheit der österreichischen E-Government Infrastruktur bei.

Literatur

[Adobe11] Adobe BrowserLab, https://browserlab.adobe.com/ (2011)

[BKA01] Bundeskanzleramt: ELAK-Konzept. http://www.digitales.oesterreich.gv.at/DocView.axd?CobId=19396 (2001)

[BS11] Browser Statistik, http://www.browser-statistik.de/ (2011)

[CBT11] Cross Browser Testing, http://crossbrowsertesting.com/ (2011)

[CeOB10] M. Centner, C. Orthacker, W. Bauer: Minimal-Footprint Middleware for the Creation of Qualified Signatures. In: Proceedings of the 6th International Conference on Web Information Systems and Technologies (2010) 64-69.

[CCMB09] Common Criteria for Information Technology Security Evaluation, CC v3.1. Release 3, http://www.commoncriteriaportal.org/cc/ (2009)

[DigÖ08] Digitales Österreich, Behörden im Netz - Das österreichische E-Government ABC, http://www.bka.gv.at/DocView.axd?CobId=27782 (2008)

[EGovG04] E-Government-Gesetz (E-GovG) (2004), Bundesgesetzblatt für die Republik Öster-reich BGBl.I Nr. 10/2004

[EP99] Richtlinie1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000:013:0012:0020:DE:PDF (1999)

[HoKe06] A. Holmes, M. Kellogg: Automating Functional Tests Using Selenium, Proceedings of AGILE 2006 Conference (2006)

[HP11] Functional Testing Software, Simplify the automation of functional and regression testing https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24^1322_4000_100_(2011) zuletzt abgerufen am 28.04.2011

[IBM11] Rational Functional Tester, Performance test automation for quality driven software delivery http://www-01.ibm.com/software/awdtools/tester/performance/ (2011)

[LeHP02] H. Leitold, A. Hollosi, R. Posch: Security Architecture of the Austrian Citizen Card Concept. In: ACSAC '02, Proceedings of the 18th Annual Computer Security Applications Conference, IEEE Computer Society (2002) 391.

[LePR09] H. Leitold, R. Posch, and T. Rössler: Media-break resistant eSignatures in eGovernment: an Austrian experience. In: Javier Lopez Dimitris Gritzalis, Emerging Challenges for Security, Privacy, and Trust - 24th IFIP SEC, Springer (2009) 109-118.

[NMS11] Net Market Share, Operating System Market Share, http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=10 (2011)

[SigG99] Signaturgesetz (SigG) (1999), Bundesgesetzblatt für die Republik Österreich BGBl.I Nr. 190/1999

[VmS11] VMware vSphere Server, http://www.vmware.com/de/products/vi/esx/ (2011)

[VmC11] VMware vCenter Server, http://www.vmware.com/de/products/vi/vc/ (2011)

[VmL11] VMware vCenter Lab Manager, http://www.vmware.com/de/products/labmanager/overview.html (2011)

10 | Effizientes Testen von E-Government Komponenten in der Cloud

Conference	DACH Security 2013
Language	German
Title	Effizientes Testen von E-Government Komponenten in der Cloud
Authors	Vesna Krnjic, Philip Weber, Thomas Zefferer, Bernd Zwattendorfer
Booktitle	Proceedings DACH Security 2013

Effizientes Testen von E-Government Komponenten in der Cloud

Vesna Krnjic · Philip Weber · Thomas Zefferer · Bernd Zwattendorfer

Institut für Angewandte Informationsverarbeitung und Kommunikationstechnologie – Technische Universität Graz

{vesna.krnjic, philip.weber, thomas.zefferer, bernd.zwattendorfer}@iaik.tugraz.at

Zusammenfassung

Für softwarebasierte Anwendungen in sicherheitskritischen Bereichen wie E-Government stellt die Durchführung umfassender Tests ein wichtiges Mittel zur Sicherstellung definierter Qualitäts- und Sicherheitsmerkmale dar. Aufgrund der ständig zunehmenden Komplexität von Softwarelösungen auch in sicherheitskritischen Anwendungen, werden für die Durchführung von Testroutinen im verstärkten Maße entsprechende Frameworks und Tools eingesetzt. Vor allem im Bereich des E-Government ergeben sich jedoch mitunter besondere Anforderungen, die von bisher verfügbaren Frameworks und Tools nicht vollständig erfüllt werden können. Zur Lösung dieses Problems stellen wir in diesem Artikel ein neuartiges Testframework vor. Dieses kombiniert die Vorteile etablierter virtueller Testframeworks mit Konzepten des Cloud Computing. Damit bietet das vorgestellte Testframework Entwicklerinnen und Entwicklern von E-Government-Anwendungen eine umfassende Lösung zum systematischen Testen sicherheitskritischer Komponenten im Bereich des E-Government und trägt zur Sicherung definierter Qualitäts- und Sicherheitsmerkmale bei.

1 Einleitung

Für die professionelle Entwicklung von Software ist die Gewährleistung ihrer Qualität von zentraler Bedeutung. Nach ISO/IEC 25010:2011 [ISO11] werden für die Messung der Qualität von Software u.a. die Faktoren Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Wartbarkeit und Portabilität herangezogen. Formale Nachweise für die Einhaltung dieser Faktoren sind aufgrund der zunehmenden Komplexität von Software in der Praxis schwierig bis unmöglich zu erbringen. Aus diesem Grund stellen umfassende und ausgeklügelte Testmethoden derzeit den wichtigsten Ansatz zur Überprüfung der Einhaltung definierter Qualitätsmerkmale dar.

Die Einhaltung dieser Qualitätsmerkmale durch umfassende Tests ist besonders für sicherheitskritische Applikationen von zentraler Bedeutung, da hier auftretende Mängel unmittelbar zu Schäden unterschiedlicher Natur führen können. Prominente Beispiele für sicherheitskritische Anwendungen sind beispielsweise Softwarelösungen in den Bereichen E-Government oder E-Banking. Fehlerhafte bzw. mangelhafte Software kann in diesen Bereichen zu erheblichen finanziellen Schäden (E-Banking), oder auch zur Kompromittierung privater und schützenswerter Daten (E-Government) führen. Problematisch ist dies auch vor allem dann, wenn Software zur Erstellung qualifizierter – und damit rechtsgültiger – elektronischer Signaturen gemäß EU Signaturrichtlinie [EP99] fehlerhaft ist. Speziell für die Bereiche E-Banking und

P. Schartner · P. Trommler (Hrsg.) · DACH Security 2013 · syssec (2013) pp-pp.

E-Government sind daher geeignete Testmethoden zur Sicherstellung der erforderlichen Qualität im Zuge der Entwicklung von Softwarelösungen unumgänglich.

Vor allem im Bereich E-Government ergeben sich zumeist komplexe und verteilte Softwarelösungen bestehend aus sich ergänzenden Server- und Client-Komponenten. So werden E-Government-Dienste häufig über zentrale Web- und Applikationsserver angeboten. Für die Authentifizierung an diesen Diensten kommen jedoch auch clientseitige Komponenten wie zum Beispiel Chipkarten und Middleware-Komponenten [CeOB10] für den lokalen Zugriff auf diese Chipkarten zum Einsatz.

Da Server-Komponenten in der Regel in kontrollierten Umgebungen zum Einsatz kommen, können diese im Zuge der Softwareentwicklung relativ einfach und effizient getestet werden, da deren Zielplattform und Infrastruktur bekannt oder beeinflussbar ist. Für Client-Komponenten stellt sich die Situation als schwieriger dar. Da diese direkt am System der Endbenutzerin bzw. des Endbenutzers installiert und betrieben werden, müssen hier eine Vielzahl an möglichen Systemkonfigurationen berücksichtigt werden, die sich potentiell unterschiedlich auf die Funktionalität und damit auf die Qualität der jeweiligen Softwarelösung auswirken. Relevante Unterschiede können sich hier beispielsweise in Bezug auf das verwendete Betriebssystem oder den verwendeten Web-Browser ergeben.

Im Zuge der Softwareentwicklung ergibt sich bei der Erstellung clientseitiger E-Government-Lösungen damit die Herausforderung, qualitativ hochwertige Lösungen für eine möglichst große Anzahl unterschiedlicher Systemkonfigurationen zu implementieren. Dafür bedarf es geeigneter Testmethoden, über die unterschiedliche Konfigurationen potentieller Ziel-Systeme effektiv und effizient abgedeckt werden können.

In der Vergangenheit zeigte sich, dass virtuelle Testframeworks [ZeKZ11] im Unterschied zu herkömmlichen Frameworks [HP11][IBM11] ein geeignetes Mittel darstellen, um der großen Anzahl unterschiedlicher zu testender Systemkonfigurationen Herr zu werden. Bei diesem Ansatz werden verschiedene Systemkonfigurationen durch virtuelle Maschinen repräsentiert, die durch geeignete Software effizient verwaltet und verwendet werden können. Obwohl dieser Ansatz rasch zu den erwünschten Ergebnissen führt, manifestieren sich im praktischen Einsatz nach einiger Zeit einige zentrale Nachteile virtueller Testframeworks. Beispielsweise wächst durch die rasch steigende Anzahl an unterschiedlichen Betriebssystem- und Browserversionen auch die Anzahl möglicher und zu berücksichtigender Systemkonfigurationen und damit die Anzahl der für die Durchführung umfangreicher und möglichst vollständiger Tests benötigten virtuellen Maschinen. Damit stoßen virtuelle Testframeworks mit begrenzten lokalen Ressourcen relativ rasch an ihre Grenzen und sind nicht mehr in der Lage, alle zu testenden Systemkonfigurationen abzudecken.

Um dieser Problematik zu begegnen, stellen wir in diesem Artikel eine Weiterentwicklung des Konzepts der virtuellen Testframeworks vor. Dabei bedienen wir uns des Konzepts des Cloud-Computing, welches in den letzten Jahren enorm an Bedeutung gewonnen hat. Wir zeigen, dass durch die Kombination des Konzepts virtueller Testframeworks mit jenem des Cloud-Computing das bestehende Problem der mangelnden Skalierbarkeit elegant gelöst werden kann. Die praktische Umsetzbarkeit dieses neuen Ansatzes wird anhand einer konkreten Implementierung gezeigt. Diese Implementierung ist bereits im produktiven Einsatz und wird seit einiger Zeit erfolgreich für systematische Tests österreichischer E-Government-Komponenten eingesetzt. Damit trägt dieses System zur Gewährleistung der Qualität und damit auch der Sicherheit österreichischer E-Government-Lösungen bei.

2 Anforderungen an Testframeworks

Die stetig wachsende Verbreitung von Informations- und Kommunikationstechnologien (IKT) bewirkte in den letzten Jahren Veränderungen in vielen Bereichen des täglichen Lebens. Diese Entwicklung machte auch vor der öffentlichen Verwaltung nicht halt. Unter dem Begriff E-Government entstanden in den letzten Jahren zahlreiche Lösungen, die unter Verwendung von IKT Bürgerinnen und Bürgern die Möglichkeit bieten, Amtsgeschäfte und Behördenwege effizient auf elektronischem Wege abzuwickeln. Durch die strikten Sicherheitsanforderungen, die sich vor allem für transaktionale E-Government-Dienste ergeben, weisen E-Government-Lösungen mitunter einen beachtlichen Grad an Komplexität auf und machen die Integration zusätzlicher sicherheitssteigernder Konzepte wie beispielsweise Chipkarten notwendig [CeOB10].

Als zusätzlich erschwerender Faktor kommt hinzu, dass Technologieunabhängigkeit in der Regel eine zentrale Anforderung an E-Government-Dienste darstellt. Die Bereitstellung technologieunabhängiger Lösungen ist wichtig, um eine möglichst breite Masse an potentiellen Benutzerinnen und Benutzern zu erreichen und niemanden von der Verwendung bereitgestellter Dienste auszuschließen. In einigen Ländern wie Österreich ist die Notwendigkeit technologieunabhängiger Lösungen sogar gesetzlich verankert [EGov04].

Für die Entwicklung von E-Government-Diensten und Applikationen stellt die Forderung nach Technologieunabhängigkeit eine ernstzunehmende Herausforderung dar. Vor allem im Bereich clientseitiger Komponenten ergibt sich durch die Forderung nach Technologieunabhängigkeit eine Vielzahl möglicher Endbenutzersystemkonfigurationen, die im Zuge von Funktionstests entsprechend berücksichtigt werden müssen. In diesem Zusammenhang zeigt die langjährige Erfahrung in Entwicklung und Betrieb von E-Government-Lösungen, dass vor allem die Integration von Chipkarten, gegebene Abhängigkeiten von Java-Laufzeitumgebungen, die stetig steigende Anzahl an im Umlauf befindlichen Web-Browser-Varianten und Version, sowie unerwartete Änderung von im Umlauf befindlichen Betriebssystemen die Funktionalität von E-Government-Lösungen beeinträchtigen können [ZeKZ11].

Bisherige Testframeworks, die Entwicklerinnen und Entwickler dabei unterstützten, der stetig wachsenden Anzahl an unterschiedlichen Systemkonfigurationen Herr zu werden, versuchen hauptsächlich, folgende Kriterien zu erfüllen [ZeKZ11]:

- **Dynamische Adaptierbarkeit:** Updatezyklen von Betriebssystemen und Software sind oft unregelmäßig und anlassbezogen. Ein bestehendes Testframework muss daher sehr rasch und einfach in Hinblick auf neue Versionen und damit auf neue Systemkonfigurationen adaptierbar sein.
- **Effiziente Verwaltung:** Durch ständig neue Versionen von Betriebssystemen und anderen Softwarekomponenten steigt die Anzahl potentieller Systemkonfigurationen stets an. Effiziente Mechanismen zur einfachen und übersichtlichen Verwaltung der einzelnen im Testframework abgebildeten Systemkonfigurationen ist daher ein weiteres wichtiges Kriterium.
- **Zentrale Verfügbarkeit:** In der Regel arbeiten mehrere Entwicklerinnen und Entwickler bzw. verschiedene Teams an der Erstellung von E-Government-Diensten und Applikationen. Um Ressourcen zu sparen, sollte das Testframework zentral zugänglich sein und allen berechtigten Entwicklerinnen und Entwicklern zur Verfügung stehen.

- **Nachvollziehbarkeit:** Durchgeführte Tests sollten jederzeit wieder nachvollziehbar sein, um im Falle von Problemen, die direkt bei Endbenutzerinnen oder Endbenutzern auftreten, entsprechende Systemkonfigurationen einfach wiederherstellen zu können.

Obwohl diese Kriterien durchaus ihre Berechtigung haben und ohne Zweifel die Grundlage entsprechender Testframeworks bilden müssen, zeigt die Erfahrung im praktischen Umgang mit entsprechenden Umsetzungen, dass diese Kriterien in dieser Form unvollständig sind und einige wichtige Aspekte außer Acht lassen. Dementsprechend schlagen wir eine Erweiterung der o.g. Liste an Kriterien und Anforderungen an Testframeworks um folgende Punkte vor:

- **Kompatibilität mit Chipkartentechnologie:** Obwohl laufend neue technologische Ansätze zur Umsetzung qualifizierter Signaturlösungen entwickelt und erprobt werden, stellen Chipkarten nach wie vor eine zentrale und häufig genutzte Technologie im Rahmen von E-Government-Lösungen dar. Für systematische Tests – und hier vor allem für automatische Tests – stellt dies eine bedeutende Herausforderung dar. Testframeworks, die für die Evaluierung von E-Government-Lösungen verwendet werden sollen, müssen daher in der Lage sein, mit Chipkarten entsprechend umgehen zu können.
- **Kompatibilität zu Java:** Vor allem im Zusammenhang mit der Integration von Chipkartentechnologie stellt Java nach wie vor eine wichtige Technologie dar. Zudem erlaubt Java die Erstellung plattformunabhängiger Lösungen, was einen Einsatz dieser Technologie vor allem im Bereich des E-Government als sinnvoll erscheinen lässt. Testframeworks, die in diesem Bereich eingesetzt werden sollen, müssen daher in der Lage sein, mit Java entsprechend umgehen zu können, um Probleme, die sich im Umgang mit dieser Technologie ergeben können, geeignet abbilden und simulieren zu können.
- **Skalierbarkeit:** Durch die relativ kurzen Updatezyklen gängiger Betriebssysteme und Anwendungen ergibt sich in kurzer Zeit eine relativ große Anzahl an potentiellen Systemkonfigurationen, die durch das Testframework abgedeckt werden müssen. Die Erfahrung zeigte, dass bestehende Testframeworks hier oft sehr rasch an ihre Grenzen stoßen. Nachhaltige Testframeworks müssen daher einen entsprechenden Grad an Skalierbarkeit aufweisen, um mit einer ständig wachsenden Anzahl an potentiellen Systemkonfigurationen umgehen zu können.
- **Einfache Wartung der Testumgebungen:** Softwarekomponenten werden in unregelmäßigen Abständen mit Patches versorgt, die beispielsweise kritische Sicherheitslücken in Web-Browsern schließen. Derartige Patches müssen in alle Systemkonfigurationen, die von einem Testframework unterstützt werden, eingespielt werden, um diese auf einem aktuellen Stand zu halten. Mit einer steigenden Anzahl an unterstützten Konfigurationen kann dies zu einem erheblichen Wartungsaufwand führen. Testframeworks sollten daher Möglichkeiten vorsehen, diesen Wartungsaufwand so gering wie möglich zu halten.
- **Kosteneffizienz:** Betrieb und Wartung eines Testframeworks kann mit zunehmender Komplexität (d.h. steigender Anzahl an unterstützten Systemkonfigurationen) signifikante Kosten verursachen, da mitunter umfangreiche Hardware-Ressourcen benötigt werden. Testframeworks sollten dementsprechend so ausgelegt werden, dass Ressourcen möglichst effizient und sparsam genutzt und Kosten gespart werden.

Im Laufe der letzten Jahre wurden diverse Testframeworks entwickelt, die zum Ziel haben, Entwicklerinnen und Entwickler bei der Erstellung komplexer Softwarelösungen zu unterstüt-

zen. Ob und inwieweit diese Frameworks in der Lage sind, die hier definierten Anforderungen zu erfüllen, wird im folgenden Kapitel näher untersucht.

3 Bestehende Lösungen

Das systematische Testen von Softwarekomponenten zur Qualitätssicherung und Qualitätsverbesserung ist keine neue Thematik und betrifft nicht nur den speziellen Bereich des E-Government. Aus diesem Grund existieren bereits eine Vielzahl an Strategien und Werkzeugen, die sich mit dem Thema des qualitativen und automatisierten Testens beschäftigen. Im Rahmen dieses Kapitels werden daher überblicksmäßig existierende Lösungen vorgestellt und deren Tauglichkeit für einen Einsatz zum Testen von E-Government-Komponenten analysiert.

3.1 Überblick

Im Folgenden wird ein kurzer Überblick über existierende Systeme und Frameworks gegeben, die das Testen Web-basierter Anwendungen auf unterschiedlichen Client-Systemen erlauben.

3.1.1 Selenium

Selenium [HoKe06] ist ein auf HTML und JavaScript basierendes Testframework, welches speziell für das Testen von Web-basierten Anwendungen optimiert ist. Wesentliche Idee dabei ist die Aufzeichnung und automatische Wiederholung von Benutzerinteraktionen. Nebenbei bietet Selenium noch die Möglichkeit, Testfälle in unterschiedlichen Programmiersprachen wie Java, PHP oder Perl zu schreiben, welche gegen die meisten modernen Web-Browser durchgeführt werden können. Aufgrund seines Web-basierten Kerns ist Selenium betriebssystemunabhängig und kann daher auch mit den meisten Browsern verwendet werden.

3.1.2 TestingBot

TestingBot[1] setzt ein automatisiertes Testframework mittels des zuvor vorgestellten Frameworks Selenium in der Cloud um. Dabei können Webseiten rund um die Uhr bezüglich Browser-Kompatibilität getestet werden. TestingBot bietet dabei 96 Browser-Betriebssystem-Kombinationen in der Cloud an und unterstützt unterschiedliche Plug-Ins wie z.B. für PHP oder Ruby. Testfälle können einfach über das Selenium IDE Add-On erstellt und in das Test-Lab von TestingBot importiert werden. Wesentlicher Vorteil dieser Lösung ist die Möglichkeit, unterschiedliche Browser-Betriebssystem-Kombinationen parallel in der Cloud zu testen.

3.1.3 testCloud

Im Gegensatz zu den zuvor erwähnten Lösungen verfolgt testCloud[2] einen kontroversen Ansatz. Dabei werden Tests über eine Crowd von menschlichen Testern letztendlich manuell durchgeführt. Im Gegensatz zu technologiebasierten Cloud-Lösungen werden in diesem Fall jedoch nicht nur IT-Ressourcen, sondern auch „echte“ Menschen und „echte“ Geräte zum Testen bei Bedarf bereitgestellt. TestCloud bietet explorative Tests sowie Testfälle für alle Browser und Betriebssysteme an. Von den „gemieteten“ Testern gefundene Fehler werden über spezielle Bug-Report-Tools wie z.B. Jira[3] oder Redmine[4] bereitgestellt und können von

[1] http://testingbot.com

[2] https://www.testcloud.de

[3] http://www.atlassian.com/software/jira/overview

Kundinnen und Kunden einfach exportiert werden, um diese an die entsprechenden Entwicklerinnen und Entwickler weiterzuleiten.

3.1.4 Virtuelles Testframework für E-Government-Komponenten

Das Virtuelle Testframework für E-Government-Komponenten (VT-EGOV) [ZeKZ11] stellt bereits einen fundamentalen Ansatz dar, um E-Government-Komponenten sicher und zuverlässig zu testen. Dieser Ansatz verwendet als Basis virtuelle Maschinen, auf denen unterschiedliche Betriebssysteme, Browser und E-Government-Komponenten installiert werden können. Zentrale Idee dabei ist, der Dynamik sich ständig verändernder Umgebungen von Client-Systemen gerecht zu werden und somit ein effizientes Testen von sicherheitskritischen Komponenten zu ermöglichen.

3.2 Analyse

Im Rahmen dieses Unterabschnitts werden die zuvor beschriebenen Test-Frameworks auf deren Tauglichkeit hinsichtlich der in Abschnitt 3 beschriebenen Anforderungen analysiert. Tabelle 1 gibt einen prägnanten Überblick, inwiefern die einzelnen Anforderungen von den vorgestellten Tools erfüllt werden können. Ein Pluszeichen „+“ bedeutet dabei, dass die Anforderung erfüllt werden kann, ein Minuszeichen „-“, dass sie nicht erfüllt werden kann.

Tabelle 1 - Analyse bestehender Test-Systeme

Anforderung	Selenium	TestingBot	testCloud	VT-EGOV
Dynamische Adaptierbarkeit	-	-	-	+
Effiziente Verwaltung	-	+	-	+
Zentrale Verfügbarkeit	-	+	+	+
Nachvollziehbarkeit	+	+	+	+
Kompatibilität mit Chipkartentechnologie	-	-	-	+
Kompatibilität zu Java	-	-	-	+
Skalierbarkeit	-	+	-	-
Einfache Wartung der Testumgebungen	-	+	-	-
Kosteneffizienz	-	+	+	-

[4] http://www.redmine.org

Die Anforderung der dynamischen Adaptierbarkeit kann im Wesentlichen nur vom VT-EGOV-Framework erfüllt werden, da hier einfach und schnell neue Systemkonfigurationen erstellt werden können. Die anderen Tools sind hingegen zu allgemein für Browser-basierte Anwendungen gehalten, sodass die gewünschte Flexibilität nicht erreicht werden kann. TestingBot bietet zwar über 96 Browser-Betriebssystem-Kombinationen an, jedoch kann keine individuelle Kombination erstellt werden. Eine effiziente Verwaltung ist nur bei zwei der vorgestellten Lösungen gegeben (TestingBot und VT-EGOV), da nur diese mit der Möglichkeit ausgestattet sind, mit zahlreichen Browser-Betriebssystem-Kombinationen effizient umzugehen. Eine zentrale Verfügbarkeit des Testsystems ist mit allen Systemen bis auf Selenium möglich. Dieses Framework alleine, ohne weiteres Rahmenwerk wie beispielsweise bei TestingBot, ist speziell nur auf die Konfiguration jenes Rechners, auf dem Selenium installiert ist, zugeschnitten. Alle vorgestellten Systeme erfüllen das Kriterium der Nachvollziehbarkeit. Das heißt, dass durchgeführte Tests und Testkonfigurationen jederzeit reproduzierbar sind und Test einfach wiederholt werden können. Eine Kompatibilität zu Chipkartentechnologien ist hingegen nur beim VT-EGOV gegeben, da die anderen Systeme keine Möglichkeit anbieten, Zugriffe auf Kartenlesegeräte für Chipkarten in ihr Test-Framework einzubinden. Eine Kompatibilität zu Java ist ebenfalls nur beim VT-EGOV gegeben, da dieses System das einzige ist, welches erlaubt, Java in das Test-Framework einzubinden. Alle anderen Systeme sind im Wesentlichen für reine Web-Anwendungen optimiert.

Skalierbar ist im Wesentlichen nur die vorgestellte Cloud Lösung TestingBot, da hier das Cloud Computing-Paradigma vollständig ausgenutzt werden kann und mehrere Tests parallel abgearbeitet werden können. Benötigte IT-Ressourcen wie z.B. Arbeitsspeicher spielen dabei keine Rolle, wohingegen die anderen Lösungen in diesem Bereich an ihre Grenzen stoßen können. Auch das Kriterium der einfachen Wartung kann nur von der vorgestellten Cloud-Lösung erfüllt werden, da sich hier der Cloud-Anbieter um die unter dem Test-Framework liegende Infrastruktur kümmert. Bei allen anderen Lösungen muss die Wartung selbständig von den Entwicklerinnen und Entwicklern oder Administratorinnen und Administratoren, welche für das Testsystem zuständig sind, übernommen werden. Letztendlich ergibt sich die beste Kosteneffizienz bei der Cloud-Lösung TestingBot und bei der Crowd-Lösung testCloud. Bei beiden Lösungen ist es möglich, nur die wirklich benötigten Ressourcen anzufordern. Dadurch werden auch nur die wirklich konsumierten Leistungen verrechnet, was wiederum die Kosteneffizienz sicherstellt.

Insgesamt kann festgehalten werden, dass Lösungen basierend auf virtuellen Testumgebungen (VT-EGOV) bzw. Cloud-basierte Lösungen klare Vorteile gegenüber anderen Ansätzen aufweisen. Im folgenden Kapitel zeigen wir, wie durch eine Kombination dieser beiden Ansätze ein Testframework geschaffen werden kann, das allen in Kapitel 2 definierten Anforderungen genügt.

4 Cloud-basiertes Testframework

Testlösungen im Bereich des E-Government müssen besonderen und vor allem sich ständig verändernden Anforderungen genügen. In Österreich wurde zu diesem Zweck in den vergangenen Jahren ein virtuelles Testframework basierend auf der in Kapitel 3 vorgestellten Lösung VT-EGOV eingesetzt. Im praktischen Einsatz stellte sich jedoch heraus, dass dieses Testframework trotz des vielversprechenden Ansatzes einige Mängel aufweist, was sich vor allem in der mangelhaften Skalierbarkeit des Systems widerspiegelt. Um hier eine Verbesserung der Situation herbeizuführen und die vorhandene zentrale virtuelle Lösung an die flexib-

len sich ständig ändernden Anforderungen anzupassen, wurde die bestehende Lösung adaptiert und erweitert, um ein einfaches und effizientes Testen von E-Government-Komponenten zu ermöglichen. Die resultierende Lösung, welche den Ansatz virtueller Testframeworks weiterverfolgt und diesen um Aspekte des Cloud Computing erweitert, wird in diesem Kapitel näher vorgestellt.

Die Architektur der vorgeschlagenen Lösung wird anhand eines konkreten in Österreich im produktiven Einsatz befindlichen Testframeworks veranschaulicht. Dieses Framework verwendet Komponenten und Module des Unternehmens VMware[5]. Das grundlegende Konzept des vorgestellten Testframeworks ist jedoch nicht auf Komponenten dieses speziellen Herstellers beschränkt, sondern kann prinzipiell mit Hilfe beliebiger virtueller Cloud-Infrastrukturen umgesetzt werden.

Grundidee des hier vorgestellten Frameworks ist es, Benutzerinnen und Benutzern entsprechend dem VT-EGOV Konzept vorkonfigurierte Testumgebungen in Form virtueller Maschinen zur Verfügung zu stellen. Als Basis für die virtuelle Architektur dienen bei der hier vorgestellten Implementierung VMware ESXi Server [VMwS13], die ein gemeinsames Set von virtuellen Maschinen anbieten. Aufbauend darauf stellt VMware vCenter Server [VMwC13] eine zentrale Plattform für das Management der gesamten virtuellen Infrastruktur dar. Die Verwaltung der einzelnen für Testzwecke vorgesehenen virtuellen Maschinen wird über die Komponente vCloud Director [VMwD13], die Benutzerinnen und Benutzern einen rollenbasierten Zugang über eine Web-Konsole bietet, durchgeführt.

Die verwendeten VMware-Produkte bilden eine Private Cloud und ermöglichen so eine gemeinsame Nutzung der Infrastruktur. In Abb. 1 werden die wichtigsten Komponenten der virtuellen Cloud-Infrastruktur gezeigt. Sowohl ESXi Server als auch vCenter Server sind für Endbenutzerinnen und Endbenutzer transparent und nur für Administratorinnen und Administratoren des Testframeworks sichtbar. Zentraler Zugangspunkt für Benutzerinnen und Benutzer ist die vCloud Director Komponente.

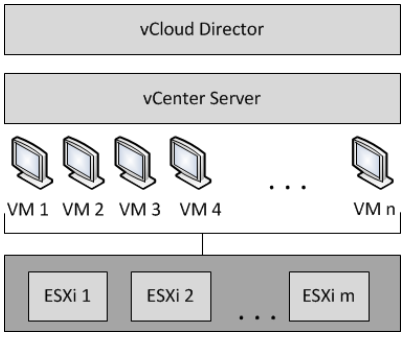


Abb. 1: Verwendete Architektur

Die grundlegende Architektur, die für die Bereitstellung des vCloud Directors notwendig ist, ist in Abb. 2 dargestellt. Benutzerinnen und Benutzer erreichen den vCloud Director über einen Web-Browser. Mehrere vCloud Director Server-Instanzen können bei Bedarf breitgestellt werden. Diese verwenden eine gemeinsame Datenbank und verbinden sich mit einem oder mehreren vCenter Servern.

[5] http://www.vmware.com

Zentrale Einheit bei der Verwaltung virtueller Maschinen über den vCloud Director sind sogenannte Organisationen. Eine Organisation ist eine Verwaltungseinheit, die Benutzerinnen und Benutzer, Gruppen und Rechenressourcen umfasst. Eine solche Einheit wird von Organisationsadministratorinnen und Organisationsadministratoren verwaltet.

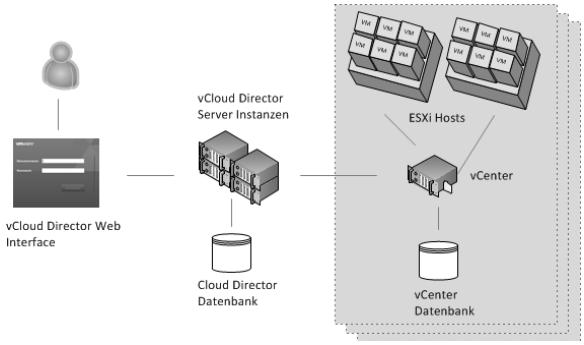


Abb. 2: Architektur des vCloud Directors

Administratorinnen und Administratoren können Organisationen Ressourcen wie Netzwerke, CPU-Kerne und Speicher zur Verfügung stellen. Dabei werden Ressourcen zu entsprechenden Klassen zusammengefasst. Abb. 3 illustriert dies für die vordefinierten Klassen Gold und Silber. Die Gold-Klasse beinhaltet CPUs mit erweiterten Befehlssätzen für kryptographische Operationen und Solid-State Disks für bessere Performance. Virtuelle Maschinen und vApps – Ansammlungen von virtuellen Maschinen – mit erhöhten Anforderungen können sehr einfach der entsprechenden Klasse zugewiesen werden. Intern werden die vorhandenen Ressourcen dynamisch und entsprechend der jeweiligen Klasse auf die zugewiesenen vApps aufgeteilt, wodurch eine effiziente Nutzung von Hardware-Ressourcen gewährleistet wird.

Eine weitere Effizienzsteigerung bei der Verwendung von Ressourcen kann durch das Konzept des Over-Provisioning erreicht werden. Dabei werden vorhandene Ressourcen mehrfach an Organisationen vergeben. Die Ressourcennutzung der Organisationen wird stetig überwacht, um bei Engpässen entsprechend rasch reagieren zu können. Dadurch ist es möglich, Ressourcen anhand des Nutzungsverhaltens der Organisationen entsprechend nur bei wirklichem Bedarf zu kaufen. Frühere Systeme wurden oft überdimensioniert und Ressourcen blieben dadurch ungenutzt.

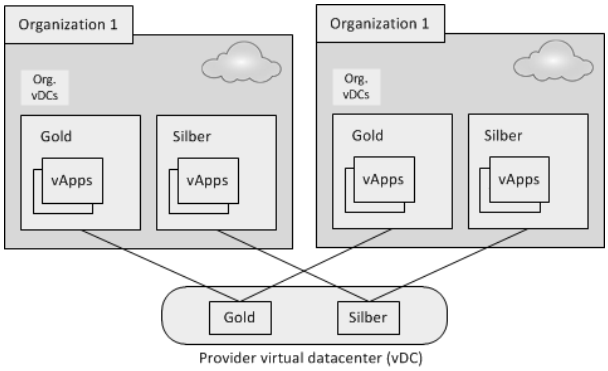


Abb. 3: Organisationen eines vCloud Directors

Benutzerinnen und Benutzer können über die Web-Konsole mit den Ressourcen der ihnen zugeteilten Organisation interagieren und neue virtuelle Maschinen und vApps erzeugen, verwenden und verwalten. Vordefinierte virtuelle Maschinen und vApps, die in typischen Testszenarien im E-Government-Bereich verwendet werden, sind in Katalogen gespeichert, die bei der Erstellung virtueller Maschinen als Vorlage dienen können. Je nach zugeteilter Rolle und Bedarf kann eine Benutzerin oder ein Benutzer neue Kataloge anlegen, vApps erstellen, vApps vom Katalog in die eigene Cloud hinzufügen, bestehende vApps der zugeteilten Gruppe starten, oder nur speziell für die Benutzerin oder den Benutzer vordefinierte vApps verwenden. Die virtuellen Maschinen werden in der Cloud gestartet. Der Zugriff auf die gestarteten Maschinen erfolgt über die Web-Browser-Konsole oder über Remote-Desktop-Protokolle. Bei der Bereitstellung der virtuellen Maschinen wird bereits festgelegt, wie lange vApps maximal ausgeführt werden können. Damit kann verhindert werden, dass nicht verwendete vApps unnötigerweise Ressourcen verbrauchen.

5 Evaluierung

Durch die Kombination des Ansatzes virtueller Testsysteme mit den Konzepten des Cloud Computing erreicht die hier vorgestellte Lösung im Vergleich zu anderen Ansätzen ein höheres Maß an Flexibilität und Funktionalität. Um die Tauglichkeit der hier vorgestellten Lösung im Kontext des E-Government zu evaluieren, wird diese im Folgenden hinsichtlich der im Kapitel 2 definierten Anforderungen analysiert.

- **Dynamische Adaptierbarkeit:** Durch den Einsatz von Cloud Computing und damit verbundener zentraler Lösungen, kann der Aufwand für die Zusammenstellung neuer Systemkonfigurationen gering gehalten werden. Benutzerinnen und Benutzer können aus einer großen Anzahl an virtuellen Maschinen eine eigene Testumgebung dynamisch erstellen. Dafür werden vordefinierte vApps, die eine oder mehrere virtuelle Maschinen beinhalten können, herangezogen. Typische E-Government-Testszenarien können durch diese vorkonfigurierten virtuellen Maschinen rasch nachgestellt werden. Die Zusammensetzung der vApps beginnt bei Systemen mit neuinstallierten Betriebssystemen, reicht über Systeme mit vorinstallierten Web-Browsern und geht bis hin zu Systemen mit speziellen vorinstallierten Softwarekomponenten aus dem E-Government-Bereich.
- **Effiziente Verwaltung:** Die VMware-Komponente vCloud Director bietet umfangreiche Mechanismen für die Verwaltung komplexer Testlandschaften. Virtuelle Maschinen können dabei sehr schnell benutzerspezifischen vApps hinzugefügt, verlinkt und wieder entfernt werden. Eine dynamische Erweiterung der Testumgebung durch neue VMs ist einfach möglich. Der praktische Einsatz dieser Lösung zeigt, dass die vordefinierten Systeme, die typische Testszenarien aus dem E-Government-Bereich repräsentieren, Benutzerinnen und Benutzern viel Zeit bei der Erstellung der gewünschten Testumgebung ersparen. Die integrierte Benutzerverwaltung und das umfangreiche Rechtemanagementsystem können dynamisch angepasst werden. Je nach Anforderung besteht so die Möglichkeit, Benutzerinnen und Benutzern entsprechende Rechte zuzuweisen.
- **Zentrale Verfügbarkeit:** Eines der Hauptmerkmale von Cloud Computing ist die zentrale Verfügbarkeit. Durch die Integration von Cloud Computing-Aspekten erfüllt auch das hier vorgestellte Testframework diese Anforderung vollständig. Der Einsatz dieser Technologie ermöglicht eine gemeinsame Nutzung von Ressourcen. Sowohl

das Testframework als Ganzes, als auch einzelne virtuelle Maschinen können von mehreren Benutzerinnen und Benutzern gleichzeitig verwendet werden.

- **Nachvollziehbarkeit**: Jede Benutzerin und jeder Benutzer besitzt seine eigene Test-Cloud mit eigenen vApps. Bestimmte Systemkonfigurationen, die für weitere Tests und eine spätere Nachvollziehbarkeit nützlich sind, können längerfristig gespeichert oder als Basiskatalog für alle Benutzerinnen und Benutzer zu Verfügung gestellt werden. Testsysteme können somit jederzeit einfach und rasch aus Basiskatalogen geklont werden, wodurch die spätere Nachvollziehbarkeit von Tests gewährleistet ist.
- **Kompatibilität mit Chipkartentechnologie:** Benutzerinnen und Benutzer können sich zu einer virtuellen Maschine über Remote-Desktop-Protokolle verbinden, welche eine Weiterleitung von USB-Geräten unterstützen. Somit ist gewährleistet, dass auch USB-Kartenlesegeräte am Testsystem verwendet werden können. Auf diese Weise können auch Tests mit Chipkarten sehr einfach durchgeführt werden.
- **Kompatibilität mit Java:** Java-Technologien können nach Belieben in das Testsystem eingebunden werden. Benutzerinnen und Benutzer können innerhalb einer virtuellen Maschine jede beliebige Technologie verwenden, die für die Durchführung der Tests notwendig ist.
- **Skalierbarkeit**: Durch die dahinterliegende Cloud-Infrastruktur ist das Testsystem in jeder Hinsicht beliebig erweiterbar. Sowohl Hardware- als auch Softwarekomponenten können dem bestehenden System jederzeit hinzugefügt werden.
- **Einfache Wartung der Testumgebung:** Sowohl der Wartungsaufwand für Administratorinnen und Administratoren des Systems über den zentralen vCenter Server als auch die Wartung der vApps jeder einzelnen Benutzerin bzw. jedes einzelnen Benutzers kann effizient und teilweise automatisch durchgeführt werden. Der zentrale Ansatz und die gemeinsam genutzte Infrastruktur begünstigt eine kostengünstige und einfache Wartung.
- **Kosteneffizienz**: Das Testsystem besteht aus einer virtuellen Private Cloud-Lösung, die geringe Infrastrukturkosten hat. Storage-Kosten sind die einzigen Hardwarekosten, die bei dieser Lösung anfallen. Die Linked Clone-Technologie [VMwD13] ermöglicht das Klonen von Basis-vApps in untergeordnete vApps, indem nur die Änderungen gespeichert werden, die von den untergeordneten vApps stammen, während die restlichen Daten aus den Basis-vApps verwendet werden. Durch den Einsatz dieser Technologie können Storage-Kosten reduziert werden. Durch die einfache Bereitstellung der virtuellen Maschinen und die effiziente Wartung der Testumgebung können ebenfalls Kosten gespart werden.

6 Fazit

Umfangreiche und systematische Tests stellen vor allem auch bei sicherheitskritischen IT-Lösungen eine wichtige Methode zur Gewährleistung der Einhaltung gegebener Qualitätsstandards dar. Vor allem im Bereich des E-Government zeigt die langjährige Erfahrung, dass die Durchführung systematischer Tests durch spezielle Anforderungen von E-Government-Lösungen oftmals schwierig und aufwändig ist. Bisher vorgestellte und etablierte Testframeworks können die zahlreichen Anforderungen oft nur teilweise und auf unbefriedigende Art und Weise erfüllen.

Um dieser Problematik geeignet zu begegnen, wurde in diesem Artikel ein neuartiges Konzept zur Durchführung effizienter und effektiver Tests im Bereich des E-Government vorge-

stellt. Dieses Konzept kombiniert Aspekte etablierter virtueller Testsysteme mit jenen des Cloud Computing und erlaubt so die Entwicklung eines Testframeworks, das in der Lage ist, allen Anforderungen an E-Government-Testsysteme zu genügen. Die praktische Umsetzbarkeit der hier vorgestellten Lösung wurde anhand einer konkreten Implementierung positiv evaluiert. Diese Implementierung basiert auf Virtualisierungskomponenten und Cloud-Lösungen der Firma VMware. Das vorgeschlagene und in diesem Artikel im Detail diskutierte Konzept ist jedoch prinzipiell auch über beliebige andere Komponenten mit entsprechender Funktionalität umsetzbar. Unabhängig von der gewählten Umsetzung bietet das vorgeschlagene Konzept Entwicklerinnen und Entwicklern von E-Government-Komponenten die Möglichkeit, bereitgestellte Lösungen effektiven und effizienten Tests zu unterziehen und dadurch geforderte Qualitäts- und Sicherheitsmerkmale einzuhalten.

