

Interoperability of Electronic Documents

Next-Generation Technologies and Applications for
Electronic Documents

Klaus Stranacher

Interoperability of Electronic Documents

Next-Generation Technologies and Applications for Electronic Documents

Ph.D. Thesis

at

Graz University of Technology

submitted by

Klaus Stranacher

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology
A-8010 Graz, Austria

June 2014

© Copyright 2014 by Klaus Stranacher

Assessors

O.Univ.-Prof. Reinhard Posch

Priv.Doiz. Ass.Prof. Stefan Rass

Advisor

Dr. Arne Tauber



Interoperabilität von Elektronischen Dokumenten

Next-Generation Technologien und Applikationen für Elektronische Dokumente

Doktorarbeit
an der
Technischen Universität Graz

vorgelegt von

Klaus Stranacher

Institut für Angewandte Informationsverarbeitung und Kommunikationstechnologie (IAIK),
Technische Universität Graz
A-8010 Graz

Juni 2014

© Copyright 2014, Klaus Stranacher

Diese Arbeit ist in englischer Sprache verfasst.

Begutachter

O.Univ.-Prof. Reinhard Posch
Priv.Doiz. Ass.Prof. Stefan Rass

Betreuer

Dr. Arne Tauber



Abstract

Electronic documents are a vital element for electronic communication. They are used to store, exchange and process different kinds of information and data. To assure the authenticity and integrity of electronic documents, the current means of choice are electronic signatures whereas their legal framework is defined by the EU Signature Directive. Due to the increasing globalisation and mobility of citizens and enterprises, new challenges for electronic documents and electronic signatures, emerge. These challenges comprise the need for cross-border interoperability in the area of electronic documents and electronic signatures in particular. Furthermore, these challenges also include the need for next-generation applications across borders. These needs are also underpinned by the Digital Agenda for Europe and the e-Government Action Plan 2011-2015.

To face these upcoming challenges next-generation core-technologies for processing electronic documents have been developed. These technologies comprise an interoperability framework for exchanging documents across borders and advanced signature verification facilities. Additionally, the technologies include a mechanism to allow an efficient processing of electronic documents as well as signature technologies, which allow subsequent modifications of signed data by still retaining the validity of the original signature (called editable signatures).

These core-technologies are used to introduce next-generation applications for electronic documents. These applications are: (a) an approach for a trusted open government data, which enables a reliable and trustworthy publication of public sector data; (b) a user-centered identity management for electronic identities enabling selective disclosure, whereas the qualified and authentic identity data is still verifiable, even if only a subset of the identity data is revealed; (c) an approach for secure, authentic and efficient public administration procedures across borders based upon the requirements of the EU Services Directive. The presented technologies and applications are compliant to the needs given by the European Interoperability Framework (EIF). All of these applications have been implemented on a prototype basis and tested - partly and as far as possible within real life environments - to evaluate their applicability.

Kurzfassung

Elektronische Dokumente sind ein essentielles Element elektronischer Kommunikation. Sie werden benutzt um verschiedene Formen von Informationen und Daten zu speichern, auszutauschen oder weiterzuverarbeiten. Um die Authentizität und Integrität von elektronischen Dokumenten sicherzustellen, sind elektronische Signaturen derzeit die vorherrschende Technologie. Deren rechtliches Rahmenwerk wird dabei von der EU Signaturrechtlinie bestimmt. Aufgrund der ansteigenden Globalisierung und der erhöhten Mobilität der Bürgerinnen und Bürger, als auch der Unternehmen, entstehen neue Herausforderungen sowohl für elektronische Dokumente als auch für elektronische Signaturen. Diese Herausforderungen umfassen dabei die Notwendigkeit von grenzüberschreitender Interoperabilität im Bereich von elektronischen Dokumenten und elektronischen Signaturen im Speziellen. Zusätzlich beinhalten diese Herausforderungen die Notwendigkeit einer neuen Generation von grenzüberschreitenden Anwendungen. Diese Notwendigkeiten werden auch durch die Digitale Agenda für Europa und den E-Government Aktionsplan 2011-2015 untermauert.

Um sich nun diesen Herausforderungen stellen zu können, wurde eine neue Generation von Technologien für elektronische Dokumente entwickelt. Diese Technologien umfassen ein Interoperabilitätsframework für den grenzüberschreitenden Austausch von elektronischen Dokumenten, sowie erweiterte Möglichkeiten zur Prüfung der Gültigkeit von elektronischen Signaturen. Des Weiteren wurden Mechanismen zur effizienten Weiterverarbeitung von elektronischen Dokumenten entwickelt, als auch Signaturtechnologien, die eine nachträgliche Modifikation der signierten Daten ermöglichen und das bei gleichzeitigem Erhalt der Gültigkeit der Originalsignatur (editierbare Signaturen genannt).

Diese Kern-Technologien werden anschließend genutzt um eine neue Generation an Anwendungen vorzustellen. Diese Anwendungen sind: (a) Ein vertrauenswürdiges Open Government Data, das die sichere und authentische Veröffentlichung von Daten des öffentlichen Sektors ermöglicht; (b) Ein benutzerzentriertes Identitätsmanagement, das auch nur eine teilweise Bekanntgabe der Identität ermöglicht bei gleichzeitigem Erhalt der Prüfbarkeit der authentischen und qualifizierten Identitätsdaten; (c) Eine Umsetzung für eine sichere, authentische und effiziente Verfahrensabwicklung der öffentlichen Verwaltung über die Landesgrenzen hinweg basierend auf der EU Dienstleistungsrichtlinie. Die präsentierten Technologien und Anwendungen orientieren sich dabei am Europäischen Interoperabilitätsframework (EIF). Alle Entwicklungen wurden zur Evaluierung ihrer Anwendbarkeit prototypisch umgesetzt und - soweit wie möglich in operativen Umgebungen und unter realen Bedingungen - getestet.

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.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.


Graz

06.06.2014

Place/Ort

Date/Datum

Signature/Unterschrift

Signaturwert	xU1Swrpf7ZvyaL/qDZAKiHhuoDMcBxieZ8q14K2wv5Y+1QEhkYcrSpbOn+YFe0YqL548LtLhgH3MnymhSSoNLg==	
	Unterzeichner	DI Klaus Stranacher
	Aussteller-Zertifikat	CN=a-sign-premium-mobile-03,OU=a-sign-premium-mobile-03,O=A-Trust Ges. f. Sicherheitssysteme im elektr. Datenverkehr GmbH,C=AT
	Serien-Nr.	685117
	Methode	urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0
	Parameter	etsi-bka-atrust-1.0:ecdsa-sha256:sha256:sha1
Prüfinformation	Signaturprüfung unter: http://www.signaturpruefung.gv.at	
Hinweis	Dieses mit einer qualifizierten elektronischen Signatur versehene Dokument ist gemäß § 4 Abs. 1 Signaturgesetz einem handschriftlich unterschriebenen Dokument grundsätzlich rechtlich gleichgestellt.	
Datum/Zeit-UTC	2014-06-05T12:56:03Z	

Contents

Contents	iii
List of Figures	vii
List of Tables	ix
Acknowledgements	xi
List of Acronyms	xiii
1 Introduction	1
1.1 Motivation	2
1.2 Methodology and Thesis Outline	2
I Documents and Their Applications in Services	5
2 From Traditional to Electronic Administrative Services	7
2.1 Introduction	8
2.2 Traditional Services	8
2.3 Transition to Electronic Services	9
2.4 Electronic Signatures	11
3 The Need for Next-Generation Technologies and Applications	23
3.1 Introduction	24
3.2 European Initiatives	24
3.3 Next-Generation Technologies and Applications	38