Literatur

[CeOB10] M. Centner, C. Orthacker, W. Bauer: Minimal-Footprint Middleware for the Creation of Qualified Signatures. In: Proceedings of the 6th International Conference on Web Information Systems and Technologies 64-69, (2010)

[EGov04] E-Government-Gesetz (E-GovG), Bundesgesetzblatt für die Republik Österreich BGBl.I Nr. 10/2004, (2004)

[EP99] Richtlinie1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen, http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000-:013:0012:0020:DE:PDF, (1999)

[HoKe06] A. Holmes, M. Kellogg: Automating Functional Tests Using Selenium, Proceedings of AGILE 2006 Conference, (2006)

[HP11] Functional Testing Software, Simplify the automation of functional and regresion testing https://h10078.www1.hp.com/cda/hpms/display/main/hpms_-content.jsp?zn=bto&cp=1-11-127-24^1322_4000_100_, (2011)

[IBM11] Rational Functional Tester, Performance test automation for quality driven software delivery http://www-01.ibm.com/software/awdtools/tester/-performance/, (2011)

[ISO11] ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_-detail_ics.htm?csnumber=35733, (2011)

[VMwC13] VMware vCenter Server, http://www.vmware.com/products/vcenterserver/-overview.html, (2013)

[VMwD13] VMware vCloud Director, Key Features and Functionality, http://www.vmware.com/products/vcloud-director/features.html, (2013)

[VMwS13] VMware vSphere ESX and ESXi Info Center, http://www.vmware.com/products/vsphere/esxi-and-esx/overview.html, (2013)

[ZeKZ11] T. Zefferer, V. Krnjic, B. Zwattendorfer: Ein Virtuelles Testframework für E-Government Komponenten. In D-A-CH Security 2011, (2011)

Part IV

Publications Application of Electronic Signatures - Privacy

11 | Trust and Reliability for Public Sector Data

Journal	International Journal of Computer, Electrical, Automation, Control and Information Engineering
Language	English
Title	Trust and Reliability for Public Sector Data
Authors	Klaus Stranacher, Vesna Krnjic, Thomas Zefferer
Publisher	World Academy of Science, Engineering and Technology

World Academy of Science, Engineering and Technology
International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:7, No:1, 2013

Trust and Reliability for Public Sector Data

Klaus Stranacher, Vesna Krnjic, and Thomas Zefferer

***Abstract*—**The public sector holds large amounts of data of various areas such as social affairs, economy, or tourism. Various initiatives such as Open Government Data or the EU Directive on public sector information aim to make these data available for public and private service providers. Requirements for the provision of public sector data are defined by legal and organizational frameworks. Surprisingly, the defined requirements hardly cover security aspects such as integrity or authenticity.

In this paper we discuss the importance of these missing requirements and present a concept to assure the integrity and authenticity of provided data based on electronic signatures. We show that our concept is perfectly suitable for the provisioning of unaltered data. We also show that our concept can also be extended to data that needs to be anonymized before provisioning by incorporating redactable signatures. Our proposed concept enhances trust and reliability of provided public sector data.

***Keywords*—** Trusted Public Sector Data, Integrity, Authenticity, Reliability, Redactable Signatures.

I. Introduction

DURING the past few years, various developments in the IT sector have been significantly influenced by the so called "open movement". For instance, Open Source has become a well-known term that describes the philosophy of making source code publicly available to everybody. Also related concepts such as Open Access or Open Content have continuously gained popularity during the past years. Recently, especially the concept of *Open Data* has attracted attention. The general idea behind Open Data is that data should be freely available for everyone to be used and republished. Focus is thereby mainly put on non-textual data such as maps, genomes, or statistics, to name but a few.

Considering the different categories of data that are potentially affected by Open Data, it is hardly surprising that the public sector represents one of the most relevant data sources. The importance of governments and related public sector institutions is emphasized by different initiatives that deal with the provision of open data by the public sector.

An example is the *Open Government Data* (OGD) initiative. OGD can be seen as a subset of Open Data and pertains to data being under control of governmental institutions. Numerous OGD initiatives have been started recently in various countries and allow the provision of services based on data supplied by governmental organizations. For instance, in Vienna, Austria, more than 40 applications[1] that make use of OGD provided by the city government are already available for citizens ranging from various mobile smartphone apps to complex applications for desktop computers.

In addition, the public sector collects, creates, reproduces, and disseminates comprehensive sets of data in many areas such as social affairs, economy, weather, tourism, business, and education. Based on these data, new digital-content based products and services can be developed. The European Union considers this as a key factor for accessing and acquiring knowledge as well as rapid job creation, especially in small emerging companies. Therefore, the Directive 2003/98/EC of the European Parliament and of the Council of 17 November 2003 on the re-use of public sector information [10] (hereinafter referred to as *PSI Directive*) has been published. The directive defines a common legal framework for the provision of public data and the re-use of information sources enabling the "*harmonisation of the rules and practices in the Member States relating to the exploitation of public sector information"* [10].

In general, the term *public sector data* denotes all kinds of electronic data being produced, collected, provided, or simply processed by the public sector. In this paper we focus on public sector data used in the context of the OGD initiative and the PSI Directive. However, the methods proposed in this paper are not limited to these use cases.

Given the growing relevance and popularity of using public sector data in the public domain, security issues have been astonishingly rarely discussed so far. In literature, several requirements have been defined for OGD solutions [9]. However, security aspects such as data integrity or authenticity are hardly ever mentioned. Also the PSI Directive defines a set of basic requirements for solutions dealing with public sector information but does not clearly define data integrity or authenticity as a requirement.

Security in general and selected security aspects such as data integrity and authenticity in particular are without doubt important factors that should also be considered by public sector data based solutions. The use of forged data might for instance lead to resource claims. In such cases, the supplier of data should be able to proof that originally provided data has been altered. Current solutions based on public sector data usually do not support this feature.

In this paper, we propose a method that makes use of electronic-signature concepts in order to assure the integrity and authenticity of provided public sector data and information. Electronic signatures rely on public-key cryptography and basically represent the electronic equivalent

Klaus Stranacher is with the E-Government Innovation Center, Inffeldgasse 16a, 8010 Graz, Austria (e-mail: klaus.stranacher@egiz.gv.at).

Vesna Krnjic is with the E-Government Innovation Center, Inffeldgasse 16a, 8010 Graz, Austria (e-mail: vesna.krnjic@egiz.gv.at).

Thomas Zefferer is with the Secure Information Technology Center - Austria, Inffeldgasse 16a, 8010 Graz, Austria (e-mail: thomas.zefferer@a-sit.at).

[1] See http://www.data.gv.at/

to hand-written signatures. By applying a cryptographic method incorporating a private key to a set of data, the data is unambiguously linked to (i.e. signed by) the holder of the private key. The electronic signature can be verified using the corresponding public key. The verification process can only succeed, if the correct public key is used, and if the signed data is unaltered. Each modification of the signed data immediately breaks the electronic signature. This way, illegitimate alterations of signed data can be detected easily.

While the proposed solution is able to assure the integrity and authenticity of open public sector data, the application of electronic signatures also raises new challenges. In order to meet privacy requirements, data provided for public use needs to be redacted[2], i.e. altered. Of course, the modification of data would break any electronic signature on these data. To overcome this issue, we extend our approach by replacing the concept of electronic signatures by redactable signatures. Redactable signatures are a special kind of electronic signatures that allow for a (limited) modification of signed data without breaking the applied signature. We use redactable signatures to assure the integrity and authenticity of redacted data. This way, the proposed concepts enhance the overall security of solutions relying on public sector data.

The remainder of this paper is structured as follows. In Section II we discuss common requirements of public sector data and argue the need for additional requirements that cover data integrity and authenticity. We discuss electronic signature concepts that will be employed to meet these additional security requirements in Section III. Based on this theoretical foundation, we introduce our concepts to assure integrity and authenticity of public sector data using electronic and redactable signatures in Section IV. Final conclusions are drawn in Section V.

II. Common Requirements for Public Sector Data

OGD and PSI are main areas regarding the publishing and provisioning of public sector data. There are already a number of well-defined requirements for OGD as well as for the re-use of public sector information. In 2007, the Open Government Working Group [9] published a set of fundamental principles for Open Government Data. Also the PSI Directive establishes a minimum set of rules governing the re-use of existing documents[3] held by public sector bodies of the EU Member States.

In general, provision of government data in public sector should fulfil a set of requirements in order to assure an appropriate level of quality. In this context, the following aspects should be considered:

1) *Completeness*: OGD principles specify that all government data that are not subject to privacy or security restrictions should be made publicly available. The PSI Directive does not mention completeness of data explicitly. Provision of all appropriate documents held by the public sector is one of the goals of PSI. With regard to privacy, the PSI Directive states that: *"The Directive should be implemented and applied in full compliance with the principles relating to the protection of personal data in accordance with Directive 95/46/EC of the European Parliament and the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and of the free movement of such data."* [10].

2) *Primary Source*: OGD principles state that: *"Data should be published and collected at the source with the finest possible level of granularity, not in aggregate or modified forms."* [9]. The PSI Directive does not explicit provide any guidance for a primary source of data. It can be assumed that data provided by a public sector body fulfil this requirement.

3) *Timely Available*: OGD should be made available as fast as possible to the public. The benefit for the public can be enhanced through real-time update of time-dependent data. For PSI, there are no explicit rules for regulating the timely provision of documents. In the PSI Directive (12) is recorded that *"public sector bodies should make the documents available in a time-frame that allows their full economical potential to be exploited."* [10].

4) *Accessibility*: Public data must be made available barrier-free to widest range of users. The need for physical access to data (e.g. the attendance of special premises) should be avoided as well as the use of special electronic technologies. PSI data are not constricted to electronic data so the access to these data is not only through the Internet. Article 3 of the PSI Directive states that *"Where possible, documents shall be made available through electronic means."* [10].

5) *Machine Processible*: OGD should be stored in widely used file formats so that they could be automatically processed in order to ensure an easy integration in software applications. If data were normalized a sufficient documentation should be provided about the used file format. Likewise, the raw data should be available, which can be downloaded automatically. Article 5 of the PSI Directive states that *"Public sector bodies shall make their documents available [...] through electronic means where possible and appropriate."* [10].

6) *Data Access*: An anonymous access to the OGD should be possible for all users at any time. The access to the data should not be restricted to certain organizations or groups of people. Furthermore, users should not be forced to use certain software applications. In general PSI documents are not available for free. Therefore public sector documents usually need request for reuse (e.g. licence is needed).

7) *Non-Proprietary*: OGD specify the use of open standards to ensure that reading and processing of provided data does not require specific software. In most cases, it is necessary to provide data in different formats. PSI Directive states that *"Public sector bodies shall make their documents available in any pre-existing format or language [...]."* [10].

8) *License*: Open Government Data are license-free and not subject to any copyright. While in contrast the re-use of PSI imposes no strict guidelines. The Directive (Article 8)

[2] Redact means to make (portions of) a text unrecognizable (anonymization) or to substitute it with another text.

[3] The PSI Directive defines documents as *"any content whatever its medium (written on paper or stored in electronic form or as a sound, visual or audiovisual recording)"* [10]. For our following considerations we refer on electronically available data, which come under the Directive.

proposes that: *"Public sector bodies may allow for re-use of documents without conditions or may impose conditions, where appropriate through a licence, dealing with relevant issues."* and *"In some cases the re-use of documents will take place without a licence being agreed. In other cases a licence will be issued imposing conditions on the re-use by the licensee dealing with issues such as liability, the proper use of documents, guaranteeing non-alteration and the acknowledgement of source."* [10]. In addition, the Directive states that charges *"shall not exceed the cost of collection, production, reproduction and dissemination […]"* [10].

Table I summarizes the different requirements of public sector data and compares their impact on OGD and PSI.

TABLE I
OVERVIEW OGD AND PSI DIRECTIVE REQUIREMENTS

Requirement	Open Government Data	PSI Directive
Completeness	Data must be complete and privacy regulations must be taken into account.	Privacy regulations must be taken into account.
Primary Source	Data must originate from the primary source.	Not explicit mentioned, but public sector body should count as primary source.
Timely available	Data should be published as fast as possible.	Data should be provided in an appropriate time-frame.
Accessibility	Data should be published barrier-free and the need for physical access avoided.	Data is not restricted to electronic data, but shall be made available electronically.
Machine processible	Data should be in automatically proccesible formats.	Data should be provided through electronic means (where possible and appropriate)
Data Access	An anonymous access for anybody at any time should be provided.	Data is usually not publicly available and a request for reuse is needed.
Non-Proprietary	Data formats should base upon open standards to ensure the long-term readability.	Data should be available in any pre-existing format.
License	Data must be license-free and not subject to any copyright	No strict guidelines defined. Data may be provided under designated and non-discriminatory conditions

The focus of the above-mentioned principles of OGD and the re-use of PSI targets on completeness, timeliness, and accessibility of data. Security aspects have not been included, except the usage restriction of personal data. However, depending on the use case scenario we strongly recommend compliance with appropriate security requirements, especially for providing and publishing public sector data. We consider the previously defined requirements for public sector data (for certain scenarios) as incomplete and hence insufficient. Therefore, we extend the general principles by the following two requirements in order to appropriately consider security aspects:

1) *Authenticity and Integrity*: The authenticity and integrity of public sector data should be ensured by the use of appropriate cryptographic procedures. This shall establish that recipients of these data can check unauthorized modification (integrity) and beyond everyone can identify the provider of the data unambiguously (authenticity).

2) *Authenticity and Integrity for Redacted Government Data*: As defined in previous section of general requirements for public sector data, personal data must not be published as Open Government Data or be provided as public sector information because they underlie data privacy constraints. Often, the general information linked to these personal data can be of interest for the public and still be useful. Therefore, such data should be redacted in an appropriate way and thereafter be published without any privacy violation. This requirement must not be in conflict with the demand for authenticity and integrity. In any case, the authenticity and integrity of the redacted data must be ensured.

The discussed requirements extension for public sector data is a serious challenge for public sector bodies. A consideration of these extensions will necessarily include the integration of well-established and upcoming electronic signature concepts.

Therefore, the following Section III will present and discuss the cryptographic concepts that allow consideration of the defined extended requirements. Concrete concepts to implement appropriate procedures to take account of the extended requirements are finally discussed in Section IV.

III. CONCEPTS FOR ELECTRONIC SIGNATURES

In general, electronic signatures are used to provide a proof of genuineness for electronic data. Hence, electronic signatures represent the counterpart to hand-written signatures on paper based documents. Electronic signatures basically assure authenticity, data integrity, and non-repudiation of origin. The receiver of a signed document is able to uniquely identify the creator of the signature[4] (authenticity) and is able to verify that the signed data has not been modified (integrity). At the same time, the creator of an electronic signature cannot deny to have signed the data (non-repudiation). Especially the validation of data integrity becomes important for security-critical applications. For instance, in case of an electronically signed contract the content of the contract cannot be unilaterally modified without invalidating the electronic signature. We use the properties of electronic signatures to ensure both integrity and authenticity for public sector data. In the European Union, electronic signatures are widely used in transactional e-government processes and rely on a common legal basis formed by the EU Signature Directive [11] and their national implementations.

During the past decades, different forms of electronic signatures with different properties and characteristics have been developed. The security-enhancing concepts proposed in this paper basically rely on conventional electronic signatures and redactable signatures. We discuss relevant properties of these cryptographic methods in the following subsections.

[4] The creator of a signature is also called signatory.

A. Conventional Electronic Signatures

The technical basis for electronic signatures is public key cryptography. The creator of an electronic signature holds a private and a public key. The creator has sole control over the private key, which is used to create the signature[5]. Fig. 1 (a) illustrates the basic principle of a typical signature-creation process. In a first step, the signed data is mapped to a hash value of a fixed length using a so called hash function[6]. This hash value is then signed using the signatory's private key. The corresponding public key is published[7] and the receiver of the signed data is able to verify the validity of the signature by means of this public key.

Usually, the receiver of signed data wants to verify the validity of the obtained electronic signature. Therefore, the receiver executes a signature verification process as shown in Fig. 1 (b). As a first step, a hash value comparison is carried out. To do so, the verifier computes a hash value over the received signed data. The resulting hash value is then compared to the original hash value that can be extracted out of the obtained electronic signature. If these two hash values match, the data has not been altered[8]. If there is a difference between the two hash values, the data has obviously been modified after the signing process. In a second step (if the hash values are equal) the verifier checks if the public key matches the private key by applying the public key on the obtained signature value. If there is a match[9], the signature is called valid, otherwise invalid.

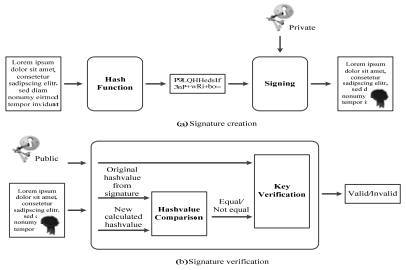


Fig. 1 Basic principle of electronic signatures

A fundamental property of conventional electronic signatures is that each modification of signed data leads to an invalidity of the signature. During the signature verification process, the hash value of the modified data is compared to the hash value of the original data. As the hash value of the modified data differs from the original hash value, the verification process results in an invalid signature. This way, the receiver of the signed data is able to detect modifications during the signature verification process.

For conventional electronic signatures a variety of different signature formats have been developed. For instance, XMLDSIG [12] and XAdES [13] are well established XML based signature formats. Similarly, Adobe PDF signatures [14] or PAdES [15] are commonly used signature formats for the signing of PDF documents.

B. Redactable Signatures

There exist use cases, in which a modification of signed data should be possible without leading to an invalidity of the signature. Such a use case is for instance the anonymization of data including personal and private data, which must not be published for legal and privacy reasons. *Redactable signatures* are a cryptographic concept, which allows a subsequent modification of signed data without invalidating the original signature. The person who is able to perform such modification is called the *redactor*.

The concept of redactable signatures is discussed in detail in [1]. The authors of this article define different properties of redactable signatures. These properties can be used to classify the different existing schemes for redactable signatures. The following properties exist:

1) *Property P1 – Designated Redactor*: This property defines if signed data can be modified by everybody or exclusively by a designated redactor, which is explicitly defined by the signatory.

2) *Property P2 – Replacement of Blocks*: This property defines if a redactor is able to delete (blacken out) text blocks only, or if the redactor is also able to replace it with other text blocks.

3) *Property P3 – Designated Parts*: A signatory is able to determine if a redactor is able to redact all text blocks or only designated blocks.

4) *Property P4 – Recognizable Modification*: This property defines if a modification of a redactor is recognizable afterwards.

5) *Property P5 – Controlled Replacement*: This property specifies if a signatory is able to control which text blocks can be used for the replacement (e.g. a certain text block can be defined to be replaceable by the text blocks "Yes" or "No" only).

By combining these properties, several different redactable signature schemes can be derived. The following Table I gives an overview of different redactable signature schemes and compares their properties.

[5] An important characteristic is that the private key cannot be determined out of the public key and is infeasible to guess.

[6] A hash function is a one-way function, which creates a fixed length checksum (hash value) out of arbitrary length data. Fundamental properties of hash functions are that it is neither possible to determine the original data out of a given hash value (pre-image resistance), nor to find another data, which maps to the same hash value (second-pre-image and collision resistance). The main reason for applying a hash function is that, in general, the data to be signed is quite large and signing large data is very inefficient and time consuming for practical applications.

[7] The public key is usually published via a trusted third party using an electronic certificate. This certificate holds the public key of the signatory and binds the signatory's identity to this key.

[8] This means the integrity of the data has been successfully checked.

[9] This means the signatory is the very person he or she claims to be.

TABLE I
REDACTABLE SIGNATURE SCHEMES AND THEIR PROPERTIES [1]

Signature Schema	P1	P2	P3	P4	P5
Content Extraction Signatures [2]	No	No	Partly	Yes	No
Sanitizable Signatures [3]	Yes	No	Yes	No	No
Homomorphic Signature Schemes [4]	No	No	No	Yes	No
Extended Sanitizable Signatures [5]	Yes	Yes	Yes	Yes	Yes
Extended Sanitizable Signature Schemes [6]	Yes	Yes	Yes	Yes	Yes
Generalizations and Extensions of Redactable Signatures [7]	No	Yes	Partly	Yes	Yes
Efficient signature schemes [8]	No	No	Yes	Yes	No

Independent from the respective scheme, the basic principle of all redactable signatures is the same. All schemes base on retention of the hash value of the original and unmodified data. For conventional electronic signatures, a different hash value indicates a modification of the signed data and leads to an invalid signature. However, if the original hash value is retained and used during the signature verification (instead of the new calculated hash value) the original signature can be validated successfully.

Fig. 2 shows the basic principle of redactable signatures by means of a simple example. It explains how a text is signed, afterwards redacted, and finally verified successfully. For the signature creation, a message *m* is divided into five text blocks m_1 to m_5. For each of these blocks, a hash function *H* is applied and the hash values h_1 to h_5 are computed. Based on these hash values, a total hash value H_{TOTAL} is calculated. This total hash value is then signed to create the signature *S*.

According to the example shown in Fig. 2, the text block *"redacted"* is then blacked out (message block $m_{4'}$). This leads to a hash value h_4, which differs from the original hash value[10] and would result in an invalid signature. To avoid this invalidity, the original hash value is used during the signature verification process[11].

Of course, this approach requires the receiver of the signature, who usually performs the signature verification, to receive the hash value h_4 in addition to the signature. Hence, the receiver, who only knows the redacted message, is able to verify the original signature without knowing the text block m_4. Due to the one-way functionality of the hash function, the receiver is not able to determine the redacted text block. By using conventional hash functions in association with a small number of potential text blocks (e.g. if only first names are possible for the redacted text block), there is a risk that the redacted text can be reconstructed by just trying all possible combinations. Thus, for real applications randomized hash functions are used. These hash functions are using an additional random value to calculate the hash value, which hinders a simple guessing of the text block.

Conventional electronic signatures are able to ensure authenticity and integrity for public sector data, whereas redactable signatures work well to fulfil the requirement for authentic and integrity-protected redacted public sector data as defined in Section II. In the following section we introduce our concept for a *trusted public sector data*, which bases on conventional and redactable signatures.

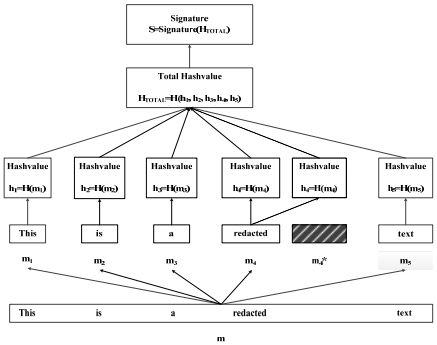


Fig. 2 Basic principle of redactable signatures

IV. TRUSTED PUBLIC SECTOR DATA

The objective of the presented concept is to ensure authenticity and integrity for public sector data including the possibility to anonymize or redact (parts of) these data. To fulfil these requirements, the proposed concept integrates conventional and redactable signature schemes as outline in Section III. In the following we discuss details of this concept and show how providers as well as recipients of public sector data benefit from this approach.

By using electronic signatures for public sector data, two general use cases can be distinguished. Depending on the use case, our concept makes use of different schemes for electronic signatures. In the following, the two general use cases covered by our concept are presented in detail.

A. Use Case 1: Ensuring Authenticity and Integrity for Public Sector Data

In this scenario we show how a provider of public sector data is able to provide authentic and integrity-protected data. Providing such secured data has following advantages:

1) *Integrity of the Data*: By ensuring the integrity of data, subsequent modifications of the data can be detected. Both, the data provider and the recipient of the data benefit from this feature. The recipient is able to trust the validity and correctness of the provided data. For the provider this feature guarantees that recipients cannot claim to have received incorrect data.

2) *Authenticity of the Data Provider*: The recipient of the public sector data is able to reliably determine the identity of the data provider. This leverages the trust in the reliability and trustworthiness of the provided data.

[10] $H(m_4)$ is unequal to $H(m_{4'})$.
[11] I.e. $H(m_4)$ is used for calculating H_{TOTAL} instead of $H(m_{4'})$.

The means of choice for implementing authentic and integrity-protected public sector data are conventional signature schemes. Fig. 3 illustrates the basic approach. The original public sector data source is located in the domain of the public sector data provider. These data is signed with the private signature key of the provider. Depending on the data format, different signature formats are possible. For instance, XML-based or PDF-based signatures are promising candidates, but basically each suitable signature format is applicable. Afterwards, the signed data is provided or published through appropriate communication channels as trusted public sector data[12].

To verify the authenticity and integrity of the data, the recipient can verify the electronic signature. In case of a valid signature the recipient has evidence that the data has not been altered or modified. Additionally, the recipient is assured that the data has been provided by the respective provider.

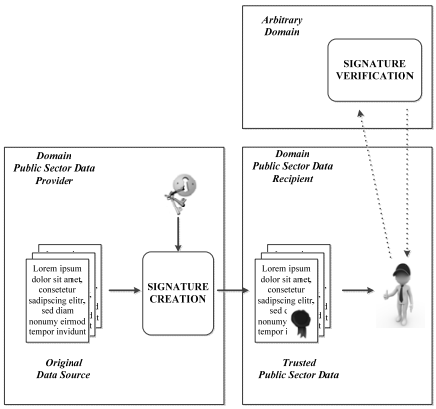


Fig. 3 Use case 1 – Ensuring authenticity and integrity for public sector data

B. Use Case 2: Authenticity and Integrity for Redacted Public Sector Data

This use case covers all applications, in which the original data set contains personal and private data. Usually, such data is prohibited for processing due to legal and privacy reasons. However, there exist applications where general data being linked to the private data is suitable to be reused. Hence, there is a need to anonymize or to redact the original personalized or private data.

For use case 1, a concept using conventional signatures to achieve authenticity and integrity has been proposed. This approach is not practical for the second use case. The anonymization process leads to a modification of the signed data and therefore to an invalid signature. In order to achieve trusted public sector data, the anonymized or redacted data must be signed again. For some applications, this is however no practical or feasible approach. For instance, the original signatory could not be available or a renewed signature creation could not be possible for other reasons. At this point redactable signatures produce a relief.

Fig. 4 shows the basic principle of trusted public sector data based on redactable signatures. The provider of the public sector data uses its private key to create a redactable signature. The redactor anonymizes or redacts the data and updates the redactable signature (i.e. adding the appropriate original hash values of the redacted blocks). For this purpose, the redactor must use his or her private key, if the provider has defined that only designated redactors are able to modify the signed data. After this, the redactable signature and the modified data are made available for the recipient. The recipient is able to verify the original signature without gaining access to the anonymized or redacted data. In case of a positive signature verification result, the recipient can again trust on the authenticity and integrity of the obtained data.

Depending on the concrete use case, different redactable signature schemes may be used. Depending on the properties of the chosen signature scheme, the provider is able to define designated redactors (property P1) or may define which parts of the data can be anonymized or redacted (property P2).

V. Conclusion

In this paper, we have proposed two concepts to assure the integrity and authenticity of public sector data being provided for re-use. Our first concept makes use of conventional electronic signatures to guarantee integrity and authenticity of arbitrary provided data. As conventional electronic signatures cannot be successfully applied if the signed data needs to be modified after the signature-creation process, this concept is not suitable for the provision of data that needs to be redacted. For these scenarios, we have proposed a second concept that relies on redactable signature schemes. These signature schemes allow for a successful signature-verification process, even if signed data needs to be modified.

By applying the proposed concepts, providers of public sector data can easily assure the integrity and authenticity of data intended for re-use by external parties. This leverages an appropriate level of security for solutions based on provided public sector data and is advantageous for both providers and recipients of data. Recipients can be sure that obtained data is genuine, unmodified, and stems from the expected source. At the same time, data providers benefit from the application of electronic signatures, as recipients cannot claim to have obtained incorrect data.

The different signature schemes, which the proposed concepts rely on, are already well established and frequently used in various security-sensitive fields of application. Software modules that facilitate the creation and verification of electronic signatures are publicly available and already frequently used in e-government and related fields of application. The implementation of a prototype application

[12] E.g. in the context of Open Government Data, these data may be published through publicly accessible interfaces. For data based on the PSI Directive the provider may give the recipient an appropriate access to the data.

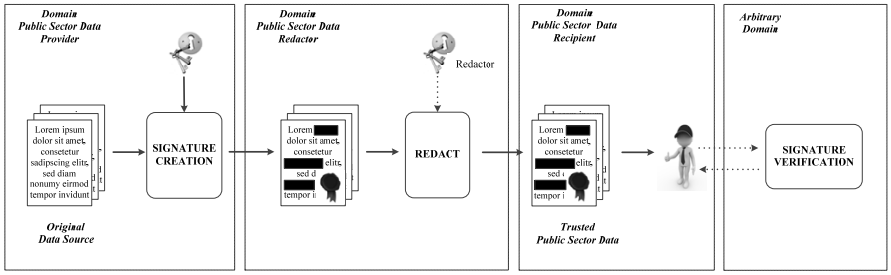


Fig. 4 Use case 2 – Authenticity and integrity for redacted public sector data

that relies on existing software solutions and incorporates the concepts proposed in this paper will demonstrate the practical applicability of our approach and is regarded as future work.

REFERENCES

[1] D.Slamanig und S.Rass, "Redigierbare Signaturen: Theorie und Praxis" in: Datenschutz und Datensicherheit, Bd. 35, Nr. 11, S. 757-762.
[2] R. Steinfeld, L. Bull und Y. Zheng: Content Extraction Signatures. ICISC, LNCS 2288, S. 285-304. Springer, 2001.
[3] G. Ateniese, D. H. Chou, B. de Medeiros und G. Tsudik. Sanitizable Signatures. ESORICS, LNCS 3679, S. 159-177. Springer, 2005.
[4] R. Johnson, D. Molnar, D. X. Song und D. Wagner. Homomorphic Signature Schemes. CTRSA, LNCS 2271, S. 244-262. Springer, 2002.
[5] M. Klonowski und A. Lauks. Extended Sanitizable Signatures. ICISC, LNCS 4296, S. 343-355. Springer, 2006.
[6] S. Canard und A. Jambert. On Extended Sanitizable Signature Schemes. CT-RSA, LNCS 5985, S. 179-194. Springer, 2010.
[7] D. Slamanig und S. Rass. Generalizations and Extensions of Redactable Signatures with Applications to Electronic Healthcare. CMS, LNCS 6109, S. 201-213. Springer, 2010.
[8] S. Haber, Y. Hatano, et al.: Efficient signature schemes supporting redaction, pseudonymization, and data identification. ASIACCS, S. 353-362. ACM, 2008.
[9] Open Government Working Group, 8 Principles of Open Government Data, http://www.opengovdata.org/home/8principles, 2007.
[10] The European Parliament and the Council of the European Union: Directive 2003/98/EC of the European Parliament and the Council of 17 November 2003 on the re-use of public sector information, Official Journal of the European Union L 345/90, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32003L0098:EN:NOT, 2003.
[11] The European Parliament and the Council of the European Union: Directive 1999/93/EC of the European Parliament and the Council of 13 December 1999 on a Community framework for electronic signatures, Official Journal of the European Union L 13/12, http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&numdoc=31999L0093&model=guichett&lg=en, 2000.
[12] W3C Recommendation: XML-Signature Syntax and Processing (Second Edition), http://www.w3.org/TR/xmldsig-core/, 2008.
[13] ETSI TS 101 903, Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES), V1.4.2, 2010
[14] Adobe Corporation, Document management — Portable document format — Part 1: PDF 1.7, First Edition, 2008.
[15] ETSI TS 102 778-1, Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview - a framework document for PAdES, V1.1.1, 2009.

Klaus Stranacher finished his MSc with distinction in Telematics at the University of Technology in the year 2006. Since 2005 he is working at the E-Government Innovation Center (EGIZ) in Graz. His main activities are in the area of E-Government and IT security especially on electronic identities, electronic documents and interoperability. During his activities he participates in several European research projects. He was involved the European electronic identity large scale pilot STORK (Secure IdentTity acrOss boRders linKed) and he is the leader of work package 2 (eDocuments) in the European large scale pilot SPOCS (Simple Procedures Online for Crossborder Services) under the ICT-PSP (Policy Support Programme), co-founded by EU. Additional he is working on his PhD thesis on interoperability of electronic documents, which is also his main research interest.

Vesna Krnjic finished her BSc with distinction in Informatics at the University of Technology Graz in the year 2010. Since 2010 she is working at the Institute for Applied Information Processing and Communications at the University of Technology Graz. Her main activities are in the area of E-Government and IT security with focus on usability and testing. Additionally she is working on her master thesis, which is about visual programming languages on smartphones, especially developed for children and teenagers.

Thomas Zefferer finished his MSc with distinction in Telematics at the University of Technology in the year 2007. Since 2007 he is working at the Institute for Applied Information Processing and Communications at the University of Technology Graz. Her main activities are in the area of E-Government and IT security. In the last years he was involved in many projects and activities of the E-Government Innovation Center (EGIZ) and the Secure Information Technology Center – Austria (A-SIT). His research focus lies on smartphone security and mobile E-Government processes, which is also the subject of his PhD thesis.

12 Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data

Conference	13th European Conference on e-Government
Language	English
Title	Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data
Authors	Klaus Stranacher, Vesna Krnjic, Bernd Zwattendorfer, Thomas Zefferer
Publisher	ACPI
Booktitle	Proceedings of the 13th European Conference on e-Government

Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data

Klaus Stranacher, Vesna Krnjic, Bernd Zwattendorfer
E-Government Innovation Center (EGIZ)[1], Graz University of Technology, Austria
Klaus.Stranacher@egiz.gv.at, Vesna.Krnjic@egiz.gv.at, Bernd.Zwattendorfer@egiz.gv.at

Thomas Zefferer
Secure Information Technology Center (A-SIT), Austria
Thomas.Zefferer@a-sit.at

Abstract. Due to the increased application of information and communication technologies in the public sector, the amount of data being produced and processed by the public sector has been constantly growing during the past years. As these data can also be useful for the general public and the corporate sector, current initiatives attempt to make these data publicly available. Recent work on this topic has shown that publishing of public sector data potentially raises several issues regarding data integrity and authenticity. These issues render the implementation of solutions based on trusted and reliable public sector data difficult. However, recent work has proposed electronic signatures in general and redactable electronic signatures in particular as adequate means to address these issues. While a variety of redactable signature schemes has been introduced in literature, their capabilities to assure the integrity and authenticity of published public sector data has not been assessed so far. This renders a concrete implementation of solutions based on redactable signatures impossible.
To overcome this problem, this paper first identifies and discusses legal, organisational, and technical requirements that need to be met by redactable signature schemes when applied to public sector data to be published. Afterwards, different existing redactable signature schemes are examined and discussed in more detail. Based on the previously identified requirements, the different redactable signature schemes are then assessed in detail. The conducted assessment reveals that sanitizable signature schemes, which represent a subset of redactable signature schemes, are especially suited to meet the predefined requirements. Among the wide set of existing sanitizable signature schemes, the conducted survey has revealed two concrete schemes to be best suited to assure the integrity and authenticity of public sector data to be published. The results obtained from the conducted survey will serve as input and basis for the implementation of solutions based on trusted and reliable public sector data.

Keywords: eGovernment, Redactable Signatures, Sanitizable Signatures, Public Sector Data

1. Introduction

The public sector produces, collects, processes, and provides large amounts of electronic data. These public sector data can be of interest also for the general public as well as for the corporate sector. In the area of e-Government, two main approaches have evolved to take up the challenge of providing public sector data. The Open Government Data (OGD) initiative bases on the concept of open data and claims that data should be freely available for everyone's use. In addition, the EU Directive on the re-use of public sector information (PSI Directive) (European Union, 2003) defines a legal framework for the provision of public data within the European Union.

Both approaches define partly different requirements for applications dealing with OGD and PSI related data. Surprisingly, security related aspects such as data integrity of authenticity of data are not part of these requirements. To bridge this gap, supplementary security requirements have been defined in literature recently (Stranacher et al., 2013). In this work, the authors have also proposed a concept to meet these additional requirements in practice. The proposed concept employs electronic signatures to allow for the realization of trusted and reliable public sector data. Furthermore, the proposed concept also includes a mechanism to assure the integrity and authenticity of data even if these data need to be redacted. For instance, a redaction can be necessary if the data contain security-sensitive or individual-related information. For such scenarios Stranacher et al. (2013) propose the use of redactable signature schemes, which allow third parties (redactors) to modify signed data without invalidating the original signature.

Redactable signature schemes have already proven their usefulness in different fields of application. During the past years, especially the e-Health sector has turned out to be predestinated

[1] EGIZ is a joint initiative of the Austrian Federal Chancellery and the Graz University of Technology

for an application of redactable signature schemes (Bauer et al., 2009) (Slamanig and Rass, 2010). So far, several different redactable signature schemes have been proposed and discussed in literature. These schemes differ in various fundamental properties, such as the possibility to explicitly define a designated redactor, or to allow the redacting of predefined data blocks only. Unfortunately, current concepts that propose a use of redactable signatures in order to assure authenticity and integrity of public sector data lack on an assessment and definition of appropriate redactable signature schemes so far.

In this paper we bridge this gap by assessing existing redactable signature schemes and evaluating their capabilities to meet the requirements of public sector data. For this purpose, we first recap the concept of trusted and reliable public sector data in Section 2. In Section 3, we then derive concrete requirements that have to be met by redactable signature schemes when being applied to the concept of trusted and reliable public sector data. Potential candidates of redactable signature schemes are examined in Section 4. In Section 5, we map the derived requirements to the examined redactable signature schemes in order to assess them schemes' capabilities to meet the given requirements.

2. Trusted and Reliable Public Sector Data

This section comprises a brief overview of the findings of Stranacher et al. (2013). Since the re-use of public sector information and the open publishing of governmental data do not define new issues, several requirements for such data provisioning techniques have already emerged over the past years. For instance, the Open Government Working Group (2007) has published eight fundamental principles for open government data. While also the PSI Directive includes some general and common requirements for providing public sector data, security requirements have not been defined.

Stranacher et al. (2013) define security requirements, namely data integrity and authenticity, when publishing public sector data. Both requirements ensure data consumers that published data have not been altered and are provided by a trustworthy authority. The authors also propose a concept for trusted and reliable public sector data. They distinguish two main use cases. In the first use case public sector data are published as it is. To ensure data integrity and authenticity, conventional electronic signatures are applied to these data. In the second use case, the public sector data contain personal and private data that need to be anonymized before publishing. Redactable signatures are used in this case. Figure 1 illustrates this use case and shows how trusted and reliable anonymization of public sector data without applying a new signature to the modified data is achieved. Avoiding the re-generation of electronic signatures e.g. might be useful if the person, who has originally signed the data, is not available anymore for re-signing for some reason.

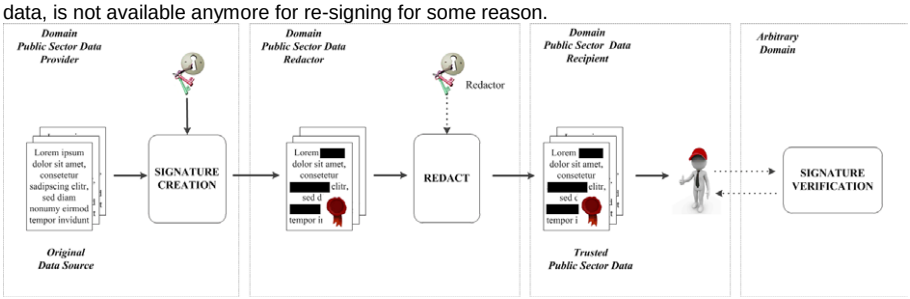


Figure 1: Authenticity and integrity for redacted public sector data (Stranacher et al., 2013)

In the following Section 3 we define concrete requirements redactable signatures for this use case. Additionally we give some more details on different redactable signature schemes and their applicability for public sector data in the sections 4 and 5.

3. Requirements for Redactable Signature Schemes

The proposed concept of Stranacher et al. (2013) for anonymized public sector data elaborates on the different properties of redactable signature schemes, but lacks on defining concrete requirements for

redactable signature schemes applied to anonymized public sector data. In order to close this gap, this section defines legal, organisational and technical requirements for redactable signature schemes.

3.1. General Legal Requirements

The concept of trusted and reliable public sector data bases on electronic signatures. The legal basis for electronic signatures is formed by the Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures (European Union, 1999). In addition, the national regulatory authorities are responsible for implementation of the Signature Directive on the national level. Therefore, following general legal requirements are defined:

- **Advanced Electronic Signatures**: Such a signature defines, among other things, that the signature is *"uniquely linked to the signatory"* and *"is capable of identifying the signatory"*. There a redactable signature scheme must satisfy the requirements of an advanced electronic signature as defined by European Union (1999). This is a prerequisite for accountability and to identify the original signer.
- **Qualified Electronic Signature**: In addition to the requirements for advanced electronic signatures a qualified signature requires to base on a qualified certificate and must be created using a secure signature creation device. These additional requirements are not necessarily needed for the public sector data use cases. Nevertheless a redactable signature scheme may, optionally, meet also the requirements for qualified electronic signatures as defined by European Union (1999).
- **Accountability**: In case of a dispute the signatory must be able to prove that certain modifications have been done by a certain redactor. Accountability can be achieved by technical means (see also technical requirements below).

3.2. General Organisational Requirements

Beside legal requirements, there exist also some general requirements on organisational level. These requirements concern mainly the role of the redactors and the signatory, i.e. the party, which holds the public sector data. So, following general organisational requirements are defined:

- **Definition and Revocation of Redactors**: Designated redactors should be easily definable by using existing systems (to avoid additional investments) and the signatory should also have the opportunity to revoke redactors.
- **Non-Disclosure Agreement**: Designated redactors must sign an appropriate confidentiality agreement. In particular regarding the data protection as redactors usually have access to private and personal data, which is governed by data protection regulations.
- **Responsibilities:** Responsibilities must be clearly defined both by the signatory and the redactors (e.g. who is allowed to sign/redact, who is responsible in case of a dispute).
- **Service Level Agreement/Security Compliance**: Redactors must ensure to redact data within an appropriate time frame (especially for real time data). In addition, redactors must be compliant to current security regulations as they operate on private and personal data.

3.3. Technical Requirements

On a technical level there exists also some requirements, which are tightly bound the particular redactable signature schemes. Therefore, we have defined following technical requirements:

- **Designated Redactors**: Designated redactors must be able to be specified by the redactable signature scheme. That means that the signatory must be able to determine who is allowed to modify the signed data. Persons except the signatory and the designated redactors must not be able to redact data without breaking the originally signature applied. Any change of the data by unauthorized persons must be recognizable.
- **Privacy**: The redactable data as well as the original signature must not allow revealing the redacted message blocks.

- **Designated Parts**: The signatory must be able to specify which data blocks may be modified. Editing unauthorized data must be recognized and must lead to an invalid signature.
- **Accountability**: See definition in legal requirements.
- **Applicability**: The scheme must be applicable on structured data such as XML (W3C Recommendation, 2008).
- **Compatibility:** The signature scheme should be compatible with existing signature standards, such as XMLDSIG (W3C Recommendation, 2008) or XAdES (ETSI, 2010).

4. Examination

Redactable Signatures provide a cryptographic mechanism to allow redactors to apply modifications to signed messages without invalidating the original signature and have been introduced by Steinfeld et al. (2001) and Johnson et al. (2002). This mechanism has many applications in electronic healthcare as shown by Slamanig and Rass (2010) and several other areas presented in Ateniese et al. (2005). A main property of redactable signatures is that they only allow blacking certain parts of the signed data. To remove or replaced designated parts of the signed messages with an arbitrary string, Ateniese et al. (2005) proposed Sanitizable Signatures. Sanitizable signatures can be seen as a small subset of redactable signatures, as they are basically redactable signatures where the replacement part is permanently exchanged.

Figure 1 shows an overview of about the most relevant redactable and sanitizable signature schemes proposed in the last years and their relation to each other. There exist also other schemes (not shown in Figure 1), but either they have been the basis for one of the mentioned schemes or they have been proven as insecure or not applicable. For instance, the authors of Yuen et al. (2008) lacks on accountability of the proposed schema or Pöhls et al. (2011) contains only minor updates on the property transparency (which is not of special interest for our use cases).

For our following examination we have looked initially on the redactable signature schemes proposed by Steinfeld et al. (2001), Johnson et al. (2002), Slamanig and Rass (2010), Chang et al. (2009) and Brzuska et al (2010a). Right at the beginning of the examination we have figured out that all of these schemes do not support the specification of designated redactors. As this is one of the main requirements for the public sector data use cases, all of these schemes are not applicable for these scenarios. Therefore we omitted an in-depth analysis of these schemes and concentrated on sanitizable signature schemes instead.

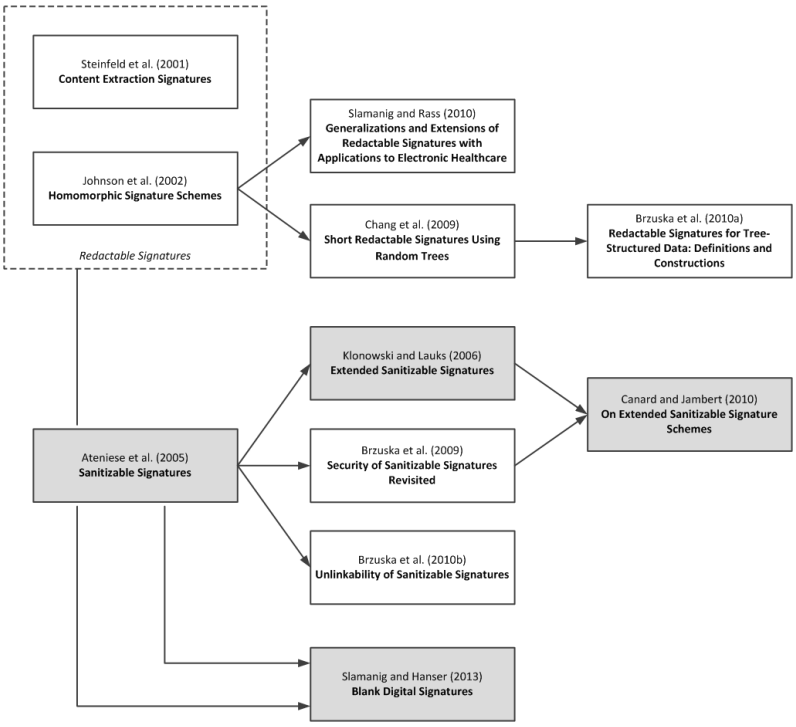


Figure 2: Overview about redactable and sanitizable signature schemes

Figure 2 shows the sanitizable signature schemes we have chosen for our examination (highlighted in grey). A few sanitizable signature schemes we have skipped from our examination due to following reasons:

- Brzuska et al. (2009) proposed a rigorous security model. This model has been incorporated by Canard and Jambert (2010), which is examined below. Therefore we have skipped it from our analysis.
- Brzuska et al. (2010b) proposed an update of Ateniese (2005) which does not permit creating a link between different signatures over the same original message. This functionality is not of interest for the public sector use cases, so we have skipped this scheme.

Following sub-sections give the examination of the chosen sanitizable signature schemes. In addition, we examine on the proposal of Slamanig and Hanser (2013) on Blank Digital Signature, which incorporates the findings of redactable and sanitizable signatures.

4.1. Sanitizable Signatures by Ateniese et al. (2005)

The basic principle of redactable signatures bases upon commitments[2], which in turn build upon hash-functions. This principle basis upon retaining the original hash values for redacted message blocks and to use them during the signature verification process (instead of calculating a new hash value over the redacted data). This process is described in Stranacher et al. (2013) and in more detail in Johnson et al. (2002) and Steinfeld et al. (2001).

[2] Commitments are often used in cryptographic protocols. They allow a committer to publish a commitment (= a value), which binds the committer to a certain message, but without revealing it. If a verifier wants to check if the message is consistent with the commitment, the committer may open the commitment to reveal the message.

Ateniese et al. (2006) proposed the first scheme for sanitizable signatures, where a designated redactor is able to modify designated parts of a signed message. Here the basic principle bases on chameleon hash-functions instead of conventional hash-functions for conventional signatures. Such chameleon hash-functions are parameterized with the public key of the redactor. Because of the parameterization, the redactor is able to compute collisions. This means the redactor is able to generate messages, which lead to the same hash value as for the data, which is going to be redacted. Based on this mechanism the redactor can replace message blocks with arbitrary message blocks and the verification of the original signature will not fail. In this case it is neither possible to detect if a message has been redacted nor it is possible to detect which message blocks have been modified. Therefore the authors propose to add non-redactable meta information after each redactable message block indicating the restriction for the message to be replaced. Obviously, this is a very inefficient solution.

4.2. Extended Sanitizable Signatures by Klonowski and Lauks (2006)

Klonowski and Lauks (2006) extended the scheme of Ateniese et al (2005). They omitted the added meta information and extended the schema itself to allow the signatory to limit the message blocks which are modifiable by the redactor and to limit the messages which are replaced. This scheme also bases on chameleon hash-functions. For the message replacement restrictions they propose to use accumulators[3] or bloom filters[4].

4.3. On Extended Sanitizable Signature Schemes by Canard and Jambert (2010)

Canard and Jambert (2010) presented a second approach to limit the modification of message blocks and the message to be replaced by the scheme itself. As for the other sanitizable signature schemes, the authors base their proposal on chameleon hash-functions. In addition, they use pseudorandom generators and accumulators to implement the message replacement restrictions.

4.4. Blank Digital Signatures by Slamanig and Hanser (2013)

Slamanig and Hanser (2013) proposed a new signature scheme, which bases on redactable and sanitizable signatures. They specified a message template, which is defined by an originator and describe a message containing fixed message blocks and multiple choices of message blocks, which are exchangeable. This template is signed by the originator. A proxy[5] is then able to sign an instantiation of this template, i.e. selecting concrete message blocks of the defined choices. Finally, the resulting message can be verified by a third party using the originator's and proxy's verification keys. Their proposal builds upon conventional signature schemes, elliptic curve cryptography and polynomial commitments[6].

5. Assessment

5.1. Legal and Organisational Assessment

In this section, we evaluate redactable and sanitizable signature schemes based on legal and organisational requirements. In order to use redactable and sanitizable signatures for ensuring trusted

[3] An accumulator is a one-way hash function which satisfies a quasi-commutative property. See Benaloh and Mare (1994) for details.
[4] Bloom filters are data structures which allow to efficient test whether an element is a member of a certain set or not. See Bloom (1970) for details.
[5] For the public sector use cases the proxy can be seen as the redactor.
[6] Polynomial commitments are conventional commitments applied to polynomial functions.

and reliable public sector data, all defined requirements must and can be fulfilled by the proposed signature schemes.

The European Union has published the EU Signature Directive (European Union, 1999) to define how electronic documents can achieve statutory trust within its Member States. While this directive primarily considers conventional electronic signatures, the use of redactable and sanitizable signatures compliant with this directive has been only slightly discussed so far. Höhne et al. (2012) and Brzuska et al. (2012), for instance, examine legal consequences of redactable and sanitizable signatures. They especially argue that redactable and sanitizable signatures are compliant to advanced electronic signatures but cannot be used for qualified electronic signatures according to the EU Signature Directive. The reason for being not compliant with qualified electronic signatures constitutes missing displaying possibilities for the signatory. According to the Signature Directive, the data to be signed must be viewable by the signatory before the signature creation process. This requirement cannot be fulfilled by redactable and sanitizable signatures as modifications of signed data are possible also after signature creation, which the signatory cannot be aware of at the time of the signature creation process regardless the signatory is able to define which message parts are able to be modified and how they can be modified. Another legal requirement to be fulfilled by the proposed signature schemes is accountability. Accountability means that redactors, who used her private keys to modify signed data, can be determined. This requirement cannot be met by all described signature schemes (see following Section 5.2).

Equal to legal requirements, several organisational requirements must be met by the proposed signature schemes in order to successfully apply redactable and sanitizable signatures to public sector or open government data. In fact, all organisational requirements identified in Section 3.2 are independent of the technical implementation of the proposed signature schemes. While some organisational requirements may be fulfilled using technical means, others require solutions on organisational level. For instance, the requirement on revoking designated redactors can be fulfilled on technical level as all of the proposed schemes rely on a public key infrastructure (PKI) and hence on existing and well-established revocation mechanisms. However, other organisational requirements still require organisational measures. This particularly means that a fulfilment of those requirements requires e.g. some kind of contractual agreements between all involved parties. Within such agreements, especially individual responsibilities, signature validity limitations, or liability questions must be thoroughly elaborated.

5.2 Technical Assessment

This sub-section comprises the technical assessment of the examined sanitizable signature schemes according to the defined requirements in Section 3. In the following, the schemes are assessed in detail and Section 5.2.5 summarizes the findings of this technical assessment.

5.2.1 Assessment of Sanitizable Signatures by Ateniese et al. (2005)

Ateniese et al. (2005) states "[…] as a secure digital signature scheme that allows a semi-trusted censor to modify certain designated portions of the message […]"[7]. That means the requirement for designated redactor and designated parts is fulfilled. In addition the privacy is also fulfilled as "[…] the indistinguishhability requirement provides for privacy". The author also state that "accountability follows from the unforgeability requirement", but this has been proven by Brzuska et al. (2009) as not true. So the Ateniese sanitizable signature scheme does not provide accountability.

5.2.2 Assessment of Extended Sanitizable Signatures by Klonowski and Lauks (2006)

The extended sanitizable signature scheme of Klonowski and Lauks (2006) provides a designated redactor and designated parts as stated by the authors: *"[…] in this scheme the designated censor can change the content of designated (so called mutable) parts of a signed message […]"*. They also state that privacy is fulfilled due to the basement of their extended scheme on Ateniese et al. (2005).

[7] They used the name censor for the redactor.

Concerning accountability we have to distinguish between the two characteristics of this scheme. The accumulator technique provides accountability whereas bloom filter does not. Nevertheless, the authors miss a concrete security model and proofs for their proposed schema.

5.2.3 Assessment of Extended Sanitizable Signature Schemes by Canard and Jambert (2010)

As this scheme strongly bases on Ateniese et al. (2005), it provides designated redactors as needed by our defined requirements. In addition, Canard and Jambert (2010) state that *"[…] to force some admissible blocks of a signed message to be modified only into a predefined set of sub-messages."*[8] and *"[…] privacy is also included by transparency in the extended model."*. Thus, the scheme fulfils the requirements for designated parts and privacy. In addition, the authors prove that *"Unforgeability (and thus accountability) is reached thanks to the computation of a new tag per message."*. This is one of the major extensions of Ateniese et al. (2005).

5.2.4. Assessment of Blank Digital Signatures by Slamanig and Hanser (2013)

Slamanig and Hanser (2013) state that *"Immutability guarantees that no malicious proxy can compute message templates or templates instantiations not intended by the signer."* and *"[…] is called private, if for any polynomial-time algorithm A the probability of winning Game 2 is negligible as a function of security parameter k."* It follows that the proposed scheme provides a designated redactor and privacy. The requirement, that designated parts must definable, is fulfilled because of the proposed template mechanism, where the signatory defines a message template. Additionally accountability is also fulfilled as the proxy signs the template instantiations with a conventional signature, which provides accountability.

5.2.5. Technical Assessment Summary

The requirements for applicability to structured data and compatibility with existing signature standards can be assessed together for all examined schemes. Pöhls et al. (2011) have shown several implementations of sanitizable signatures based upon XML and the W3C Recommendation (2008) on XML-Signature Syntax and Processing (XMLDSIG). The authors have proven that sanitizable signatures are applicable to structured data and fit into XMLDSIG without invalidating the recommendation. In addition, the findings of Pöhls et al. (2011) may be applied to the examined schemes with slight changes.

Table 1 summarizes the results of the assessment. It shows that Ateniese et al. (2005) lacks on the requirement on accountability. Furthermore Klonowski and Lauks (2006) miss a security model and proofs for the proposed scheme. Therefore these two schemes are assessed to be not suitable for the public sector data use cases.

In contrast, the sanitizable signature schemes of Canard and Jambert (2010) and Slamanig and Hanser (2013) meet all technical requirements. Hence these schemes are appropriate to the use cases of redacted public sector data as defined in Stranacher et al. (2013).

Table 1: Technical assessment of examined sanitizable signature schemes

Signature Scheme	Design. Redactor	Privacy	Design. Parts	Account-ability	Applicable to Structured Data	Compatibility	Comment
Ateniese et al. (2005)	Yes	Yes	Yes	No	Yes	Yes	
Canard and Jambert (2010)	Yes	Yes	Yes	Yes	Yes	Yes	

[8] Message parts which can be modified by a redactor are often called admissible blocks.

Klonowski and Lauks (2006)	Yes	Yes	Yes	Yes[9]	Yes	Yes	No security model and no proofs are given
Slamanig and Hanser (2013)	Yes	Yes	Yes	Yes	Yes	Yes	

6. Conclusions

The emerging trend to make public sector data available to the general public and to the corporate sector raises the demand for innovative techniques to meet arising security requirements. Electronic signatures in general and redactable electronic signature schemes in particular have recently been proposed as adequate enabler for such security preserving techniques.

In this paper we have made the next step towards a concrete implementation of these techniques by evaluating different proposed schemes for redactable signatures and by assessing their capabilities to enhance the security of publishing (anonymized) public sector data. The assessment has been based on a set of legal, organisational, and technical requirements, which have previously been defined and discussed. The conducted assessment of existing redactable signature schemes has revealed that especially sanitizable signature schemes, which represent a subset of redactable signatures schemes, are well suited to enhance the security of published public sector data. Among the set of evaluated sanitizable signature schemes, especially two schemes proposed by Canard and Jambert (2010) and by Slamanig and Hanser (2013) have turned out to be able to meet given legal, organisational, and technical requirements.

The results that have been obtained from the conducted assessment pave the way for several future activities in this field. In a next step, the two most promising schemes that have been identified by the conducted assessment will be implemented and integrated into approved electronic signature schemes such as XMLDSIG. This implementation will then serve as basis for the development of solutions based on trusted and reliable public sector data.

References

Ateniese, G., Chou, D. H., de Medeiros, B., Tsudik, G. (2005), *Sanitizable Signatures*, in European Symposium on Research in Computer Security ESORICS 2005, LNSC, vol. 3679, pp. 159-177, Springer.

Bauer, D., Blough, D., Mohan, A. (2009), *Redactable Signatures on Data with Dependencies and their Application to Personal Health Records*. In: Proc. of the 8th ACM Workshop on Privacy in the Electronic Society, WPES '09, pp. 91–100. ACM Press, New York

Benaloh, J., Mare, M., (1994), *One-Way Accumulators: A Decentralized Alternative to Digital Signatures*, in Advances in Cryptology — EUROCRYPT 1993, LNCS, vol. 765, pp. 274-285, Springer.

Bloom, B. (1970), *Space/time trade-offs in hash coding with allowable errors*, in Communication of ACM, vol. 13, no. 7, pp. 422-426

Brzuska, C., Fischlin, M., Freudenreich, T., Lehmann, A., Page, M., Schelbert, J., Schröder, D., Volk, F. (2009), *Security of sanitizable signatures revisited*, in Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 317–336. Springer.

Brzuska, C., Busch, H., et al. (2010a), *Redactable Signatures for Tree-Structured Data: Definitions and Constructions*, in Applied Cryptography and Network Security 2010, LNCS, vol. 6123, pp. 87-104, Springer.

Brzuska, C., Fischlin, M., Lehmann, A., Schröder, D. (2010b), *Unlinkability of Sanitizable Signatures*, in Public Key Cryptography – PKC 2010, LNCS, vol. 6056, pp. 444-461, Springer

Brzuska, C. Pöhls, H., Samelin, K. (2012), Non-Interactive Public Accountability for Sanitizable Signatures, in Proceedings of the 9th European PKI Workshop: Research and Applications (EuroPKI 2012), Springer, Note: to appear.

Canard, S., Jambert, A. (2010), *On Extended Sanitizable Signature Schemes*, in Topics in Cryptology - CT-RSA 2010, LNCS, vol. 5985, pp. 179-194, Springer.

Chang, E., Lim, C., Xu, J. (2009), *Short Redactable Signatures Using Random Trees*, in Topics in Cryptology – CT-RSA 2009, LNCS, vol. 5473, pp. 133-147, Springer.

[9] This scheme supports accountability only for the version where accumulators are used. In case the bloom filter is used accountability is no achievable.

ETSI (2010), *Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)*, V1.4.2.
European Union (1999) *Directive 1999/93/EC on a Community framework for electronic signatures*.
European Union (2003) *Directive 2003/98/EC of the European Parliament and the Council of 17 November 2003 on the re-use of public sector information*
Höhne, F., Pöhls, H., Samelin, K. (2012), Rechtsfolgen editierbarer Signaturen, in Datenschutz und Datenrecht (DuD), vol. 36(6), pp. 485-491
Johnson, R., Molnar, D., Song, D. X., Wagner, D. (2002), *Homomorphic Signature Schemes*, in Topics in Cryptology CT-RSA 2002, LNCS 2271, pp. 244-262, Springer.
Klonowski, M., Lauks, A. (2006), *Extended sanitizable signatures*, in: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 343–355. Springer.
Open Government Working Group (2007), *8 Principles of Open Government Data*, http://www.opengovdata.org/home/8principles.
Pöhls, H., Samelin, K., Posegga, J. (2011), *Sanitizable Signatures in XML Signature — Performance, Mixing Properties, and Revisiting the Property of Transparency*, in Applied Cryptography and Network Security, LNCS, vol. 6715, pp. 166-182, Springer.
Slamanig, D., Hanser, C. (2013), *Blank Digital Signatures*, in Proceedings of 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2013), Note: to appear.
Slamanig D., Rass, S. (2010), *Generalizations and Extensions of Redactable Signatures with Applications to Electronic Healthcare*, in Communications and Multimedia Security 2010, LNCS, vol. 6109, pp. 201-213. Springer.
Steinfeld R., Bull, L., Zheng, Y, (2001), *Content Extraction Signatures*, in Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 285–304. Springer.
Stranacher, K., Krnjic, V., Zefferer, T. (2012), Vertrauenswürdiges Open Government Data, in 1.OGD D-A-CH-LI Konferenz, pp. 27-39.
Stranacher, K., Krnjic, V., Zefferer, T. (2013), Trust and Reliability for Public Sector Data, Note: to appear.
W3C Recommendation (2008), *XML-Signature Syntax and Processing (Second Edition)*, http://www.w3.org/TR/xmldsig-core/
Yuen, T., Susilo, W., Liu, J., Mu, Y. (2008), *Sanitizable Signatures Revisited*, in Cryptology and Network Security, LNCS, vol. 5339, pp. 80-97, Springer.

13 Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data

Journal	Electronic Journal of e-Government, Volume 11
Language	English
Title	Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data
Authors	Klaus Stranacher, Vesna Krnjic, Bernd Zwattendorfer, Thomas Zefferer
Publisher	Academic Conferences Limited
Booktitle/Journal	Electronic Journal of e-Government

Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data

Klaus Stranacher[1], Vesna Krnjic[1] and Bernd Zwattendorfer[1] and Thomas Zefferer[2]
[1]E-Government Innovation Center (EGIZ)[1], Graz University of Technology, Austria
[2]Secure Information Technology Center (A-SIT), Austria
Klaus.Stranacher@egiz.gv.at
Vesna.Krnjic@egiz.gv.at
Bernd.Zwattendorfer@egiz.gv.at
Thomas.Zefferer@a-sit.at

Abstract: Due to the increased application of information and communication technologies in the public sector, the amount of data being produced and processed by the public sector has been constantly growing during the past years. As these data can also be useful for the general public and the corporate sector, current initiatives attempt to make these data publicly available. Recent work on this topic has shown that publishing of public sector data potentially raises several issues regarding data integrity and authenticity. These issues render the implementation of solutions based on trusted and reliable public sector data difficult. However, recent work has proposed electronic signatures in general and editable electronic signatures in particular as adequate means to address these issues. While a variety of editable signature schemes has been introduced in literature, their capabilities to assure the integrity and authenticity of published public sector data has not been assessed so far. This renders a concrete implementation of solutions based on editable signatures impossible. To overcome this problem, this paper identifies and discusses legal, organisational, and technical requirements that need to be met by editable signature schemes when applied to public sector data to be published. Afterwards, different existing editable signature schemes are examined and discussed in more detail. Based on the previously identified requirements, the different editable signature schemes are then assessed in detail. The conducted assessment reveals that blank digital signatures, which are a novel approach representing a subset of editable signature schemes, are especially suited to meet the predefined requirements. The results obtained from the conducted survey served as input and basis for the implementation of solutions based on trusted and reliable public sector data.

Keywords: e-government, redactable signatures, editable signatures, blank digital signatures, public sector data

1. Introduction

The public sector produces, collects, processes, and provides large amounts of electronic data. These public sector data can be of interest also for the general public as well as for the corporate sector. In the area of e-Government, two main approaches have evolved to take up the challenge of providing public sector data. The Open Government Data (OGD) initiative bases on the concept of open data and claims that data should be freely available for everyone's use. In addition, the EU Directive on the re-use of public sector information (PSI Directive) defines a legal framework for the provision of public data within the European Union. In June 2013 an amendment of the pre-existing PSI Directive (European Union, 2003) has been published (European Union, 2013). The pre-existing Directive has been published before the emergence of open data. Thus this Directive had a more traditional view on public sector information, which has led to partly different requirements for applications dealing with OGD and PSI related data. This has been consolidated in the updated PSI Directive, which explicitly refers to open (government) data. Nevertheless, security related aspects such as data integrity of authenticity of data are not part of the requirements defined by open data and the updated PSI Directive. To bridge this gap, supplementary security requirements have been defined in literature recently (Stranacher et al., 2013). In this work, the authors have also proposed a concept to meet these additional requirements in practice. The proposed concept employs electronic signatures to allow for the realization of trusted and reliable public sector data. Furthermore, the concept also includes a mechanism to assure the integrity and authenticity of data even if these data need to be redacted. For instance, a redaction can be necessary if the data contain security-sensitive or individual-related information. For such scenarios Stranacher et al. (2013) propose the use of redactable signature schemes, which represent a subset of editable signatures. Editable signatures allow third parties (redactors) to modify signed data without invalidating the original signature. These signature schemes have already proven their usefulness in different fields of application. During the past years, especially the e-Health sector has turned out to be predestinated for an application of editable

[1] EGIZ is a joint initiative of the Austrian Federal Chancellery and the Graz University of Technology

ISSN 1479-439X 360
Reference this paper as: Klaus Stranacher et al "Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data" *Electronic Journal of e-Government* Volume 11 Issue 1 2013, (pp360 - 372), available online at www.ejeg.com

signature schemes (Bauer et al., 2009) (Slamanig and Rass, 2010). So far, several different editable signature schemes have been proposed and discussed in literature. These schemes differ in various fundamental properties, such as the possibility to explicitly define a designated redactor, or to allow the redacting of predefined data blocks only. Unfortunately, current concepts that propose a use of editable signatures in order to assure authenticity and integrity of public sector data lack on an assessment and definition of appropriate editable signature schemes so far.

In this paper we bridge this gap by assessing existing editable signature schemes and evaluating their capabilities to meet the requirements of public sector data. For this purpose, Section 2 gives the legal and technical status quo on (conventional) electronic signatures and editable signature in particular. In Section 3, we recap the concept of trusted and reliable public sector data. Then Section 4 derives concrete requirements that have to be met by editable signature schemes when being applied to the concept of trusted and reliable public sector data. Potential candidates of editable signature schemes are examined in Section 5. In Section 6, we map the derived requirements to the examined editable signature schemes in order to assess them schemes' capabilities to meet the given requirements. Finally, we summarize the findings and outline the ongoing and scheduled research activities.

2. Electronic signatures status quo

Authentication methods are used to assure authenticity and integrity. Basically two main authentication methods – electronic signatures and challenge-response authentication – exist. Whereas latter methods are mainly used in (low level) protocols, electronic signatures are commonly used in various e-Business applications. Especially the e-Government sector uses electronic signatures as a core technology enabling trusted services.

In general, electronic signatures are used to provide a proof of genuineness for electronic data. They basically assure authenticity, data integrity, and non-repudiation of origin. The receiver of a signed document is able to uniquely identify the creator of the signature (authenticity) and is able to verify that the signed data has not been modified (integrity). At the same time, the creator of an electronic signature cannot deny to have signed the data (non-repudiation). Especially the validation of data integrity becomes important for security critical applications. During the past decades, different forms of electronic signatures with different properties and characteristics have been developed. The following sub-sections briefly discusses

2.1 Conventional signatures

Electronic signatures base on public key cryptography. The creator of an electronic signature holds two keys, a private and a public key. The private key is used to create the signature and is under the creator's sole control. The corresponding public key is used by the verifier of the electronic signature to verify the signature's validity.

A typical signature creation process consists of two steps. At first, the data to be signed is mapped to a fixed length hash value. This mapping is done via a so called hash function[2]. Secondly, this hash value is signed using the creator's private key to create the signature. During the verification of the signature it is verified if the received data corresponds to the originally signed data by comparing the received hash value and the hash value computed out of the received data. If these values differ, the data has been modified. If the data has not been modified, the signature itself is verified by means of the creator's public key.

The legal basis for electronic signatures within the European Union is formed by the Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures (European Union, 1999). In particular, the Directive defines three basic types of signatures:

- **Electronic Signature**: Electronic signatures are defined as *"data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication"*.
- **Advanced Electronic Signatures**: The requirements for such a signature are, that the signature is *"uniquely linked to the signatory"*, *"is capable of identifying the signatory"*, *"is created using means that the*

[2] A hash function is a one-way function creating a fixed length data set out of a data set with arbitrary length. Given a hash value, the initial data cannot be determined or re-constructed. The main reason for using a hash function in electronic signature schemes is to reduce the length of the data to be signed, as signing of large data is inefficient and time consuming.

signatory can maintain under his sole control" and *"is linked to the data to which it relates in such a manner that any subsequent change of the data is detectable"*. These requirements are usually fulfilled by conventional signature schemes basing upon a suitable public key infrastructure.

- **Qualified Electronic Signature**: In addition to the requirements for advanced electronic signatures a qualified signature requires to base on a qualified certificate and must be created using a secure signature creation device. The requirements for qualified certificates and secure signature creation devices are also determined in the Signature Directive (Annex I and Annex III). In addition, Article 5 of the Directive defines legal effects of electronic signatures. In particular, it is defined that qualified electronic signatures are legally equivalent to handwritten signatures.

To meet the requirements for advanced electronic signatures, different signature formats have been specified, covering the most wide-spread data formats. These formats are: CAdES[3], XAdES[4] and PAdES[5]. Due to the complexity of these signature formats, which hinders interoperability especially on cross-border level, the European Commission established reference formats for advanced electronic signatures. These reference formats represent appropriate profiles (i.e. subsets) of the mentioned signature formats (European Commission, 2011).

2.2 Editable signatures

Editable signatures provide means to allow (certain) modifications within electronic signatures. Basically editable signatures can be categorized into redactable signatures and blank digital signatures.

2.2.1 Redactable signatures

Redactable signatures have been invented by Johnson et al. (2002) and Steinfeld et al. (2001). In case of conventional signatures, modifications of the signed data are detectable due to an altered hash value. Thus redactable signatures' basic principle bases on retaining the hash value of the original and unmodified data. A main property of these redactable signature schemes is that they only allow blackening out certain message blocks of a signed message. To allow also deletion and replacement of message blocks with other message blocks, Ateniese et al. (2005) introduced the concept of sanitizable signatures, which represent a subset of redactable signatures, but the basic technical concept stays the same.

Figure 1 illustrates this basic principle. First of all, a message m is divided into several message blocks. For illustration, we assume a split into m_1-m_5. For each of these message blocks a hash function H is applied, creating the hash values h_1-h_5. These hash values are concatenated to a total hash value $Hash_{TOTAL}$. Finally, this total hash value is signed to create signature S. At this point we still have created a conventional electronic signature.

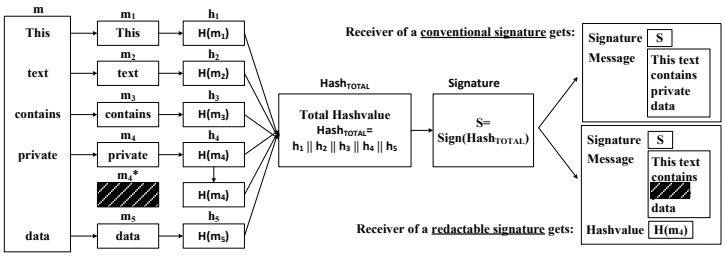


Figure 1: Basic principle of redactable signature schemes

According to the example in Figure 1, the message block "private" (message block m_4*) is redacted. The person, who is allowed to redact message blocks, is usually called redactor. Computing the hash value of the redacted message block will lead to a hash value, which differs from the original hash value and would result in an invalid signature. To avoid this behaviour, the original hash value is retained and used during the signature

[3] CMS Advanced Electronic Signature (ETSI, 2013)
[4] XML Advanced Electronic Signatures (ETSI, 2010a)
[5] PDF Advanced Electronic Signatures (ETSI, 2010b)

verification process[6]. Obviously, the redacted signature must include the original hash value $H(m_4)$. So, the receiver is able to verify the redacted message, but is not able to determine the redacted message block due to the one-way functionality of the hash function. Several redactable signatures schemes do exist, which all base on this basic principle of retaining the original hash values.

2.2.2 Blank digital signatures

Blank digital signatures are a novel scheme invented by Hanser and Slamanig (2013). These signatures have comparable properties to redactable signatures, but the concept behind differs.

Figure 2 illustrates the basic principle of blank digital signatures. An originator defines and signs a message template. This template consists of fixed parts of a message and multiple choices of exchangeable parts. Then a redactor[7] is given the permission to create a message instance. In the instantiation process the redactor selects certain choices of the exchangeable message parts. Finally the redactor signs the message instance. This resulting signature can be publicly verified using the originator's and redactor's public keys. If the verification is positive, it is proven that the message has not been altered as well as the message is compliant to the message template.

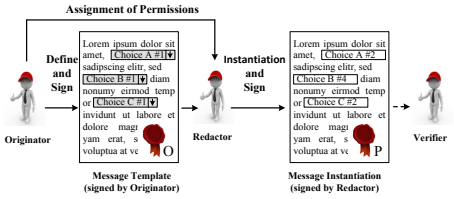


Figure 2: Basic principle of blank digital signature schemes

3. Trusted and reliable public sector data

This section comprises a brief overview of the findings of Stranacher et al. (2013). Since the re-use of public sector information and the open publishing of governmental data do not define new issues, several requirements for such data provisioning techniques have already emerged over the past years. For instance, the Open Government Working Group (2007) has published eight fundamental principles for open government data. While also the (updated) PSI Directive includes some general and common requirements for providing public sector data, security requirements have not been defined.

[6] That means $H(m_4)$ instead of $H(m_4^*)$ is used for calculating $Hash_{TOTAL}$ during signature verification.
[7] The authors use the term proxy for redactor in their proposal.

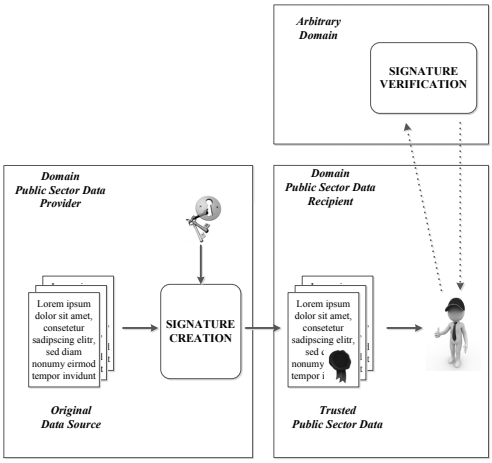


Figure 3: Ensuring authenticity and integrity for public sector data (Stranacher et al., 2013)

Stranacher et al. (2013) define security requirements, namely data integrity and authenticity, when publishing public sector data. Both requirements ensure data consumers that published data have not been altered and are provided by a trustworthy authority. The authors also propose a concept for trusted and reliable public sector data. They distinguish two main use cases. As illustrated in Figure 3, in the first use case public sector data are signed by the data provider before publishing. By using conventional electronic signatures, data integrity and authenticity is ensured.

In the second use case, the public sector data contain personal and private data that need to be anonymized before publishing. Figure 4 illustrates this use case and shows how trusted and reliable anonymization of public sector data without applying a new signature to the modified data is achieved. The original data have been signed by using an editable signature scheme to ensure authenticity and integrity of the entire data set. In case of, these data contain private or personal data, but the remaining data are still useful to publish, the applied editable signature avoids a re-signing process. Avoiding such a re-generation of an electronic signature is useful if the person, who has originally signed the data, is not available anymore for re-signing for some reason.

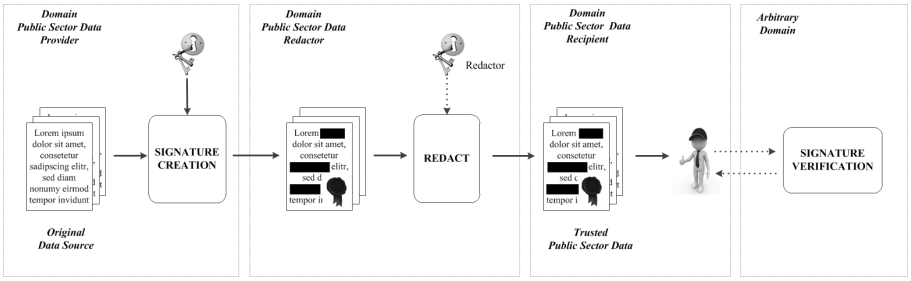


Figure 4: Authenticity and integrity for redacted public sector data (Stranacher et al., 2013)

In the following Section 4 we define concrete requirements for editable signatures applied in this second use case. Additionally we give some more details on different editable signature schemes and their applicability for public sector data in the sections 5 and 6.

4. Requirements for editable signature schemes

The proposed concept of Stranacher et al. (2013) for anonymized public sector data elaborates on the different properties of editable signature schemes, but lacks on defining concrete requirements for editable

signature schemes applied to anonymized public sector data. In order to close this gap, this section defines legal, organisational and technical requirements for editable signature schemes.

4.1 General legal requirements

The EU Signature Directive (European Union, 1999) does not differ between conventional signatures, editable signatures or any other signature type. Therefore the regulations and requirements, defined in the Directive, also applies for editable signatures. Therefore, following general legal requirements are defined:

- **Advanced Electronic Signatures**: An editable signature scheme must satisfy the requirements of an advanced electronic signature as defined by Signature Directive. This is a prerequisite for accountability and to identify the original signer.
- **Qualified Electronic Signature**: These additional requirements are not necessarily needed for the public sector data use cases. An editable signature scheme may, optionally, meet also the requirements for qualified electronic signatures as defined by the Signature Directive.
- **Accountability**: In case of a dispute the signatory must be able to prove that certain modifications have been done by a certain redactor. This is of major importance in case of a dispute, being able to give evidence who has signed or redacted specific data (as legal consequences may arise). Accountability can be achieved by technical means (see also technical requirements below).

4.2 General organisational requirements

Beside legal requirements, there exist also some general requirements on organisational level. These requirements concern mainly the role of the redactors and the signatory, i.e. the party, which holds the public sector data. So, following general organisational requirements are defined:

- **Definition and Revocation of Redactors**: Designated redactors should be easily definable by using existing systems (to avoid additional investments) and the signatory should also have the opportunity to revoke redactors.
- **Non-Disclosure Agreement**: Designated redactors must sign an appropriate confidentiality agreement. In particular regarding the data protection as redactors usually have access to private and personal data, which is governed by data protection regulations.
- **Responsibilities:** Responsibilities must be clearly defined both by the signatory and the redactors (e.g. who is allowed to sign/redact, who is responsible in case of a dispute).
- **Service Level Agreement/Security Compliance**: Redactors must ensure to redact data within an appropriate time frame (especially for real time data). In addition, redactors must be compliant to current security regulations as they operate on private and personal data.

4.3 Technical requirements

On a technical level there exists also some requirements, which are tightly bound the particular editable signature schemes. Therefore, we have defined following technical requirements:

- **Designated Redactors**: Designated redactors must be able to be specified by the editable signature scheme. That means that the signatory must be able to determine who is allowed to modify the signed data. Persons except the signatory and the designated redactors must not be able to redact data without breaking the originally signature applied. Any change of the data by unauthorized persons must be recognizable.
- **Privacy**: The redactable data as well as the original signature must not allow revealing the redacted message blocks.
- **Designated Parts**: The signatory must be able to specify which data blocks may be modified. Editing unauthorized data must be recognized and must lead to an invalid signature.

- **Accountability**: See definition in legal requirements.
- **Applicability**: The scheme must be applicable on open and structured data such as XML (W3C Recommendation, 2008)
- **Compatibility:** The signature scheme must be compatible to (at least one of) the reference signature formats defined in European Commission (2011).

5. Examination

In the following, we examine various editable signature schemes. Figure 5 shows an overview on the most relevant[8] editable signature schemes proposed in the last years and their relation to each other. A main requirement for editable signature schemes to be used in e-Business services is to support the definition of designated redactors. Redactable signature schemes, such as Steinfeld et al. (2001) and Johnson et al. (2002)[9], do not offer the definition of designed redactors. Therefore, these schemes have been skipped from a more in-depth analysis. In contrast, sanitizable signature and blank digital signature schemes allow for more complex definitions of modification options and designated redactors. Thus, the following sub-sections examine selected editable signature schemes only. The selected signature schemes, which are marked grey in Figure 5, have been chosen for examination. In addition, following signature schemes have been skipped from the examination:

- Brzuska et al. (2009) proposed a rigorous security model. This model has been incorporated by Canard and Jambert (2010), which is examined below. Therefore we have skipped it from our analysis.
- Brzuska et al. (2010b) proposed an update of Ateniese (2005) which does not permit creating a link between different signatures over the same original message. This functionality is not of interest for the public sector use cases, so we have skipped this scheme.

5.1 Sanitizable signatures by Ateniese et al. (2005)

The basic principle of sanitizable signatures bases upon commitments[10], which in turn build upon hash-functions. Ateniese et al. (2006) proposed the first scheme for sanitizable signatures, where a designated redactor is able to modify designated parts of a signed message. Here the basic principle bases on chameleon hash-functions instead of conventional hash-functions for conventional signatures. Such chameleon hash-functions are parameterized with the public key of the redactor. Because of the parameterization, the redactor is able to compute collisions. This means the redactor is able to generate messages, which lead to the same hash value as for the data, which is going to be redacted. Based on this mechanism the redactor can replace message blocks with arbitrary message blocks and the verification of the original signature will not fail. In this case it is neither possible to detect if a message has been redacted nor it is possible to detect which message blocks have been modified. Therefore the authors propose to add non-redactable meta information after each redactable message block indicating the restriction for the message to be replaced. Obviously, this is a very inefficient solution.

[8] Relevant in terms of citation rate and author's reputation (mainly based on h-index).
[9] This also applies for Slamanig and Rass (2010), Chang et al. (2009) and Brzuska et al. (2010a), which all base on Johnson et al. [13].
[10] Commitments are often used in cryptographic protocols. They allow a committer to publish a commitment (= a value), which binds the committer to a certain message, but without revealing it. If a verifier wants to check if the message is consistent with the commitment, the committer may open the commitment to reveal the message.

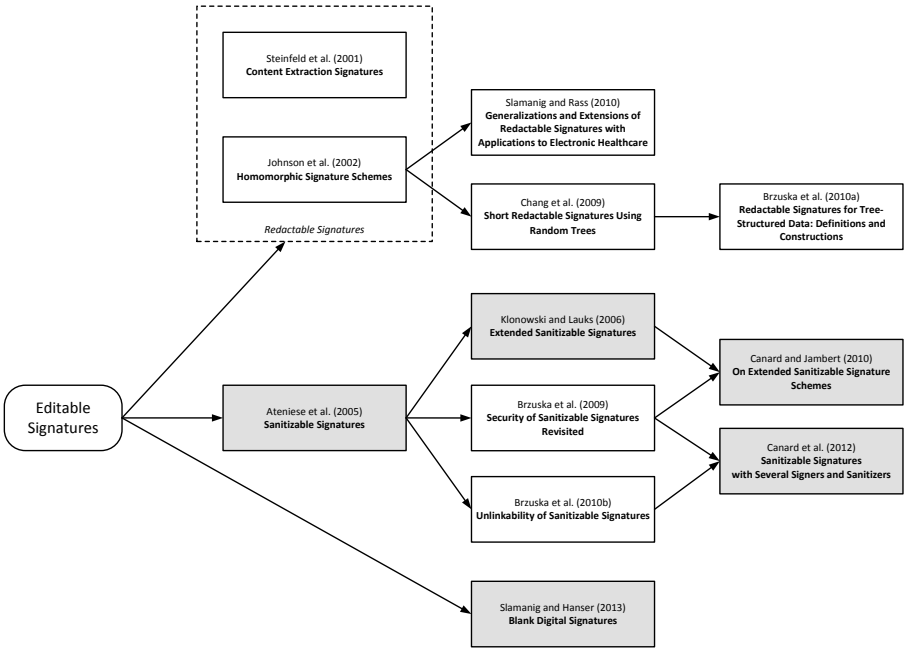


Figure 5: Overview about editable signature schemes

5.2 Extended sanitizable signatures by Klonowski and Lauks (2006)

Klonowski and Lauks (2006) extended the scheme of Ateniese et al (2005). They omitted the added meta information and extended the schema itself to allow the signatory to limit the message blocks which are modifiable by the redactor and to limit the messages which are replaced. This scheme also bases on chameleon hash-functions. For the message replacement restrictions they propose to use accumulators[11] or bloom filters[12].

5.3 On extended sanitizable signature schemes by Canard and Jambert (2010)

Canard and Jambert (2010) presented a second approach to limit the modification of message blocks and the message to be replaced by the scheme itself. As for the other sanitizable signature schemes, the authors base their proposal on chameleon hash-functions. In addition, they use pseudorandom generators and accumulators to implement the message replacement restrictions.

5.4 Sanitizable signatures with several signers and sanitizers by Canard et al. (2012)

Canard et al. (2012) builds upon the findings of Brzuska et al. (2009) and Brzuska et al. (2010b). The proposed scheme allows defining multiple signers and multiple redactors. To support multiple signers and redactors, the authors make use of group signatures[13]. Their scheme also provides group anonymity. That means a signer (resp. redactor) is anonymous for other entities, which are not in the group of signers (resp. redactors).

[11] An accumulator is a one-way hash function which satisfies a quasi-commutative property. See Benaloh and Mare (1994) for details.
[12] Bloom filters are data structures which allow to efficient test whether an element is a member of a certain set or not. See Bloom (1970) for details.
[13] Group signatures give a group of signers signing rights.