II	Next-Generation Technologies for Electronic Documents	45
4	Electronic Documents Interoperability	47
4.1	Introduction	48
4.2	Interoperable Electronic Document Framework	49
4.3	Implementation	53
4.4	Deployment and Evaluation	57
4.5	Cross-border Signature Validation	59
4.6	Summary and Conclusions	71
5	Examination and Assessment of Editable Signatures	73
5.1	Introduction	74
5.2	Status Quo of Editable Signatures	75
5.3	Requirements	83
5.4	Examination	85
5.5	Assessment	87
5.6	Summary and Conclusions	90
6	An Advanced Editable Signature Scheme	93
6.1	Introduction	94
6.2	Requirements	94
6.3	Existing BDS Core Implementation	95
6.4	Architecture	98
6.5	Implementation	99
6.6	Evaluation and Conclusions	103
7	Electronic Document Processing	105
7.1	Introduction	106
7.2	Requirements	106
7.3	Architectures	107
7.4	Implementation	113
7.5	Evaluation and Conclusions	120
III	Next-Generation Applications for Electronic Documents	123
8	Next-Generation Applications for Open Government Data	125
8.1	Introduction	126
8.2	Common Requirements	127
8.3	Trusted Open Government Data - Concept	130
8.4	Trusted Open Government Data - Architectures	132
8.5	Trusted Open Government Data - Implementations	138
8.6	Evaluation and Conclusions	140

9	Next-Generation Applications for Identity Management	143
9.1	Introduction	144
9.2	Selective Disclosure	145
9.3	Requirements	146
9.4	The Model	146
9.5	The Austrian eID System	149
9.6	Application to the Austrian eID System	151
9.7	Evaluation and Conclusions	156
10	Next-Generation Public Administration Procedures	159
10.1	Introduction	160
10.2	Issues and Challenges	161
10.3	Requirements	163
10.4	General Architecture and Process Model	164
10.5	Implementation	170
10.6	Evaluation and Conclusions	172
11	Summary and Conclusions	175
11.1	Summary	176
11.2	Conclusions	180
IV	Appendices	185
A	Advanced Editable Signature Examples	187
A.1	Template Signature (enveloping)	187
A.2	Message Signature (enveloping)	191
A.3	Final XSL Transformed Result	195
B	Identity Management Example	197
B.1	Example Identity Link	197
B.2	Example Identity Link*	199
C	Publications	207
C.1	Thesis related Publications	207
C.2	Other Publications	209
	Bibliography	222

List of Figures

2.1	Wax seal	9
2.2	Typical structure of public administration processes	11
2.3	Basic principle of signature-creation	12
2.4	Basic principle of signature-verification	13
2.5	Certificate validation	14
2.6	SignedData container type	20
2.7	XMLDSIG structure	21
2.8	XAdES-BES/EPES structure	22
2.9	PDF signature	22
3.1	Timeline EU initiatives	25
3.2	EIF 2.0: Interoperability layers	30
3.3	EU Services Directive and SPOCS	36
3.4	E-Document usage	39
3.5	Interoperability issues electronic documents	40
4.1	Multi-layered container format OCD	51
4.2	Visual representation of metadata and authentication layer	52
4.3	OCD creation module	54
4.4	OCD validation and verification module	56
4.5	OCD extraction module	56
4.6	Overall evaluation of “fitness for purpose”	58
4.7	SWOT analysis of OCD modules	58
4.8	TSL library architecture	61
4.9	Process flow: TSL import module	63
4.10	Process flow: TSL verification module	64
4.11	MOA-SP architecture	67
4.12	Process flow: startup and TSL unit initialization	68
4.13	Process flow: signature verification and TSL based certificate validation	69

4.14	Results certificate validation	70
4.15	Distribution of TSLs	71
5.1	Basic principle of redactable signatures	77
5.2	Sequence diagram of redactable signatures	78
5.3	Basic principle of redactable signatures	79
5.4	Sequence diagram of sanitizable signatures	81
5.5	Basic principle of blank digital signatures	81
5.6	Sequenze diagram of blank digital signatures	82
5.7	Overview about editable signature schemes	85
6.1	Process flow: BDS core implementation	97
6.2	BDS template format	97
6.3	BDS message format	98
6.4	Architecture of the advanced editable signature scheme	99
6.5	Example: Advanced editable signature used for redaction	101
6.6	SignedInfo element of a template and message signature	103
7.1	Architecture: Data validation	108
7.2	Sequence diagram: Data validation	110
7.3	Architecture: Data extraction	111
7.4	Sequence diagram: Data extraction	112
7.5	Data validation request	114
7.6	Data validation response	115
7.7	Data validation implementation	116
7.8	Data validation configuration	116
7.9	Data extraction request	117
7.10	Data extraction response	118
7.11	Data extraction implementation	119
7.12	Data extraction configuration	120
8.1	Overview OGD and PSI Directive requirements	129
8.2	Concept for Trusted OGD - use case 1	131
8.3	Concept for Trusted OGD - use case 1	132
8.4	Server-side architecture for Trusted OGD	136
8.5	Client-side architecture for Trusted OGD	138
8.6	Server-side implementation for Trusted OGD	139
8.7	Screenshot Android app	140
9.1	A user-centric and selective disclosure enabling model for eIDs	147

9.2	The Austrian eID system	150
9.3	Sequence diagram of registration process	153
9.4	Sequence diagram of identification and authentication processes	155
10.1	Architecture: Use case 1	165
10.2	Process model: Use case 1	167
10.3	Architecture: Use case 2	169
10.4	Process model: Use case 2	170
10.5	Implementation next-generation public administration procedures	172
11.1	Thesis summary	176

List of Tables

2.1	Evaluation result against the identified requirements	16
2.2	CADES signature formats	18
5.1	Assessment summary (legal and technical) of examined editable signature schemes .	91
6.1	BDS parameter analysis	98
6.2	BDS parameter analysis	102
6.3	Evaluation result against the identified requirements	104
7.1	Evaluation result against the identified requirements	121
8.1	Summary signing capabilities	135
8.2	Evaluation result against the identified requirements	141
9.1	Evaluation result against the defined requirements	157
10.1	Evaluation result against the identified requirements	174
11.1	Key action points vs. thesis results	177
11.1	Key action points vs. thesis results	178
11.1	Key action points vs. thesis results	179
11.1	Key action points vs. thesis results	180
11.2	Individual conclusions of the main findings	181
11.2	Individual conclusions of the main findings	182
11.2	Individual conclusions of the main findings	183
C.1	Publications in journals	207
C.2	Publications at conferences	208
C.3	Other publications	209

Acknowledgements

This thesis would not have been possible without the support of my colleagues at the IAIK. Following alphabetical - but gender adjusted - list highlights colleagues and friends, which supported me in particular:

Vesna Krnjic: For being co-author of many publications. In addition, I want to thank her for all exiting discussions during photo walks and after-work *drinks*. These traditions will hopefully pursue (not only to exchange rumours) even if her family counter has been increased.

Herbert Leitold: For his support during the initial phase of my thesis and for giving me the opportunity to work as work package leader for the EU large scale pilot SPOCS.

Thomas Lenz: For our regular coffee breaks and being co-author of some publications. Furthermore, I want to thank him for giving me advices based upon his endless knowledge about nearly everything (e.g. how to detonate post-boxes or how to perform an abdominal delivery).

Christian Maierhofer: For his valuable support in implementing some of the proposed concept and models in this thesis. Additionally, I want to thank him for the inspiring discussions about photography and camera equipment, even if some of these discussions led to expensive investments.

Reinhard Posch: For being my assessor and his comprehensive knowledge about e-Government and e-Business processes.

Stefan Rass: For being my second assessor and his valuable comments on this thesis.

Christof Rath: For not being co-author of any publications, but for being a supportive friend. In addition, he deserves endless credits for being my (donkey) brother in spirit and for being the second co-founder of the cracker barrel for the first victims of feministic IAIK isolation politics.

Thomas Rössler: For his valuable and omnifarious inputs during the conception of my thesis. Additionally, I want to thank him for learning much about project management skills.