5.5 Blank digital signatures by Slamanig and Hanser (2013)

Blank digital signatures, proposed by Hanser and Slamanig (2013), are a new signature scheme, which makes use of elliptic curve pairings[14] and polynomial commitments[15]. In contrast to redactable signatures, blank digital signatures make use of conventional signatures for signing the message template and the message instance. For the definition of the message template polynomials are used. The message instantiation bases upon polynomial commitments. Finally, for the verification of the polynomial commitments pairings are used.
In addition, the authors have published an updated version of this scheme[16]. This update includes a simplified construction of the signatures allowing significantly performance enhancements. Finally, this update incorporates full security proofs.

6. Assessment

6.1 Legal assessment

In this section, we assess editable signature schemes based on legal and organisational requirements. Concerning the legal assessment, the EU Signature Directive defines the legal framework. While this directive primarily considers conventional electronic signatures, the use of sanitizable signatures compliant with this directive has been slightly discussed by Höhne et al. (2012) and Brzuska et al. (2012). The authors examined legal consequences of sanitizable signatures. They especially argue that sanitizable signatures are compliant to advanced electronic signatures but cannot be used for qualified electronic signatures according to the EU Signature Directive. The reason for being not compliant with qualified electronic signatures constitutes missing displaying possibilities for the signatory. According to the Signature Directive, the data to be signed must be viewable by the signatory before the signature creation process. This requirement cannot be fulfilled by sanitizable signatures as modifications of signed data are possible also after signature creation, which the signatory cannot be aware of at the time of the signature creation process regardless the signatory is able to define which message parts are able to be modified and how they can be modified.

Legal considerations for blank digital signatures do not exist yet. Following the argumentation of Höhne et al. (2012) and Brzuska et al. (2012), blank digital signatures are compliant to advanced electronic signatures. The reason for that is mainly based upon the use of public key cryptography. In contrast to sanitizable signatures, blank digital signatures are considered to be compliant with requirements defined for qualified signatures. The reason for being compliant is based upon the usage of conventional signatures for the message template and the message instance signature.
Another legal requirement to be fulfilled by the proposed signature schemes is accountability. Accountability means that redactors, who used her private keys to modify signed data, can be determined. This requirement cannot be met by all described signature schemes (see following Section 6.2).

6.2 Organisational assessment

Equal to legal requirements, several organisational requirements must be met by the proposed signature schemes in order to successfully apply editable signatures to public sector or open government data. In fact, all organisational requirements identified in Section 3.2 are independent of the technical implementation of the proposed signature schemes. While some organisational requirements may be fulfilled using technical means, others require solutions on organisational level. For instance, the requirement on revoking designated redactors can be fulfilled on technical level as all of the proposed schemes rely on a public key infrastructure (PKI) and hence on existing and well-established revocation mechanisms. However, other organisational requirements still require organisational measures. This particularly means that a fulfilment of those requirements requires e.g. some kind of contractual agreements between all involved parties. Within such agreements, especially individual responsibilities, signature validity limitations, or liability questions must be thoroughly elaborated.

[14] Pairings are bilinear mappings as defined by Silverman (1986).
[15] Conventional commitments applied to polynomial functions are called polynomial commitments (see Kate et al. (2010) for details).
[16] https://online.tugraz.at/tug_online/voe_main2.getvolltext?pCurrPk=69904

6.3 Technical assessment

The technical assessment concerning applicability to structured data and the signature format compliance to the European Commission Decision 2011/130/EU can be done for all examined schemes together. Pöhls et al. (2011) have implemented several editable signature schemes based upon XML and the W3C Recommendation on XML signatures (W3C Recommendation, 2008). Hence, they have proven that editable signatures are applicable to structured data, such as XML. Nevertheless, implementations of editable signature schemes fulfilling the requirements for the advanced electronic signatures format XAdES, CAdES or PAdES do not yet exist.
The following sub-sections comprise the further technical assessment of the different editable signature schemes.

6.3.1 Assessment of sanitizable signatures by Ateniese et al. (2005)

Ateniese et al. (2005) states *"[…] as a secure digital signature scheme that allows a semi-trusted censor to modify certain designated portions of the message […]"*[17]. That means the requirement for designated redactor and designated parts is fulfilled. In addition the privacy is also fulfilled as *"[…] the indistinguishhability requirement provides for privacy"*. The author also state that *"accountability follows from the unforgeability requirement"*, but this has been proven by Brzuska et al. (2009) as not true. So the Ateniese sanitizable signature scheme does not provide accountability.

6.3.2 Assessment of extended sanitizable signatures by Klonowski and Lauks (2006)

The extended sanitizable signature scheme of Klonowski and Lauks (2006) provides a designated redactor and designated parts as stated by the authors: *"[…] in this scheme the designated censor can change the content of designated (so called mutable) parts of a signed message […]"*. They also state that privacy is fulfilled due to the basement of their extended scheme on Ateniese et al. (2005). Concerning accountability we have to distinguish between the two characteristics of this scheme. The accumulator technique provides accountability whereas bloom filter does not. Nevertheless, the authors miss a concrete security model and proofs for their proposed schema. This implies an unpredictable security risk, which disqualifies this scheme.

6.3.3 Assessment of extended sanitizable signature schemes by Canard and Jambert (2010)

As this scheme strongly bases on Ateniese et al. (2005), it provides designated redactors as needed by our defined requirements. In addition, Canard and Jambert (2010) state that *"[…] to force some admissible blocks of a signed message to be modified only into a predefined set of sub-messages."*[18] and *"[…] privacy is also included by transparency in the extended model."*. Thus, the scheme fulfils the requirements for designated parts and privacy. In addition, the authors prove that *"Unforgeability (and thus accountability) is reached thanks to the computation of a new tag per message."*. This is one of the major extensions of Ateniese et al. (2005).

6.3.4 Assessment of sanitizable signatures with several signers and sanitizers by Canard et al. (2012)

The scheme of Canard et al. (2012) supports the definition of designated redactors as the authors state that *"[…] a model where one signer (among n) can choose a set of sanitizers (among m)"*. Furthermore the scheme also provides to define designated blocks due to *"Given a message m of length l and divided into t blocks […], which will be modifiable by the sanitizer"*. As this scheme strongly bases on Brzuska et al. (2009) and Brzuska et al. (2010b), the requirement privacy is supported as well. Finally the authors also proofs that their scheme is accountable.

6.3.5 Assessment of blank digital signatures by Slamanig and Hanser (2013)

The proposed template mechanism by Hanser and Slamanig (2013) fulfils the requirement for designated parts, as the originator defines the message template, i.e. only the exchangeable parts, defined by the originator, are modifiable. In addition, the designated redactor requirement is fulfilled as *"Immutability guarantees that no malicious proxy can compute message templates or templates instantiations not intended*

[17] They used the name censor for the redactor.
[18] Message parts which can be modified by a redactor are often called admissible blocks.

by the signer". They even prove that their scheme supports the privacy requirement. Finally, the scheme fulfils the accountability requirement, as the redactor signs the message template instance with a conventional signature (which provides accountability in any case).

6.4 Assessment summary

Table 1 summarizes the results of the legal and technical assessment. It shows that Ateniese et al. (2005) lacks on the requirement on accountability. Furthermore Klonowski and Lauks (2006) miss a security model and proofs for the proposed scheme. Therefore these two schemes are assessed to be not suitable for the public sector data use cases.

In contrast, the sanitizable signature schemes of Canard and Jambert (2010) and Canard et al. (2012) as well as blank digital signatures of Slamanig and Hanser (2013) meet all technical requirements. Hence these schemes are appropriate to the use cases of redacted public sector data as defined in Stranacher et al. (2013). In addition, blank digital signatures fulfil the nice-to-have requirement on qualified electronic signature.

Nevertheless, obstacles hindering an application of these schemes in public sector data applications exist. Concrete implementations for these signature schemes do not exist yet or are not compliant to the standard advanced signature formats defined by the European Commission Decision 2011/130/EU.

Table 1: Assessment summary (legal and technical) of examined editable signature schemes

Signature Scheme	Legal Requirements			Technical Requirements				
	Account-ability	***Advanced Signature***	***Qualified Signature***	***Designated Redactor***	***Designated Parts***	***Privacy***	***Applicab. Structured Data***	***Compliance 2011/130/ EU***
Ateniese et al. (2005)	No	Yes	No	Yes	Yes	Yes	Yes	Partly
Canard and Jambert (2010)	Yes[19]	Yes	No	Yes	Yes	Yes	Yes	Partly
Canard et al. (2012)	Yes	Yes	No	Yes	Yes	Yes	Yes	Partly
Klonowski and Lauks (2006)	Yes	Yes	No	Yes	Yes	Yes	Yes	Partly
Slamanig and Hanser (2013)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Partly

7. Conclusions

The emerging trend to make public sector data available to the general public and to the corporate sector raises the demand for innovative techniques to meet arising security requirements. Electronic signatures in general and editable electronic signature schemes in particular have recently been proposed as adequate enabler for such security preserving techniques.

In this paper we have made the next step towards a concrete implementation of these techniques by evaluating different proposed schemes for editable signatures and by assessing their capabilities to enhance the security of publishing (anonymized) public sector data. The assessment has been based on a set of legal, organisational, and technical requirements, which have previously been defined and discussed. The conducted assessment of existing editable signature schemes has revealed that especially blank digital signatures by Slamanig and Hanser (2013) are well suited to enhance the security of published public sector data.

[19] This scheme supports accountability only for the version where accumulators are used. In case the bloom filter is used accountability is no achievable.

The results that have been obtained from the conducted assessment pave the way for several future activities in this field. The blank digital signature scheme that has been identified by the conducted assessment has been implemented on a prototype basis and allows for creation of XAdES-based signatures. Currently, this implementation serves as basis for the development of solutions based on trusted and reliable public sector data.

References

Ateniese, G., Chou, D. H., de Medeiros, B., Tsudik, G. (2005), *Sanitizable Signatures*, in European Symposium on Research in Computer Security ESORICS 2005, LNSC, vol. 3679, pp. 159-177, Springer.

Bauer, D., Blough, D., Mohan, A. (2009), *Redactable Signatures on Data with Dependencies and their Application to Personal Health Records*. In: Proc. of the 8th ACM Workshop on Privacy in the Electronic Society, WPES '09, pp. 91–100. ACM Press, New York

Benaloh, J., Mare, M., (1994), *One-Way Accumulators: A Decentralized Alternative to Digital Signatures*, in Advances in Cryptology — EUROCRYPT 1993, LNCS, vol. 765, pp. 274-285, Springer.

Bloom, B. (1970), *Space/time trade-offs in hash coding with allowable errors*, in Communication of ACM, vol. 13, no. 7, pp. 422-426

Brzuska, C., Fischlin, M., Freudenreich, T., Lehmann, A., Page, M., Schelbert, J., Schröder, D., Volk, F. (2009), *Security of sanitizable signatures revisited*, in Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 317–336. Springer.

Brzuska, C., Busch, H., et al. (2010a), *Redactable Signatures for Tree-Structured Data: Definitions and Constructions*, in Applied Cryptography and Network Security 2010, LNCS, vol. 6123, pp. 87-104, Springer.

Brzuska, C., Fischlin, M., Lehmann, A., Schröder, D. (2010b), *Unlinkability of Sanitizable Signatures*, in Public Key Cryptography – PKC 2010, LNCS, vol. 6056, pp. 444-461, Springer

Brzuska, C. Pöhls, H., Samelin, K. (2012), Non-Interactive Public Accountability for Sanitizable Signatures, in Proceedings of the 9th European PKI Workshop: Research and Applications (EuroPKI 2012), Springer, Note: to appear.

Canard, S., Jambert, A. (2010), *On Extended Sanitizable Signature Schemes*, in Topics in Cryptology - CT-RSA2010, LNCS, vol. 5985, pp. 179-194, Springer.

Canard, S., Jambert, A., Lescuyer, R. (2012), Sanitizable Signatures with Several Signers and Sanitizers, AFRICACRYPT'12 Proceedings of the 5th international conference on Cryptology in Africa, pp. 35-52, Springer.

Chang, E., Lim, C., Xu, J. (2009), *Short Redactable Signatures Using Random Trees*, in Topics in Cryptology – CT-RSA 2009, LNCS, vol. 5473, pp. 133-147, Springer.

ETSI (2010a), *Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)*, V1.4.2.

ETSI (2010b), *Electronic Signatures and Infrastructures (ESI);PDF Advanced Electronic Signature Profiles; Part 3: PAdES Enhanced - PAdES-BES and PAdES-EPES Profiles*, V1.2.1.

ETSI (2013), *Electronic Signatures and Infrastructures (ESI);CMS Advanced Electronic Signatures (CAdES)*, V2.2.1.

European Commission (2011), European Commission Decision, *Establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market*, notified under document C(2011) 1081, 2011/130/EU, 25.02.2011.

European Union (1999), *Directive 1999/93/EC on a Community framework for electronic signatures*.

European Union (2003), *Directive 2003/98/EC of the European Parliament and the Council of 17 November 2003 on the re-use of public sector information*

European Union (2013), *Directive 2013/98/EC of the European Parliament and the Council of 26 June 2013 on the re-use of public sector information*

Höhne, F., Pöhls, H., Samelin, K. (2012), Rechtsfolgen editierbarer Signaturen, in Datenschutz und Datenrecht (DuD), vol. 36(6), pp. 485-491

Johnson, R., Molnar, D., Song, D. X., Wagner, D. (2002), *Homomorphic Signature Schemes*, in Topics in Cryptology CT-RSA 2002, LNCS 2271, pp. 244-262, Springer.

Kate, A., Zaverucha, G. M., Goldberg I. (2010), *Constant-size commitments to polynomials and their applications*. In Advances in Cryptology - ASIACRYPT 2010, pp. 177-194, 2010

Klonowski, M., Lauks, A. (2006), *Extended sanitizable signatures*, in: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 343–355. Springer.

Open Government Working Group (2007), *8 Principles of Open Government Data*, http://www.opengovdata.org/home/8principles.

Pöhls, H., Samelin, K., Posegga, J. (2011), *Sanitizable Signatures in XML Signature — Performance, Mixing Properties, and Revisiting the Property of Transparency*, in Applied Cryptography and Network Security, LNCS, vol. 6715, pp. 166-182, Springer.

Silverman, J. (1986),*The Arithmetic of Elliptic Curves*, volume 106 of Graduate Texts in Mathematics, 1986, Springer.

Slamanig, D., Hanser, C. (2013), *Blank Digital Signatures*, in Proceedings of 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2013), pp. 95-106, ACM.

Slamanig D., Rass, S. (2010), *Generalizations and Extensions of Redactable Signatures with Applications to Electronic Healthcare*, in Communications and Multimedia Security 2010, LNCS, vol. 6109, pp. 201-213. Springer.

Steinfeld R., Bull, L., Zheng, Y, (2001), *Content Extraction Signatures*, in Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 285–304. Springer.

Stranacher, K., Krnjic, V., Zefferer, T. (2012), Vertrauenswürdiges Open Government Data, in 1.OGD D-A-CH-LI Konferenz, pp. 27-39.

Stranacher, K., Krnjic, V., Zefferer, T. (2013), Trust and Reliability for Public Sector Data, Proceedings of International Conference on e-Business and e-Government, pp. 124-132.

W3C Recommendation (2008), *XML-Signature Syntax and Processing (Second Edition)*, http://www.w3.org/TR/xmldsig-core/

Yuen, T., Susilo, W., Liu, J., Mu, Y. (2008), *Sanitizable Signatures Revisited*, in Cryptology and Network Security, LNCS, vol. 5339, pp. 80-97, Springer.

14 Vertrauenswürdiges Open Government Data - Authentizität und Integrität für öffentliche Verwaltungsdaten

Conference	OGD D-A-CH-LI
Language	German
Title	Vertrauenswürdiges Open Government Data - Authentizität und Integrität für öffentliche Verwaltungsdaten
Authors	Klaus Stranacher, Vesna Krnjic, Thomas Zefferer
Book	1.OGD D-A-CH-LI Konferenz

Vertrauenswürdiges Open Government Data

Authentizität und Integrität für öffentliche Verwaltungsdaten

Klaus Stranacher*, Vesna Krnjic**, Thomas Zefferer***

*Inffeldgasse 16a, A-8010 Graz, E-Government Innovationszentrum (EGIZ), klaus.stranacher@egiz.gv.at

** Inffeldgasse 16a, A-8010 Graz, E-Government Innovationszentrum (EGIZ), vesna.krnjic@egiz.gv.at

*** Inffeldgasse 16a, A-8010 Graz, Zentrum für sichere Informationstechnologie – Austria (A-SIT), thomas.zefferer@a-sit.at

Kurzzusammenfassung: *Open Government Data hat sich in den letzten Jahren zu einem wichtigen Trend in verschiedenen Bereichen des E-Government entwickelt. Rahmenbedingungen und Richtlinien für die öffentliche Bereitstellung von Datenbeständen durch Behörden werden dabei zumeist durch allgemein anerkannte Anforderungslisten definiert. Überraschenderweise enthalten diese Listen kaum Anforderungen zur Sicherstellung der Vertrauenswürdigkeit öffentlich bereitgestellter Daten. So haben Bezieherinnen und Bezieher von Open Government Data in der Regel keine Möglichkeit, die Integrität und Authentizität der erhaltenen Daten zu verifizieren. Open Government Data kann daher in der Regel nur bedingt als vertrauenswürdig angesehen werden.*

Um eine Steigerung der Vertrauenswürdigkeit zu erreichen, präsentieren wir in diesem Beitrag ein Konzept zur Integration kryptographischer Methoden in Open Government Data. Durch die Verwendung elektronischer Signaturen kann die Integrität öffentlich bereitgestellter Daten sichergestellt und die Authentizität dieser Daten gewährleistet werden. Wir zeigen außerdem, wie durch die Verwendung editierbarer elektronischer Signaturen die Integrität und Authentizität von Open Government Data auch bei einer zum Schutz privater Daten durchgeführten Anonymisierung weiterhin gewährleistet werden kann. Damit sichert das vorgestellte Konzept die Vertrauenswürdigkeit von Open Government Data und ebnet so den Weg für eine sichere zukünftige Verwendung.

Schlüsselwörter: Vertrauenswürdiges Open Government Data, Integrität, Authentizität, Editierbare Signaturen

1. Einleitung

„Data is the new gold! In short, ladies and gentlemen, my message today is that data is gold. We have a huge goldmine in public administration. Let's start mining it."

(Neelie Kroes,

Vizepräsidentin der Europäischen Kommission, 2011)

Dieses Zitat von Neelie Kroes, Vizepräsidentin der Europäischen Kommission, drückt sehr treffend die aktuelle Erwartungshaltung aus, die den Konzepten *Open Data* bzw. *Open Government Data* entgegengebracht wird. Das zugrundeliegende Konzept von Open Data selbst ist dabei denkbar einfach und beschreibt im Wesentlichen *„Datenbestände, die im Interesse der*

Allgemeinheit der Gesellschaft ohne jedwede Einschränkung zur freien Nutzung, zur Weiterverbreitung und zur freien Weiterverwendung frei zugänglich gemacht werden" [14]. Open Government Data (OGD) kann als Teilbereich von Open Data verstanden werden und bezieht sich auf jene *„Datenbestände des öffentlichen Sektors, die von Staat und Verwaltung im Interesse der Allgemeinheit ohne jedwede Einschränkung zur freien Nutzung, zur Weiterverbreitung und zur freien Weiterverwendung frei zugänglich gemacht werden"* [14].

Open Government Data ist noch eine relativ neue Initiative, die erst in den letzten Jahren Berücksichtigung von offizieller Seite erfahren durfte. Auf europäischer Ebene wurde die OGD erstmals in der Mitteilung der Europäischen Kommission vom 12. Dezember 2011 aufgegriffen, in der die Relevanz von Open Data für Europa erörtert, Herausforderungen und Chancen dieses Konzepts identifiziert und entsprechende Maßnahmen auf europäischer Ebene erarbeitet wurden [10]. Ergänzend dazu ist für die OGD auch die Richtlinie 2003/98/EG des Europäischen Parlaments und des Rates vom 17. November 2003 über die Weiterverwendung von Informationen des öffentlichen Sektors [11] zu erwähnen, deren Ziel ebenfalls ein Abbau von Barrieren bei der Entwicklung von Informationsprodukten und -diensten anhand von Informationen des öffentlichen Sektors ist, die dabei jedoch die Konzepte Open Data und Open Government Data nicht explizit erwähnt. Basierend auf diesen europäischen Richtlinien entwickelten sich in den letzten Jahren zahlreiche – auch nationale – Initiativen, deren Ziel eine verstärkte Berücksichtigung des OGD-Konzepts ist (für Österreich siehe beispielsweise [12] . In Anbetracht dieses gesteigerten Interesses und der umfangreichen Datenmengen, über die Behörden und verwandte öffentliche Einrichtungen verfügen und die im Zuge der verstärkten Verfolgung des OGD-Konzepts freigegeben werden könnten, ist absehbar, dass sich Open Data bzw. Open Government Data zu einem wichtigen zukünftigen Trend im Bereich E-Government entwickeln wird. Die geeignete Aufbereitung und Bereitstellung verfügbarer Daten für eine öffentliche Verwendung wird eine der Herausforderungen sein, denen sich Behörden in absehbarer Zukunft stellen werden müssen.

Der geeignete Schutz von Daten spielt in Anwendungen des E-Government seit jeher eine zentrale Rolle. Zum Schutz von privaten und sicherheitskritischen Daten kommen daher bei der Umsetzung und Durchführung von E-Government-Transaktionen in der Regel erprobte kryptographische Methoden zur Anwendung. In klassischen E-Government-Anwendungen dienen diese Methoden hauptsächlich der sicheren Authentifizierung von Bürgerinnen und Bürgern, sowie dem Schutz zu übertragender Daten. In diesem Zusammenhang hat sich die Verwendung elektronischer Signaturen bewährt. Über elektronische Signaturkonzepte kann die Integrität von Daten – d.h. der Schutz vor unerlaubter Veränderung – sichergestellt werden. Darüber hinaus kann über elektronische Signaturen – ähnlich zur handschriftlichen Unterschrift – Nichtabstreitbarkeit erreicht werden. Eine Unterzeichnerin bzw. ein Unterzeichner hat keine Möglichkeit, die Kenntnis eines von ihr bzw. ihm elektronisch unterzeichneten Datensatzes abzustreiten. Damit kann auch die Authentizität dieser Daten sichergestellt werden. Der Relevanz elektronischer Signaturen für das E-Government wurde bereits im Jahr 2000 durch Veröffentlichung der Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen [13] Rechnung getragen. Gemäß dieser Richtlinie sind qualifizierte elektronische Signaturen – das sind elektronische Signaturen, die definierten Sicherheitsanforderungen genügen müssen – handschriftlichen Unterschriften rechtlich gleichgestellt.

Da elektronische Signaturen einen integralen Bestandteil vieler bestehender E-Government Anwendungen darstellen, liegt der Gedanke nahe, diese auch auf das im Bereich E-Government

noch relativ neue Konzept des Open Government Data anzuwenden. Dadurch können OGD-Lösungen auf ein höheres Sicherheitsniveau gehoben und damit das gebotene Service an Bürgerinnen und Bürgern insgesamt verbessert werden. In diesem Beitrag stellen wir einen Ansatz zur Erweiterung des Konzepts Open Government Data um elektronische Signaturen vor. Durch Verwendung einer speziellen Klasse elektronischer Signaturen gewährleistet dieser Ansatz die Authentizität und Integrität von öffentlich zur Verfügung gestellten Daten und ermöglicht gleichzeitig deren teilweise Anonymisierung. Damit trägt der in diesem Beitrag vorgestellte Ansatz dazu bei, die Vertrauenswürdigkeit von OGD zu erhöhen.

Dieser Beitrag ist wie folgt strukturiert. Abschnitt 2 fasst allgemeine Anforderungen an OGD zusammen und erweitert diese um Anforderungen in Bezug auf Vertrauenswürdigkeit. Die kryptographischen Konzepte elektronischer Signaturen und editierbarer elektronischer Signaturen, welche zur Erfüllung dieser Anforderungen herangezogen werden, werden in Abschnitt 3 erläutert. Abschnitt 4 stellt schließlich ein konkretes Konzept zur Integration (editierbarer) elektronischer Signaturen in OGD-Anwendungen vor. Abschnitt 5 fasst abschließend die Kernaussagen dieses Beitrags zusammen und gibt einen Ausblick auf mögliche zukünftige Entwicklungen in diesem Bereich.

2. Anforderungen an Open Government Data

Für OGD-basierte Anwendungen existiert eine Reihe von Anforderungen, die bei der Bereitstellung von Daten für die Öffentlichkeit berücksichtigt werden sollten. Ende 2007 veröffentlichte die Open Government Arbeitsgruppe [9] acht Prinzipien, die bei der Veröffentlichung von Open Government Data soweit wie möglich eingehalten werden sollten. Folgende Aspekte wurden dabei berücksichtigt:

1. **Vollständigkeit**: Alle öffentlichen Daten sollen so vollständig wie möglich der Öffentlichkeit zugänglich gemacht werden, sofern dies den Datenschutz nicht verletzt.
2. **Verwendung von Primärquellen**: Die Daten sollen an ihrem Ursprung mit dem höchstmöglichen Freiheitsgrad und in nicht aggregierten oder modifizierenden Formaten gesammelt werden.
3. **Aktualität**: Daten sollen nach ihrer Entstehung zum schnellstmöglichen Zeitpunkt der Öffentlichkeit zugänglich gemacht werden. Werden Daten bereitgestellt, deren Nutzen zeitabhängig ist, sollte die Veröffentlichung dieser Daten priorisiert werden. Der Nutzwert für die Öffentlichkeit kann durch eine Echtzeitaktualisierung der Daten erhöht werden.
4. **Uneingeschränkter Zugang:** Veröffentlichte Daten sollen sowohl infrastrukturell als auch elektronisch so einfach wie möglich und barrierefrei zugänglich gemacht werden. Die Notwendigkeit eines physischen Zugangs zu Daten (z.B. der Besuch spezieller Räumlichkeiten) ist ebenso zu vermeiden wie der Einsatz spezieller elektronischer Zugangstechnologien.
5. **Maschinenlesbarkeit:** Daten sollen automatisch weiterverarbeitet werden können, um eine einfache Einbindungen in Softwareanwendungen zu gewährleisten. Daten sollen in offenen und weitverbreiteten Dateiformaten abgespeichert werden. Falls Daten normalisiert wurden, soll ausreichende Dokumentation über das verwendete Dateiformat zur Verfügung gestellt werden. Ebenso sollen die Rohdaten bereitgestellt werden, welche automatisch maschinell bezogen werden können.

6. **Nichtdiskriminierender Zugang:** Ein anonymer Zugang zu Daten soll für jede Person jeder Zeit möglich sein. Der Zugriff auf die Daten soll nicht auf bestimmte Organisationen oder Personenkreise eingeschränkt werden. Darüber hinaus soll der Zwang zur Nutzung bestimmter Softwareapplikationen nicht gegeben sein.
7. **Nichtproprietäre Datenformate:** Durch die Verwendung von offenen Standards soll gewährleistet werden, dass Daten nicht nur durch ausgewählte Softwarekomponenten gelesen und verarbeitet werden können. Meistens wird es dazu notwendig sein, die Daten in unterschiedlichen Formaten zur Verfügung zu stellen.
8. **Lizenzfreiheit:** Daten sollen entgeltlos zur Verfügung gestellt werden. Die Einhebung von Gebühren würde andernfalls die Nutzergruppe einschränken.

Hauptaugenmerk bei der Definition dieser Grundsätze wurde augenscheinlich auf Vollständigkeit, Aktualität und einfache Zugriffsmöglichkeiten gelegt. Sicherheitsaspekte wurden – mit Ausnahme der in Punkt 1 angeführten Einschränkung der Vollständigkeit um personenbezogene Daten – nicht berücksichtigt. Derartige Aspekte finden auch in adaptierten Versionen der Anforderungsliste, wie beispielsweise der in [15] angeführten Aufzählung, kaum Erwähnung. Abhängig vom jeweiligen Anwendungsszenario kann jedoch die Berücksichtigung entsprechender Sicherheitsanforderungen zielführend bei der Veröffentlichung von OGD sein. Aus diesem Grund betrachten wir die bisher übliche Definition von Anforderungen an Open Government Data für gewisse Szenarien als unvollständig und erweitern diese um folgende Punkte.

A. **Integrität und Authentizität:** Die Integrität und Authentizität der veröffentlichten Daten soll durch die Verwendung entsprechender technischer Verfahren gewährleistet werden. Dadurch sollen Bezieherinnen und Bezieher dieser Daten diese jederzeit auf unerlaubte Veränderung prüfen (Integrität) und darüber hinaus die Bereitstellerin bzw. den Bereitsteller der Daten zweifelsfrei feststellen können (Authentizität). Für die Bereitstellerin bzw. den Bereitsteller der Daten ergibt sich der Vorteil, dass Bezieherinnen und Bezieher auf diese Weise nicht behaupten können, falsche Daten erhalten zu haben.
B. **Anonymisierung**: Wie in Punkt 1 der allgemeinen Anforderungen an OGD definiert, dürfen personenbezogene Daten nicht als Open Data veröffentlicht werden, da dies den Datenschutz untergraben würde. Oft können mit diesen personenbezogenen Daten verknüpfte allgemeine Daten für die Öffentlichkeit dennoch von Interesse und Nutzen sein. Derartige Daten sollen daher geeignet anonymisiert und in weiterer Folge datenschutzrechtlich unbedenklich veröffentlicht werden können. Diese Anforderung darf dabei nicht im Konflikt der Forderung nach Integrität und Authentizität treten. Die Integrität und Authentizität anonymisierter Daten soll in jedem Fall weiterhin gewährleistet bleiben.

Die diskutierte und aus sicherheitstechnischer Sicht sinnvolle Erweiterung der Anforderungen an Open Government Data stellt Behörden, die als Anbieter von OGD fungieren, vor neue Herausforderungen. Eine Berücksichtigung dieser Erweiterungen macht eine Integration elektronischer Signaturkonzepte notwendig. Im folgenden Abschnitt werden daher zunächst jene kryptographischen Konzepte vorgestellt und diskutiert, die eine Berücksichtigung der definierten erweiterten Anforderungen ermöglichen. Konkrete Konzepte zur Umsetzung entsprechender Verfahren zur Berücksichtigung der erweiterten Anforderungen werden schließlich in Abschnitt 4 diskutiert.

3. Signaturkonzepte

Grundsätzlich dienen elektronische Signaturen dem Echtheitsnachweis von elektronischen Dokumenten. Damit repräsentieren elektronische Signaturen das Pendant zur handschriftlichen Unterschrift auf Papierdokumenten. Elektronische Signaturen ermöglichen der Empfängerin bzw. dem Empfänger eines signierten Dokuments die eindeutige Identifikation der Erstellerin bzw. des Erstellers der Signatur (Authentizität) und den Nachweis, dass die signierten Daten nicht verändert wurden (Integrität). Speziell die Überprüfung der Integrität spielt in sicherheitskritischen Anwendungen oft eine zentrale Rolle. So kann bei Verwendung elektronischer Signaturlösungen beispielsweise der Inhalt eines signierten Vertrages nicht einseitig verändert werden, ohne dass die elektronische Signatur über den Inhalt des Vertrags ungültig wird. Durch diese Eigenschaften scheinen elektronische Signaturen für die Umsetzung entsprechender Verfahren zur Gewährleistung der Integrität und Authentizität von OGD geeignet zu sein.

Die technische Basis für elektronische Signaturen bilden Public-Key-Verfahren. Bei diesen Verfahren besitzt die Erstellerin bzw. der Ersteller einer Signatur (Unterzeichnerin bzw. Unterzeichner) sowohl einen privaten als auch einen öffentlichen Schlüssel. Der private Schlüssel ist im alleinigen Besitz der Erstellerin bzw. des Erstellers, welche sie bzw. er zur Erzeugung der Signatur verwenden. Dabei werden die zu signierenden Daten zuerst mittels einer so genannten Hash-Funktion[2] auf einen Wert fixer Länge – den sogenannten Hash-Wert – abgebildet[3]. Der so ermittelte Hash-Wert wird anschließend von der Unterzeichnerin bzw. dem Unterzeichner mit dem privaten Schlüssel signiert. Der dazugehörige öffentliche Schlüssel wird veröffentlicht[4] und dient Empfängerinnen und Empfängern der signierten Daten zur Verifikation der Gültigkeit der elektronischen Signatur.

Bei konventionellen Signaturverfahren führt jede Änderung der signierten Daten unweigerlich zu einer ungültigen Signatur. Durch die geänderten Daten ergibt sich im Zuge der Verifikation ein im Vergleich zur Signaturerstellung unterschiedlicher Hash-Wert. Die Signatur kann über diesen geänderten Hash-Wert nicht mehr positiv verifiziert werden. Die Empfängerin und der Empfänger der signierten Daten können daher die Modifikation der Daten im Zuge der Signaturverifikation eindeutig feststellen.

Es existieren jedoch Anwendungsfälle, in denen ein nachträgliches Ändern der Signaturdaten sehr wohl ermöglicht werden soll, ohne dass die aufgebrachte Signatur ihre Gültigkeit verliert. Ein solcher Anwendungsfall ist beispielsweise das Schwärzen bestimmter Textstellen, wie dies etwa bei der Anonymisierung von OGD zur Anwendung kommt. Das kryptographische Konzept der editierbaren Signaturen ermöglicht eine nachträgliche Änderung an Signaturdaten, ohne dass eine bestehende Signatur dadurch ungültig wird. Als *Redigiererin* bzw. *Redigierer* wird in diesem Zusammenhang jene Person bezeichnet, die in der Lage ist, Veränderungen der Daten vorzunehmen, ohne bestehende Signaturen ungültig zu machen.

---

[2] Eine Hash-Funktion ist eine Einwegfunktion, die von Daten beliebiger Länge eine Prüfsumme (Hash-Wert) konstanter Länge erzeugt. Es kann weder von einem gegebenen Hash-Wert auf die ursprünglichen Daten rückgeschlossen werden, noch kann einfach ein anderer Datensatz gefunden werden, der auf denselben Hash-Wert abbildet.

[3] Der Hauptgrund für das Anwenden dieser Hash-Funktion liegt darin, dass die zu signierenden Daten üblicherweise aus einem langen Text bestehen und ein Signieren dieses langen Textes für praktische Anwendungen sehr ineffizient und zeitintensiv ist.

[4] Aus dem öffentlichen Schlüssel kann nicht auf den privaten Schlüssel rückgeschlossen werden. Ebenso ist eine Signaturerstellung mit dem öffentlichen Schlüssel alleine nicht möglich. Ein über eine vertrauenswürdige dritte Partei ausgestelltes elektronisches Zertifikat enthält den öffentlichen Schlüssel der Signatorin und bindet diesen an ihre Identität.

Das Konzept editierbarer Signaturen wird in [1] ausführlich diskutiert. Die Autoren dieses Artikels definieren die folgenden Eigenschaften editierbarer Signaturen, anhand derer die unterschiedlichen existierenden Verfahren, die auf dem Konzept editierbarer Signaturen beruhen, klassifizieren werden können.

- **Eigenschaft E1 - Designierte Redigiererin/ Designierter Redigierer:** Diese Eigenschaft definiert, ob Daten von jeder Person oder ausschließlich von Redigierern bzw. Redigiererinnen, die durch die Unterzeichnerin bzw. den Unterzeichner definiert wurden, verändert werden können.
- **Eigenschaft E2 - Ersetzung von Blöcken:** Diese Eigenschaft legt fest, ob eine Redigiererin bzw. ein Redigierer bestimmte Textblöcke lediglich entfernen bzw. ausschwärzen, oder auch durch andere Textblöcken ersetzen kann.
- **Eigenschaft E3 - Designierte Teile:** Eine Unterzeichnerin oder ein Unterzeichner kann festlegen, ob eine Redigiererin oder ein Redigierer sämtliche Nachrichtenblöcke oder nur designierte Blöcke verändern darf.
- **Eigenschaft E4 - Erkennbare Veränderung:** Diese Eigenschaft definiert, ob eine Veränderung durch eine Redigiererin oder einen Redigierer im Nachhinein feststellbar ist.
- **Eigenschaft E5 - Kontrolliertes Ersetzen:** Diese Eigenschaft legt fest, ob eine Unterzeichnerin oder ein Unterzeichner bestimmen kann, mit welchen konkreten Textblöcken ein anderer Textblock ersetzt werden kann.

Durch die beliebige Kombination dieser Eigenschaften ergibt sich eine Vielzahl unterschiedlicher Verfahren. Diese Verfahren werden mitunter auch als Schema bezeichnet. Die folgende Tabelle bietet einen Überblick über verschiedenen Schemen und stellt deren Eigenschaften vergleichend gegenüber.

Tabelle 1: Editierbare Signatur Schemen und ihre Eigenschaften [1]

Signatur-Schema	**E1**	**E2**	**E3**	**E4**	**E5**
Content Extraction Signatures [2]	Nein	Nein	Teilw.	Ja	Nein
Sanitizable Signatures [3]	Ja	Nein	Ja	Nein	Nein
Homomorphic Signature Schemes [4]	Nein	Nein	Nein	Ja	Nein
Extended Sanitizable Signatures [5]	Ja	Ja	Ja	Ja	Ja
Extended Sanitizable Signature Schemes [6]	Ja	Ja	Ja	Ja	Ja
Generalizations and Extensions of Redactable Signatures [7]	Nein	Ja	Teilw.	Ja	Ja
Efficient signature schemes [8]	Nein	Nein	Ja	Ja	Nein

Unabhängig vom jeweiligen Schema beruht die prinzipielle Funktionsweise von editierbaren Signaturen auf der Beibehaltung des Hash-Werts der veränderten Daten. Eine Veränderung des Hash-Werts deutet auf veränderte Daten hin und würde bei konventionellen Signaturverfahren erwartungsgemäß zu einer Ungültigkeit der Signatur dieser Daten führen. Wird jedoch der

ursprüngliche Hash-Wert beibehalten und zur Verifikation der Signatur herangezogen, so kann die originale Signatur trotz veränderter Daten erfolgreich geprüft werden.

Abbildung 1 veranschaulicht das generelle Prinzip editierbarer Signaturen anhand eines einfachen Beispiels. Hierbei wird zur Signaturerstellung eine Nachricht *m* in fünf Blöcke m_1 bis m_5 unterteilt. Jeder dieser Nachrichtenblöcke wir anschließend über eine Hash-Funktion *H* in entsprechende Werte h_1 bis h_5 umgewandelt. Danach wird aus diesen Werten ein Gesamt-Hashwert H_{GES} berechnet, der in weitere Folge signiert wird. Wird nun wie in Abbildung 1 dargestellt der Textblock mit der Bezeichnung *„geschwärzter“* ausgeblendet (Nachrichtenblock m_4*), würde bei der Verifikation der Signatur ein anderer Hashwert h_4 berechnet werden[5], was in weiterer Folge zu einer ungültigen Signatur führen würde. Um dies zu verhindern, wird für den Verifikationsprozess der ursprüngliche Hashwert h_4 herangezogen. Dafür muss natürlich der Empfängerin bzw. dem Empfänger der Signatur, die üblicherweise die Verifikation durchführt, neben der Signatur selbst auch der Hashwert h_4 übermittelt werden. Somit erhält die Empfängerin bzw. der Empfänger die geschwärzte Nachricht und kann ohne Kenntnis des geschwärzten Textes die originale Signatur erfolgreich überprüfen. Durch die Einwegfunktionalität der Hash-Funktion bleibt zudem der Originaltext (in der Abbildung das Wort *„geschwärzter“*) der Empfängerin bzw. dem Empfänger verborgen. Bei herkömmlichen Hash-Funktionen in Verbindung mit einer kleinen Anzahl von möglichen Texten (wenn beispielsweise nur Vornamen als Text in Frage kommen) besteht die Gefahr, dass anonymisierte Texte durch einfaches Probieren aller möglichen Kombinationen rekonstruiert werden können. Für den realen Einsatz solcher Signaturschemen werden daher so genannte randomisierte Hash-Funktionen verwendet, die durch die Verwendung von Zufallswerten ein Erraten des anonymisierten Textes verhindern.

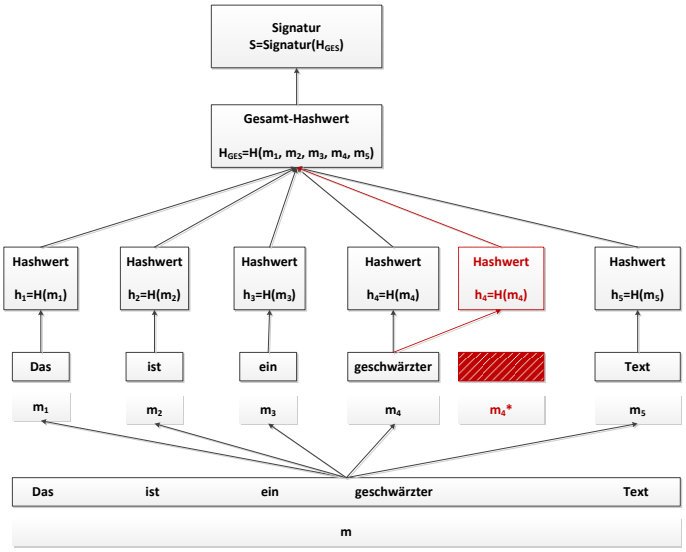


Abbildung 1: Prinzip editierbarer Signaturen

[5] $H(m_4)$ ist ungleich $H(m_4^*)$

Während herkömmliche elektronische Signaturen die Integrität und Authentizität von OGD gewährleisten können, eignet sich das kryptographische Konzept der editierbaren Signaturen vorzüglich, um die in Abschnitt 2 ebenfalls definierte Anforderung an eine Anonymisierung von OGD zu erfüllen und eine entsprechende Lösungen umzusetzen. Im folgenden Abschnitt stellen wir unser Konzept zur Integration herkömmlicher und editierbarer Signaturen in OGD-Lösungen vor.

3. Vertrauenswürdiges Open Government Data

Ziel des im Folgenden dargelegten Konzepts ist die Gewährleistung von Integrität und Authentizität von Open Government Data bzw. die Ermöglichung einer Anonymisierung dieser Daten. Zur Erfüllung dieser zusätzlichen Anforderungen sieht unser Konzept eine Integration der in Abschnitt 3 erläuterten Signaturkonzepte vor. Im Folgenden diskutieren wir Details unseres Konzepts und zeigen auf, wie sowohl Bereitstellerinnen und Bereitsteller als auch Bezieherinnen und Bezieher von Open Government Data von diesem Ansatz profitieren können.