Daniel Slamanig: For giving me detailed mathematical insights into blank digital signatures, even if elliptic curves will stay as some mystery for me. In addition, I want to thank him for being a critical reviewer of some publications, which is sometimes a bit annoying, but definitely increase the quality of the publication.

Arne Tauber: For his valuable comments on this thesis and for giving me time and resources to work on my thesis. Hopefully, he will not break his neck during exercising one of his adrenalin-fuelled hobbies.

Thomas Zefferer: For being co-author of many publications and for his talent to produce pageful texts based upon the input of two single words. As tribute I will let him win the next badminton match against me - maybe.

Bernd Zwattendorfer: For being co-author of many publications and his precise and critical reviews of publications. For his future ice hockey career I wish him less injuries and a universal “body checks preventer”.

Additionally, I want to thank my mother Ingrid and my father Manfred for their selfless support. Finally, and last but not least, I want to thank my live-in partner Jessica for being my greatest supporter.

Klaus Stranacher
Graz, Austria, 2014

List of Acronyms

AdES	Advanced Electronic Signature
API	Application Programming Interface
BGBI	Bundesgesetzblatt
BDS	Blank Digital Signature
BDSS	Blank Digital Signature Scheme
CAeS	Cryptographic Message Syntax Advanced Electronic Signature
CA	Certification Authority
CEM	Certified Electronic Mail
CERT	Computer Emergency Response Team
CIP	Competitiveness and Innovation Framework Programme
CRL	Certificate Revocation List
CRR	Central Residents Register
CSP	Certification Service Provider
CSV	Comma-Separated Values
DOCX	Office Open XML Format
e-Business	Electronic Business
ECAS	European Commission Authentication System
e-CODEX	e-Justice Communication via Online Data Exchange
e-Delivery	Electronic Delivery
e-Document	Electronic Document
e-Government	Electronic Government

e-Health Electronic Healthcare
e-Justice Electronic Justice
e-mail Electronic Mail
e-Medication Electronic Medication
e-Prescription Electronic Prescription
e-Procurement Electronic Procurement
e-Safe Electronic Safe
e-Signature Electronic Signature
EC European Commission
ECMA European Computer Manufacturers Association
EDI Electronic Data Interchange
EDIAKT Electronic Data Interchange Akt
EDIFACT Electronic Data Interchange For Administration, Commerce and Transport
eID Electronic Identity
EIF European Interoperability Framework
ENISA European Network and Information Security Agency
epSOS Smart Open Services for European Patients
etc. et cetera
ETSI European Telecommunications Standards Institute
EU European Union
EUPL European Union Public License
GIS Geographic Information System
GML Geography Markup Language
GUI Graphical User Interface
HSM Hardware Security Module
ICT Information and Communication Technologies
ICT-PSP ICT-Policy Support Programme
IDA Interchange of Data across Administrations

IDABC Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens

IdM Identity Management

IdP Identity Provider

IEC International Electrotechnical Commission

IETF Internet Engineering Task Force

ISO International Organization for Standardization

ISO/IEC International Organization for Standardization/International Electrotechnical Commission

IT Information Technology

KISS Keep it simple and smart

KML Keyhole Markup Language

LSP Large Scale Pilot

MIME Multipurpose Internet Mail Extensions

MOA-ID Modules for Online Applications - Identification

MOA-SP Modules for Online Applications - Signature Verification

MOA-SS Modules for Online Applications - Signature Creation

MS Member State

OCSP Online Certificate Status Protocol

OCD Omnifarious Container for e-Documents

ODF Open Document Format

OGD Open Government Data

OID Object Identifier

PAeS PDF Advanced Electronic Signature

PDF Portable Document Format

PEPPOL Pan-European Public Procurement Online

PIN Personal Identification Number

PKCS Public Key Cryptography Standards

PKI Public Key Infrastructure

PKIX Public Key Infrastructure Exchange

PSC Point of Single Contact

PSI Public Sector Information

QC Qualified Certificate

QES Qualified Electronic Signature

RTR (Austrian) Regulatory Authority for Broadcasting and Telecommunications

SAML Security Assertion Markup Language

SHP Shapefile

SLA Service Level Agreement

SOA Service Oriented Architecture

SOAP Simple Object Access Protocol

SP Service Provider

SPOCS Simple Procedures Online for Cross-border Services

SSCD Secure Signature Creation Device

ssPIN sector-specific PIN

STORK Secure Identity Across Borders Linked

SVG Scalable Vector Graphics

SWOT Strengths, Weaknesses, Opportunities and Threats

TAN Transaction Number

TSL Trust-service Status List

TTP Trusted Third Party

UN United Nations

UTC Coordinated Universal Time

VCD Virtual Company Dossier

WP Work Package

WS Web Services

XAdES XML Advanced Electronic Signature

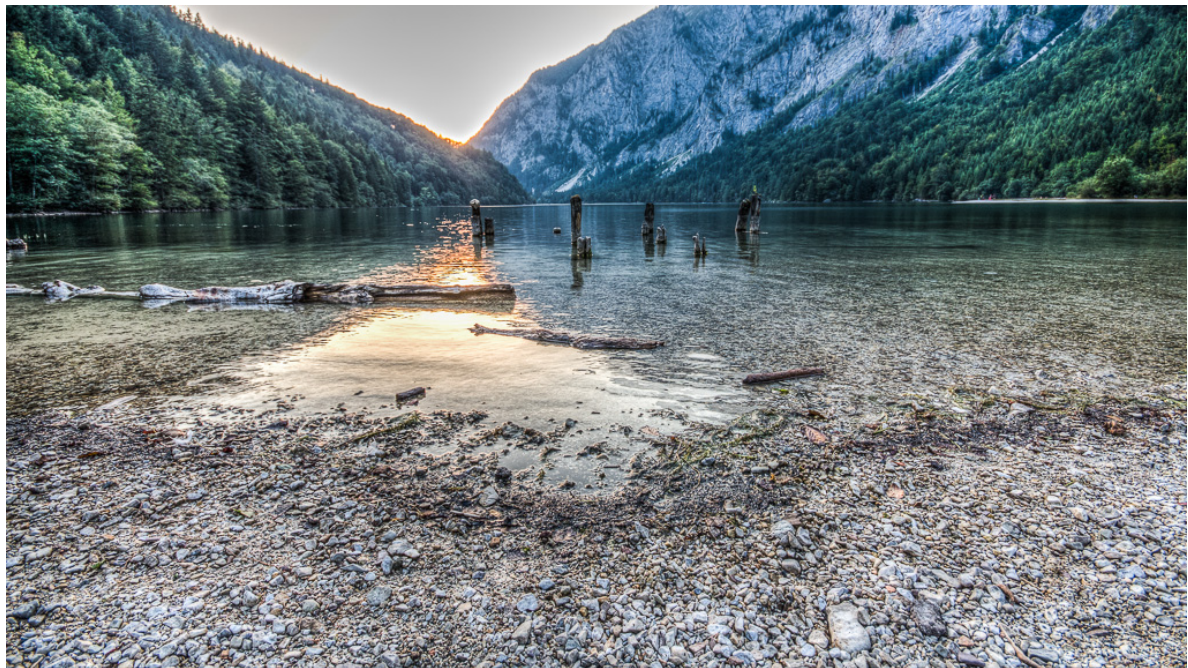
XML Extensible Markup Language

XMLDSIG XML Digital Signature

XSLT Extensible Stylesheet Language Transformations

Chapter 1

Introduction



“Bureaucracy gives birth to itself and then expects maternity benefits.”

[Dale Dauten]

1.1 Motivation

Electronic documents (e-Documents) are used to store, process, exchange or archive different kinds of information and data. For the term document different definitions exist. Oxford Dictionaries¹ defines a document as

“A piece of written, printed, or electronic matter that provides information or evidence or that serves as an official record.”

In contrast, Thefreedictionary² gives a more computer science related definition:

“A computer file that is not an executable file and contains data for use by applications.”

Due to the variety of different definitions, the term e-Document, as it will be used in this thesis, is defined as:

Definition 1 *“E-Documents” are any piece of electronic data, which can be exchanged, processed and used in applications.*

E-Documents and the exchange of e-Documents are a central element in electronic communication. Quite early questions concerning the security, authenticity and integrity of e-Documents have emerged. Hence, appropriate authentication mechanisms have been introduced to achieve reliable and trustworthy e-Documents. Thereby, electronic signatures have evolved as the most used authentication mechanism for e-Documents. Nevertheless, due to the increasing globalisation and the increasing mobility of citizens, new challenges arise.

These challenges are manifold and comprise areas such as the efficient and trustworthiness processing of e-Documents or the document exchange and mutual recognition of e-Documents across borders. As a consequence, the need of interoperability and next-generation applications for e-Documents emerges. These challenges have been taken up and motivated to the present thesis.

1.2 Methodology and Thesis Outline

The thesis has been structured based upon a well-defined methodology. Hence, the different parts and chapters reflect this methodology, which has been applied in each stage of the thesis. Hence, the thesis consists of following main parts:

- Part I - Documents and Their Applications in Services
- Part II - Next-Generation Technologies for Electronic Documents
- Part III - Next-Generation Applications for Electronic Documents

¹<http://www.oxforddictionaries.com/definition/english/document>

²<http://www.thefreedictionary.com/document>

Part I introduces the thesis and elaborates on the transition of traditional paper based services to electronic services. In addition, the need for next-generation applications for electronic documents is emphasised. In detail, this part consists of following chapters:

- Chapter 2 - From Traditional to Electronic Administrative Services
- Chapter 3 - The Need for Next-Generation Technologies and Applications

Chapter 2 illustrates the movement from traditional services to electronic administrative services. It explains where documents are used by means of a traditional administrative procedure. In particular, this chapter elaborates on the authenticity and integrity of paper based documents and how these documents are processed. Finally, the transition to electronic documents (including electronic signatures) and electronic services in the area of e-Government and e-Administration are treated.

Chapter 3 highlights the need for next-generation technologies and applications in the area of e-Documents. First of all, European initiatives in the IT-sector with focus on the public administrations are discussed. This involves historic initiatives as well as current ones. Among other things the Digital Agenda for Europe and the different EU large scale pilot projects are main topics of this chapter. In addition, this chapter elaborates on issues and challenges for future e-Document based e-Government and e-Administration processes. Thereby, the focus lies on the authenticity and integrity of e-Documents on the one side. On the other side it is discussed which restrictions exist with currently deployed electronic signatures. This mainly concerns that signed data is not modifiable without invalidating the original applied signature. Nevertheless, it is explained that applications exist, which need such functionalities. Finally, it is highlighted that current public administration procedures lack on security, authenticity and efficiency especially in a cross border context. Hence, the need for next-generation public administrative procedures arises.

Part II covers next-generation technologies for e-Documents, which focus on the authenticity and efficient processing of e-Documents. These technologies serve as core elements for the next-generation applications, developed in the last part of this thesis. Hence, following chapters exist:

- Chapter 4 - Electronic Documents Interoperability
- Chapter 5 - Examination and Assessment of Editable Signatures
- Chapter 6 - An Advanced Editable Signature Scheme
- Chapter 7 - Electronic Document Processing

Chapter 4 is twofold. On the one hand this chapter presents an interoperability framework for cross-border exchange of e-Documents, which has been developed by the author of this thesis in the course of the EU large scale pilot SPOCS³. On the other side this chapter elaborates on the signature verification, especially for cross border use cases.

Chapter 5 deals with editable signature schemes. Such schemes allow modifications of signed data, but preserve the authenticity and integrity of the unchanged data. In the last years a variety of editable signature schemes has been introduced in literature, but their capabilities to assure the integrity and authenticity in e-Administration and e-Government use cases has not been assessed so far.

³The author of this thesis was leading work package 2 (“e-Documents”) of SPOCS from July 2010 to December 2012.

Hence, this chapter evaluates and assesses selected editable signature schemes based upon identified requirements.

Chapter 6 presents the implementation of an advanced editable signature scheme. Based upon an existing core implementation, extensions have been specified and implemented to be applicable in the e-Government domain. This mainly concerns the applicability to XML-based data and that the created signatures base upon a defined and open standard for advanced electronic signatures.

Chapter 7 introduces a concept and implementation for an efficient processing of e-Documents. This comprises a comprehensive data validation, a data extraction out of available e-Documents and a data re-integration. These functionalities are combined to a processing unit, which enables an efficient processing of e-Documents due to a reduced necessity of manual interactions.

Part III represents the last part of the thesis and combines the developed next-generation technologies to create next-generation applications. Thereby, this part comprises following chapters:

- Chapter 8 - Next-Generation Applications for Open Government Data
- Chapter 9 - Next-Generation Applications for Identity Management
- Chapter 10 - Next-Generation Public Administrations Procedures

Chapter 8 discusses next-generation applications for open government data (OGD). In the last years open government data has emerged and has significantly influenced the IT sector. Based upon the definition of e-Documents, open government data fall into this category too. Given the growing relevance and popularity of OGD, security issues have been astonishingly rarely discussed so far. Hence, this chapter introduces a novel approach for a trusted OGD by using conventional and editable signatures.

Chapter 9 treats on next-generation applications in the area of identity management. Identity management bases upon identity data, which represent an identity document. This chapter introduces a novel approach for a user-centered identity management enabling selective disclosure of identity attributes. This approach focuses on the application on national eID solutions, which have been rolled out for a long time. Core elements of this approach are editable signatures, which allow to reveal only a subset of available identity data by remaining their verifiability.

Chapter 10 presents a new approach to achieve secure, authentic and efficient public administration procedures across borders. Thereby the focus lies on the use cases of the EU Services Directive. This approach bases upon the developed next-generation technologies - namely the interoperability framework for cross-border exchange of e-Documents, editable signatures and the efficient processing of e-Documents.

Finally, *Chapter 11* summarizes the thesis and draws conclusions.

Remark: Appendix C contains the publication list of the thesis author. Additionally, for all thesis-related publications, the relations to the corresponding chapters are given. That means it is highlighted which publications served as a basis for the respective chapter.

Part I

Documents and Their Applications in Services

Chapter 2

From Traditional to Electronic Administrative Services



“The science of today is the technology of tomorrow.”

[Edward Teller]

2.1 Introduction

Public administration has a long history and started already in the ancient times. Hence, this chapter elaborates on ancient public administrations in Section 2.2 briefly. Additionally, this section treats on traditional - paper-based - public administration processes. In the last decade many public administration moved to the electronic world and created an electronic version of public administration. This transition to electronic government (e-Government) is treated in Section 2.3. In particular, this section deals with the typical structure of electronic public administration procedures. Already in traditional services authenticity and reliability played a vital role and this continues for e-Government. The means of choice to achieve trustworthiness in the electronic world are electronic signatures. Section 2.4 elaborates in detail on electronic signatures as they are also a key aspect for the remainder of the present thesis.