Unmittelbar ergeben sich für eine Verwendung von elektronischen Signaturen in OGD-Anwendungen zwei allgemeine Anwendungsfälle. Je nach Anwendungsfall werden dabei unterschiedliche Varianten elektronischer Signaturen eingesetzt. Die beiden Anwendungsfälle werden im Folgenden näher diskutiert.

a. Anwendungsfall 1: Authentische und integritätsgesicherte Daten

In diesem Szenario zeigen wir auf, wie Bereitstellerinnen und Bereitsteller von Open Government Data authentische und integritätsgesicherte Daten bereitstellen können. Derartig gesicherte Daten bieten dabei generell folgende Vorteile:

- *Integrität der Daten*

  Durch die Integrität der Daten ist sichergestellt, dass eine nachträgliche Änderung der Daten entdeckt werden kann. Hiervon profitieren sowohl die Bezieherin bzw. der Bezieher der OGD (sie können auf die Richtigkeit der zur Verfügung stehenden Daten vertrauen) als auch die jeweilige Bereitstellerin bzw. der jeweilige Bereitsteller (eine Bezieherin ober ein Bezieher kann nicht behaupten, falsche Daten erhalten zu haben).

- *Authentizität der Bereitstellerin bzw. des Bereitstellers*

  Die Bezieherin bzw. der Bezieher der OGD kann die Identität der Bereitstellering oder des Bereitstellers eindeutig feststellen und somit auf die Richtigkeit und Vertrauenswürdigkeit der Daten vertrauen.

Das geeignete Mittel zur Umsetzung authentischer und integritätsgesicherter Daten sind herkömmliche elektronische Signaturen. Abbildung 2 illustriert die prinzipielle Vorgehensweise zur Erzielung von authentischen und integritätsgesicherten Open Government Data. In der Domäne der Bereitstellerin oder des Bereitstellers befindet sich die Original-Datenquelle der Daten, die als Open Government Data veröffentlicht werden sollen. Diese Daten werden mit dem privaten Signaturschlüssel der Bereitstellerin oder des Bereitstellers signiert. Abhängig vom Datenformat der vorliegenden Daten kann dies beispielsweise eine XML-basierte oder PDF-basierte Signatur sein. An dieser Stelle ist aber prinzipiell jedes weitere geeignete Signaturformat vorstellbar. Die signierten Daten können nun an geeigneter Stelle veröffentlicht werden und stehen der Bezieherin oder dem Bezieher als authentisches und integritätsgesichertes OGD zur Verfügung. Zur Überprüfung der Integrität und Authentizität kann die Bezieherin bzw. der

Bezieher die elektronische Signatur der signierten Daten überprüfen. Fällt diese Prüfung positiv aus, so kann die Bezieherin bzw. der Bezieher der Daten darauf vertrauen, dass die Daten nicht verändert oder modifiziert wurden und dass die Daten von der entsprechenden Bereitstellerin bzw. vom entsprechenden Bereitsteller zur Verfügung gestellt wurden.

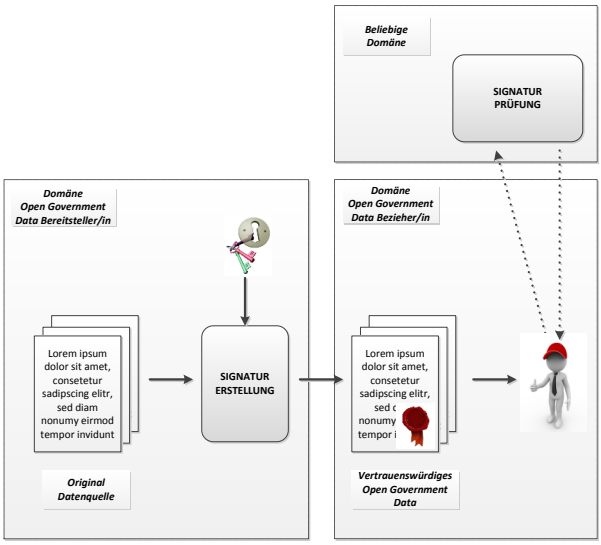


Abbildung 2: Anwendungsfall 1 – Authentisches und integritätsgesichertes Open Government Data

b. Anwendungsfall 2: Authentische und integritätsgesicherte Anonymisierung

Dieser Anwendungsfall deckt jene Fälle ab, in denen sich in der Original-Datenquelle personalisierte und private Daten befinden, die damit verknüpften allgemeinen Daten aber dennoch als Open Government Data veröffentlicht werden sollen. Es besteht hierbei also die Notwendigkeit einer Anonymisierung der Original-Daten, da private Daten aus Datenschutzgründen von einer Veröffentlichung ausgeschlossen sind.

Eine wie in Abschnitt 4.1 dargestellte Gewährleistung der Authentizität und Integrität dieser Daten über herkömmliche elektronische Signaturen ist nicht umsetzbar. Der Anonymisierungsprozess würde eine Änderung der zugrundeliegenden Daten bedingen und damit eine über diese Daten berechnete Signatur ungültig machen. Um dennoch authentisches und integrationsgesichertes Open Government Data zu erreichen, müssten die anonymisierten Daten erneut signiert werden. In manchen Fällen – wenn beispielsweise die ursprüngliche Unterzeichnerin bzw. Unterzeichner nicht greifbar oder eine erneute Signaturauslösung nicht umsetzbar ist – ist dies jedoch keine gangbare Alternative. An dieser Stelle können editierbare Signaturen Abhilfe schaffen.

Abbildung 3 zeigt das generelle Prinzip einer Anonymisierung von OGD basierend auf editierbaren Signaturen. Die Bereitstellering bzw. der Bereitsteller der OGD erstellt mit seinem Signaturschlüssel eine editierbare Signatur über die Original-Daten, welche private Informationen beinhalten. Die Redigiererin bzw. der Redigierer anonymisiert den Datensatz und aktualisiert die editierbare Signatur. Hierzu verwendet sie ihren privaten Schlüssel, wenn die

Bereitstellerin bzw. der Bereitsteller angegeben hat, dass nur bestimmte Redigiererinnen bzw. Redigierer Daten anonymisieren dürfen. Die auf diese Weise erstellte Signatur wird der Bezieherin bzw. dem Bezieher zusammen mit den anonymisierten Daten zugänglich gemacht. Die Bezieherin bzw. der Bezieher ist in weiterer Folge in der Lage, die Signatur der bezogenen Daten erfolgreich zu verifizieren. Hierbei wird die Signatur über die Original-Daten geprüft, ohne der Bezieherin oder dem Bezieher Zugriff auf die privaten Daten zu geben. Die Bezieherin bzw. der Bezieher kann bei einer positiven Signaturprüfung wiederum von sicheren und vertrauenswürdigen Daten ausgehen, d.h. sie können sowohl auf die Authentizität der Bereitstellerin bzw. des Bereitstellers als auch auf der Integrität der Daten vertrauen.

Je nach konkretem Anwendungsfall können in diesem Anwendungsszenario unterschiedliche Schemen editierbarer Signaturen eingesetzt werden. Abhängig von den Eigenschaften des gewählten Signaturschemas kann die Bereitstellerin bzw. der Bereitsteller der Open Government Data beispielsweise eine explizite Redigiererin bzw. einen expliziten Redigierer bestimmten (Eigenschaft „Designierte Redigiererin/Designierter Redigierer“), oder festlegen, dass nur bestimmte Teile der Daten editiert werden dürfen (Eigenschaft „Designierte Teile“). Des Weiteren kann prinzipiell auch ein Signaturschema eingesetzt werden, das nicht nur das Schwärzen sondern auch das Ersetzen von Daten ermöglicht (Eigenschaft „Ersetzung von Blöcken“). Im Bereich von Open Government Data erscheint diese Eigenschaft aber aktuell noch keine legitime und sinnvolle Anwendung zu finden.

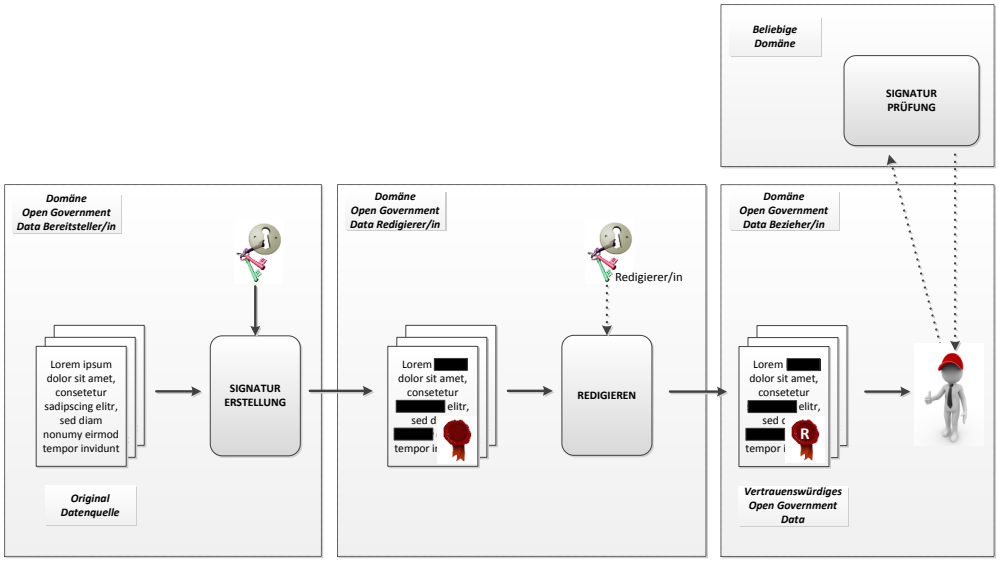


Abbildung 3: Anwendungsfall 2 – Authentische und integritätsgesicherte Anonymisierung

4. Fazit und Ausblick

Das in diesem Beitrag vorgestellte Konzept ermöglicht die Umsetzung von vertrauenswürdigem Open Government Data. Durch den Einsatz von elektronischen Signaturen kann die Authentizität und Integrität von OGD sichergestellt werden. Darüber hinaus kann eine eventuell notwendige Anonymisierung von derartigen Daten durch die Verwendung editierbarer elektronischer Signaturen gewährleistet werden.

Die Gewährleistung der Authentizität und Integrität von OGD ist prinzipiell für alle Formen und Ausprägungen von OGD (wie beispielsweise Linked Open Data) eine interessante Option, durch die die Vertrauenswürdigkeit bereitgestellter Daten insgesamt entscheidend erhöht werden kann. Die Möglichkeit einer Anonymisierung von OGD bei gleichzeitiger Beibehaltung derer Authentizität und Integrität ist vor allem für jene Datenbestände interessant, die private oder personenbezogene Daten enthalten, jedoch zumindest partiell der Allgemeinheit zugänglich gemacht werden sollen. Als mögliche Beispiele können hier Datenauszüge aus dem Grundbuch, detaillierte Bevölkerungsstatistiken, Budget- und Steuerstatistiken, oder auch Protokolle aus Gemeinderatssitzungen genannt werden.

Der in diesem Beitrag vorgestellt Ansatz zur Umsetzung von vertrauenswürdigem Open Government Data liegt derzeit als Konzept vor. In einem nächsten Schritt ist nun die Evaluierung der praktischen Durchführbarkeit dieses Konzepts anhand einer prototypischen Umsetzung geplant. Unabhängig vom Resultat dieser Evaluierung kann bereits festgehalten werden, dass die Authentizität, Integrität und damit die Vertrauenswürdigkeit von Open Government Data wichtige Aspekte darstellen, die die Zukunft von OGD entscheidend mitbeeinflussen und neue Herausforderungen aufwerfen werden. Die in diesem Beitrag vorgestellten Konzepte sind ein erster Schritt, um diesen Herausforderungen entsprechend zu begegnen.

Referenzen

[1] D.Slamanig und S.Rass, “Redigierbare Signaturen: Theorie und Praxis“ in: Datenschutz und Datensicherheit, Bd. 35, Nr. 11, S. 757-762

[2] R. Steinfeld, L. Bull und Y. Zheng: Content Extraction Signatures. ICISC, LNCS 2288, S. 285-304. Springer, 2001.

[3] G. Ateniese, D. H. Chou, B. de Medeiros und G. Tsudik. Sanitizable Signatures. ESORICS, LNCS 3679, S. 159-177. Springer, 2005.

[4] R. Johnson, D. Molnar, D. X. Song und D. Wagner. Homomorphic Signature Schemes. CTRSA, LNCS 2271, S. 244-262. Springer, 2002.

[5] M. Klonowski und A. Lauks. Extended Sanitizable Signatures. ICISC, LNCS 4296, S. 343-355. Springer, 2006.

[6] S. Canard und A. Jambert. On Extended Sanitizable Signature Schemes. CT-RSA, LNCS 5985, S. 179-194. Springer, 2010.

[7] D. Slamanig und S. Rass. Generalizations and Extensions of Redactable Signatures with Applications to Electronic Healthcare. CMS, LNCS 6109, S. 201-213. Springer, 2010.

[8] S. Haber, Y. Hatano, et al.: Efficient signature schemes supporting redaction, pseudonymization, and data identification. ASIACCS, S. 353-362. ACM, 2008.

[9] 8 Principles of Open Government Data, www.opengovdata.org/home/8principles, 2007

[10] Europäische Kommission: Mitteilung der Kommission an das Europäische Parlament, den Rat, den Europäischen Wirtschafts- und Sozialausschuss und den Ausschuss der Regionen – Offene Daten: Ein Motor für Innovation, Wachstum und transparente Verwaltung, http://ec.europa.eu/information_society/policy/psi/docs/pdfs/opendata2012/open_data_communication/de.pdf, 2011.

[11] Europäisches Parlament und der Rat der Europäischen Union: Richtlinie 2003/98/EG des Europäischen Parlaments und des Rates vom 17. November 2003 über die Weiterverwendung von Informationen des öffentlichen Sektors, Amtsblatt der Europäischen Union, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2003:345:0090:0096:DE:PDF, 2003.

[12] Open Government Data Austria, http://gov.opendata.at/site/, 2012.

[13] Europäisches Parlament und der Rat der Europäischen Union: Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen, Amtsblatt der Europäischen Union, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000:013:0012:0020:DE:PDF, 2000.

[14] Jörn von Lucke, Christian P. Geiger: Open Government Data – Frei verfügbare Daten des öffentlichen Sektors, http://www.zu.de/deutsch/lehrstuehle/ticc/TICC-101203-OpenGovernmentData-V1.pdf, 2010

[15] Johann Höchtl, Peter Reichstädter: Linked Open Data – A Means for Public Sector Information Management, Proceedings of the Second international conference on Electronic government and the information systems perspective, 330-343, 2011.

Über die Autoren

Klaus Stranacher

Klaus Stranacher absolvierte das Studium der Telematik an der Technischen Universität Graz. Im Jahr 2005 wurde er Mitarbeiter des E-Government Innovationszentrums (EGIZ) in Graz und beschäftigt sich derzeit mit aktuellen Themen im Bereich des E-Government und der IT-Sicherheit, im Speziellen mit elektronischen Identitäten, elektronischen Dokumenten und Interoperabilität. Während seiner Tätigkeit wirkte er unter anderem in den folgenden EU Projekten mit: FP6 Projekt eGov-Bus, LSP Projekt STORK und LSP Projekt SPOCS (als Leiter des Arbeitspakets „elektronische Dokumente“).

Vesna Krnjic

Vesna Krnjic absolvierte das Bachelorstudium der Informatik an der Technischen Universität Graz. Im Jahr 2010 wurde sie Mitarbeiterin des Instituts für Angewandte Informationsverarbeitung und Kommunikationstechnologie (IAIK) an der TU Graz im Umfeld des E-Government und der IT-Sicherheit. Im Speziellen arbeitet sie im Bereich der Usability und Testen. Zusätzlich beschäftigt sie sich, im Rahmen ihrer Masterarbeit, mit visuellen Programmiersprachen am Smartphone speziell entwickelt für Kinder und Jugendliche.

Thomas Zefferer

Thomas Zefferer absolvierte das Studium der Telematik an der Technischen Universität Graz (TU Graz). Seit dem Jahr 2007 ist er Mitarbeiter des Instituts für Angewandte Informationsverarbeitung und Kommunikationstechnologie (IAIK) der TU Graz und in den Bereichen IT-Sicherheit und E-Government tätig. In den letzten Jahren war er in zahlreiche Projekte und Tätigkeiten des E-Government Innovationszentrums (EGIZ) und des Zentrums für sichere Informationstechnologie – Austria (A-SIT) involviert. Sein aktueller Forschungsschwerpunkt liegt auf Smartphone-Security und mobile E-Government-Verfahren.

15 | Authentische und integritätsgesicherte Verwaltungsdaten

Journal	eGovernment Review
Language	German
Title	Authentische und integritätsgesicherte Verwaltungsdaten
Authors	Klaus Stranacher, Vesna Krnjic, Thomas Zefferer

Authentische und integritätsgesicherte Verwaltungsdaten

Klaus Stranacher I Vesna Krnjic I Thomas Zefferer

abstract

In den letzten Jahren konnte Open Government Data zunehmend an Bedeutung gewinnen. Für die öffentliche Bereitstellung von Verwaltungsdaten wurden bisher jedoch keinerlei Anforderungen hinsichtlich der Authentizität und Integrität solcher Daten festgelegt. Zur Steigerung der Vertrauenswürdigkeit offener Verwaltungsdaten bieten sich herkömmliche elektronische Signaturen an. Im Falle einer zum Schutz privater Daten notwendigen Anonymisierung kann durch den Einsatz sogenannter Redigierbarer Signaturen die Vertrauenswürdigkeit bereitgestellter Daten weiterhin zuverlässig sichergestellt werden.

Open Data ist eine Entwicklung der letzten Jahre, die auch vor dem E-Government-Sektor nicht halt gemacht hat. Der Begriff Open Government Data (OGD) bezeichnet dabei jene „Datenbestände des öffentlichen Sektors, die von Staat und Verwaltung im Interesse der Allgemeinheit ohne jedwede Einschränkung zur freien Nutzung, zur Weiterverbreitung und zur freien Weiterverwendung frei zugänglich gemacht werden“[4]. Innerhalb Österreichs wurde im Juli 2011 die Cooperation Open Government Data Österreich[5] gegründet, deren Ziel die Schaffung einer gemeinsamen Basis für die Veröffentlichung von offenen Verwaltungsdaten ist.

Anforderungen an Open Government Data. Zur Bereitstellung von offenen Daten im Rahmen von Open Government Data wurde eine Reihe von allgemeinen Anforderungen definiert. Die Open Government Arbeitsgruppe[2] veröffentlichte hierzu acht Prinzipien, die so weit wie möglich eingehalten werden sollen. Diese Prinzipien sind: (a) Vollständigkeit der Daten, (b) Verwendung von Primärquellen, (c) Aktualität der Daten, (d) uneingeschränkter Zugang, (e) Maschinenlesbarkeit, (f) nichtdiskriminierender Zugang, (g) Verwendung nichtproprietärer Datenformate und (h) Lizenzfreiheit.

Diese Grundsätze enthalten jedoch keinerlei Anforderungen hinsichtlich der Sicherstellung der Vertrauenswürdigkeit der Daten. Abhängig vom jeweiligen Anwendungsszenario können jedoch zwei weitere Anforderungen formuliert werden: (i) Datenintegrität und Authentizität sowie (ii) Anonymisierung. Die erste Anforderung ermöglicht es Bezieherinnen und Bezieher von OGD, unerlaubte Veränderungen der veröffentlichten Daten zu überprüfen (Datenintegrität), wohingegen Authentizität die Feststellung der Identität der OGD-Bereitstellerin bzw. des OGD-Bereitstellers erlaubt. Die zweite Anforderung definiert die Möglichkeit, Daten unter Beibehaltung der Datenintegrität und Authentizität zu anonymisieren. Da personenbezogene Daten im Allgemeinen von einer Veröffentlichung ausgeschlossen sind, kann eine entsprechende Anonymisierung der zu veröffentlichenden Daten notwendig sein.

Elektronische Signaturen. Das geeignete Mittel zur Wahrung der Integrität und Authentizität von Daten sind elektronische Signaturen. Bei konventionellen Signaturverfahren führt dabei jede Änderung der signierten Daten unweigerlich zu einer ungültigen Signatur, d. h. die Empfängerin bzw. der Empfänger kann nicht mehr von der Vertrauenswürdigkeit der erhaltenen Daten ausgehen. Es existieren jedoch Anwendungsfälle, in denen eine nachträgliche Änderung der signierten Daten – für designierte Personen – ermöglicht werden soll, ohne dass die aufgebrachte Signatur durch die durchgeführten Änderungen ihre Gültigkeit verliert. Dieses Konzept wurde unter dem Namen Redigierbare Signaturen bekannt[3].

Vertrauenswürdiges Open Government Data. Durch die Erfüllung der erweiterten Anforderungen an Open Government Data kann eine Steigerung der Vertrauenswürdigkeit von Open Government Data erreicht werden. Basierend auf den oben definierten erweiterten Anforderungen können prinzipiell zwei Anwendungsfälle unterschieden werden.

Abbildung 1 zeigt den ersten Anwendungsfall. In der Domäne der OGD-Bereitstellerin bzw. des Bereitstellers befindet sich die Original-Datenquelle. Diese Daten werden nun vor der Veröffentlichung von der Bereitstellerin bzw. dem Bereitsteller signiert. Der Bezieherin bzw. dem Bezieher stehen diese Daten anschließend als vertrauenswürdiges OGD zur Verfügung. Zur Über-

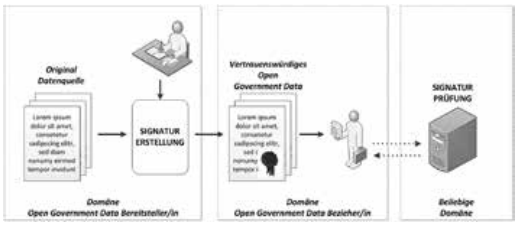


Abb. 1: Authentisches und integritätsgesichertes OGD (Anwendungsfall 1)

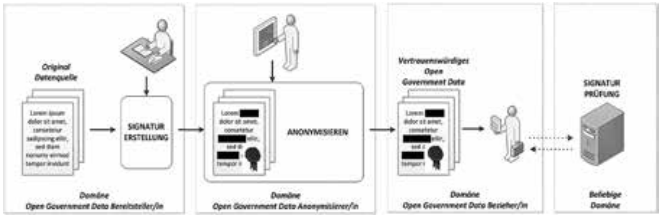


Abb. 2: Authentische und integritätsgesicherte Anonymisieren (Anwendungsfall 2)

prüfung der Authentizität und Integrität kann die elektronische Signatur geprüft werden. Fällt diese Prüfung positiv aus, so wurden die Daten nach ihrer Veröffentlichung nachweislich nicht verändert und wurden auch nachweislich von der angegebenen Stelle veröffentlicht.

Der zweite Anwendungsfall wird in Abbildung 2 dargestellt und deckt jene Fälle ab, in denen sich in der Original-Datenquelle personenbezogene Daten befinden, die damit verknüpften allgemeinen Daten aber dennoch als OGD veröffentlicht werden sollen. Die personenbezogenen Daten müssen daher anonymisiert werden. Um die Gültigkeit der Signatur nach dem Anonymisierungs-Prozess zu erhalten, können Redigierbare Signaturen eingesetzt werden. Die Bereitstellerin bzw. der Bereitsteller der Daten erstellt eine Signatur über die nicht-anonymisierten Daten und legt eine (oder mehrere) Personen fest, die eine Anonymisierung durchführen dürfen. Im Anonymisierungs-Prozess führt diese Person eine Schwärzung der entsprechenden Textstellen durch und aktualisiert die Signaturdaten. Anschließend werden die anonymisierten Daten und die Signaturdaten der Bezieherin bzw. dem Bezieher zugänglich gemacht. Die Bezieherin bzw. der Bezieher kann in Folge die Signatur über die Original-Daten prüfen (ohne Wissen der personenbezogenen Daten) und kann – bei einer positiven Signaturprüfung – von authentischen und integritätsgesicherten Daten ausgehen.

Fazit. Das hier präsentierte Konzept trägt zu einer Steigerung der Vertrauenswürdigkeit offener Verwaltungsdaten bei und wurde auch bei der 1. Open Government Data Konferenz D-A-CH-LI vorgestellt. Der entsprechende Beitrag findet sich im Tagungsband der Konferenz[1] und gibt einen detaillierteren Einblick. Die vorgestellte Lösung lässt sich auch auf Anwendungen außerhalb von OGD adaptieren. So befasst sich der Beitrag in[6] auch mit der Gewährleistung der Authentizität und Integrität für Daten, die im Rahmen der EU PSI-Richtlinie[7] zur Verfügung gestellt werden. ■

DI Klaus STRANACHER
E-Government Innovationszentrum (EGIZ);
klaus.stranacher@egiz.gv.at

Vesna KRNJIC BSc
E-Government Innovationszentrum (EGIZ);
vesna.krnjic@egiz.gv.at

DI Thomas ZEFFERER
Zentrum für sichere Informationstechnologie – Austria (A-SIT);
thomas.zefferer@a-sit.at

literatur

[1] Klaus Stranacher, Vesna Krnjic, Thomas Zefferer; Vertrauenswürdiges Open Government Data – Authentizität und Integrität für öffentliche Verwaltungsdaten; Open Government Data Konferenz D-A-CH-LI; Seite 27-39; Oktober 2012.

[2] Open Government Working Group, 8 Principles of Open Government Data; 2007.

[3] D.Slamanig, S.Rass; Redigierbare Signaturen: Theorie und Praxis; Datenschutz und Datensicherheit; Bd. 35; Nr. 11; Seite 757-762.

[4] Jörn von Lucke, Christian P. Geiger: Open Government Data – Frei verfügbare Daten des öffentlichen Sektors; 2010.

[5] Cooperation OGD Österreich, data.gv.at – offene Daten Österreichs; 2012; http://www.data.gv.at./

[6] Klaus Stranacher, Vesna Krnjic, Thomas Zefferer; Trust and Reliability for Public Sector Data; ICBG - International Conference on e-Business and e-Government; Jänner 2013; (Veröffentlichung im Jänner 2013).

[7] Richtlinie 2003/98/EG des Europäischen Parlaments und des Rates vom 17. November 2003 über die Weiterverwendung von Informationen des öffentlichen Sektors; 2003.

16 | Secure and Efficient Processing of Electronic Documents in the Cloud

Conference	IAIDIS International Conference e-Society
Language	English
Title	Secure and Efficient Processing of Electronic Documents in the Cloud
Authors	Klaus Stranacher, Bernd Zwattendorfer, Vesna Krnjic
Booktitle	Proceedings of IAIDIS International Conference e-Society

SECURE AND EFFICIENT PROCESSING OF ELECTRONIC DOCUMENTS IN THE CLOUD

Klaus Stranacher, Bernd Zwattendorfer, Vesna Krnjic
Graz University of Technology, E-Government Innovation Center, EGIZ
Inffeldgasse 16a, 8010 Graz, Austria

ABSTRACT

Electronic documents are often exchanged in e-Government and e-Business processes. In e-Government, the usage and importance of electronic documents has significantly increased particularly in cross-border scenarios, especially due to the implementation of the EU Services Directive. To ensure genuineness, in many situations electronic signatures are applied on the documents exchanged. Besides of their application, verification of electronic signatures is essential. Current solutions for the verification of electronic signatures usually support a subset of existing signature formats only. In addition, electronic documents require some kind of detailed description on higher level, e.g. through meta data. If corresponding meta data are recognized as incomplete or wrong during document exchange, additional costs and time delays may occur. Here, the need of previous data validation arises. To overcome these issues, we introduce an approach for secure and efficient processing of electronic documents, particularly focusing on signature and meta data verification. Our solution follows a generic concept and is not limited to certain use cases. Nevertheless, we present our approach based on the findings of the EU Large Scale Pilot Project SPOCS. Finally, we elaborate on the movement of verification and validation services into the cloud.

KEYWORDS

Electronic Documents, Signature Verification, Data Validation, Cloud Computing

1. INTRODUCTION

Electronic documents are important parts of most e-Government and e-Business processes. Their significance particularly increased with the progressing **implementation of the EU Services Directive** (European Union, 2006). The main objective of the directive is to establish a framework for easily setting up and exercise a service in another EU Member State by using **electronic procedures. Here, electronic documents are one of the key enablers to achieve this goal**. To guarantee authenticity and integrity of electronic documents, usually electronic signatures are applied to electronic documents. The validity of an electronic signature can be unambiguously determined by the receiver of a signed document using signature verification services. Current existing signature verification services are limited to verify only certain signature formats. In general, they support several kinds of standard signature formats as defined by the European Commission Decision on *"establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Services Directive"* (European Commission, 2011). In addition, they are able to verify a few other formats suitable for their field of application. Nevertheless, there exists a lack on comprehensive signature verification services supporting a wider field of signature formats.

In addition to the verification of genuineness of electronic documents, automatic processing of electronic documents is essential for a cost reducing, time saving, and efficient public administration. The basis for automatic processing is availability of machine-readable data, i.e. structured electronic documents and appropriate meta data. Nevertheless, additional costs and time delays may arise if electronic documents or meta data are recognized as incomplete or wrong. Here, the need of a previous data validation arises.

The European Large Scale Pilot SPOCS[1] developed an electronic document interoperability framework and document container format, called OCD - Omnifarious Container for eDocuments (SPOCS, 2011). Among other things, this framework defines how to verify and validate an OCD container and all affiliated electronic documents. The focus of the OCD interoperability framework has been given on the verification of the electronic signatures applied to the container and the contained electronic documents. In this paper we present and propose mechanisms for secure and efficient processing of electronic documents.

The remainder of this paper is organized as follows. In Section 2, we describe the electronic document interoperability framework and the Omnifarious Container for eDocuments (OCD). Additionally, we point out existing solutions for verifying and validating electronic documents. Section 3 elaborates on external signature verification services to support an extended set of signature formats. In addition, a data validation mechanism is proposed. These verification and validation facilities base on the OCD but are not limited to this use case. The subsequent Section 4 elaborates possibilities to transfer verification and validation services into the cloud. Finally, we draw conclusions including an evaluation of our proposed solution and discuss future work.

2. RELATED WORK

Basically, electronic documents can be divided into structured, unstructured, and container formats. The content of structured document formats follows a well-defined schema and is therefore machine-readable and can be easily processed. The most popular structured eDocument format is XML. In contrast, unstructured electronic documents, such as the PDF format, cannot be automatically processed. They are mainly used for visual representation of document content. Container formats specify how different types of data are stored in one container. Additionally, all required information, which third parties would need for processing the documents, is stored in the container. One of the first container formats was MIME[2]. In the meanwhile, formats such as Open Document Format[3] (ODF) and Office Open XML[4] (OOXML) have increased in popularity.

Looking at the e-Government landscape in Europe, every country has its own eDocument infrastructure deployed based on existing standards and technologies. Many national applications are using XML-based specifications for information and document exchange. However, national XML specifications cannot be automatically processed nor automatically interpreted by any third party without the knowledge of the schema of the particular document. Due to the EU Services Directive the need of interoperability for electronic documents, especially on a cross-border level, has significantly increased. This need for interoperability has also been discussed by Rössler and Tauber (2010).

The challenge on interoperability has been taken up by the European Large Scale Pilot SPOCS. Here, an interoperability concept has been introduced, which bases on the individual national infrastructures of the participating EU countries and builds an interoperability layer on top of it. This concept is called Omnifarious Container for eDocuments (OCD) and represents an interoperable multi-layer framework for cross-border exchange of electronic documents. The container supports all formats and technologies of electronic documents and is easily extendable to support new formats and technologies too. Additionally, semantic interoperability and authentication mechanisms for guaranteeing the authenticity of an OCD container are provided.

The specification of the OCD container (SPOCS, 2011) consists of a logical and a physical structure. Thereby, the logical structure consists of a payload layer, a meta data layer, and an authentication layer. The payload layer stores all kind of electronic documents, which should be transported in the OCD container. To

[1] SPOCS (Simple Procedures Online for Cross-border Services) is an EU co-funded project out of the EU ICT Policy Support Programme and aims to overcome the obstacles raised by the EU Services Directive. http://www.eu-spocs.eu/.

[2] MIME (Multipurpose Internet Mail Extensions) are extensions of the standard RFC 822 and defined in RFC 2045, RFC 2046, RFC 2047, RFC 2048 and RFC 2049.

[3] ODF is a standard developed by the standardization organization OASIS and is specified in ISO/IEC 26300.

[4] OOXML is a standard developed by Microsoft and is specified in ISO/IEC 29500.

support automatic processing, the meta data layer has been introduced on two levels. The first level describes each payload document, while the second level describes the container itself. In addition to the signed payload documents, the whole container can be signed as well. This authentication layer is optional and enables the support of authenticity of OCD containers.

Two different physical structures are defined to implement the logical structure. The ZIP based OCD relies on the ETSI specification on Associated Signature Containers - ASiC (ETSI, 2012) and uses XAdES signatures (ETSI, 2009) for the authentication layer. This ZIP based OCD is primarily suitable for back office applications. The second structure is a PDF based OCD where the master PDF represents the meta data and the payload documents are added as attachments. Here, PAdES signatures (ETSI, 2010) are used for the authentication layer. This technology is especially suitable for applications where citizens are directly involved.

To handle OCD in real live scenarios, operations on the core elements of OCDs are defined. The OCD Creation method defines how an OCD container is created. As input, this method takes arbitrarily signed or unsigned electronic documents with appropriate meta data. The resulting OCD container can be signed optionally. The OCD Validation and Verification method defines how an OCD container is validated and how signature verification is carried out. This method takes an OCD container as input. The output of this method represents the corresponding validation and verification report. The described methods have been implemented as open source software modules and are freely available for download on Joinup[5].

In addition to OCD, several other signature verification activities have been established. The European Commission published a tool, called SD-DSS[6], which is capable to verify signature formats based on the European Commission Decision on standard signature formats (European Commission, 2011). Furthermore the EU Large Scale Pilot PEPPOL[7] addressed issues concerning the signature verification in the field of e-Procurement and developed a suitable signature verification service[8]. Nevertheless, these services do not support verification of national and proprietary signature formats.

3. VERIFICATION AND VALIDATION SERVICES

Verification of the genuineness of electronic documents is important to trust the authenticity and data integrity of these documents. Usually, electronic signatures are the means of choice for guaranteeing authenticity and integrity. Verification of these signatures is essential for their further processing. In addition, data validation, i.e. the validation whether the present data are appropriate and correct or not, gains more and more importance. Both, signature verification and data validation are necessary for a secure and efficient processing in e-Government or e-Business scenarios.

The following sub-sections elaborate on signature verification and data validation of electronic documents incorporating external verification and validation services. Thereby, we concentrate on the use cases related to the EU Services Directive and the implementations of the large-scale pilot project SPOCS, focusing on OCD container verification and validation. Nevertheless, our approach is not limited to these use cases and applies for all processes where electronic documents are involved and must be processed. In addition, we show external dependencies to our methods, which are able to be outsourced to cloud computing, enabling high scalability and cost savings.

3.1 Signature Verification

For electronic signatures various data formats exist. On the one hand, there are signature formats which are tightly bound to specific document formats, such as PDF signatures. On the other hand, there exist signature formats which can be used with almost every document format, e.g. XML and XAdES signatures. Based on the EU Services Directive (European Union, 2006) the European Commission established minimum requirements for the cross-border processing of documents signed electronically by competent

[5] http://joinup.ec.europa.eu/site/spocs/eDocuments/
[6] http://joinup.ec.europa.eu/software/sd-dss/home/
[7] PEPPOL (Pan-European Public Procurement Online), http://www.project.peppol.eu/
[8] http://www.peppol.eu/peppol_components/esignature/esignature

authorities under the Services Directive. Article 1 (1) of this decision defines three signature formats, namely *"XML or a CMS or a PDF advanced electronic signature in the BES or EPES format"* (European Commission, 2011), as minimum or standard formats to be processed by EU Member States. In addition, Article 1 (2) states that *"Member States whose competent authorities sign the documents referred to in paragraph 1 using other formats of electronic signatures than those referred to in that same paragraph, shall notify to the Commission existing validation possibilities that allow other Member States to validate the received electronic signatures online, free of charge and in a way that is understandable for non-native speakers […]"* (European Commission, 2011).

Actual existing signature verification services are limited to verify certain signature formats. In general, they support the standard signature formats and a few other formats suitable for their field of application. So usually national and proprietary formats, as mentioned in Article 1 (2) of the EC Decision, are not supported. Stranacher and Kawecki (2012) presented a signature verification service, which introduced the concept of external signature verification services. This service can be used to integrate the verification of national and proprietary signature formats. Their concept bases upon the OCD Validation and Verification Module but lacks on a concrete implementation of this mechanism.

In Figure 1 we show the concrete mechanism to integrate external signature verification services on the basis of the OCD Validation and Verification Module. As an OCD container can be signed itself and usually contains signed electronic documents, the container signature and the document signatures are divided. A format detection unit analyzes the signature and recognizes the signature format. The verification of standard signature formats is covered by the internal signature verification unit. National and proprietary formats are verifiable via external verification services. These external services have to be defined in the configuration of the module. Within the configuration, a mapping between the MIME type representing the signature format and the respective external service is given. Based on this mapping, the verification of the national and proprietary formats are outsourced to the external service via a connector. This connector creates the request to the external services and receives the corresponding response. Additionally, the connector converts and transforms the response into the OCD module internal verification result format. Finally, the result generator unit collects all results, including additional validation results from a basic validation[9] (not shown in the figure for clarity), and generates an XML based verification report.

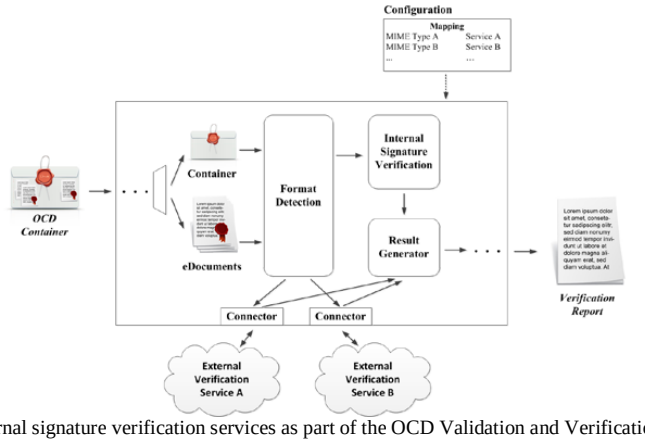


Figure 1. External signature verification services as part of the OCD Validation and Verification Module

Examples for external signature verification services supporting national and proprietary formats are the:

- Lithuanian verification service: This service supports the verification of the ADOC format specified by Director General of the Lithuanian Archives Department (2009).
- Austrian verification service: This service supports the verification of the PDF-AS format. PDF-AS is a proprietary Austrian format based on PDF and explained by Leitold et al (2009). Based on the solution of Zefferer et al. (2011), a Web-Service of this service based on SOAP will be available soon.

[9] The basic validation validates if the OCD container is compliant to the OCD specification.

These verification services may also be maintained within a cloud. An evaluation of this cloud-based approach is given in section 4.

3.2 Data Validation

Electronic documents are usually received by a service or application to be used for further processing. For instance, a public authority receives a request for opening a business and forwards it to the relevant competent authorities, which actually handle the request. If these documents are recognized as incomplete or wrong during further processing, the entire process must be stopped. To avoid associated costs and time delays, which may occur in such situations due to the necessity of manual interactions, a previous automated data validation is necessary. Data validation simply means that the data is verified if it fulfills the requirements for the subsequent process.

Basically, two different kinds of data can be distinguished. On the one hand, *meta data* provides information about the accompanied data such as the creation date or the creator of the data. Usually, meta data is available as machine-readable data. For instance, the OCD container comprises a meta data layer which describes the container itself (so-called meta data level 2) and the included electronic documents (so-called meta data level 1). On the other hand, the OCD specification (SPOCS, 2011) defines *document data* as a unified and machine-readable description of the content, optionally including the real content data. This document data introduces a mechanism to describe the content of an electronic document, which is available in a non-machine-readable format only[10], but still in a structured way though.

Document data defines a set of information on the level of electronic documents for storing machine-readable content. This set of information includes:

- A type identifier, which indicates the type of the document data, e.g. this is a birth certificate.
- A description of the structure, e.g. a birth certificate must contain the name and date of birth of the person as well as the names of her parents.
- The extracted values out of the original electronic documents satisfying the defined structure, e.g. the real name and date of birth of the person as well as the real names of her parents.

Figure 2 shows the basic principles of meta data (a.) and document data validation (b.). For meta data validation, meta data to be validated serve as input for the validation. For instance, such meta data can be extracted from an OCD container. In addition, a meta data profile ID selects a certain pre-configured meta data profile. Such a profile defines the meta data structure, i.e. which meta data must be present (e.g. meta data must contain a sender and a subject) and optionally which content must be present in the corresponding meta data fields (e.g. the sender must be “John Doe”). Based on this profile, the meta data is validated. First, the structure of the meta data is validated. In the second and optional step, the contents of the meta data fields are validated against the selected profile. Based on these validation steps, a common validation result is generated.

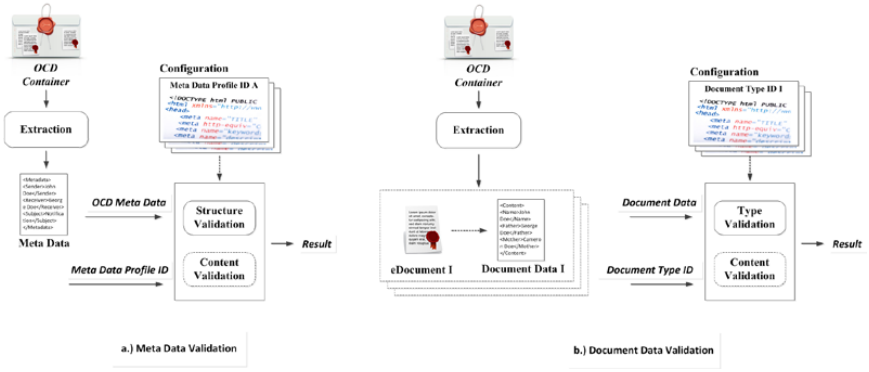


Figure 2. Meta data and document data validation

[10] E.g. a PDF document containing a scanned copy of a birth certificate

Document data validation is carried out according to a similar principle. Document data to be validated serve as input. For example, document data can be extracted from an OCD container. As second input, a document type ID is given, selecting a pre-configured document type profile. Thereby, a document type profile indicates the structure and optionally the contents of the document type (e.g. a birth certificate). Subsequently, document data are validated checking compliance against the profile, i.e. the data represent the given document type and – optionally – contain the required content. Finally, a validation result is generated.

Both meta data and document data validation base on XML schemata. During the validation process the data are verified if it is compliant to the given XML schema.

Figure 3 shows the integration of meta data and document data validation based on the OCD Validation and Verification Module. Here, meta data validation is an internal part of the module as the meta data scheme is OCD specific. Nevertheless, the concept of the proposed meta data validation is adaptable and can be used in various scenarios where validation of meta data is necessary. In addition, the validation of document data is linked to an external service as this validation is not OCD specific and thus follows a universal approach. Finally, the results of the meta data and document data validation are added to the verification report. External document data validation may also be maintained within a cloud. Section 4 elaborates on a possible cloud-based approach.