2.2 Traditional Services

Already the ancient Egyptians, Romans and Greeks had public administration or predecessors in place. These public administrations have changed and improved over the centuries and have led to modern public administrations in the 20th century. Following list gives a brief historical overview about the evolution of traditional public administrations (according to [Mathes, 2008; Wikipedia.org, 2014a,c,d,b]):

- The ancient Egypt had a very strong structured civil service. Their code of conduct was defined by numerous doctrines. If civil servants followed these rules, they had a high renown. To undersign documents, i.e. to proof their authenticity and integrity, the civil servants used seals, which were handed over by the pharaoh.
- In contrast, the ancient Greeks were the first who built up their public administration upon legislation. That means all civil servants were obliged to follow the rules given by the different laws.
- In the ancient Rome also civil servants were responsible for the public administration. Their principles strongly differed if they were applied to Roman citizens or to residents in the conquered regions. That means for the Roman citizens a duty of care were in force, whereas the other residents were treated as subjects, which had to pay taxes and duties. During the Empire, the civil servants were in debt of the emperor.
- This royal obligation continued until the Early Middle Ages, whereas the king gained more and more influence. In the Late Middle Ages the kings lost ground and the princes evolved to the most important decision makers. Hence, the royal civil servants lost influence to the princely civil servants. To sign documents wax seals were used (see Figure 2.1), whereas the colour indicate the rank of the person.
- In the 18th century the princes lost more and more influence due to the French Revolution and the regent gained on power. As consequence, civil service systems evolved, whereas the civil servants were not in debt of the regent. In addition, the wax seal was replaced by rubber stamps and hand-written signatures.



Figure 2.1: Wax seal

- In the 19th century these civil service systems started to evolve to the modern public administration systems of the 20th century, which is characterised by constantly changes to achieve more modernisation.

2.3 Transition to Electronic Services

2.3.1 Electronic Government

In the mid-90s public administration started to move to the electronic world. The booming Internet and the new electronic means, which have been provided by the information and communication technologies (ICT), created new opportunities but also additional challenges for public administration. This transition to the electronic world has become known as *electronic government (e-Government)*. For e-Government various definitions exist. For the further considerations, it is referred to following definition. According to the eEurope 2005 action plan (cf. Section 3.2.2.1) e-Government is defined as follows¹:

Definition 2 “*E-Government*” means the use of information and communication technologies in public administrations combined with organisational changes and new skills.

In addition, e-Government can be subdivided into two categories:

¹See http://europa.eu/legislation_summaries/information_society/strategies/124226b_en.htm

Internal e-Government: That means the use of ICT in the public sector without point of contact to the citizens. This concerns for instance the back-office processing or internal applications.

External e-Government: This comprises mainly services for citizens and enterprises, which are available via the Internet.

Furthermore, e-Government takes place on several levels. The following main levels exist²:

Legal level: Public administration and thus e-Government base upon a comprehensive legal framework. This legal framework has to be taken into account by all other levels.

Technical level: This level provides the appropriate technical solutions, which are in accordance with the legal framework.

Organisational level: On this level the appropriate organisational decision must be made to assure the inclusion of new electronic solutions into the given infrastructures.

One of the main pillars of e-Government are public administration procedures. These procedures cover a lot of different applications, but have common main objective which is to have a consistent process from the application of the citizen to the delivery of the official decision back to the citizen without any media breaks. Therefore a typical and generalised structure for such media break resistant procedures can be defined. Figure 2.2 illustrates this structure. According to [Posch et al., 2011] such a procedure can be subdivided into three basic stages: (a) application, (b) processing, and (c) delivery.

The application stage constitutes the first step in a public administration procedure. The applicant³ requests an application at a portal. For this purpose, the applicant fills out an online form and attaches the required electronic documents⁴. This can be done via direct file upload, upload via an eSafe, or via any other resource (depending on the portal's functionalities and general infrastructure). These form data and the uploaded documents are then sent to the back-office. In the back-office, the processing takes place. That means the application is processed in the particular competent authorities (CAs). At the end of this stage, the last CA issues the decision concerning the application. Finally, in the delivery stage this decision is sent to the applicant. Depending on the legal regulations for the application, the decision can be delivered informally via e-mail, a portal-internal delivery system, or via an electronic delivery system.

Essential for implementing of such procedure is the authenticity and integrity of the exchanged data. Actually, the means of choice are *electronic signatures* to ensure the trustworthiness of the data. Electronic signature are used to sign data and documents. Thus they represent the electronic pendant to non-electronic means, such as a handwritten signature. Furthermore, electronic signatures play an important role in identity management. Thereby, they enable a high-level authentication of person using electronic means only. Based upon this importance of electronic signatures for e-Government in general and for the remainder of the present thesis in particular, the following section discusses electronic signatures in detail.

²For sure, also other levels exist. For instance the sociocultural level, which deals with the user acceptance of e-Government.

³An applicant can be a citizen or business or even a public official (in G2G or G2E applications for instance).

⁴Also other forms of application are possible, e.g. via e-mail.

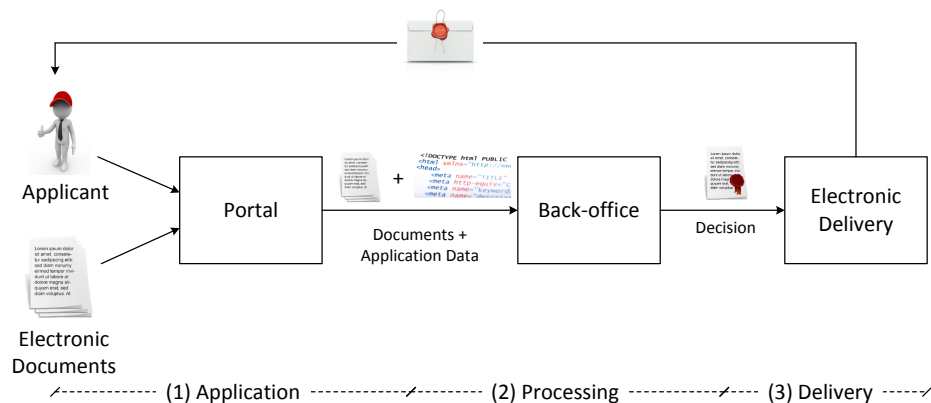


Figure 2.2: Typical structure of public administration processes

2.4 Electronic Signatures

2.4.1 Overview

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 identify⁵ the creator of the signature⁶ (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.

2.4.2 Basic Principle

The technical basis for electronic signatures, which are applied in the e-Government area, is public key cryptography. Following subsections explain the cryptographic signature creation and verification (including a brief description of the RSA algorithm as example) as well as the additional needed certificate validation.

2.4.2.1 Cryptographic Signature Creation and Verification

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 signature⁷. Figure 2.3 illustrates the basic principle of a typical signature-creation process. In a first step, the data to be signed is mapped to a hash value⁸

⁵The quality of this identification strongly depends on the identification process at the certification service provider - see Section 2.4.2.1.

⁶The creator of a signature is also called signatory.

⁷An important characteristic is that the private key cannot be determined out of the public key and is infeasible to guess.

⁸Also referred to digest value

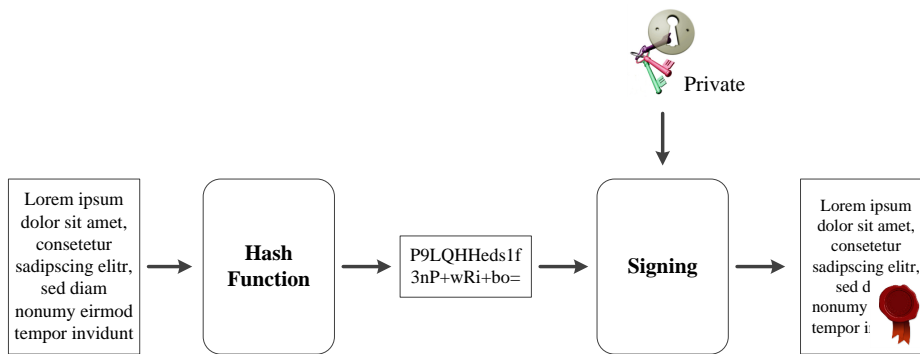


Figure 2.3: Basic principle of signature-creation

of a fixed length using a so called hash function⁹. This hash value is then signed using the signatory's private key. The corresponding public key is published and the receiver of the signed data is able to verify the validity of the signature by means of this public key.

The public key is usually published via a trusted third party - the certification service provider (CSP) - using an electronic certificate. This certificate holds the public key of the signatory and binds the signatory's identity to this key. Thereby, the quality of the identification process is essential for the quality of the issued certificate. Only, if the signatory is uniquely identified by the CPS, e.g. via a personal identification using a photo ID, the receiver is able to uniquely identify the signatory. This qualified identification is an essential requirement to issue a so called qualified certificate (cf. Section 2.4.3)

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 Figure 2.4. This process consists of a cryptographic signature verification and a so called certificate validation.

For the cryptographic signature verification, the original hash value is revealed from the signature by using the signatory's public key. Then the verifier computes the hash value over the received signed data. The resulting hash value is then compared to the original hash value. If these two hash values match, the signatures has been verified as valid, otherwise invalid.

The following subsection gives the RSA algorithm as a concrete example for a signing algorithm.

2.4.2.2 RSA Signing Algorithm

RSA has been invented by Rivest, Shamir and Adleman in the year 1977 [Rivest et al., 1978] but is still used today. Following the algorithm, which is divided into the key generation and signature creation/verification phase, is briefly explained.

⁹A hash function (or digest method) 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 reasons for applying a hash function are to prevent and easy detect forgeries as well as that the data to be signed is quite large and signing large data is very inefficient and time consuming for practical applications.

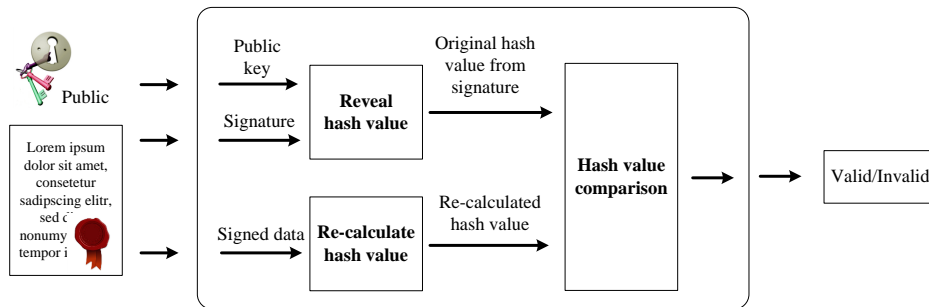


Figure 2.4: Basic principle of signature-verification

Key Generation Initially, the key generation is responsible to create the public and private key. It consists of following steps:

- Generation of two prime numbers p and q of the same size approximately.
- Calculation of the prime number product $n = p \cdot q$.
- Calculation of the Euler function $\phi(n) = (p - 1)(q - 1)$.
- Choice of a number e according to¹⁰: $\gcd(e, \phi(n)) = 1$.
- Calculation of the number d so that $e \cdot d \equiv 1 \pmod{\phi(n)}$.
- \Rightarrow **(e, n)** represents the **public** key.
- \Rightarrow **(d, n)** represents the **private** key.

Signature Creation and Verification The signature is created by the signer, holding her private key, and signs a message m via following steps:

- A hash function H is applied to the message m and creates the hash value h : $h = H(m)$.
- The signature is calculated over the hash value by using the private key: $signature = h^d \pmod{n}$

The signature and the message are sent to the respective entity. This entity, the verifier, verifies the signature as follows:

- The signed hash value h_{signed} is revealed by using the public key of the signer and calculating $h_{signed} = signature^e \pmod{n}$, whereas following applies (without proof): $h_{signed} = signature^e \pmod{n} = (h_{signed}^d)^e \pmod{n} = h_{signed}$.
- Then the hash value is re-calculated over the received message m' : $h_{re-cal} = H(m')$
- In case the signed hash value and the re-calculated hash value are the same (i.e. $h_{signed} = h_{re-cal}$), the signature is valid, otherwise invalid.

More details on the RSA algorithm can be found in Menezes et al. [1996] and Rivest et al. [1978].

¹⁰ \gcd = greatest common divisor

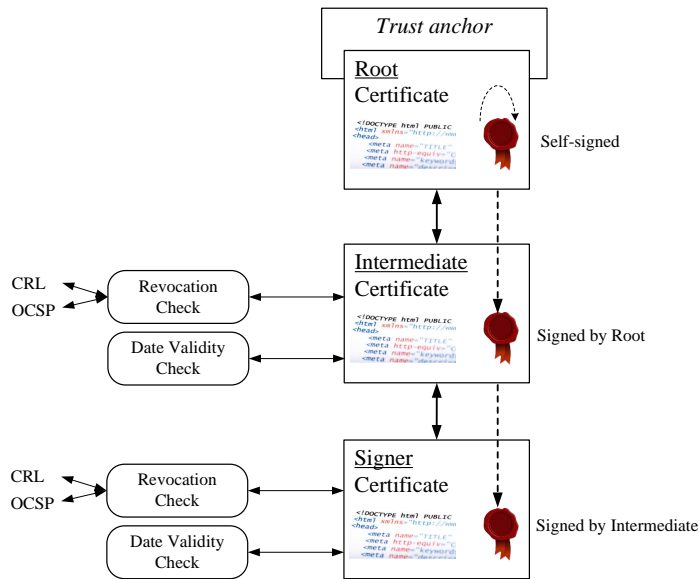


Figure 2.5: Certificate validation

2.4.2.3 Certificate Validation

Beside the cryptographic signature verification, the validation of the validity of the signer certificate is also an important part of an entire signature verification process. Thereby the certificate validation consists of following process steps (see also Figure 2.5) ¹¹:

Certificate chain: Based upon the signer certificate, a certificate chain is built up to a root certificate, which serves as trust anchor and this is usually named trusted root certificate. In Figure 2.5, this chains consist of the signer certificate via an intermediate certificate to the self-signed trusted root certificate.

Date and time verification: Verify if the verification time is within the time validity of each certificate in the certificate chain (except the trusted root certificate). That means if the verification time is between the date/time values `notBefore` and `notAfter` given in the certificate.

Revocation status verification: Verify the revocation status for each certificate in the certificate chain (except the trusted root certificate). That means, verify via the given revocation mechanism (CRL ¹² or OCSP ¹³) if the certificate is revoked.

Only if a certificate chain could be found and all certificates are valid in terms of time and all certificates are not revoked, the result of the certificate validation is positive.

¹¹According to Public Key Infrastructure Exchange (PKIX) specification [Cooper et al., 2008].

¹²Certificate Revocation List.

¹³Online Certificate Status Protocol.

2.4.3 Legal Framework

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 [The Council of the European Union, 2000] and their national implementations. The Directive formally defines three different types of signatures (see also following definitions):

- Electronic signature
- Advanced electronic signature
- Qualified electronic signature

An *electronic signature* is defined as;

“Electronic signature means data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication.”
[The Council of the European Union, 2000]

This definition is very general. In particular, it does not state anything about the identification of the signatory, which is essential for many e-Government and e-Business processes. Hence, the Directive defines an *advanced electronic signature* an electronic signature, which must meet following additional 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;”* [The Council of the European Union, 2000]

Thereby, requirement (b) is of particular importance as this enables the identification of the signatory and is therefore an important prerequisite for the legal recognition of electronic signatures. Table 2.1 compares these requirements and how they are fulfilled usually.

Concerning the legal effects the Directive states that

*“advanced electronic signatures which are based on a qualified certificate and which are created by a secure-signature-creation device:
(a) 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; [...]”* [The Council of the European Union, 2000]

In literature, such a signature is usually called *qualified electronic signature*, although this term is formally not defined in the Directive.