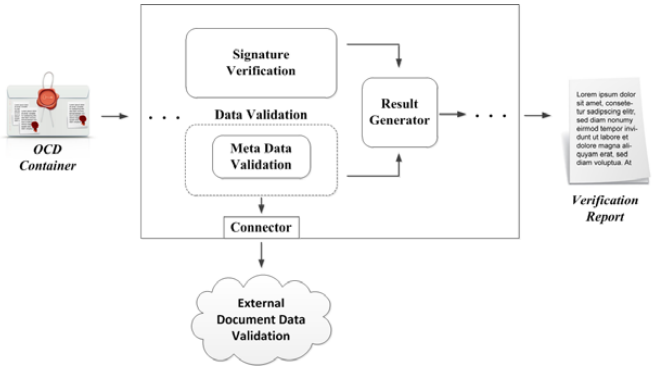


Figure 3. Data validation as part of the OCD Validation and Verification Module

4. VERIFICATION AND VALIDATION SERVICES IN THE CLOUD

Verification and validation of both electronic signatures and corresponding data applied to electronic documents are essential. In this section, we elaborate how cloud computing can be taken up to further improve verification and validation services of electronic documents.

Cloud computing (Mell and Grance, 2010) is currently on of the dominant topics within the ICT sector. The main aim of cloud computing – providing IT resources such as computing power or data storage based on a pay-as-you-go model – promises a lot of benefits. For instance, IT resources can be consumed just on demand and only effectively consumed resources are charged and must be paid. On the one hand, this provides high scalability for online services because required resources can be easily added. On the other hand, due to the flexible pricing model a lot of costs can be saved. By taking up cloud computing also for verification and validation services, these services can also take advantage of higher scalability and cost savings.

Basically, we see two main scenarios where cloud computing can particularly help improving verification and validation services, namely by deploying a

- Single external verification or validation service in the cloud or
- Brokered external verification or validation service in the cloud

We elaborate both approaches in more detail in the next sub-sections.

4.1 Single External Verification or Validation Service in the Cloud

Involving external verification services extend internal signature verification mechanisms, as the applied signature format might be proprietary and hence not supported by an internal service. This especially holds for country-specific signature formats, e.g. the Austrian and Lithuanian signature verification services as mentioned in Section 3.1. The verification of electronic signatures constitute a frequent process, hence such a national signature verification service may face a lot of requests. In particular, the amount of requests to be processed may not be constant. I.e. situations may occur where such national services have to cope with a high load. In such situations, simple verification services may not be able to handle load peaks and may tend to break down. More severely, in a worst case this can lead to a denial of service.

To bypass such bottlenecks, the verification service could be easily deployed as cloud service. The cloud guarantees nearly an independent amount of resources. Hence such potential bottlenecks could be easily overcome. In addition, applying the cloud computing paradigm offers some cost savings potential, as only the consumed amount of resources has to be paid. An imaginable scenario would be the implementation and deployment of a central cloud service per country, which is capable of processing individual signature verification requests. A similar approach, where countries host single and central gateways for individual data processing, can be found in the European Large Scale Pilot Projects STORK[11] and epSOS[12].

While scalability issues can easily be solved by applying cloud computing, the use of the cloud might bring up other issues in terms of security or privacy (Zissis and Lekkas, 2012). Before deploying such verification and validation services in the cloud, a thorough analysis on the cloud model to be applied is required. While public clouds offer the highest cost savings potential, private or community clouds might be favored as they allow higher control on the data to be processed (Catteddu and Hogben, 2009).

Finally, applying such a model is not limited to signature verification services. Needless to say that data validation services could follow such an approach too.

4.2 Brokered External Verification or Validation Service in the Cloud

While single external verification services in the cloud bypass the issue on scalability, they still leave the issue on heterogeneity of external verification services unresolved. Applying the single external verification services in the cloud model can lead to situations, where verification modules still have to manage several different interfaces to those external services. In other words, verification modules must support and implement the interface for connecting to the Austrian verification service, the interface to the Lithuanian verification service, etc. Such a model does not perfectly scale, hence we propose a brokered external verification service in the cloud similar to the brokered approach described by the Cloud Security Alliance (2011) as a second option. In this model, the verification module needs to support one interface to an external verification service only, namely to the brokered external verification service in the cloud. In addition, the brokered external service incorporates several other external verification services interfaces, e.g. the interfaces of several countries. In other words, such a service acts like a broker or hub between the verification and validation module and several external services. Summarizing, this approach provides two main advantages. The first advantage is scalability as the service is deployed in the cloud. The second advantage is the support of individual other external verification services to avoid heterogeneity.

However, this approach has also to deal with privacy and security concerns. Probably, private companies might take up this approach and hence data could be processed in a public cloud, which provide a lower security or privacy level. To bypass these concerns, it might also be feasible that the European Commission itself sets up such a service. Hence, to ensure higher control on the data to be processed such a scenario relates more to the application of a private or community cloud. Again, this approach is valid for both signature verification and data validation services.

5. CONCLUSIONS

[11] STORK (Security Across Borders Linked), https://www.eid-stork.eu/
[12] epSOS (Smart Open Services for European Patients), http://www.epsos.eu/

Secure and efficient processing of electronic documents and its affiliated meta data are crucial requirements for efficient and security-sensitive applications. Our described approach shows solutions which are capable to fulfill these requirements. We have introduced a mechanism which enables existing signature verification services to integrate external services supporting the verification of national and proprietary signature formats. This facilitates the dynamic enhancement of supported signature formats. In addition, we have highlighted the need for previous data validation and have presented a solution on validating meta and document data. Although we have presented our solutions on the basis of the OCD container format and its software modules, they are also applicable for several other use cases where electronic documents must be exchanged and processed. Anyhow, our presented approach contributes to more efficient, time saving, and cost reducing e-Government and e-Business applications.

Additionally, we have elaborated possibilities to make verification and validation services available via cloud computing. The movement of these services to the cloud allows for additional cost savings and enables higher scalability. The incorporation of encrypted OCD containers and documents in the presented approach as well as the definition of interoperable document data types are subjects to be addressed in our future work. This might also help bypassing security and privacy concerns with respect to cloud computing.

REFERENCES

Catteddu, D., and Hogben, G. (2009). *Cloud Computing - Benefits, risks and recommendations for information security*. ENISA

Cloud Security Alliance. (2011). SECURITY GUIDANCE FOR CRITICAL AREAS OF FOCUS IN CLOUD COMPUTING V3.0.

Director General of the Lithuanian Archives Department, 2009. *Specification ADOC-V1.0 of the electronic document signed by the electronic signature*. Valstybės žinios 108-4574, https://signa.mitsoft.lt/static/signa-web/webResources/docs/ADOC_specification_approved20090907_EN.pdf

ETSI, 2009. *ETSI TS 101 903, XML Advanced Electronic Signatures (XAdES)*. Version 1.4.1.

ETSI, 2010. *ETSI TS 102 778-3. Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 3: PAdES Enhanced - PAdES-BES and PAdES-EPES Profiles*. Version 1.2.1.

ETSI, 2012. *ETSI TS 102 918, Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC)*. Version 1.2.1.

European Commission, 2011. *European Commission Decision on Establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market, notified under document C(2011) 1081. 2011/130/EU*. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:053:0066:0072:EN:PDF.

European Union, 2006. *Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market*. Official Journal of the European Union. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:376:0036:0068:en:PDF.

Mell, P., and Grance, T. (2010). The NIST definition of cloud computing. NIST. http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf

Leitold H. et al, 2009, *Mediabreak resistant eSignatures in eGovernment - An Austrian experience*. Emerging Challenges for Security, Privacy, and Trust - 24th IFIP SEC. Volume IFIP AICT 297 of IFIP Advances in Information and Communication Technologies, pp. 109-118.

Rössler T., Tauber A., 2010. *The SPOCS Interoperability Framework: Interoperability of eDocuments and eDelivery Systems taken as Example*. ISSE 2010 Securing Electronic Business Processes, pp. 122-130.

SPOCS, 2011. *SPOCS Deliverable D2.2 - Standard Document and Validation Common Specifications*. Version 1.4.0. http://joinup.ec.europa.eu/site/spocs/eDocuments/references/D2.2_Standard_document_and_validation_common_specifications.zip

Stranacher K., Kawecki T., 2012. *Interoperable Electronic Documents*. Electronic Government and Electronic Participation - Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart 2012, Kristiansand, Norway, pp. 81-88.

Zefferer T. et al, 2011, *Secure and Reliable Online-Verification of Electronic Signatures in the Digital Age*, Proceedings of the IADIS International Conference WWW/INTERNET 2011, pp. 269-276.

Zissis, D., and Lekkas, D. (2012). *Addressing cloud computing security issues*. Future Generation Computer Systems, 28(3), 583–592.

Part V

Publications and Outlook
The Catrobat Project

17 | Purely Visual and Hybrid Visual/Textual Formula Composition: A Usability Study Plan

Conference	Mobile and Touch PRoMoTo 2013
Language	English
Title	Purely Visual and Hybrid Visual/Textual Formula Composition: A Usability Study Plan
Authors	Annemarie Harzl, Vesna Krnjic, Franz Schreiner, Wolfgang Slany
Booktitle	Proceedings of Programming for Mobile and Touch PRoMoTo 2013

Purely Visual and Hybrid Visual/Textual Formula Composition: A Usability Study Plan

Annemarie Harzl, Vesna Krnjic, Franz Schreiner, and Wolfgang Slany
Graz University of Technology
aharzl@ist.tugraz.at, vesna.krnjic@iaik.tugraz.at, franz.schreiner@student.tugraz.at, wolfgang.slany@tugraz.at

Abstract

Only very few end users have the skills to develop mobile apps such as games or animations. Visual programming languages can be very supportive for casual and first-time users, allowing the users to concentrate on the programming task rather than learning complex syntax. This is why visual programming languages are often used where children are concerned. Nevertheless, studies have shown that the advantage of visual languages tends to decrease on larger tasks or mathematical formulas. The paper distinguishes different approaches for creating formulas with end user programming languages, namely purely textual, purely visual, and hybrid approaches. In our paper we introduce Pocket Code, a new approach with visual programming and hybrid formula editing, which combines the easiness of visual programming with the effectiveness and clarity of textual formula displaying. Additionally, we present a proposal of an evaluation of the different approaches to formula manipulation in visual programming languages for smartphones.

1. Introduction

Visual programming languages (VPL) are important for end user programming. They empower end users with little, no, or only casual programming experience to develop programs for their individual use. The ability to write applications, to create one's own games, or to automate small tasks, can be easily learned. Due to the features of VPL they are often used when it comes to children. Especially for younger children it seems to be easier to drag & drop bricks together like in Scratch[1] than to learn a textual programming language. They can learn the basic principles of programming without bothering with the sometimes restrictive syntax of a textual programming language. There are other important benefits of visual languages as well, e.g., being able to see what command blocks are available and thus might be employed, thereby suggesting their use without the user having to know or even remember them. Regarding programming and the writing of formulas there are two clearly distinct approaches: the textual and the visual approach. With traditional textual programming languages like C or Java, developers enter statements with a standard keyboard, though modern IDEs to some degree simplify the entering of statements through context sensitive statement completion. Visual languages like Scratch use the visual approach, where even formulas are created with pre-defined graphical blocks. When complex formulas are involved, this latter approach can become unwieldy and even confusing. Some programming languages like TouchDevelop[2], which is mainly textual, pursue an approach where code and formulas are entered via an interface that reminds of pocket calculators: Statements and operators are chosen from a set of visually differentiated alternatives, but the actual visualization of the resulting statements and formulas is done in a purely textual way. In our paper we introduce Pocket Code[3], a new approach with visual programming and textual formula representation, which combines the easiness of visual programming with the effectiveness and clarity of textual formula displaying. One of the reasons for introducing this new combination is that the presented visual programming language is optimized for the use on smartphones with their touch screens and small display sizes, where Scratch-like blocks cannot easily be accommodated due to the narrowness of the screens and the difficulty to drag and drop blocks closely nested together with one's fingers compared to when using a mouse pointer. Furthermore we will discuss different programming language approaches for editing and manipulating formulas and present a proposal how these approaches could be compared and evaluated regarding their efficiency, effectiveness, and user satisfaction.

2. Related Work

User studies have shown mixed results on the superiority of visual languages over text languages. However most work has focused on the desktop so far. For instance one empirical study [6], which has been done comparing constructability of programs in textual versus visual languages, concludes that matrix manipulation programs can be more easily constructed in a visual language (e.g., Formd3) than in a textual language (e.g., OSU-APL and Pascal). Another empirical study [1] concludes that visual languages provide a better user experience, reduce perceived workload, and increase perceived success. On the other hand, Green et al. [4] show that dataflow visual programming languages are not consistently superior to text languages. Their study shows that some visual notations, for example the gate notation are, in fact, worse than equivalent textual notations. Neither textual languages nor visual languages are perfect and are both perceived as not supportive or confusing by end users without (much) programming experience. However the comparison of the two approaches needs to be considered for smart phones and small displays as well.

[1] http://scratch.mit.edu/

[2] https://www.touchdevelop.com/

[3] http://www.pocketcode.org/

Promoto '13, October 26, 2013, Indianapolis, Indiana, USA.

http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn

Figure 1. Visual formula editing in Scratch.

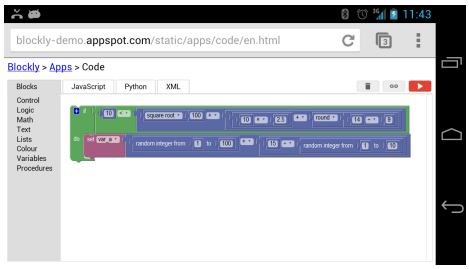


Figure 2. Blockly in landscape mode on a mobile browser.

When creating formulas with end user programming languages there are three main approaches:

1. The purely textual approach like in Microsoft's Excel, where formulas are created and displayed textually. Excel is a spreadsheet application, designed mainly for adults and the use on traditional computers, where users can calculate values with formulas entered in Excel.
2. The purely visual approach, like in Scratch [7], Snap![4] [3], and Blockly[5] where pre-defined visual segments are used to create and to display a formula.
3. The "hybrid" approach like in Microsoft's TouchDevelop and Pocket Code's formula editor, where formulas are created using visual elements similar to a pocket calculator, but are displayed textually.

In the course of our paper we will focus on the latter two approaches.

Scratch was designed for children and the use on traditional computers. Snap!, an extension to Scratch, was designed for children and adults and the use on traditional computers (large screen, keyboard, mouse), but works on smartphones as well. In addition to Scratch's approach (see Figure 1) Snap! highlights different nesting levels of formulas in a so called "zebra"-mode: Parts of the formula are alternatingly lighter and darker colored (see Figure 3).

Blockly[6] was also mainly designed for adults and the use on traditional computers (large screen, keyboard, mouse), but works on smartphones as well. It allows to switch seamlessly between a purely visual approach similar to Scratch to purely textual ones (alternatively JavaScript, Python, and XML) and back (see Figure 2).

TouchDevelop is an application creation environment intended particularly for students or adult hobbyist programmers. It is intended to be used primarily on smartphones. The programming language is text-based but uses a few non-ASCII graphical characters

[4] `http://snap.berkeley.edu/`

[5] `https://code.google.com/p/blockly/`

[6] `http://blockly-demo.appspot.com/static/apps/code/en.html`

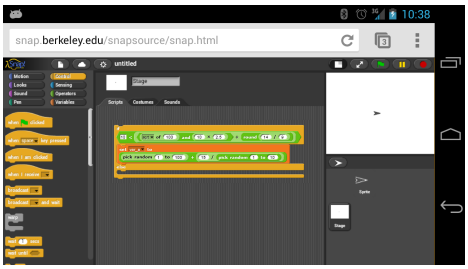


Figure 3. Visual formula editing in Snap! in landscape mode on a mobile browser.

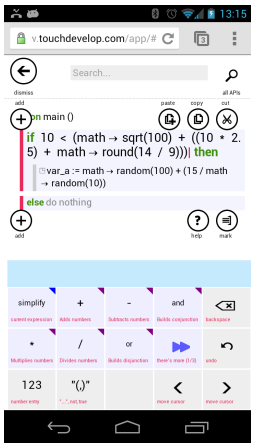


Figure 4. Textual formula editing in TouchDevelop.

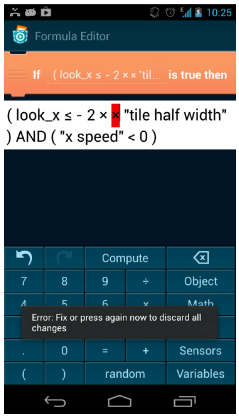
for representation of the syntax, for example arrows, a recycling symbol, etc. It resembles a traditional text-based programming language, though with a specialized editor (see Figure 4) and use of annotation of program text and automated re-formating. Formulas are entered visually but displayed textually, similar to Pocket Code's formula editor (see Figure 5(a)).

3. Pocket Code

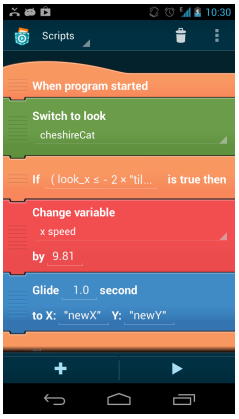
Pocket Code is a free and open source mobile visual programming system for the Catrobat language[7]. It allows users, starting from the age of eight, to develop games and animations with their smartphones. To program, the children use their Android phone, iPhone, Windows Phone, or other smartphone with an HTML5 browser. No notebook or desktop computer is needed [8]. Pocket Code is inspired by, but distinct from, the Scratch programming system developed by the Lifelong Kindergarten Group at the MIT Media Lab [7]. Similar to Scratch, our aim is to enable children and teenagers to creatively develop and share their own software. The main differences of Pocket Code in contrast to Scratch are:

1. Support and integration of multi-touch mobile devices

[7] `http://catrobat.org/`



(a) In this particular example the user entered a syntactically incorrect formula (two multiplication signs one after the other). After pressing the BACK-button an error message appears and the syntax error gets highlighted in red.



(b) Visual script of a program written using Pocket Code that contains the If-brick with the formula that was edited in Figure 5(a).

Figure 5. Textual formula editing in Pocket Code.

2. Use of mobile device's special hardware (e.g., acceleration, compass, inclination)
3. No need for a traditional PC

Additionally, there are more than 30 ongoing sub-projects mostly aiming at extending Pocket Code's functionality, e.g., a 2D physics engine that will make the programming of games similar to the popular Angry Birds type of games very easy, or an extension allowing to very easily record the screen as well as sound during execution of a program and to upload it to an online video sharing site, the high definition video being created on our server and uploaded from there to avoid high costs and lengthy file transmissions for the kids.

In the following we give a succinct overview of the design intentions of Pocket Code's formula editor:

We wanted to show the user which statements operators, variables, messages are possible, thus simplifying discoverability for the user. It should be easy to edit on a small touch screen ("use only one's thumb to enter a whole program or formula"). Text based formulas are easier to display on narrow screen (text wrapping) and at the same time well known from typical pocket calculators, calculator apps, but also from spreadsheets and even from math classes at school. It should be easy to get a preview of the current value of a formula through a "compute" button. It should provide users with an overview over current variable values. The formula editor should make it easy to position the cursor at any place in the formula, and to select larger parts of a formula. It should visualize "matching" parentheses in complicated nested expressions. The formula editor should also eliminate some syntax errors preemptively. Other syntax errors when users try to use an "unfinished" formula (see Figure 5(a)) should be highlighted. Scrolling of long formulas should be possible. Copy/cut/paste of parts of formulas should be possible. Easy undo/redo should make it easier to develop a formula.

4. Formula manipulation approaches under study

In this section we discuss two main ways of editing and displaying formulas (the third main approach, textual creation and textual visualization of formulas, was mentioned in Section 2, but will not be discussed here):

1. Visual creation and visual representation of formulas like in Scratch and Blockly. Scratch (see Figure 1) and other visual programming languages use purely visual formula editing. Different segments have to be nested within each other to compose a formula.
2. Visual creation and textual representation of formulas, like in TouchDevelop and Pocket Code. In TouchDevelop's and Pocket Code's formula editors, formulas are created via a pocket calculator-like interface (see Figures 4 and 5(a)). Statements and operators are selected visually, but the actual formula representation is purely textual. This should help save screen space and provide users with a better overview over the formula.

Visual composition of formulas can become a tedious task, because numerous visual components have to be nested within each other for more complex formulas. This is especially true for the small screens of smartphones. The screen limitations of mobile phones and the common knowledge of how to use a calculator led to our decision to display formulas textually in Pocket Code's formula editor. Most teenage and adult users know how to operate a pocket calculator and should therefore experience no problems with Pocket Code's formula editor. For smaller children, who did not use a calculator before, future usability studies will determine whether the textual or visual approach works better for them. Displaying formulas textually may be faster and easier to understand for users who are familiar with pocket calculators. In order to evaluate the two different approaches we will conduct a formal experiment that is described in the following section.

5. Evaluation of Pocket Code's formula editor

To assess the usability of Pocket Code's formula editor we followed the main objectives of User Centered Design (UCD) methods defined by ISO 9241-210:2010[8] including user research, interface design, and usability testing during the implementation cycles. According to the agile principles used by the software development team, the applied UCD methods followed the agile methods as well, such as inspection, heuristic evaluation, paper mockups, and thinking aloud tests [2]. Additionally to previously done formative testing we are planning to conduct a formal experiment in order to gain a summative assessment of the formula editor. This section describes the methodology that will be applied to evaluate the usability of Pocket Code's formula editor and compare it with three different programming language approaches.

5.1 Methodology

The purpose of the planned experiment is to provide scientific evidence to support or revoke the assertions described below. The following hypotheses are stated:

1. Null hypothesis: For the manipulation of complex formulas, the calculator metaphor (the hybrid textual/visual approach) is more effective and efficient than the pure visual programming language approach.
2. Alternative hypothesis: The contrary of above null hypothesis: For the manipulation of complex formulas, the calculator

[8] http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075

metaphor (the hybrid textual/visual approach) is as good or less effective and efficient than the pure visual programming language approach.

A complex formula in this context will be clearly defined, for example something like a logical formula composed of nested expressions at least 4 levels deep, with 12 parentheses, 3 variables, 1 sensor value, 6 constants, 4 logical operators, 6 numerical constants, and 8 operators. Users will be allowed to use the phones in portrait and landscape modus. We will also experiment with different screen sizes and resolutions. Before running the experiment a pilot test will be conducted to discover errors and to obtain extra practice for the research team [9]. We will evaluate several aspects during the pilot test, like the reactions of participants, discovery of errors in the test setup, and the procedure for data processing and analysis. After the pilot test and resulting adaptations to the test procedure, the real test with the null hypothesis presented before will be conducted. The following subsections describe the experimental design and specify the test metrics.

5.2 Research Questions

The aim of the study is to compare four different programming language environments described in Section 2. We want to find out what type of formula creation and visualization works faster, is better understandable, and preferred by the participants, while using and manipulating complex formulas. A purely visual programming language like Snap! and Blockly, or a hybrid programming language like TouchDevelop and Pocket Code's formula editor. To answer these questions, we will conduct a comprehensive formal experiment. Details of the planned test method and the experimental setup are provided in the following section.

5.3 Experimental Design

To evaluate the hybrid programming environment approach of formula editing, we are going to conduct a counterbalanced formal experiment with repeated measures. The experiment will take place in a laboratory at Graz University of Technology. We will use the same test setup for all of the tests. For the evaluation, we will randomly select 32 participants at the age of 16, because Pocket Code is specifically targeted at teenagers. For other age groups, the programming system will be adapted to their needs. The participants will be recruited from schools in and around Graz. None of them will have any previous programming experience. The participants will be randomly distributed in four groups. In either case, the participants will spend two hours in the experiment, first learning the basics of the system from a tutorial, and then trying to accomplish a series of tasks. First, they will be asked to create a very simple program in order to get familiar with the programming environment. The order of the tasks will be counterbalanced (each group works with the different programming environments in a different order) between the groups to avoid learning bias [5]. After each task the participants will be asked to fill out a feedback questionnaire for the purpose of collecting subjective qualitative data. The dependent variables that will be measured are

1. time spent for solving each task, while using different programming environments
2. successfully finished tasks
3. tasks finished with help and
4. the number of errors occurred
5. quality of the programs created by the participants, rated by the test team.

Different applications, tasks, and time are going to be the dependent variables. In addition to the quantitative data, we will collect qualitative data as well. After each task, we are going to initiate a discussion with the participants and try to get as much information as possible about their subjective experiences, information about what they liked and disliked, and what would have made the programming language more compelling, more useful, or easier to use.

The data will be collected from three different sources. The most relevant data will be compiled during the task execution. After each task, users will fill out a feedback questionnaire to get some subjective feedback from the participants.

6. Outlook

We plan to experiment with even more hybrid constructionist interfaces for formula composition that, in addition to the way described in the present paper, allow to select visual blocks for operators and other formula elements in Pocket Code, in exactly the same way other statements can be selected, therefore providing easy discoverability to the user. Tapping on such a block will initiate a short animation that visualizes the way the corresponding formula element can be used in the formula editor. This animation can be skipped and also turned off permanently by the user.

Acknowledgments

Many thanks to the Catrobat team members[9].

References

[1] T. Booth and S. Stumpf. End-user experiences of visual and textual programming environments for Arduino. In Y. Dittrich, M. Burnett, A. Mrch, and D. Redmiles, editors, *End-User Development*, volume 7897 of *Lecture Notes in Computer Science*, pages 25–39. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38705-0. . URL `http://dx.doi.org/10.1007/978-3-642-38706-7_4`.

[2] D. Brown. *Agile User Experience Design: A Practitioner's Guide to Making It Work*. Elsevier Science, 2012. ISBN 9780123914095.

[3] D. Garcia, L. Segars, and J. Paley. Snap! (build your own blocks): tutorial presentation. *J. Comput. Sci. Coll.*, 27(4):120–121, Apr. 2012. ISSN 1937-4771.

[4] T. Green and M. Petre. When visual programs are harder to read than textual programs. In *Proceedings of the Sixth European Conference on Cognitive Ergonomics (ECCE 6)*, pages 167–180, 1992.

[5] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, and K. E. Emam. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28:721–734, 2002.

[6] R. K. Pandey and M. M. Burnett. Is it easier to write matrix manipulation programs visually or textually? An empirical study. In *IEEE SYMP. VISUAL LANGUAGES*, pages 24–27, 1993.

[7] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. S. Silver, B. Silverman, and Y. B. Kafai. Scratch: programming for all. *Commun. ACM*, 52(11): 60–67, 2009.

[8] W. Slany. A mobile visual programming system for Android smartphones and tablets. In M. Erwig, G. Stapleton, and G. Costagliola, editors, *VL/HCC*, pages 265–266. IEEE, 2012. ISBN 978-1-4673-0852-6.

[9] E. van Teijlingen and V. Hundley. The importance of pilot studies. *Nurs Stand*, 16(40):33–6, 2002.

[9] `http://catrob.at/credits`

18 | Comparing Purely Visual with Hybrid Visual/Textual Manipulation of Complex Formula on Smartphones

Conference	DMS 2013
Language	English
Title	Comparing Purely Visual with Hybrid Visual/Textual Manipulation of Complex Formula on Smartphones
Authors	Annemarie Harzl, Vesna Krnjic, Franz Schreiner, Wolfgang Slany
Booktitle	DMS 2013

Comparing Purely Visual with Hybrid Visual/Textual Manipulation of Complex Formula on Smartphones

Annemarie Harzl, Vesna Krnjic, Franz Schreiner, and Wolfgang Slany
Graz University of Technology
aharzl@ist.tugraz.at, vesna.krnjic@iaik.tugraz.at, franz.schreiner@student.tugraz.at, wolfgang.slany@tugraz.at

***Abstract*—Only very few end users have the skills to develop mobile apps such as games or animations. Visual programming languages can be very supportive for casual and first-time users, allowing the users to concentrate on the programming task rather than learning complex syntax. This is why visual programming languages are often used where children are concerned. Nevertheless, studies have shown that the advantage of visual languages tends to decrease on larger tasks or mathematical formulas. The paper distinguishes different approaches for creating formulas with end user programming languages, namely purely textual, purely visual, and hybrid approaches. In our paper we introduce Pocket Code, a new approach with visual programming and hybrid formula editing, which combines the easiness of visual programming with the effectiveness and clarity of textual formula displaying. Additionally, we present a proposal of an evaluation of the different approaches to formula manipulation in visual programming languages for smartphones.**

I. Introduction

Visual programming languages (VPL) are important for end user programming. It empowers end users with little, no, or only casual programming experience to develop programs for their individual use. The ability to write applications, to create one's own games, or to automate small tasks can be easily learned. Due to the features of VPL they are often used when it comes to children. Especially for younger children it seems to be easier to drag & drop bricks together like in Scratch[1] than to learn a textual programming language. They can learn the basic principles of programming without bothering with the sometimes restrictive syntax of a textual programming language. There are other important benefits of visual languages as well, e.g., being able to see what command blocks are available and thus might be employed, thereby suggesting their use without the user having to know or even remember them. Regarding programming and the writing of formulas there are two clearly distinct approaches: the textual and the visual approach. With traditional textual programming languages like C or Java, developers enter statements on a standard keyboard, though modern IDEs to some degree simplify the entering of statements through context sensitive statement completion. Visual languages like Scratch use the visual approach, where even formulas are created with pre-defined graphical blocks. When complex formulas are involved, this latter approach can become unwieldy and even confusing. Some programming languages like TouchDevelop[2], which is mainly textual, pursue an approach where code and formulas are entered via an interface that reminds of pocket calculators: Statements and operators are chosen from a set of visually differentiated alternatives, but the actual visualization of the resulting statements and formulas is done in a purely textual way. In our paper we introduce Pocket Code[3], a new approach with visual programming, but textual formula representation, which combines the easiness of visual programming with the effectiveness and clarity of textual formula displaying. One of the reasons for introducing this new combination is that the presented visual programming language is optimized for the use on smartphones with their touch screens and small display sizes, where Scratch-like blocks cannot easily be accommodated due to the narrowness of the screens and the difficulty to drag and drop blocks closely nested together with one's fingers compared to when using a mouse pointer. Furthermore we will discuss different programming language approaches for editing and manipulating formulas and present a proposal how these approaches could be compared and evaluated regarding their efficiency, effectiveness, and user satisfaction.

II. Related Work

Green et al. [1] argue that dataflow visual programming languages are not consistently superior to text languages. Their study shows that some visual notations, for example the gate notation, are in fact worse than equivalent textual notations.

When creating formulas with end user programming languages there are three main approaches:

- The purely textual approach like in Microsoft's Excel, where formulas are created and displayed textually. Excel is a spreadsheet application, designed mainly for adults and the use on traditional computers, where users can calculate values with formulas entered in Excel.
- The purely visual approach, like in Scratch [2], Snap![4] [3], and Blockly[5] where pre-defined visual segments are used to create and to display a formula.
- The "hybrid" approach like in Microsoft's TouchDevelop and Pocket Code's formula editor, where formulas are created using visual elements similar to a pocket calculator, but are displayed textually.

In the course of our paper we will focus on the latter two approaches. Table I gives an overview of different approaches for formula creation and visualization in different programming environments.

[1] http://scratch.mit.edu/
[2] https://www.touchdevelop.com/
[3] http://www.pocketcode.org/
[4] http://snap.berkeley.edu/
[5] https://code.google.com/p/blockly/

TABLE I. CHARACTERISTICS OF CONSIDERED PROGRAMMING SYSTEM ALLOWING TO CREATE FORMULAS

Programming system	formula creation	main target group	screen and input
Excel	textual	adults	large screen, mouse, keyboard
Scratch	visual	children	large screen, mouse, keyboard
Snap!	visual	children & adults	large screen (but also mobile browser), mouse, keyboard
TouchDevelop	hybrid	adults	smartphone, touch screen
Pocket Code	hybrid	children	smartphone, touch screen
Blockly	textual or visual	adults	large screen (but also mobile browser), mouse, keyboard

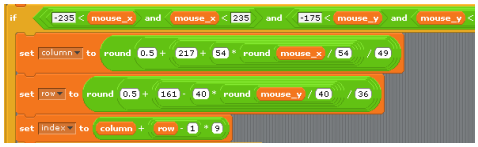


Fig. 1. Visual formula editing in Scratch

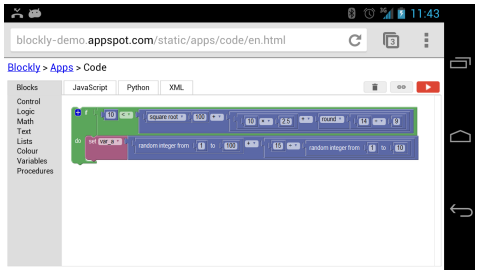


Fig. 2. Blockly in landscape mode.

Snap! is an extension to Scratch, was designed for traditional computers (large screen, keyboard, mouse), but works on smartphones as well. In addition to Scratch's approach (see Figure 1) Snap! highlights different nesting levels of formulas in a so called "zebra"-mode: Parts of the formula are alternatingly lighter and darker colored (see Figure 3).

Blockly[6] was also mainly designed for traditional computers (large screen, keyboard, mouse), but works on smartphones as well. It allows to switch seamlessly between a purely visual approach similar to Scratch to purely textual ones (alternatively JavaScript, Python, and XML) and back (see Figure 2).

TouchDevelop is an application creation environment intended particularly for students or hobbyist programmers. It is intended to be used primarily on smartphones. The programming language is text-based but uses a few non-ASCII graphical characters for representation of the syntax, for example arrows, recycling symbol, etc. It resembles a traditional text-based programming language, though with a specialized editor (see Figure 4) and use of annotation of program text and automated re-formating. Formulas are entered visually but displayed textually, similar to Pocket Code's formula editor (see Figure 5).

[6]http://blockly-demo.appspot.com/static/apps/code/en.html

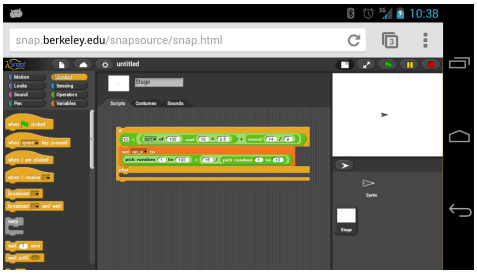


Fig. 3. Visual formula editing in Snap! in landscape mode.

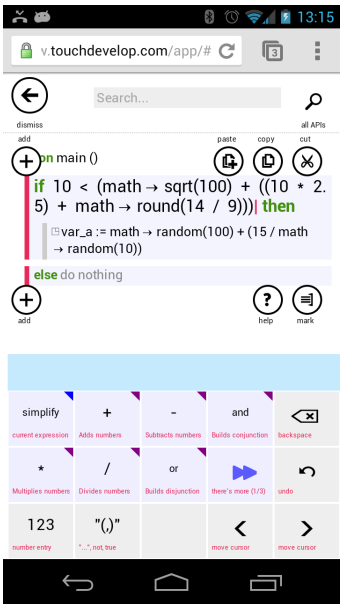


Fig. 4. Textual formula editing in TouchDevelop.

III. POCKET CODE

Pocket Code is a free and open source mobile visual programming system for the Catrobat language[7]. It allows

[7]http://catrobat.org/

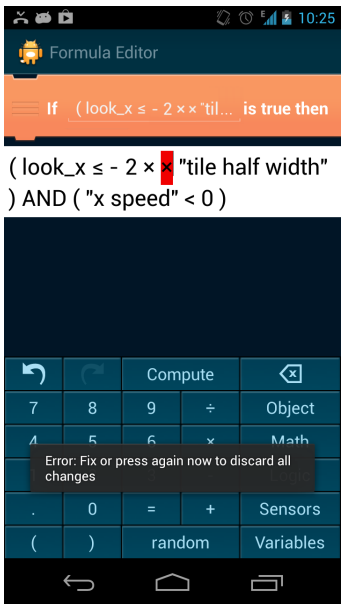


Fig. 5. Textual formula editing in Pocket Code. In this particular example the user entered a syntactically wrong formula (two multiplication signs one after the other). After pressing the BACK-button an error message appears and the syntax error gets highlighted in red.

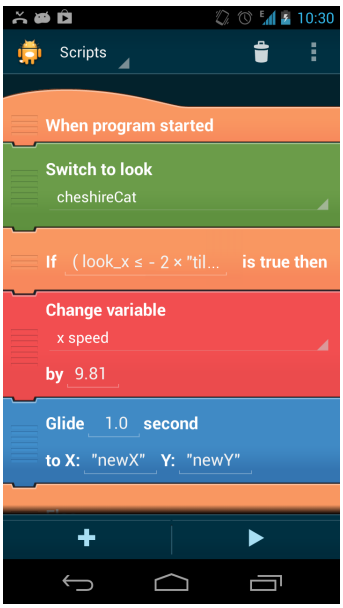


Fig. 6. Visual script of a program written using Pocket Code that contains the If-brick with the formula that was edited in Figure 5.

users, starting from the age of eight, to develop games and animations with their smartphones. To program, the children use their Android phone, iPhone, Windows Phone, or other smartphone with an HTML5 browser. No notebook or desktop computer is needed [5].

Pocket Code is inspired by, but distinct from, the Scratch programming system developed by the Lifelong Kindergarten Group at the MIT Media Lab [2]. Similar to Scratch, our aim is to enable children and teenagers to creatively develop and share their own software. The main differences of Pocket Code in contrast to Scratch are:

1) Support and integration of multi-touch mobile devices
2) Use of mobile device's special hardware (e.g., acceleration, compass, inclination)
3) No need for a traditional PC

Additionally, there are more than 30 ongoing subprojects mostly aiming at extending Pocket Code's functionality, e.g., a 2D physics engine that will make the programming of games similar to the popular Angry Birds type of games very easy, or an extension allowing to very easily record the screen as well as sound during execution of a program and to upload it to YouTube, the high definition video being created on our server and uploaded from there to avoid high costs and lengthy file transmissions for the kids.

In the following we give a succinct overview of the design intentions of Pocket Code's formula editor:

- Show the user which statements operators, variables, messages are possible, thus simplifying discoverability for the user.
- Make it easy to edit on a small touch screen ("use only one's thumb to enter a whole program or formula").
- Text based formula: easier to display on narrow screen (text wrapping) and at the same time well known from typical pocket calculators, calculator apps, but also from spreadsheets and even from math classes at school.
- Easy to get a preview of current value of a formula through the "compute" button.
- Overview over current variable values.
- Make it easy to position the cursor at any place in the formula, and to select larger parts of a formula.
- Visualize "matching" parentheses in complicated nested expressions.
- Eliminate some syntax errors preemptively.
- Highlight other syntax errors when user tries to use an "unfinished" formula (see Figure 5).
- Scrolling of long formulas.

- Allow copy/cut/paste of parts of formulas.
- Easy undo/redo.

IV. Formula manipulation approaches under study

In this section we discuss two main ways of editing and displaying formulas (the third main approach, textual creation and textual visualization of formulas, was mentioned in Section II, but will not be discussed here):

1) Visual creation and visual representation of formulas like in Scratch and Blockly. Scratch (see Figure 1) and other visual programming languages use purely visual formula editing. Different segments have to be nested within each other to compose a formula.
2) Visual creation and textual representation of formulas, like in TouchDevelop and Pocket Code. In TouchDevelop and Pocket Code's formula editor, formulas are created via a pocket calculator-like interface (see Figures 4 and 5). Statements and operators are selected visually, but the actual formula representation is purely textual. This should help save screen space and provide users with a better overview over their formula.

Visual composition of formulas can become a tedious task, because numerous visual components have to be nested within each other for more complex formulas. This is especially true for the small screens of smartphones. The screen limitations of mobile phones and the common knowledge of how to use a calculator led to our decision to display formulas textually in Pocket Code's formula editor. Most teenage and adult users know how to operate a pocket calculator and should therefore experience no problems with Pocket Code's formula editor. For smaller children, who did not use a calculator before, future usability studies will determine whether the textual or visual approach works better for them. Displaying formulas textually may be faster and easier understandable for users who are familiar with pocket calculators. In order to evaluate the two different approaches we will conduct a formal experiment that is described in the following section.

V. Evaluation of Pocket Code's formula editor

To assess the usability of Pocket Code's formula editor we followed the main objectives of User Centered Design (UCD) methods defined by ISO 9241-210:2010[8] including user research, interface design, and usability testing during the implementation cycles. According to the agile principles used by the software development team, the applied UCD methods followed the agile methods as well, such as inspection, heuristic evaluation, paper mockups, and thinking aloud tests [6]. Additionally to previously done formative testing we are planning to conduct a formal experiment in order to gain a summative assessment of the formula editor. This section describes the methodology that will be applied to evaluate the usability of Pocket Code's formula editor and compare it with three different programming language approaches.

[8]http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075

A. Methodology

The purpose of the planned experiment is to provide scientific evidence to support or revoke the assertions described below. The following hypotheses are stated:

- Null hypothesis: For the manipulation of complex formulas, the calculator metaphor (the hybrid textual/visual approach) is more effective and efficient than the pure visual programming language approach.
- Alternative hypothesis: The contrary of above null hypothesis: For the manipulation of complex formulas, the calculator metaphor (the hybrid textual/visual approach) is as good or less effective and efficient than the pure visual programming language approach.

A complex formula in this context will be clearly defined, for example something like a logical formula composed of nested expressions at least 4 levels deep, with 12 parentheses, 3 variables, 1 sensor value, 6 constants, 4 logical operators, 6 numerical constants, and 8 operators.

Users will be allowed to use the phones in portrait and landscape modus. We will also experiment with different screen sizes and resolutions.

Before running the experiment a pilot test will be conducted to discover errors and to obtain extra practice for the research team [7]. We will evaluate several aspects during the pilot test, like the reactions of participants, discovery of errors in the test setup, and the procedure for data processing and analysis.

After the pilot test and resulting adaptations to the test procedure, the real test with the null hypothesis presented before will be conducted. The following subsection describes the experimental design and specifies the test metrics.

B. Research Questions

The aim of the study is to compare four different programming language environments described in Section II. We want to find out what type of formula creation and visualization works faster, is better understandable, and preferred by the participants, while using and manipulating complex formulas: a purely visual programming language like Snap! and Blockly, or a hybrid programming language like TouchDevelop and Pocket Code's formula editor. To answer these questions, we will conduct a comprehensive formal experiment. Details of the planned test method and the experimental setup are provided in the following section.

C. Experimental Design

To evaluate the hybrid programming environment approach of formula editing, we are going to conduct a counterbalanced formal experiment with repeated measures. For the evaluation, we will randomly select 32 participants at the age of 16. The participants will be recruited from schools in and around Graz. None of them will have any previous programming experience. The participants will be randomly distributed in four groups (A, B, C, and D). In either case, the participants will spend two hours in the experiment, first learning the basics of the system from a tutorial, and then trying to accomplish

TABLE II. TASKS AND GROUPS – COUNTERBALANCED FORMAL EXPERIMENT

	Pocket Code	TouchDevelop	Snap!	Blockly
A	1	2	3	4
B	4	1	2	3
C	3	4	1	2
D	2	3	4	1

a series of tasks. First, they will be asked to create a very simple program in order to get familiar with the programming environment. The order of the tasks will be counterbalanced (see Table II) between the groups to avoid learning bias [8]. After each task the participants will be asked to fill out a feedback questionnaire for the purpose of collecting subjective qualitative data. The dependent variables that will be measured are

1) time spent for solving each task, while using different programming environments
2) successfully finished tasks
3) tasks finished with help and
4) the number of errors occurred
5) quality of the programs created by the participants, rated by the test team.