Thereby, a qualified certificate must fulfil further requirements. On the one hand these requirements concern the certificate itself (Annex I of the Signature Directive) and on the other hand it defines requirements for the certification service providers issuing qualified certificates (Annex II). The former

Table 2.1: Evaluation result against the identified requirements

Requirement	Fulfilled through
<i>“(a) it is uniquely linked to the signatory”</i>	The same private and public key must be unique (at least within an certification service provider).
<i>“(b) it is capable of identifying the signatory”</i>	This is fulfilled by: (a) it is practical impossible that a key pair is generated twice, (b) it is ensured that a signature, which is verifiable with the public key, could only be created by using the associated private key and (c) it is practical impossible that the private key can be calculated out of the public key. Therefore, an appropriate registration process at the certification service provider is needed to uniquely identify the signatory.
<i>“(c) it is created using means that the signatory can maintain under his sole control”</i>	The triggering of the signature creation must only be done by the entitled person.
<i>“(d) it is linked to the data to which it relates in such a manner that any subsequent change of the data is detectable”</i>	This is ensured by using the hash function as it is practical impossible that (a) different electronic data with the same hash value exist and (b) other electronic data have the same - given - hash value.

mainly defines which content the qualified certificate must have¹⁴. The latter define mainly reliability and accessibility requirements the certification service provider must fulfil¹⁵. An certification service provider issuing qualified certificates must be voluntary accredited or supervised by the designated Member State (or a delegated public or private body).

Finally, a secure-signature-creation device must fulfil additional requirements defined in Annex III of the Directive, which mainly define the protection of the signature creation data (e.g. the private key). These requirements define mainly the data processed must be sufficient secured and protected against forgery. The conformity of secure-signature-creation devices with these requirements shall be determined by appropriate public or private bodies. Whereas the verification of these requirements for qualified certificate and secure-signature-creation device is quite easily achievable in national and closed environments, it generates interoperability issues for cross-border applications (cf. Section 3.3.1.2).

To summarise, a qualified electronic signature is legally equivalent to an handwritten signature. Both, advanced electronic signatures and qualified electronic signature play a vital role on e-Government processes as they allow the identification of the signatory. To facilitate this identification, advanced electronic signature formats have been specified. As the identification of the signatory is carried out via the signer certificate¹⁶ and the previous registration of the signatory at the certification

¹⁴For instance, the qualified certificate must indicate that it is an qualified certificate and must contain data to identify the signatory and the certification service provider.

¹⁵For instance such requirements are that the certification service provider must operate an appropriate revocation mechanism and use trustworthy systems.

¹⁶That means the certificate of the signatory.

service provider, an appropriate protection of the signer certificate must be ensured by the advanced electronic signature format. The following subsection discusses such formats in detail.

2.4.4 Advanced Signature Formats

Different existing signature formats have led to interoperability issues. These issues mainly affects the verifiability of electronic signatures especially in a cross-border context. For instance a proprietary signature formats from Member State A, may not be verifiable in the other Member States. To eliminate these issues and to facilitate the identification of the signatory, the European Commission has published reference formats for advanced electronic signatures. Hence, these formats are of special interest for our analysis. Following reference formats are defined in this Commission Decision 2011/130/EC [European Commission, 2011a]¹⁷:

- CAAdES-BES/EPES signatures
- XAdES-BES/EPES signatures
- PAdES-BES/EPES (Part 3) signatures

Following subsections explain these formats more detailed.

2.4.4.1 CAAdES

CAAdES means CMS Advanced Electronic Signature and represents an ETSI¹⁸ standard. This standard is published in ETSI TS 101 733 [ETSI, 2013a] and bases upon CMS (Cryptographic Message Syntax [Housley, 2009]). Main objective of CAAdES is to extend CMS to be applicable as advanced electronic signature. This concerns mainly the protection of the signer certificate, which is covered by the CAAdES types Basic Electronic Signature (BES) and Explicit Policy Electronic Signature (EPES)¹⁹. The basis format CMS relies on PKCS#7 [Kaliski, 1998] and enables to encrypt and sign data. The data are encoded as ASN.1²⁰ and is stored in `ContentInfo`-containers. These containers can be nested into each other and contain a container type, whereas following types are supported:

- `Data`: This type denotes arbitrary data.
- `SignedData`: Indicates signed data.
- `EnvelopedData` and `EncryptedData`: This type indicates encrypted data.
- `DigestData`: This type is used to denote data, which is extended with a digest value to assure integrity.

¹⁷Meanwhile this decision has been ammended [European Commission, 2014a]. As these amendments shall be apply by 1st December 2014 and do not directly influence the thesis, the thesis still refers to the not amended version of the decision.

¹⁸The European Telecommunications Standards Institute (ETSI) is an European standardisation body (<http://www.etsi.org/>).

¹⁹Some additional forms exist, which focus on the long-term validation of signatures. Nevertheless, the BES and EPES form fulfil the requirements for an advanced electronic signature.

²⁰Abstract Syntax Notation 1 are a description language for defining, structuring and representing data.

- `AuthenticatedData`: Denotes data, which are secured with an MAC²¹.

Additional specifications define also other container types such as `SignedAndEnvelopedData` [Kaliski, 1998]. This type enables to sign data and then to encrypt the signed data²².

To create a CADES signature, CMS is taken as basis. Now CADES defines how to include the additional parameters, which are need for an advanced electronic signature. Thereby the CADES specification [ETSI, 2013a] defines several formats, which are listed and explained in Table 2.2.

Table 2.2: CADES signature formats

Format	Description
CADES-BES (Basic Electronic Signature)	Represents a CMS signature with the additional protection of the signer certificate.
CADES-EPES (Explicit Policy-based Electronic Signature)	Extends CMS and CADES-BES to include an explicit signature policy, which is signed too. This policy must be applied during the signature verification.
CADES-T (Electronic signature with Time)	Is a BES or EPES signature plus a signature timestamp from a trusted third timestamping service.
CADES-C (Electronic Signature with Complete validation data references)	This format adds additional references of all used certificates in the certificate chain (from the signer certificate to the root certificate).
CADES-X Long (Extended Long format)	A CADES-C signature is extended with the concrete values of the certificate and the concrete values of the revocation information.
CADES-A (Archival Form)	Extends a CADES-X Long signature with one or more archive timestamps to prevent future weaknesses of cryptographic algorithms.

In the following the container type `SignedData` is explained in more detail, as this type also contains the CADES extensions. Figure 2.6 illustrates the structure of this type, which consists of following elements:

- `Version`: Represents the version number of the underlying specification.
- `DigestAlgorithms`: Contains a set of digest algorithms, which can be used within `signerInfos`.
- `EncapsulatedContentInfo`: Represents the signed data. Beside the signed data (given in the container `eContent`) also the type (container: `eContentType`) of these data is given. In case the data are contained in the container, then it is an *enveloping* signature. If the container `eContent` is empty, then it is a *detached* signature and the signed data is externally stored.

²¹Message Authentication Code.

²²The same functionality can be achieved by nesting a `SignedData` container into an `EnvelopedData` container.

- **Certificates:** Optionally contains certificates, which are used for the signature verification.
- **Crls:** Optionally contains information about the revocation status of certificates.
- **SignerInfos:** Represents the signature itself and has the container type `SignerInfo`.

The `SignerInfo` container is also shown in Figure 2.6 and consists of following elements:

- **Version:** Represents the version number of the underlying specification.
- **SignerIdentifier:** Specifies the signer certificate or the appropriate public key.
- **DigestAlgorithm:** Indicates the digest algorithm which has been used to create the digest value.
- **SignedAttributes:** This element represents additional signed attributes for the CADES extension. This signed attributes contains the value which are needed for a CADES-BES²³ or CADES-EPES²⁴ signature.
- **SignatureAlgorithm:** Specifies the used signature algorithm.
- **Signature:** Represents the signature value.
- **UnsignedAttributes:** This element contains additional attributes, which are covered by the signature²⁵.