Different applications, tasks, and time are going to be the dependent variables. In addition to the quantitative data, we will collect qualitative data as well. After each task, we are going to initiate a discussion with the participants and try to get as much information as possible about their subjective experiences, information about what they liked and disliked, and what would have made the programming language more compelling, more useful, or easier to use.

D. Tasks

We are going to define tasks in such a way that the answers to the predefined research questions can be derived easily from the collected data. All participants will be asked to perform the following tasks:

T1 Programming some very simple and introductory task to get familiar with the environment. During the first task participants will not have to use formulas.
T2 Extending an existing formula of a program.
T3 Writing some new formulas with the programming environment.

E. Experimental Setup

The experiment will take place in a laboratory at Graz University of Technology. We will use the same test setup for all of the tests, only the smartphone for TouchDevelop will be a Windows Phone. All other programming environments will run on Android devices. Hardware:

- Windows Phone or Android smartphone with maximum screen size of 5 inches.
- Laptop
- Video camera

The smartphone will be connected to the laptop, where the screen will be mirrored for the facilitator and captured with Morae[9], a usability software, for later reexamination of the tests. The laptop camera will record the user's face during the test to get the participants' facial expressions. Additionally an external camera will record the interviews and the whole test.

F. Data Collection

The data will be collected from three different sources. The most relevant data will be compiled during the task execution. After each task, users will fill out a feedback questionnaire to get some subjective feedback from the participants.

ACKNOWLEDGMENT

Many thanks to the Catrobat team members[10].

REFERENCES

[1] T. Green and M. Petre, "When visual programs are harder to read than textual programs," in *In*, 1992, pp. 167–180.

[2] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. S. Silver, B. Silverman, and Y. B. Kafai, "Scratch: programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[3] D. Garcia, L. Segars, and J. Paley, "Snap! (build your own blocks): tutorial presentation," *J. Comput. Sci. Coll.*, vol. 27, no. 4, pp. 120–121, Apr. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2167431.2167453

[4] B. Athreya, F. Bahmani, A. Diede, and C. Scaffidi, "End-user programmers on the loose: A study of programming on the phone for the phone," in *VL/HCC*, 2012, pp. 75–82.

[5] W. Slany, "A mobile visual programming system for android smartphones and tablets." in *VL/HCC*, M. Erwig, G. Stapleton, and G. Costagliola, Eds. IEEE, 2012, pp. 265–266. [Online]. Available: http://dblp.uni-trier.de/db/conf/vl/vlhcc2012.html#Slany12

[6] D. Brown, "Agile user experience design: A practitioners guide to making it work," in *Agile User Experience Design: A Practitioners Guide to Making It Work*. Elsevier Science, 2012.

[7] E. van Teijlingen and V. Hundley, "The importance of pilot studies." *Nurs Stand*, vol. 16, no. 40, pp. 33–6, 2002.

[8] B. A. Kitchenham, I. C. Society, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and I. C. Society, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on Software Engineering*, vol. 28, pp. 721–734, 2002.

[9] http://www.techsmith.com/morae.html
[10] http://catrob.at/credits

19 | Girls Create Games: Lessons Learned

Conference	ECGBL 2019 : 13th European Conference on Games Based Learning
Language	English
Title	Girls Create Games: Lessons Learned
Authors	Bernadette Spieler, Vesna Krnjic, Wolfgang Slany

ECGBL 2019, Odense, Denmark
Paper accepted, conference in October 2019

Girls Create Games: Lessons Learned

Bernadette Spieler, Vesna Krnjic, Wolfgang Slany
Graz University of Technology, Institute of Software Technology, Graz, Austria
bernadette.spieler@ist.tugraz.at
vesna.krnjic@ist.tugraz.at
wolfgang.slany@tugraz.at

Abstract: Recent studies from all over the world show that more boys than girls play video games. The numbers are different for mobile gaming apps, where 65% of women are identified as gamers. Adapting game design activities for academic purposes is a widely applied approach at schools or off-school initiatives, like CoderDojos or similar clubs, is seen as a promising opportunity for all teenagers to learn to code in an entertaining way. This raise the questions do special girls' game-design patterns exist, and what can we learn from them? This paper describes a girl-only intervention where girls were asked to create their own games. This "Girls' Coding Week" was designed as an off-school event and took place during summer 2018 with 13 girls between 11 to 14 years old. To explain the basic steps of programming and to create personalized games, the visual coding app Pocket Code, an app developed at Graz University of Technology, was used. The girls created their own games with the help of a storyboard after receiving all important information about coding (through unplugged coding activities, challenges, and a basic introduction to game design principles). Qualitative and quantitative data was collected through open interviews, as well as created artefacts and surveys which refer to motivational aspects. The findings show that gaming elements female teenagers tend to like, create, and play, mostly follow stereotypical expectations. In contrast to our experiences in heterogeneous course settings, this was not seen as something negative by girls. Furthermore, the findings provided evidence for game-making environments for girls. Subsequently, the results contributed to the development of new featured games to be used in our app to inspire female users around the world to code their own games. The authors argue that by understanding these differences in game design, we can support girls so that they become game designers and thereby more interested in coding.

Keywords: Game Design, Gendered Design, Design Thinking, Gender-inclusive GBL, Digital Artwork

1. Introduction

The lack of diversity in technology is a serious problem all over the world; many institutions, like the European Commission, governments, and general society recognize that this problem will affect future innovations. In engineering, manufacturing and construction-related fields, male graduates count for 72.3 percent (Baker, 2019). The "Bridging the Digital Gender Divide" report by OECD (2018) states that the gender gap is present already from an early stage and continues through university level. It is stated that girls at the age of 15 underperform boys in some ICT related skills, and only 0.5% of the girls want to pursue an ICT related career (compared to 5% of the boys). Jobs in software engineering are clearly male-dominated and women do not have a great impact on new technologies. Moreover, this means that they are not part of important decisions being made in the world of tech today and that funding will not be awarded to them to develop their ideas and concepts. Therefore, it should be in the interest of the whole computing world, rather than in the interest of any specific underrepresented group, to inspire young girls for coding (Ketelaars, 2017). In addition, there is a major lack of exposure to CS at schools all over Europe (CECE, 2017). Either schools do not offer any IT courses at all or in an inadequate amount or setting (at secondary schools it is mostly compulsory in one grade or not equally distributed over the grades), or it is an optional course.

In this paper, we first take a closer look at girl's games, gendered game design, and describe our learning tool Pocket Code. Furthermore, evidence for girls-only initiatives is summarized. Section 3 presents the research questions, the approach, and the method. In the results section, Section 4, the created artefacts (i.e., the gaming apps) are presented, as well as qualitative and quantitative data. To sum up, the discussion and conclusion sections are presented and an outlook is provided.

2. Literature Review

Reasons what prevent young women from choosing a career in ICT are diverse and there are general issues that must be addressed so that girls get engaged and motivated for those fields (Medel, and Pournaghshband,

2017). For example, stereotypes are present in CS and many students assume that a fanatical interest for computers and games is required in order to be successful in this field (Gabay-Egozi, Shavit, and Yaish, 2015). Inclusive environments are the key; we must consider alternative routes to IT and multiple points of entry (Frieze and Quesenberry, 2015). Playful coding initiatives designed especially for girls can support them in their decision to choose a computer science career (Zagami et al., 2015). This section presents popular games among girls, common design patterns, and subsequently, it provides arguments and characteristics of girls-only environments.

2.1 Girl's Games and Girl's Design

The company NewZoo (2017) published numbers of the video game industry, showing that 46% of gamers across these 13 countries are women (aged 10-65) who play on different consoles (35% PC, 48% mobile, 23% console). A percentage of 12% of women who play games are 10-20 years old. Female players hear about new games from friends/family (39%, men: 27%), social networks (20%, men 17%) or reviews/game sites and advertisements (18%, men 24-26%). Figure 1 shows the preferred genres per platform and gender.

Another statistic rated family/farm simulation and Match-3 games as the most preferred for female gamers (69% of those who are playing these genres are female) (Yee, 2017a). Match 3 games can be summarized as puzzle games where you mostly need to combine three tiles together, for example, Candy Crush Saga (Julkunen, 2015). These statistics emphasize that the genre averages range from 2% to 70%, thus developers should never focus on general statistics that consider all genders.

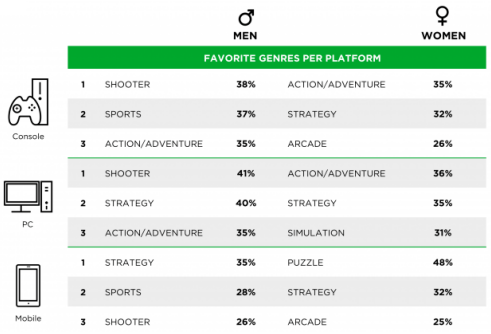


Figure 1: Statistics from (NewZoo, 2017) show women prefer mostly action/adventure genres

Ochsner (2015) collected design patterns from different girls initiatives from 1990-2007. These and literature from the same time (Agosto, 2003; Gorriz and Medina, 2000; Heeter et al., 2000) of typical design patterns, characteristics, and content girls tend to like are summarized in Figure 2.

Common design patterns	Common game characteristics	Content similarities
• exploration • collaboration • challenge • vicarious adventures • sophisticated graphics • and sound design • role-playing • realistic design	• rich narrative • roles involving positive action • appropriate levels of challenges • opportunities to design or create • engaging characters • communication and collaboration • use of strategies and skills	• storylines and character development • real-life locales • characters who are in charge of decisions and actions • to create rather than to destroy • involving simulation and identity play • chance to swap identities

Figure 2: Design, characteristics, and content among girl games.

The programming environment Scratch (https://scratch.mit.edu/) leverages game design and makes coding more accessible for a broader user group, especially novice programmers (Fields et al., 2014). A research study which observed game designs in Scratch (with a focus on racial and ethnic diversity) shows the following (Richard and Kafai., 2016): In general, most projects from female game designers focus on popular TV shows, games, or toys, or refer to mazes, dragons, and other pop culture creations and stories. Furthermore, an inspection of 52 Scratch games showed that female students used the most interactive objects that operated

through mouse clicks, keystrokes, animations, stories, and projects that included multiple genres, e.g., music with interactive objects (Kafai et al., 2012).

2.2 The Catrobat Project: Apps to Design Personal Games

For our coding workshops, we use the learning app Pocket Code (for Android: https://catrob.at/pc, for iOS: https://catrob.at/iosPC) or Luna&Cat (https://catrob.at/luna) to explain the basic steps of programming as well as to create games. The app uses a visual programming language very similar to the one in the Scratch environment but with Pocket Code, no laptop or PC are needed; only a smartphone. In addition, Pocket Code makes access of many sensors, for instance, inclination, GPS, compass direction, etc.), and has many extensions, for example, Lego NXT/EV3 robots, drone, Arduino, or programmable embroidery machines. These apps have been developed at Graz University of Technology (TU Graz) at the Institute of Software Technology as a FOSS Open Source project with the name Catrobat (https://catrobat.org).

2.3 Girls-only Interventions

Existing coding club initiatives like CoderDojos (https://coderdojo.com/foundation/) have predominantly male participation (Zagami et al., 2015). To promote initiatives for female teenagers as girls-only is therefore important. These initiatives serve as vehicles to interest girls more deeply in ICT, to foster their sense of belonging and self-efficacy (Thaler and Zorn, 2010). If such activities are promoted in schools, teachers have the conflict to provide similar activities for boys as well. Moreover, situations, where females are preferred to males, can lead to a range of negative impacts (stereotypes, threats, discrimination, etc.). During the last years, researchers have come up with numerous promising approaches to get girls encouraged with coding activities. Their findings are summarized in the following (El-Nasr et al., 2007; Sadler et al., 2012; Mann and Diprete, 2013; Wang, Eccles and Kenny, 2013; Giannakos et al., 2014; Stout and Camp, 2014; Unfried et al., 2015; Zagami et al., 2015; Alvarado et al. , 2017; Twentyman, 2018; Nichols, 2019):

- Improve girls self-efficacy, sense of belonging, interest, and engagement level in coding classes
- Encourage girls to create own projects which are presented in front of peers/others
- Raise girls' awareness of gender stereotypes in ICT
- Improve their expectations towards careers in programming
- Provide hands-on experiences and real-world examples
- Listen to girls suggestions about challenges and desires
- Focus on hands-on experience
- Provide early engagement (between 11 to 15) and create opportunities like extracurricular STEM activities for girls to build confidence in these areas
- Promote ICT careers and show positive role models and mentors they can both relate to and aspire to be
- Emphasize the creative aspects, idea creation, and design activities

Girls' initiatives create opportunities to focus on their interests and to enable them to socialize with other girls interested in computer science (Alvarado et al., 2017). Today, many coding clubs for girls exists, e.g., GirlsWhoCode (http://girlswhocode.com/) from the US, or Codefirst:Girls (https://www.codefirstgirls.org.uk/) from the UK.

Furthermore, facilitators and teachers report the difficulty to engage girls and boys equally in traditionally male-dominated subjects such as computing. Besides, in a school setting there are many constraints like time constraints. By default, CS in Austria is taught in two hours à 45 to 50 minutes weekly (Federal Ministry of Education, 2017). This means, that it is more difficult to convey a structuring concept over several weeks. Intensive off-school workshop weeks have the advantage that teenagers have an intensive learning period. Thus, coding initiatives for girls may improve girls's participation in such activities.

3. Method & Setting

The goals of the "Girls Coding Week" was twofold:

1. to provide girls with a basic set of knowledge of programming to increase their motivation, aspiration, and engagement for computer science topics (including those who do not play games), and

2. to let them design and create games in order to evaluate game design patterns used in their games to further improve girls' game design initiatives in future

For goal number 1, quantitative data was collected through questionnaires. The questionnaires (pre, daily, and post) handed out to the girls included measures of the factors regarding students' intrinsic motivators: interest, sense of belonging, self-efficacy, and engagement. The pre-questionnaires aimed to collect students' perceptions about the course, about coding, and technical fields. The daily questionnaires (handed out at the end of the first and the second double unit) had specific questions about the unit covering daily interest, fun, and achievement. Finally, a post-questionnaire asked questions about the coding week in general. For goal number 2, qualitative data was collected through interviews. Interviews were performed with all participants in groups of 2 to 4. In addition, the final gaming apps created during the course were evaluated to derive typical game design patterns from them.

The Girls Coding Week (GCW) was performed from 6th to 10th of August 2018 every day from 9 a.m. to 4 p.m. 13 girls between 11 to 14 participated in the GCW (average age: 12.8 years old). The course structure followed the PECC model (Spieler, 2018) - a gender-sensitive model that supports coding activities in the areas of Playing, Engagement, Creativity, and Coding. The workshop was conducted by two female computer science students, one female high school trainee, and the authors served as observers.

Day 1 started with an introduction game which had the goal to remember names of participants, build confidence/trust, and to get to know each other. Afterwards, participants formed groups of 4-5 members. They stayed in the same groups the whole week but they were allowed to switch groups (two did so). Each group had one facilitator allocated after each day's facilitator's switched groups. During the warm-up phase, each facilitator discussed in groups:

- Technical careers, expectations, what do programmers do, education, attributes, jobs
- What is programming, programming languages, experiences, interests
- Do you play games, apps, genres, what kind

As a result, a flipchart was created (made by the facilitator summarizing answers), which was presented and each one drew a picture of her expectation of a computer scientist. Impressions of the warm-up phase are illustrated in Figure 3.

Figure 3: Impressions of the warm-up phase of GWC.

For the first three days, the course continued with eleven units which referred to one topic, consisting of alternating activities: a) input session, b) unplugged coding activity, and c) coding session together, d) challenge is done by everyone on their own. As a programming tool, we used our app Pocket Code. The eleven coding units are summarized in Figure 4. Between the sessions were short breaks and one bigger lunch break after three hours. At the beginning of each day, a gamified revision activity was conducted (e.g., in form of a scavenger hunt), and after the lunch and half an hour before the end of each day, we played a game together outside (e.g., Werewolf). Expressions of these phases are pictured in Figure 5.

Figure 4: Coding units 0 -11, a) game outside, b) input session, c) unplugged coding activity, d) challenge.

Figure 5: a) Games during breaks, b) presentation of the unit, c) unplugged coding, and d) programming challenges solved with the Pocket Code app

On the fourth day, the girls were introduced to two more activities. First, they had a two hours session to control Lego NXT robots with Pocket Code, and second, they were allowed to stitch the patterns made in Unit 7 with programmable embroidery machines on shirts and bags, see Figure 6.

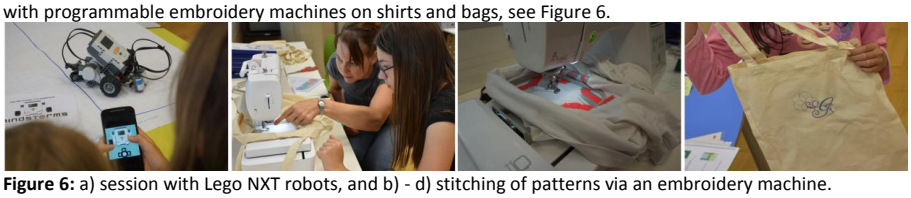


Figure 6: a) session with Lego NXT robots, and b) - d) stitching of patterns via an embroidery machine.

For the final gaming app, created on day four and five, the girls received the following supporting material in the form of storyboards:

- a graphical storyboard that was divided into four areas to help them to stick to the shape of a game and to frame the game in title, instruction, game, and end screen
- a textual storyboard that helped them to make important game decisions, e.g.,: name of the game, main character, gameplay (what is the game about?), genre, theme, goal of the game, used mechanics/dynamics (e.g., points, levels, difficulty levels, inventory, high-score, timer), amount of levels: what happens in level 1, 2, 3 (see Spieler and Slany, 2018a)

It was important to give participants time to think about a fitting concept for their games and about the story and what should happen. The supporting material should not only help them with these steps but also scaffold their design. All the games were collected on the last day by uploading them to our Catrobat community page. For the presentation, the girls' parents were invited. Every girl presented the game in front of the audience. They all felt very proud of their work.

4. Results

In reference to our first goal, during the warm-up phase (discussion in small groups) we wanted to found out more about our target group of young girls. Figure 7 summarises their discussion about their motivations to learn more about coding, their preferred apps, why they play mobile games, which programming languages they know, and what attributes the associated with women in technology.

Figure 7: Result of the group discussions during the "Warm-Up" phase.

The questionnaires (pre, daily, and post) handed out to students included measures of the various factors regarding students' intrinsic motivators: interest, sense of belonging, self-efficacy, and fun. In all cases, a 4-point Likert scale was used to measure the variables (Sullivan and Artino, 2013). The questions have been developed at the basis of literature by Schwarzer and Jerusalem (1995), the CATS Attitude Scale Items (Krieger et al., 2015), and from other research (Li and Watson, 2011). No questions were asked that could foster stereotype threats, as proposed (Krieger et al., 2015), e.g., "Girls can do technology as well as boys". The results of all questionnaires are summarized in Figure 8. The "4 Likert Scale" refers to 1: strongly disagree, 2: disagree, 3: agree and 4: strongly agree. It is recommended to use no neutral value and to use counter questions, e.g., Coding is interesting — Coding is boring, to demand their attention (McLeod, 2008). Thus, it is not always "the higher, the better". Question with "the lower, the better" are marked with "*".

The same size of this quantitative evaluation is very small (number = 13) but should serve as a first case study for our future interventions in summer 2019.

The interest in this group of girls was stable over the whole week. There is a slight increase in self-efficacy (confidence and knowledge) over the course of the week and from day to day, they felt more proud of their achievements. Also, their sense of belonging level (e.g., knowledge) slightly increased. However, there was a slight decrease in answers pre- and post- to "coding suits me" and "coding is important". They felt slightly more engaged and had fun throughout the whole course. On average, the students agreed that they want to join similar coding courses.

Figure 8: Average levels of interests, self-efficacy, sense of belonging, and fun/engagement.

The programs have been evaluated on the basis of the game design elements used (visual design, main characters) and program specific aspects (amount of scenes, objects, variables) (Spieler and Slany, 2018a). A summary of all used game elements is part of Figure 9.

genre	theme	goals	mechanics	level of control	visual design	main character	side characters
adventure (5) simulation (4) skill game (2) action (2)	nature (6) space (2) others (realistic, horror)	catch (7) avoid (2) shoot (4) keep alive (4)	points (12) levels (12) timer (2)	inclination sensors (7) buttons (2) touch sensor (7) physics (1) combination of several sensors (6)	Catrobat media library (11) Internet (1) Paint tool (3)	animals (8) pikachu (1) monster (1) girl (2) board (1)	food (11) pokemon (1) dogs (1) boy (1) animals (2)

Figure 9: Game design elements used during GCW.

The simulation games all had similar goals but used different concepts. Two of them were similar to "Tamagotchi games. The player has to feed and wash the characters, play with them, or go for a walk. One was a Pikachu simulation, the goal of which is that it evolves at the end and one was a pet simulation, where you have to care for dogs. They include "minigames", e.g., shooter, action, or catch/avoid games. The skill games were mazes and a pinball game. One of the adventure games was a text adventure which included "minigames" as well. For the main characters, girls used mostly animals like the Pocket Code family (panda, lynx, elephant, raccoon, penguin) or other animals like a dog, tiger, or horse. For side characters, all kinds of food were used, like oranges, sweets, bones, or other food for the animals to catch. Screenshots of the games are pictured in Figure 10.

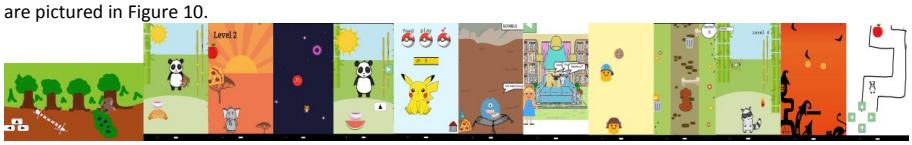


Figure 10: Games created during the GCW.

Regarding the programming itself, the girls used in their programs on average eight scenes (max: 17, min: 3), 74 scripts (max: 152, min: 18), 263 bricks (max: 502, min: 61), 40 objects (max: 87, min: 7), 43 looks (max: 94, min: 7), five sounds (max: 16, min: 0), and three variables (max: 14, min: 1). Summarized, all programs used various levels, included the shape of a game and numerous objects and looks, thus they were quite advanced.

During the interviews, the girls were asked out about graphics which are missing in our library. The answers included more animals from different angles (cats, horses, pets, hamster, dogs, birds, dragons, more bad animals, etc.), landscapes like forest and meadow, man-made creations such as houses and gardens, interiors (e.g., living room, kitchen, bathroom) and cities (e.g., café, shopping center, fitness center), foods such as cakes, vegetables, and fruits, or accessories like plates, smartphones, bags, shoes, or shopping clothes.

5. Discussion & Conclusion

The results showed that gaming elements female teenagers tend to create during the GCW mostly follow stereotypical expectations as described in Section 2.1. Compared to our previous experiences in heterogeneous groups and school classes this was not seen as something negative by girls (Spieler and Slany, 2018b).

The first aim of the paper was to provide evidence to design appropriate girls-only activities that were engaging and interesting at the same time. Results of the quantitative evaluations show the positive influence in girls' intrinsic motivation in regard to coding. This course provided a good starting point for further analysis and case studies. Even if the sample size was very small to show significant results in the quantitative evaluation, the results are still interesting. Answers to the intrinsic motivator "Self-efficiency" let us conclude that the girls had a high confidence in using the app. Furthermore, they had the feeling they learned something new and we're proud of their daily achievements. Collected factors related to the parameter "Sense of Belonging" showed that the participants learned something about technical professions and coding and hence showed a slight decrease in the feeling of the importance of coding and sense of belonging. Consequently, such a short course cannot change, for example, a strong image of stereotypes or long-held preconceptions. The fun level was high but the intention to partake in similar activities was low. Here, the study concludes that students' enjoyment has no relation to their intention to participate in similar activities again (Giannakos et al., 2014). To conclude, the quantitative evaluation shows that the values of many predictors for intrinsic motivations are located over the average. Girls agreed or strongly agreed that the coding week fostered their interests, helped them learn something about coding, helped them gain a better knowledge of technical professions and coding, and made them feel engaged while having fun during the course.

The aim of the second research question was to get more knowledge about girls design patterns. The results suggested a list of graphics be integrated into our app. New graphics (see: https://share.catrob.at/luna/media-library/looks) and featured games that have been developed on the basis of this findings together with design students from the degree program "Industrial Design" at the University of Applied Sciences in Graz (FH Joanneum). These games and some games created by the girls itself serve as feature games of the new developed Luna&Cat app. Luna&Cat is a tailored version of the app to appeal to female teenagers in particular. By showing female teenagers games designed by other young women in their age group, we help them to get ideas and inspiration to code their own programs. This is important because most girls have the feeling that the games they play are not created for them. With this customised app, our aim is to reach and build a user base of interested female teenagers who want to learn how to code.

6. Outlook

To engage girls in coding, a new project started in September 2018, with the name "Code'n'Stitch". With the option to program embroidery machines (very similar to the existing Turtlestitch project - https://www.turtlestitch.org/). In this way, self-created patterns and designs can be stitched on t-shirts, pants, or even bags. As a result, teenagers have something they can be proud of, something they can wear, and they can show to others. Starting in January 2019, we performed several design-thinking workshops to find out more about the requirements of the stitching extension, see Figure 11. These workshops start with a research unit, where students get asked to draw graphics which they want to stitch on clothes. Preliminary results

showed that girls preferred to stitch text (sayings), flowers, hearts, and animals, and boys preferred to stitch mostly text (sayings), brands, such as Nike or Adidas.

Figure 11: Impressions of the design-thinking workshops, Code'n'Stitch project

Acknowledgements

References

Agosto, D. (2003) *Girls and gaming: A summary of the research with implications for practice*. In Teacher Librarian: The Journal for School Library Professionals 31 (3), pp. 910–931.

Alvarado, C., Cao, Y., and Minnes, M. (2017) *Gender Differences in Students' Behaviors in CS Classes throughout the CS Major*. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 27–32.

Baker, S. (2019) *New data lay bare Europe's gender gap for STEM graduates*. [online] Available at: https://www.timeshighereducation.com/news/new-data-lay-bare-europes-gender-gap-stem-graduates. Last access: 06.05.2019

Committee on European Computing Education (CECE) (2017) I*nformatics Education in Europe: Are We All In The Same Boat?* Report by The Committee on European Computing Education (CECE) Jointly established by Informatics Europe & ACM Europe. [online] Available at: https://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report. Last access: 16.05.2019

El-Nasr, M S., Yuecel, I., Zupko, J., Andrea, and Smit, T.B. (2007) *Middle-to-High School Girls as Game Designers — What are the Implications?* Academic Days '07.

Federal Ministry of Education Austria (2017) [German] Lehrpläne der AHS Oberstufe - Informatik 5. Klasse. Highschool Curricula - Informatics mandatory 9th grade, BMB - Bundesministerium für Bildung. [online] Available at: https://www.bmb.gv.at/schulen/unterricht/lp/lp_neu_ahs_14_11866.pdf Last access: 02.04.2019

Fields, D.A., Giang, M., and Kafai, Y. (2014) *Programming in the wild: trends in youth computational participation in the online scratch community*. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education, pp. 2–11.

Frieze, C. and Quesenberry, J. (2015) *Kicking Butt in Computer Science: Women in Computing at Carnegie Mellon University.* Dog Ear Publishing.

Gabay-Egozi, L., Shavit, Y., and Yaish, M. (2015) *Gender Differences in Fields of Study: The Role of Significant Others and Rational Choice Motivations*. In European Sociological Review 31 (3), pp. 284–297.

Giannakos, M.N., Jacceri, L., and Leftheriotis, I. (2014) *Happy Girls Engaging with Technology: Assessing Emotions and Engagement Related to Programming Activities*. In Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences 8523,pp. 398–409.

Gorriz, C.M. and Medina, C. (2000) *Engaging girls with computers through software games*. In Commun. ACM 43 (1), pp. 42–49.

Heeter, C., Chu, K., Egidio, R., and Mishra, P. (2000) Do Girls Prefer Games Designed by Girls? In Proceedings from ISA:55 the Annual Conference of the International Communication Association, pp. 1–31.

Julkunen, J. (2015) *The Future of Match 3 — What You Need to Know — PART I*. [online] Available at: http://www.gamerefinery.com/the-future-of-match-3-what-you-need-to-know-part-i/ . Last access: 15.04.2019

Kafai, Y., Fields, D., Roque, R., Burke, W., and Monroy-Hernàndez, A. (2012) Colla*borative agency in youth online and offline creative production in Scratch*. In Research and Practice in Technology Enhanced Learning 7 (2), pp. 63–87.

Ketelaars, E. (2017) *What Place for Gender Mainstreaming in the EU's Framework on Support to Transitional Justice?* In European Foreign Affairs Review 22 (3), pp. 323–340.

Krieger, S., Allen, M., and Rawn, C. (2015) Are females disinclined to tinker in computer science? In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, pp. 102–107.

Mann, A. and Diprete, T. (2013) Trends in gender segregation in the choice of science and engineering majors. In Social Science Research 42(6), pp. 1519–1541.

McLeod, S. (2008). *Likert Scale*. [online] Available at: https://www.simplypsychology.org/likert-scale.html Last access: 4.04.2019

Medel, P. and Pouraghshband, V. (2017) *Eliminating Gender Bias in Computer Science Education Materials*. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 411–416.

NewZoo (2017) *Male and Female Gamers: How Their Similarities and Differences Shape the Games Market*. [online] Available at: https://newzoo.com/insights/articles/male-and-female-gamers-how-their-similarities-and-differences-shape-the-games-market/ Last access: 16.04.2019

Nichols J. (2019) Cl*osing the STEM Gap. Why STEM classes and careers still lack girls and what we can do about it.* Microsoft. [online] Available at: https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE1UMWz. Last access: 20.04.2019

Ochsner, A. (2015) *Lessons Learned With Girls, Games, and Design*. In Proceedings of the Third Conference on GenderIT, pp. 24–31.

OECD (2018) *Bridging the Digital Gender Divide. Include, Upskill, Innovate*. [online] Available at: http://www.oecd.org/internet/bridging-the-digital-gender-divide.pdf. Last accessed: 16.05.2019

Richard, G.T. and Kafai., Y.B. (2016) *Blind Spots in Youth DIY Programming: Examining Diversity in Creators, Content, and Comments within the Scratch Online Community*. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1473–1485.

Sadler, P.M., Sonnert, G., Hazari, Z., and Tai,, R. (2012) *Stability and volatility of STEM career interest in high school: A gender study.* In Science Education 96 (3), pp. 411–427.

Schwarzer, R. and Jerusalem, M. (1995) *Generalized Self-Efficacy scale*. In J. Weinman, S. Wright, & M. Johnston, Measures in health psychology: A user's portfolio. Causal and control beliefs, pp. 35–37.

Spieler, B. (2018). *Development and Evaluation of Concepts and Tools to Reinforce Gender Equality by Engaging Female Teenagers in Coding* (Doctoral dissertation). [online] Available at: https://catrob.at/SpielerPhD Last access: 19.05.2019

Spieler, B. and Slany, W. 2018a. Ga*me Development-Based Learning Experience: Gender Differences in Game Design*. In *Proceedings of the 12th European Conference on Games Based Learning*. pp. 616 - 625.

Spieler B., and Slany, W. (2018b).F*emale Teenagers and Coding: Create Gender Sensitive and Creative Learning Environments*. In Proceedings of Constructionism 2018, pp. 625 - 636.

Sullivan, G.M. and Artino, A.R. (2013) Analyzing and Interpreting Data From Likert-Type Scales. In Journal of Gradate Medical Education 5 (4), pp. 541–542.

Stout, J. and Camp, T. (2014) N*ow what?: action items from social science research to bridge the gender gap in computing research*. In SIGCAS Comput. Soc. 44 (4), pp. 5–8.

Thaler, A. and Zorn I. (2010). *Issues of doing gender and doing technology – Music as an innovative theme for technology education*. In European Journal of Engineering Education, 35 (4), pp. 445-454.

Twentyman, J. (2018) *Hands-on experience inspires girls to explore tech jobs*. Financial Times. [online] Available at: https://www.ft.com/content/19edb626-de17-11e8-b173-ebef6ab1374a. Last access: 14.05.2019

Unfried, A., Da Faber, M., Stanhope, D.S., and Wiebe, E. (2015) *The Development and Validation of a Measure of Student Attitudes Toward Science, Technology, Engineering, and Math (S-STEM)*. In Journal of Psychoeducational Assessment 33(7), pp. 622–639.

Yee, N. (2017). *Beyond 50/50: Breaking Down. The Percentage of Female Gamers by Genre.* [online] Available at: https://quanticfoundry.com/2017/01/19/female-gamers-by-genre/ Last access: 20.04.2019

Zagami, J., Boden, M., Keane, T., Moreton, B., and Schulz, K. (2015) G*irls and computing: Female participation in computing in schools.* In Australian Educational Computing 30 (2), pp. 1–14.

Wang, M.T., Eccles J.S., and Kenny, S. (2013) *Not Lack of Ability but More Choice : Individual and Gender Differences in Choice of Careers in Science Technology, Engineering, and Mathematics.* Psychological Science, 24 (5), pp. 770–775.

20 Outlook - Usable Security for the Catrobat Project

20.1 The Catrobat Project

Catrobat is a nonprofit Free Open Source Software (FOSS) project that was initiated in 2010 in Austria at Graz University of Technology by Prof. Wolfgang Slany. A multidisciplinary team develops free coding apps for teenagers and programming novices intending to introduce them to the world of programming. The project follows an interdisciplinary approach through worldwide collaboration and has a long term perspective of being developed sustainably over the next years. The Catrobat project can look back to a growing community of contributors since 2010, around 500 developers and 500 supporters contributed to the project.

20.1.1 Pocket Code

Pocket Code is a mobile visual coding environment designed for smartphones. This app allows children and teenagers to create their games and programs directly on their smartphones in their language and thereby teaches them fundamental programming skills. Pocket Code is an integrated development environment (IDE) running on Android and iOS and enables end-users to develop applications directly on their mobile devices. The language used in Pocket Code is based on a Lego® brick-styled visual drag-and-drop language similar to existing desktop-based frameworks such as Snap! or Scratch. In Pocket Code the bricks are categorized by their functionality such as Event, Control, Motion, Sound, Looks, Pen as shown in Figure 20.5.

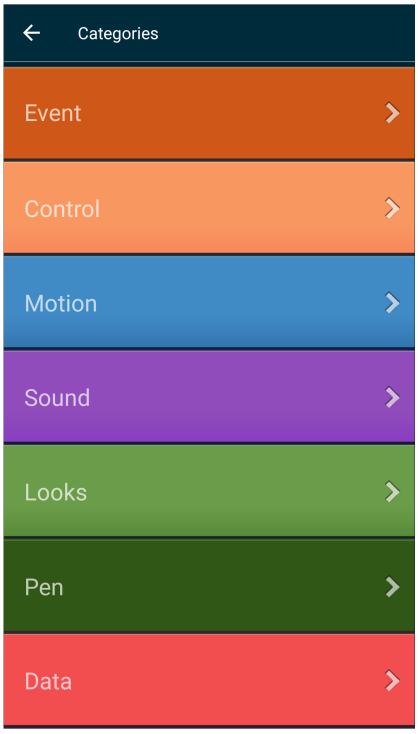


Figure 20.1: Pocket Code's default categories.

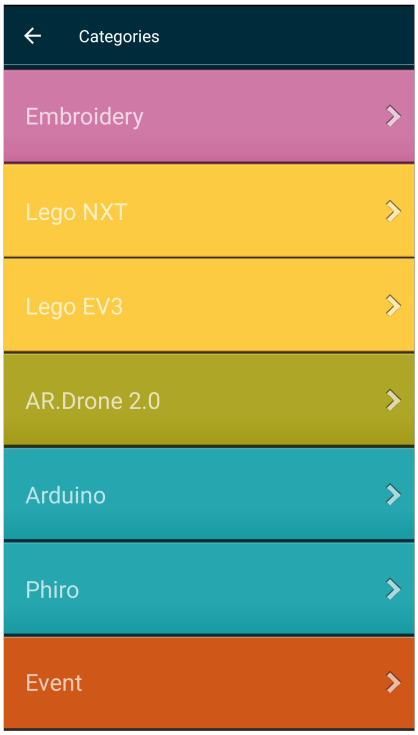


Figure 20.2: Pocket Code's extensions: Embroidery, Lego, Arduino.

Categories for particular hardware like Embroidery, Raspberry PIs, Arduino, or Lego Robotic are hidden, and users can activate them in the setting preferences if they want to use them 20.6. One of the most significant advantages comparing to desktop-based solutions is the usage of device sensors like accelerometers, gyroscope, GPS, or magnetometer. Users can upload applications written in Catrobat to the Pocket Code sharing platform, which makes them immediately publicly available. Pocket Code is an easy way to start programming.

At the beginning of the year 2019, the Pocket Code app has 660,760 downloads on Google Play. The top 10 countries are US (104K), Russia (96K), Germany (60K), India (30K), Ukraine (25K), Poland (25K), Brazil (20K), Turkey (20K), Austria (16K) and Indonesia (16K).

Pocket Code has nearly one million users from 180 countries, is natively available in more than 50 languages, including several languages not directly supported by the underlying operating system, e.g., right-to-left languages such as Farsi, or African languages such as Kiswahili.

20.1.2 Community Platform

In Pocket Code, users have the opportunity to share projects on the Catrobat community platform and interact with each other, as shown in Figure 20.3 and Figure 20.4. Without having to register, users can browse through already published projects or download them. On the community page, we divide the uploaded projects into different categories like the newest, most popular, or recommended projects.

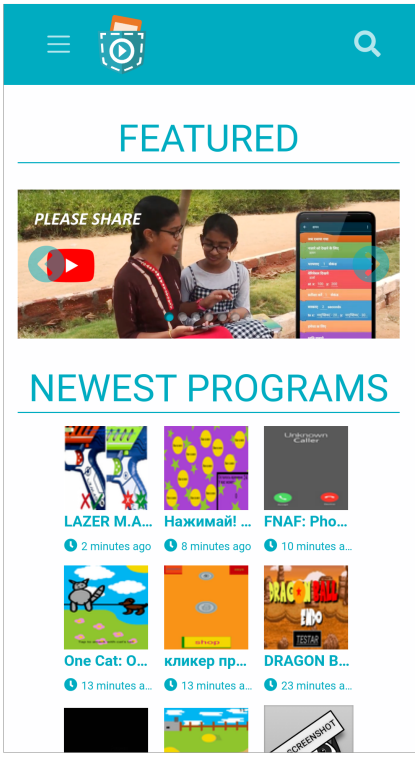


Figure 20.3: Pocket Code's community.

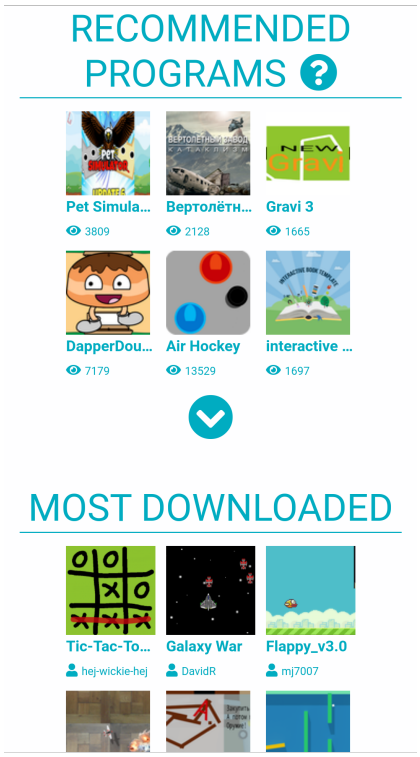


Figure 20.4: Pocket Code's community categories.

Once users want to share their projects, they need to log in with a username and password. After the successful authentication process, they can upload their projects and edit the user profile. Also the interaction with other users as well as the linking of other projects is only possible when the user is logged in.

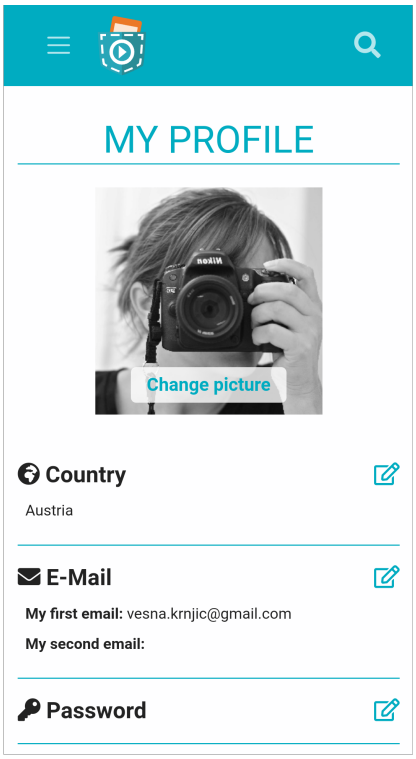


Figure 20.5: User account at Pocket Code's community.

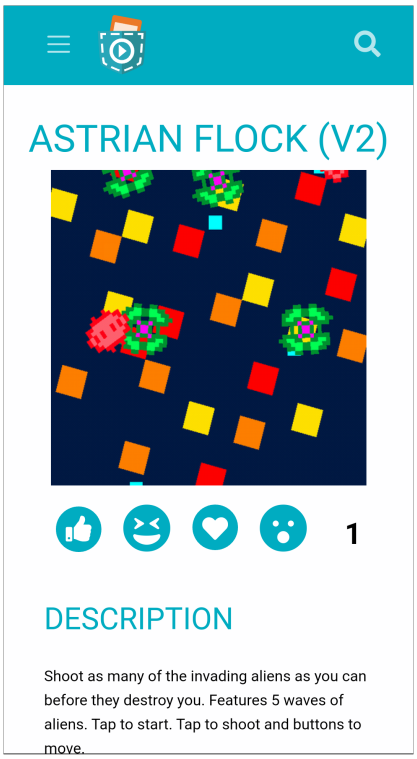


Figure 20.6: Presentation of a project on the Pocket Code's community.

On the Catrobat community, both positive, as well as negative online behaviors of youth, could be observed. On the positive side, we established cooperation and innovation. On the possible negative side, the following issues occur in the context of the Catrobat app used by teenagers:

- Excessive playing time.
- Aggression between users like sarcasm and bullying in comment sections.
- Discrimination against at-risk groups such as persons with disabilities or young people with disadvantaged social background.
- Unauthorized usage of photos or names of other people.
- Usage of inappropriate photos.
- Extremely violent.

- Horror.
- Pornographic uploaded content.
- Malicious reporting of the uploads of other users.
- Disclosure of personal date, such as name, age, address, etc.

This raises the question of how the Catrobat community can be improved to protect the privacy and decrease safety, security and privacy issues of users?

20.2 Mobile Security for Kids

Young people around the world use smartphones, intensively every day around the clock. On the one hand, they use smartphones during leisure time to communicate with their friends and on the other hand, for schools to research for homework. It is difficult to convey the security aspects of a system or software to an adult, but it is even more challenging to explain this to younger children and adolescents. Significant concerns regarding young children's and adolescents security occurred with the very rapid and widespread of smartphones. They are familiar with using the smartphone always connected to the Internet, using social media, and usually a large number of apps. They are more open to new apps and regularly app updates. Apps with malicious behavior can endanger users and harm their privacy [36]. Children and adolescents are among the most active Internet users, and unfortunately, subject to many threats. When installing an app, access to emails, contacts, photos, messages, and GPS location as well as control of Bluetooth, Wi-Fi Internet-access could be granted. The process of granting rights was introduced so that users can protect their personal information such as their phone number or privat photos. Often users, especially youngsters, are not aware of what rights an app has preserved on their mobile phone [18].

According to [18] young children and adolescents are exposed to the following potential risks while using smartphones with an Internet connection:

- Exposure to online friends or chat room "friends"
- Contact with strangers
- Access to illegal content
- Exposure to upsetting harmful content
- Exposure to phonographic content that is not suited for their age or maturity level
- Violence
- Racist content

- Advertising
- Commercial exploitation
- Misinformation
- Loss of personal information
- Attacks on privacy
- Spam emails
- Bullying

More important than blocking inappropriate material is teaching young children and teenagers safe and responsible online behavior. Education is vital in preventing online threats, so it is essential to let them know about threats, the more they know, the better.

20.3 Outlook - Usable Security for Pocket Code

Along with security and privacy, the developers of a system should always consider usability as well. As already discussed before, usability is one of the critical success factors in software systems, especially in learning tools - Chapter 2 presents background information about the importance of user research and end-user involvement in the development process. It is crucial to find out who exactly will be the end-user. In the case of Pocket Code, we have a special user group, which is adolescents between 13 and 18 years. Usability testing with youngsters faces many challenges, compared with usability testing for adults [60], [28]. For example, using heuristic evaluation for software designed for adolescents is often unrewarding because it is hard for an adult expert to put themselves in a teenage user [5].

In this section, we present the security challenges in the Pocket Code app that arise with the implementation of a brick that can access the Internet. Furthermore, we discuss the first approaches to solving this problem.

Mobile-Phones Permission Systems

Smartphones have access too many sensitive system resources such as the camera, the microphone or GPS sensors or users private data like email or saved contacts. It is important to protect this information from unauthorized access. When downloading applications to an Android device, a user can see the list of access permissions the application requests. This list shows all phone resources that the application has access to when installed [19], [54].

Bellow are the app permissions available for Android 6.0 and up. [1]

- Body Sensors

[1] https://support.google.com/googleplay/answer/6270602?hl=en

- Calendar
- Camera
- Contacts
- Location
- Microphone
- Phone
- SMS
- Storage

In contrast, the user controls the iOS access permission, meaning that every access request must be accepted or canceled. As described by Apple:

> "Users must grant permission for an app to access personal information, including the current location, calendar, contact information, reminders, and photos." [2]

Both Apple and Google, the trend over the past few years has been toward a continuous increase of user protection. The current direction is that more and more control the user posess through notifications, alerts, and permissions, eg., GPS data, or microphone [3]. The situation wasn't always like this. In the past, apps could access different functions of the smartphone, such as reading SMS or intercept interaction without permission from the user. Some mechanisms still do not require user permission, such as Internet access. Some mechanisms still do not require user permissions and are not classified as critical, such as Internet access, maybe because of economic reasons (advertisement).

Security and Privacy Warnings

Usually, security warnings inform users on the risk of allowing random applications to run on the system, help users reduce the risk of security threats, and therefore protects the system from the potential threats. Security warnings are designed to notify, inform, and advise users about the consequence of an action. The authors of the paper [13] proposed the Communication-Human Information Processing (C-HIP) model of how humans process warning messages. C-HIP is about the human experience between displaying a warning and deciding whether or not to follow it. Security warnings can be classified into five types, which are dialog box system, in-place system, notification system, balloons system, and banners system[41]. Regardless of the importance of security warnings, users tend to ignore them because they did not understand

[2] https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/requesting-permission/

[3] https://developer.android.com/about/versions/10/privacy/changes

the meaning of the warnings or are not motivated to read the warning [59], [6], [62], [70]. Usually users become accustomed with such warning and do not pay attention to them. Therefore well-considered warnings should be designed with the involvement of end-users, comprehensible description using the users' language. If the end-users are young children or teenagers, we must pay much more attention to usability.

20.3.1 Internet Access

URL Block used by Snap!

Snap! is another visual programming language that provides children and adults easy access to programming. The Snap![4] programming language has a simple brick 20.7 that allows retrieving a Uniform Resource Locator (URL). This Brick enables interaction with sensors or robots or with the World Wide Web. As input, the block receives the URL of a web page. Usually, the response is the description of the page in HTML language. The URL includes the used protocols. One website is not able to communicate with another site by Javascript security restrictions.

Figure 20.7: Snap's Internet Brick

Due to another Javascript security restriction, Snap! has no direct access to devices such as sensors or robots that have a connection to the computer. To overcome that restriction Snap! runs a separate program that provides a local HTTP server and is connected to the device. Snap! uses the server to make requests to the device. In contrast to Snap! the separate programs have access to anything on the computer. It is assumed that the external software is trustworthy.

Web Brick in Pocket Code

Unlike Snap!, Pocket Code can access all sensors supported by the smartphone. Therefore, certain security mechanisms must be provided in the app. Access to the Internet opens up some new opportunities for users, but also brings new security threats. In the design and development phase, we tried not only to focus on security issues but instead, we put the end-user in the center of the process. During the design, we differentiate between different user groups, the power user, the user who has no idea about the Web-Brick, the user downloading a third-part project containing the Web-Brick ,and the users who download an APK that includes a web brick. We tried to find a solution for all predefined user groups.

[4] https://snap.berkeley.edu/snap/snap.html

Technical Requirements: In the following section, we will describe the technical aspects of the Web-Brick **"Send web request _ and store answer in _ "** brick. The brick consists of two parameters, the URL address, which is used for server call, and a variable name to store the text for the return value (see Figure 20.8). The return value could be either a text-string (HTML, JSON, XML, etc.) or an error code.

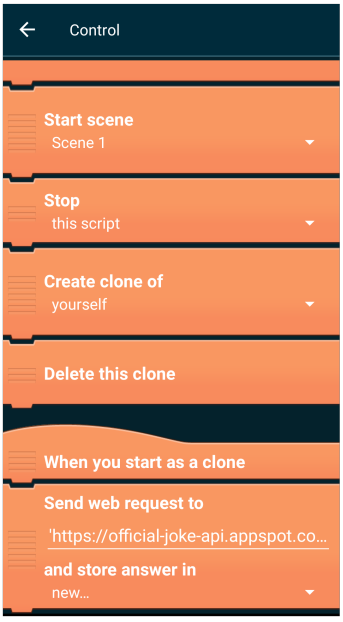


Figure 20.8: Pocket Code's Web-Brick

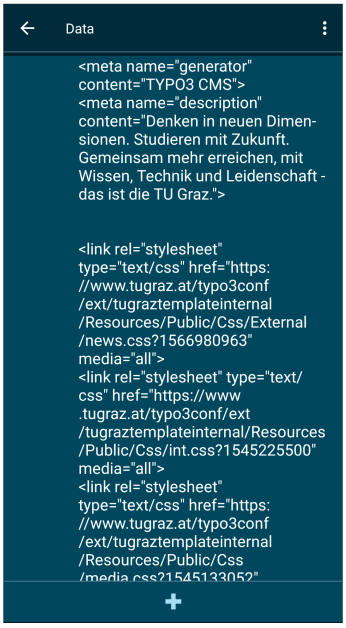


Figure 20.9: Return value of a Web-Brick

Different scripts of one project that run at the same time can contain such Web-Bricks, therefor more than one connection can be waiting for an answer at the same time. A maximum of 10 web requests can be open simultaneously at any given time by a project, in order to avoid denial-of-service attack (DoS attack), otherwise, an error code will be delivered. In case the Web-Brick returns a considerable text that fills up all the memory, the web connection will be canceled, the answer discarded, and the error code is returned.

Usable Security considerations:

During the design of the Web Brick privacy permissions, we applied the guidelines proposed by Ka-Ping Yee [29], [68] (compare with Chapter 2). When a **"Send web request to _ and store answer in _ "** brick (see Figure 20.8) is executed on the stage, the execution will be paused, and a warning is shown.

" WARNING: Do you want to allow this project to access the Internet using a **"Send web request to __ and store answer in __ "** brick with the following link?" `'http://darknet.org/gps?lat=47.070713?lon=15.439504/get-the-access-to-your-location1234567/allow-to-save-your-exact-\position/87?'` The whole link, including any parameters passed in the URL, must be included in the warning. If there is not enough space, a scroll bar must be used. The user must have the opportunity to view or copy the full link or cancel the current process 20.10. Tap on the link itself should not open the link in a browser, as otherwise the whole purpose of this warning would be defeated too easily by a single tap.

Figure 20.10: In this step the user is forced to view the URL or cancel the whole process.

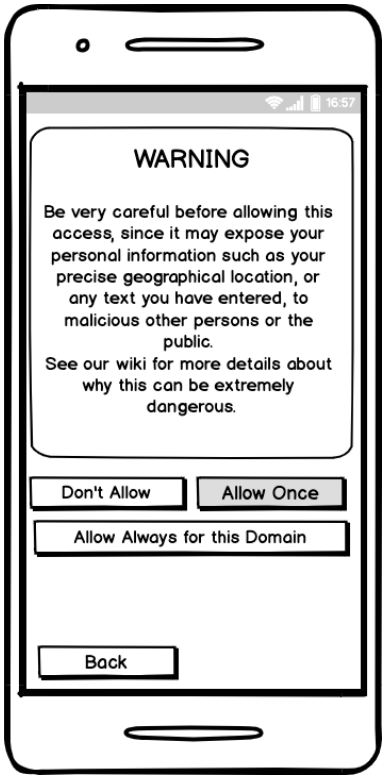


Figure 20.11: User can decide whether the URL is trusted or the whole domain.

When user selects the **Next** button, the following text is shown: **Be very careful before allowing this access, since it may expose your personal information such as your precise geographical location, or any text you have entered, to malicious other persons or the public. See our wiki for more details about why this can be extremely dangerous.** (See Figure 20.11)

Below this text, there shall be four possible options:

- **Do not allow**,
- **Allow once**,
- **Allow always for this domain**,
- or **Back**

(refer to Figure 20.11).

If **Copy full link is tapped**, the link should be copied to the Android clipboard. Another popup with the following text, warning, and button below shall be displayed (see Figure 20.12):

The link has been copied to the clipboard. Be very careful before opening this link in your browser, since it may expose your personal information such as your precise geographical location, or any text you have entered, to malicious other persons or the public. See our wiki for more information why this can be extremely dangerous.

In this step, the user must be able to go further in the process or step back, and therefore we provide two buttons, the **Next** and the **Back** button. After pressing the **Next** button, the following text is shown:

Be very careful before allowing this access, since it may expose your personal information such as your precise geographical location, or any text you have entered, to malicious other persons or the public. See our wiki for more details why this can be extremely dangerous.

Below this text, there shall be three possible options:

- **Do not allow**
- **Allow once**
- **Allow always for the domain**
- or **Cancle**.

(refer to Figure 20.11).

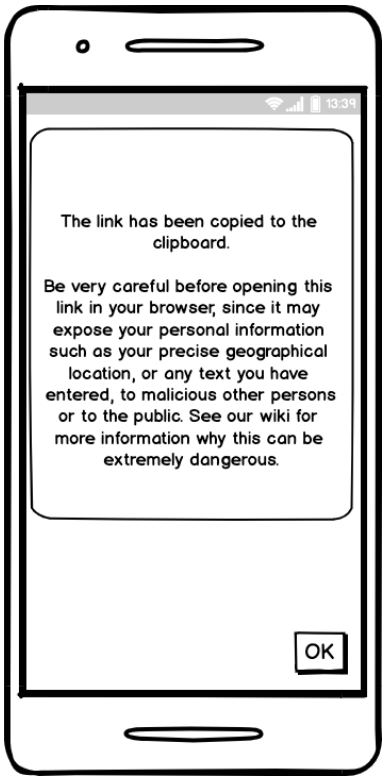


Figure 20.12: A warning is displayed before the copied link is opened.

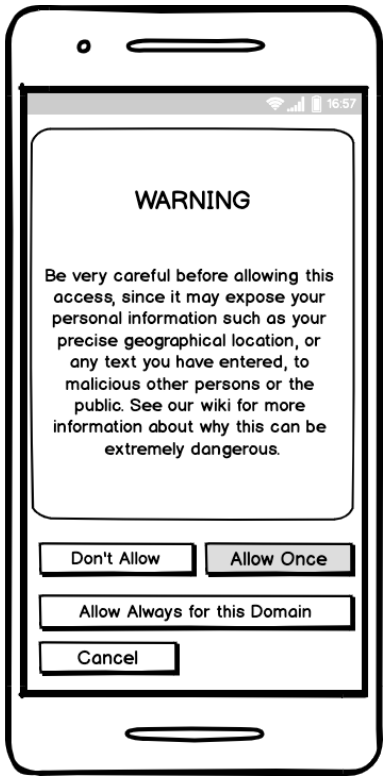


Figure 20.13: In this step, the user can decide whether to trust the URL

- If the user taps on **Do Not Allow**, the app resumes executing the project on the stage, but instead of accessing the link, the **"Send web request __ and store answer in __ "** brick returns the error code 401, which is the one for a missing authorization, see `https://en.wikipedia.org/wiki/List_of_HTTP_status_codes`, and stores it in the variable. I.e., the web is not accessed, but the project continues being executed.

- If the user taps on **Allow Once**, the app resumes executing the project on the stage, and the web link is accessed normally by the **"Send web request __ and store answer in __ "** brick. This is only allowed for this current brick, only one time, and the URL must be the same as the one shown (no new evaluation of the formula in the parameter field).

- If the user taps on **Allow Always for this Domain** `tugraz.at`, show a popup asking for confirmation, and shortly explain what happens and where this web access whitelisting for this domain can be revoked in the app.

Compiled Android Package Kit (APK) In the case of APK, the text should be: "Do you want to allow this app to access the internet with the following link?" `https://connect_four.games.com/moveinfo?board...` (See Figure 20.14)

Figure 20.14: Android Package Kit (APK) Web-Brick warning.

The whole process is summarized in Figure 20.15 as a flow-chart.

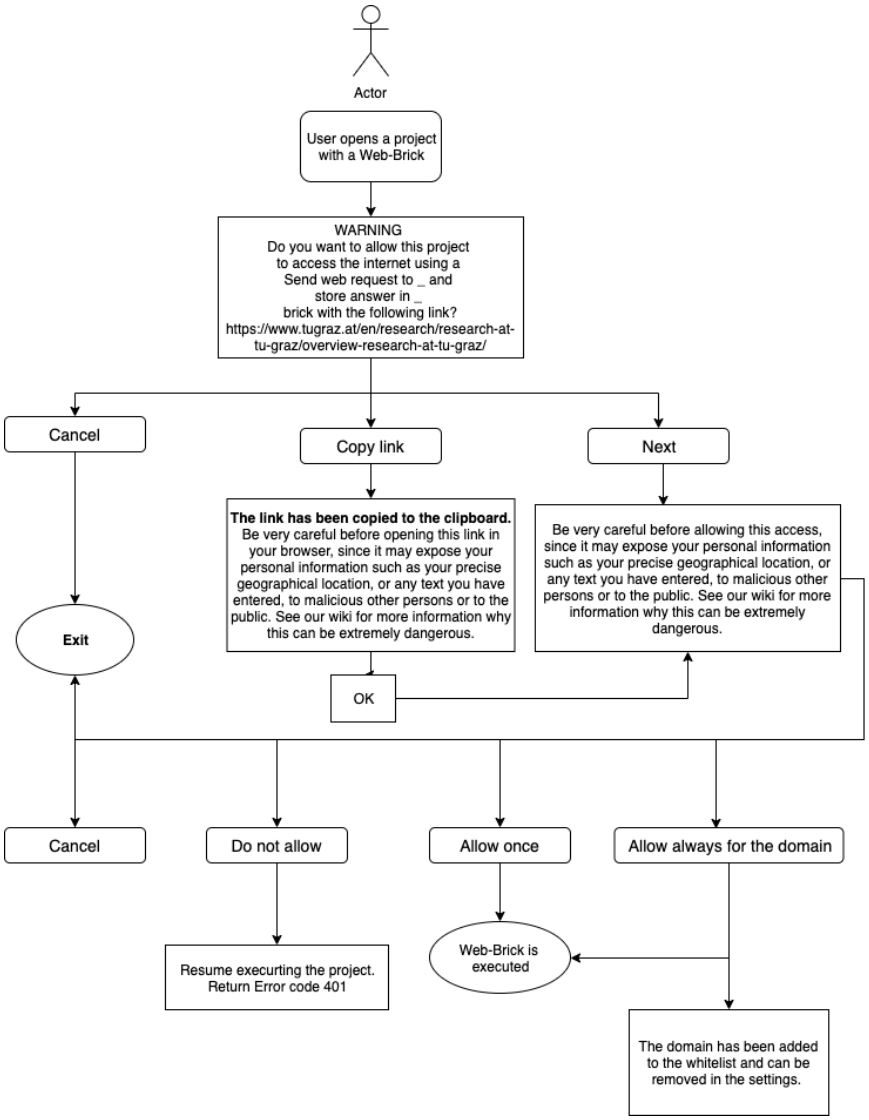


Figure 20.15: Flow chart: Warning for the Web-Brick.

Whitelist permissions

In the Pocket Code settings, there will be an entry for revoking the domain whitelisting permissions under "Web access permissions". It will look like all other lists in Catroid, with checkboxes on the left and in the action bar, a checkmark on the right. The checkmark must be grayed out if nothing is selected. On the left side of the action bar, there must be a back button and "select to remove" text.

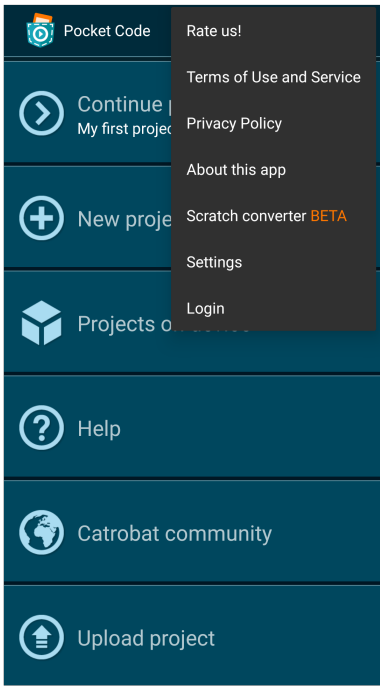


Figure 20.16: In the settings users can revoke a domain from the whitelist.

For compiled project APK's, the whitelisting cannot be revoked, since there is no settings screen, so another text is needed on the confirmation screen explaining this fact. Only by uninstalling and reinstalling it is possible to revoke the permission.

Next Steps

In the next development step of the Web-Bricks, the end-user should be involved in a user study like thinking-aloud test. From the user interviews conducted upfront, we already know that our end-users mostly ignore security warnings and do not even read them. Finding the right design for the warnings described above will be a challenging task. In the next step, we will frame design proposals and evaluate them with the target group. We have to decide together with the youngsters which warning representation is the best for them. Since many of them have noted in the interviews that they would rather watch videos than reading, one idea here is to create short videos instead of written warnings. However, classic warning dialogues, as shown in Figures 20.19 and 20.20 will

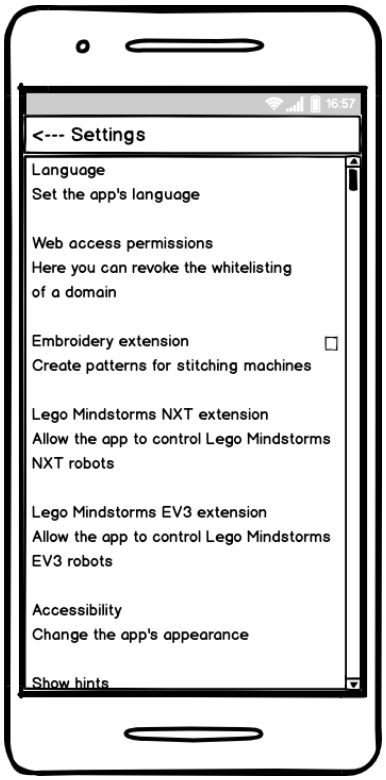


Figure 20.17: Web access permissions.

Figure 20.18: Revoke a domain from whitelist.

also be prepared.

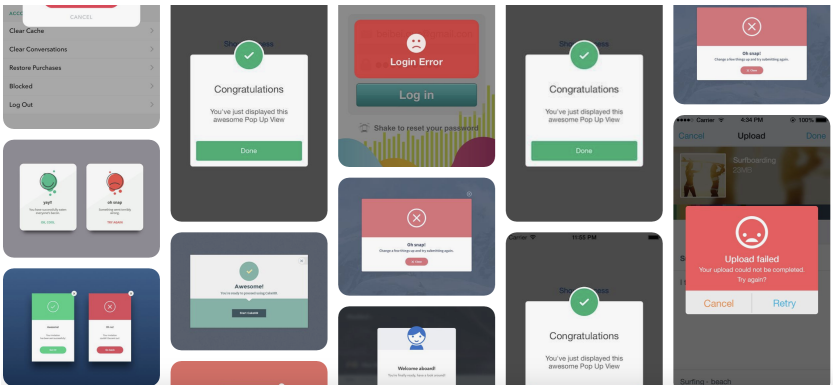


Figure 20.19: Source: https://www.pinterest.at/brentonhouse/mobile-ux-dialogs/ (Brenton House) UX design for alerts for smartphones.

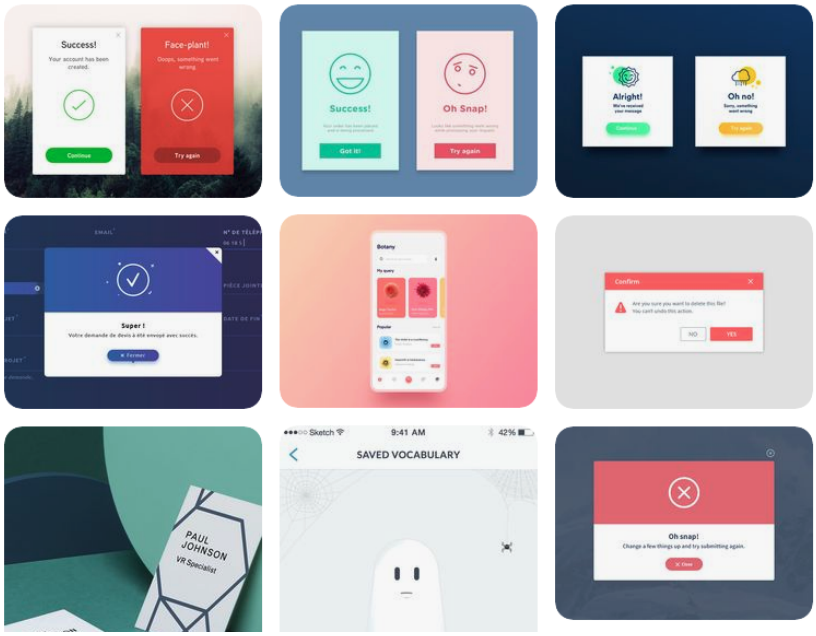


Figure 20.20: Source: https://www.pinterest.at/pin/38139928074711991/ UX design for warning dialogs.

As already discussed by Nielsen[46] it is not sufficient to only train the user regarding security.

> "However, user education should not be the main approach to countering security problems for three reasons."

First of all, it just doesn't work. The attackers are increasingly using sophisticated systems so that the user has no chance to detect an attack. Nielsen's second argument is that it puts the burden on the wrong shoulder. The Internet and computer technology was developed with the assumption that there are no attackers. Therefore, technology should change in the first place, not users. The last argument is that the Internet will never unfold its full benefit as long as users are held responsible for specific security flaws. Because of these considerations, we should not leave the whole burden with the Web-Brick to the user. In the future, therefore, consideration should also be given to automatically recognizing domains that are on blacklists. Furthermore, the user should have the possibility to report a project, including the internet address to the Catrobat team.

The Pocket Code app requires many more steps to improve the integrated security mechanisms and make them more user-friendly. Shortly, we will try to enhance the Pocket Code's authentication process. On the one hand, we should require stronger passwords from users, but also follow guidelines on how

to make the process easier for the user. Instead, we should consider whether to extend the authentication process to biometrics, such as fingerprinting or facial recognition.

21 | Conclusion and Further Work

Changing information and communication technologies influence our daily lives. Nowadays, smartphones with Internet access are an integral part of our society, affecting all age groups. The resulting digital lifestyle raises many new challenges. Identity theft, theft of sensitive data, Social Media attacks, or mobile malware threat an increasing number of users. Adequate protection of information systems and data is particularly important. At the same time it is important to improve the understanding and handling of ordinary people with regard to their competence in matters of all aspects of IC technologies. Usually, the development of security-relevant applications focuses on the resistance against malicious attacks. Security requirements are typically met by approved cryptographic methods. The level of security that can be achieved is not only determined by the technical implementation. Also, usability factors play an essential role. It is important to bear in mind that in the eyes of ordinary users security is not highly important. They always have another primary task, such as searching for information, transacting money, or only posting a photo to Social Media. Until the mid-1990s, most security researchers focused on technical issues such as the development of basic cryptographic protocols. An essential aspect for security developers should be to look at the usability of the security features they build. In the past, the focus was on training users in the handling of difficult-to-use software rather than redesigning it.

Usable security describes the interdisciplinary approach of designing security-enhancing techniques for digital products and services in such a way that users are optimally supported in their security-relevant goals. This also enables non-technical users to understand security elements and their necessity fundamentally and to use the systems in the way they were initially intended. Usable privacy focuses on technologies which assist the improvement of privacy in digital systems and platforms.

Throughout this thesis, we focused on usable security design and evaluation around user authentication, electronic documents, and privacy. Our work shows that applying usable security and privacy is crucial in any domain, whether it is an e-Government system or a visual programming environment for children and adolescents such as Pocket Code.

We presented the architecture and implementation of a dynamic provider named ALAP, Agile Authentication Provider. ALAP offers authentication factors from different categories and allows service providers to define their security requirements through policy by claiming a global Level of Assurance. Our solution assembles an authentication process for a service to meet its security requirements, whereby the users can choose the preferred authentication factor dynamically. We evaluate the system from the user's perspective by performing a multi-step usability security evaluation. The evaluation of ALAP in a real-world scenario has lead to valuable results. By observing users' interaction with ALAP and collecting user feedback through different questionnaires, we were able to identify persisting weaknesses and to create space for further improvements. Based on the usable security evaluation results, we propose an extended architecture of ALAP, the Convenient Agile Authentication (CALA), where usability plays a central role in selecting the authentication factors. CALA does not consider primarily the needs of the service provider but also those of the end-user's concern regarding the authentication process.

Signature-creation is essential for many e-Government processes. Especially the creation of qualified signatures is highly important. In this thesis, we have presented a modular architecture for adaptable signature-creation tools. Considering the needs of all user groups, reliability, usability, adaptability, and modularity are identified as core requirements for signature-creation tools. To achieve a high impact our solution is based on the Austrian Citizen Card concept. In order to make our system secure and usable, we have followed the user-centered design method where the security designers and usability experts work together from the very beginning. Several usability iterations have been carried out to ensure user-friendliness. The tool was officially introduced in Austria[1] and is still in use, years later.

Based on examples such as ALAP or the signature solution, we have demonstrated that if usable security and privacy is considered from the design phase onwards, the tensions between security and usability can be reduced and thus more secure and usable systems will be developed.

Further, we have shown that it is particularly important to evaluate security-sensitive systems like the Austrian Citizen Card Environment. Our usability evaluations delivered more in-depth insights into the usability of core components of the Austrian eGovernment from the citizens' point of view. By collecting user feedback via questionnaires and interviews, we were able to identify persisting weaknesses and found further room for improvement. Valuable findings have also been obtained from an analysis of recorded user sessions. For example, the handling of SSL certificates used by the smart card-based solutions needs to be improved. A direct comparison of the three CCS implementations shows that the Mobile Phone Signature appears to be the most secure and trustworthy solution, followed by MOCCA Local and MOCCA Online. All results are incorporated into future releases of the CCS implementations. Thus, the conducted usability studies contribute to the security and usability of MOCCA

[1] https://www.buergerkarte.at/en/pdf-signature-mobile.html

Local, MOCCA Online, and the Mobile Phone Signature and hence, to more efficient e-Government services.

The trend to make public sector data available to the general public and the corporate sector raises the demand for innovative techniques to meet rising security and privacy requirements. We discuss the importance of integrity or authenticity of public sector data and present a concept to assure the integrity and authenticity of data based on electronic signatures. Moreover, we show that our concept can also be extended to data that needs to be anonymized, to meet privacy requirements, by incorporating redactable signatures. We propose electronic signatures in general and editable electronic signature schemes, in particular as an adequate enabler for such security preserving techniques. The conducted assessment reveals that blank digital signatures, which are a novel approach representing a subset of editable signature schemes, are uniquely suited to meet the predefined requirements.

In the last part of this dissertation, we discuss the security challenges the visual programming environment Pocket Code has to overcome by introducing a Web-Brick. If such a web brick is used in a Catrobat project, access to the Internet is possible. Connection to the Internet opens up some new opportunities for users but is accompanied by security threats at the same time. To inform users about the Internet access a Catrobat Project has, we propose a warning procedure designed following well defined usable security guidelines. In our recommended solution, the user can decide whether the project gains Web access or not. Further, the user can trust an entire domain that is whitelisted, but could be revoked at any time.

In summary, we can state that significant improvements can be achieved by designing security and usability together and by reconciling the mental model of the target audience with that of the developers. Evaluating the usability of a security-critical system after it has been developed usually does not lead to satisfying results, neither with regard to security nor usability. Instead, usability and security experts must work throughout the entire development cycle.

The area of usable security and privacy is a young field of research, so there is still a lot of work to be done. For example user authentication especially passwords remains an unsolved usable security problem. Email encryption is still not accepted by end-users as it is far too difficult to use. Further research is needed to investigate the mental models of children and adolescents regarding security and privacy. Applied usable security and privacy methods are often designed for adult end-users, the adaptation to the specific user group children and adolescents has not yet been subject of serious research.

Bibliography

[1] Abanumy, A., Al-Badi, A., and Mayhew, P. e-government website accessibility: In-depth evaluation of saudi arabia and oman. *The Electronic Journal of E-Government 3* (01 2005).

[2] Acar, Y., Fahl, S., and Mazurek, M. L. You are not your developer, either: A research agenda for usable security and privacy research beyond end users. In *2016 IEEE Cybersecurity Development (SecDev)* (Nov 2016), pp. 3–8.

[3] Adams, A., and Sasse, M. A. Users are not the enemy. *Commun. ACM 42*, 12 (Dec. 1999), 40–46.

[4] Alrayes, F., and Abdelmoty, A. I. Privacy concerns in location-based social networks.

[5] Alsumait, A., and Al-Osaimi, A. Usability heuristics evaluation for child e-learning applications. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications &Amp; Services* (New York, NY, USA, 2009), iiWAS '09, ACM, pp. 425–430.

[6] Amran, A., Zaaba, Z. F., Singh, M. M., and Marashdih, A. W. Usable security: Revealing end-users comprehensions on security warnings. *Procedia Computer Science 124* (2017), 624 – 631. 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia.

[7] Balfanz, D., Durfee, G., Smetters, D., and Grinter, R. In search of usable security: Five lessons from the field. *Security and Privacy, IEEE 2* (10 2004), 19 – 24.

[8] Benisch, M., Kelley, P. G., Sadeh, N., and Cranor, L. F. Capturing location-privacy preferences: Quantifying accuracy and user-burden tradeoffs. *Personal Ubiquitous Comput. 15*, 7 (Oct. 2011), 679–694.

[9] Bonneau, J., and Preibusch, S. The password thicket: technical and market failures in human authentication on the web. In *9TH WORKSHOP ON THE ECONOMICS OF INFO SECURITY (WEIS 2010)* (2010).

[10] Brodie, C., Karat, C.-M., Karat, J., and Feng, J. Usable security and privacy: A case study of developing privacy management tools. In *Proceedings of the 2005 Symposium on Usable Privacy and Security* (New York, NY, USA, 2005), SOUPS '05, ACM, pp. 35–43.

[11] Carroll, J. M., and Olson, J. R., Eds. *Mental Models in Human-computer Interaction: Research Issues About What the User of Software Knows*. National Academy Press, Washington, DC, USA, 1987.

[12] Christensen, C., of Management, S. S., and of Technology, M. I. *Finding the Right Job for Your Product*. MIT Sloan Management Review. MIT Sloan Management Review, 2007.

[13] Conzola, V., and Wogalter, M. A communication–human information processing (c–hip) approach to warning effectiveness in the workplace. *Journal of Risk Research 4* (10 2001), 309–322.

[14] Cooper, A. *The Inmates Are Running the Asylum*. Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 1999.

[15] Cranor, L., and Garfinkel, S. *Security and Usability*. O'Reilly Media, Inc., 2005.

[16] Deyan, G. 60+ smartphone statistics in 2019. `https://techjury.net/stats-about/smartphone-us`, last visited November, 2019.

[17] Dmitrienko, A., Liebchen, C., Rossow, C., and Sadeghi, A.-R. Security analysis of mobile two-factor authentication schemes. *Intel Technology Journal, ITJ66 Identity, Biometrics, and Authentication Edition 18* (Juli 2014).

[18] Etaher, N., and Weir, G. R. S. Understanding children's mobile device usage. In *2016 IEEE International Conference on Cybercrime and Computer Forensic (ICCCF)* (June 2016), pp. 1–7.

[19] Felt, A., Ha, E., Egelman, S., Haney, A., Chin, E., and Wagner, D. Android permissions: User attention, comprehension, and behavior. *SOUPS 2012 - Proceedings of the 8th Symposium on Usable Privacy and Security* (07 2012).

[20] Florêncio, D., and Herley, C. Where do security policies come from? In *Proceedings of the Sixth Symposium on Usable Privacy and Security* (New York, NY, USA, 2010), SOUPS '10, ACM, pp. 10:1–10:14.

[21] folmer, e., van Gurp, J., and Bosch, J. Scenario-based assessment of software architecture usability.

[22] Garcia, A. C., Maciel, C., and Pinto, F. A quality inspection method to evaluate e-government sites. vol. 3591, pp. 198–209.

[23] Garfinkel, S., and Lipford, H. R. *Usable Security: History, Themes, and Challenges*. Morgan & Claypool Publishers, 2014.

[24] Garfinkel, S. L., and Miller, R. C. Johnny 2: A user test of key continuity management with s/mime and outlook express. In *Proceedings of the 2005 Symposium on Usable Privacy and Security* (New York, NY, USA, 2005), SOUPS '05, ACM, pp. 13–24.

[25] Harbach, M., Fahl, S., and Smith, M. Who's afraid of which bad wolf? a survey of it security risk awareness. In *2014 IEEE 27th Computer Security Foundations Symposium* (July 2014), pp. 97–110.

[26] Hausawi, Y. M., and Allen, W. H. An assessment framework for usable-security based on decision science. In *Proceedings of the Second International Conference on Human Aspects of Information Security, Privacy, and Trust - Volume 8533* (New York, NY, USA, 2014), Springer-Verlag New York, Inc., pp. 33–44.

[27] Hof, H.-J. User-centric it security - how to design usable security mechanisms.

[28] Joyce, A. Usability testing with minors: 16 tips. `https://www.nngroup.com/articles/usability-testing-minors/`, last visited November, 2019.

[29] Ka-Ping Yee. Aligning security and usability. *IEEE Security Privacy 2*, 5 (Sep. 2004), 48–55.

[30] Kainda, R., Flechais, I., and Roscoe, A. Usability and security of out-of-band channels in secure device pairing protocols.

[31] Kainda, R., Flechais, I., and Roscoe, A. Security and usability: Analysis and evaluation. pp. 275–282.

[32] Konoth, R. K., van der Veen, V., and Bos, H. How Anywhere Computing Just Killed Your Phone-Based Two-Factor Authentication. In *FC* (Feb. 2016).

[33] Kuo, C., Romanosky, S., and Cranor, L. F. Human selection of mnemonic phrase-based passwords. In *Proceedings of the Second Symposium on Usable Privacy and Security* (New York, NY, USA, 2006), SOUPS '06, ACM, pp. 67–78.

[34] Lance, B., and Anthony, U. The customer-centered innovation map. *Harvard business review 86* (06 2008), 109–14, 130.

[35] Laubheimer, P. Personas vs. jobs-to-be-done. `https://www.nngroup.com/articles/personas-jobs-be-done/`, last visited November, 2019.

[36] Livingstone, S., and Helsper, E. Balancing opportunities and risks in teenagers' use of the internet: the role of online skills and internet self-efficacy. *New Media & Society 12*, 2 (2010), 309–329.

[37] Ma, H.-Y. T., and Zaphiris, P. The usability and content accessibility of the e-government in the uk.

[38] Mahatody, T., Sagar, M., and Kolski, C. State of the art on the cognitive walkthrough method, its variants and evolutions. *Int. J. Hum. Comput. Interaction 26*, 8 (2010), 741–785.

[39] Mairiza, D., and Zowghi, D. An ontological framework to manage the relative conflicts between security and usability requirements. In *2010 Third International Workshop on Managing Requirements Knowledge* (Sep. 2010), pp. 1–6.

[40] Micro, T. 12 most abused android app permissions. `http://about-threats.trendmicro.com/us/library/image-gallery/12-most-abused-android-app-permissions#`, last visited November, 2019.

[41] Microsoft. Warning messages. `https://docs.microsoft.com/de-de/windows/win32/uxguide/mess-warn?redirectedfrom=MSDN`, last visited November, 2019.

[42] Mihajlov, M., Josimovski, S., and Jerman, B. A conceptual framework for evaluating usable security in authentication mechanisms - usability perspectives. pp. 332–336.

[43] Nielsen, J. Introduction to usability. `https://www.nngroup.com/articles/usability-101-introduction-to-usability/`, last visited November, 2019.

[44] Nielsen, J. Mental models. `https://www.nngroup.com/articles/mental-models/`, last visited November, 2019.

[45] Nielsen, J. Survey new u.s. smartphone growth by age and income. `https://www.nielsen.com/us/en/insights/article/2012/survey-new-u-s-smartphone-growth-by-age-and-income/`, last visited November, 2019.

[46] Nielsen, J. User education is not the answer to security problems. `https://www.nngroup.com/articles/security-and-user-education/`, last visited November, 2019.

[47] Nielsen, J., and Molich, R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1990), CHI '90, ACM, pp. 249–256.

[48] NIST. Nist special publication 800-63b, digital identity guidelines authentication and lifecycle management. https://pages.nist.gov/800-63-3/sp800-63b.html#sec8, last visited November, 2019.

[49] O'Connor, K. Personas: The foundation of a great user experience. https://uxmag.com/articles/personas-the-foundation-of-a-great-user-experience, last visited November, 2019.

[50] Patil, S., Norcie, G., Kapadia, A., and Lee, A. Reasons, rewards, regrets: Privacy considerations in location sharing as an interactive practice. *SOUPS 2012 - Proceedings of the 8th Symposium on Usable Privacy and Security* (07 2012).

[51] Payne, B., and Edwards, W. A brief introduction to usable security. *Internet Computing, IEEE 12* (06 2008), 13–21.

[52] Petsas, T., Tsirantonakis, G., Athanasopoulos, E., and Ioannidis, S. Two-factor authentication: is the world ready?: quantifying 2fa adoption. In *Proceedings of the Eighth European Workshop on System Security, EuroSec 2015, Bordeaux, France, April 21, 2015* (2015), pp. 4:1–4:7.

[53] Pruitt, J., and Grudin, J. Personas: Practice and theory. In *Proceedings of the 2003 Conference on Designing for User Experiences* (New York, NY, USA, 2003), DUX '03, ACM, pp. 1–15.

[54] Reardon, J., Feal, A., Wijesekera, P., On, A. E. B., Vallina-Rodriguez, N., and Egelman, S. 50ways to leak your data: An exploration of apps' circumvention of the android permissions system. In *Proceedings of the 28th USENIX Conference on Security Symposium* (Berkeley, CA, USA, 2019), SEC'19, USENIX Association, pp. 603–620.

[55] Rohrer, C. When to use which user-experience research methods. https://www.nngroup.com/articles/which-ux-research-methods/, last visited November, 2019.

[56] Sahar F. Tradeoffs between usability and security. *IACSIT International Journal of Engineering and Technology 5*, 4 (2013).

[57] Saltzer, J. H., and Schroeder, M. D. The protection of information in computer systems, 1975.

[58] Schultz, E. Statistiken zur smartphone-nutzung in Österreich. https://de.statista.com/themen/3654/smartphone-nutzung-in-oesterreich/, last visited November, 2019.

[59] Sharek, D., Swofford, C., and Wogalter, M. Failure to recognize fake internet popup warning messages. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 52* (09 2008).

[60] Sherwin, K., and Nielsen, J. Children's ux: Usability issues in designing for young people. `https://www.nngroup.com/articles/childrens-websites-usability-issues/`, last visited November, 2019.

[61] Standardization, I. O. F. *ISO 9241-11 - Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Part 11: Guidance on Usability*. ISO, Berlin, 1998.

[62] Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., and Cranor, L. F. Crying wolf: An empirical study of SSL warning effectiveness. In *18th USENIX Security Symposium, Montreal, Canada, August 10-14, 2009, Proceedings* (2009), pp. 399–416.

[63] Sørum, H. An empirical investigation of user involvement, website quality and perceived user satisfaction in egovernment environments. pp. 122–134.

[64] Uzun, E., Karvonen, K., and Asokan, N. Usability analysis of secure pairing methods. pp. 307–324.

[65] Wash, R., and Zurko, M. E. Usable security. *IEEE Internet Computing 21*, 3 (May 2017), 19–21.

[66] Wharton, C., Rieman, J., Lewis, C., and Polson, P. Usability inspection methods. John Wiley & Sons, Inc., New York, NY, USA, 1994, ch. The Cognitive Walkthrough Method: A Practitioner's Guide, pp. 105–140.

[67] Whitten, A., and Tygar, J. D. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8* (Berkeley, CA, USA, 1999), SSYM'99, USENIX Association, pp. 14–14.

[68] Yee, K.-p. User interaction design for secure systems.

[69] Yeratziotis, A., Pottas, D., and Greunen, D. V. A usable security heuristic evaluation for the online health social networking paradigm. *International Journal of Human–Computer Interaction 28*, 10 (2012), 678–694.

[70] Zaaba, Z. F., and Boon, T. Examination on usability issues of security warning dialogs.