2.4.4.2 XAdES

Parallel to CADES the XAdES (XML Advanced Electronic Signatures) standard has been developed. It bases upon the W3C recommendation on XML-signatures (XMLDSIG) [Bartel et al., 2008] and enables the creation of XML-based advanced electronic signatures. XAdES is an ETSI standard and is published in ETSI TS 101 903 [ETSI, 2010b]. As for CADES, XAdES defines different signature forms (analogous to Table 2.2), whereas the BES and EPES form fulfil the requirements of advanced electronic signatures.

The structure of XMLDSIG is illustrated in Figure 2.7. This structure consists of following elements:

- **SignedInfo:** This element encapsulates all information about the signature and represents the element, which is used as input for the signature calculation.
 - **CanonicalizationMethod:** Defines an algorithm for a data normalisation²⁶ of the `SignedInfo` element.
 - **SignatureMethod:** Specifies the used signature algorithm.

²³For instance: attribute `signingTime`, which indicates the signing time.

²⁴For instance: attribute `id-aa-ets-sigPolicyId`, which specifies the signature policy.

²⁵This element is used to include additional attributes for the other CADES format, such as the signature timestamp for CADES-T.

²⁶This normalisation deletes for instance whitespace, which may cause problem during verification of the signature.

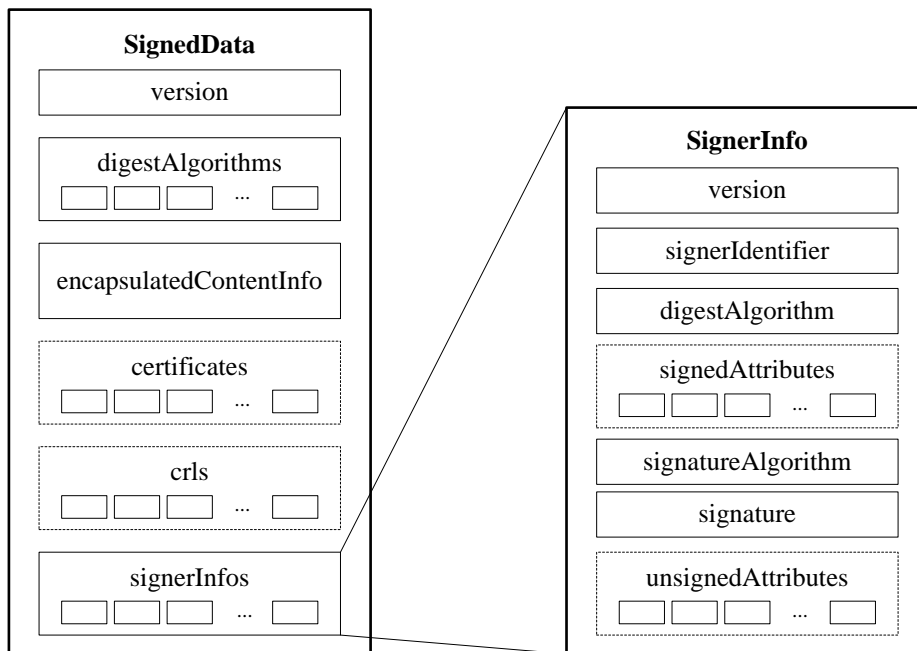


Figure 2.6: SignedData container type

- **Reference:** This element references signed data (data objects), whereas several `Reference` elements can be specified. Usually, the data objects are referenced via the attribute `URI`. This mechanism allows creating enveloping, detached and enveloped signatures²⁷. Except for the `URI` attribute following elements can or must be specified:
 - * **Transforms:** As an option, transformation can be defined. These transformations are applied on the reference data before calculating the digest value. Thereby the result of the last transformation serves as input for the digest calculation.
 - * **DigestMethod:** Specifies the digest method.
 - * **DigestValue:** Represents the digest value.
- **SignatureValue:** Represents the signature value calculated over the `SignedInfo` element including the digest values of the specified references.
- **KeyInfo:** This optional element usually holds the signing certificate.
- **Object:** This element includes data objects, which are usually referenced via the `Reference` element in `SignedInfo`.

Based upon this structure, extensions to create a XAdES signature are specified. This is done via an additional `Reference` element, which refers to an `Object` element containing the needed extension information. Thus this information is signed too. This information is usually called the XAdES properties. These properties define additional data to fulfil the requirements for advanced

²⁷Compared to CAAdES, XAdES allows also enveloped signatures, which means that the signature itself is placed within the signed data.

```

<ds:Signature ID?>
  <ds:SignedInfo>
    <ds:CanonicalizationMethod/>
    <ds:SignatureMethod/>
    (<ds:Reference URI? >
      (<ds:Transforms>)?
      <ds:DigestMethod/>
      <ds:DigestValue/>
    </ds:Reference>)+
  </ds:SignedInfo>
  <ds:SignatureValue/>
  (<ds:KeyInfo>)?
  (<ds:Object>)*
</ds:Signature>

```

Figure 2.7: XMLDSIG structure

electronic signatures. Figure 2.8 illustrates the structure of an XAdES-BES/EPES signature, whereas the given `Object` element represents the XAdES properties. These properties consist of following elements:

- `SigningTime`: Contains the signing time as UTC format.
- `SingingCertificate`: This element contains a unique reference to the signing certificate.
- `SignaturePolicyIdentifier`: To create a XAdES-EPES signature a signature policy can be given.
- `DataObjectFormat`: This element indicates the MIME type of the signed data.

As for CADES, XAdES also defines further signature formats, which are compliant to the CADES formats given in Table 2.2.

2.4.4.3 PAdES

PAdES are PDF Advanced Electronic Signatures and are also specified by ETSI. The standard is published in ETSI TS 102 778 [ETSI, 2010a] and defines an advanced electronic signature format for PDF based signatures. The PDF standard specifies that internally a CMS signature is created, which is then embedded into the PDF document. This embedding is done via a so called `signature directory` (see Figure 2.9). This `signature directory` consists of two entries. The first entry defines a `ByteRange` and the second entry contains the signature itself. The `ByteRange` indicates which bytes in the PDF document are signed. As the signature itself is included, the signature must be excluded from the signed data obviously. That means, in the Figure 2.9 the bytes 520 to 870 are not set to zero, whereas this zero values are replaced with the signature after the signature creation.

PAdES signatures use the same mechanism to embed the signature into the `signature directory`. Thereby, PAdES has the same capabilities as XAdES or CADES signatures. It defines how the current PDF specification is used to create an advanced electronic signature according to the formats given in Table 2.2.

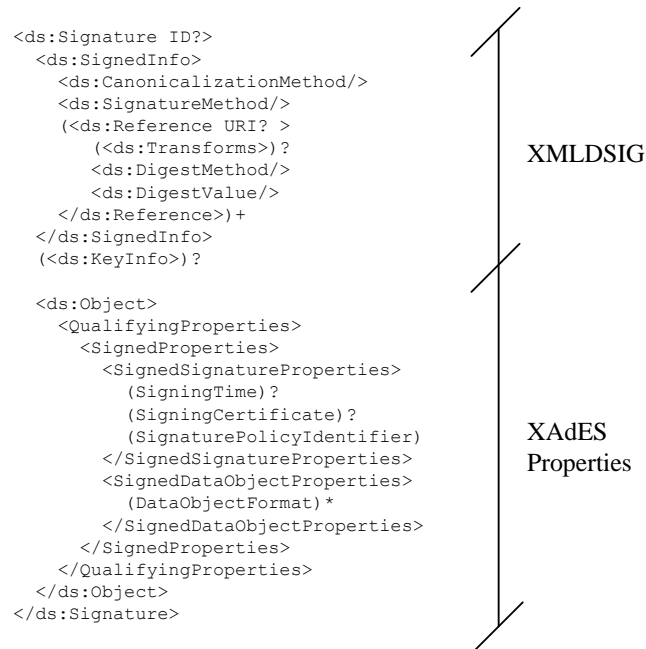


Figure 2.8: XAdES-BES/EPES structure

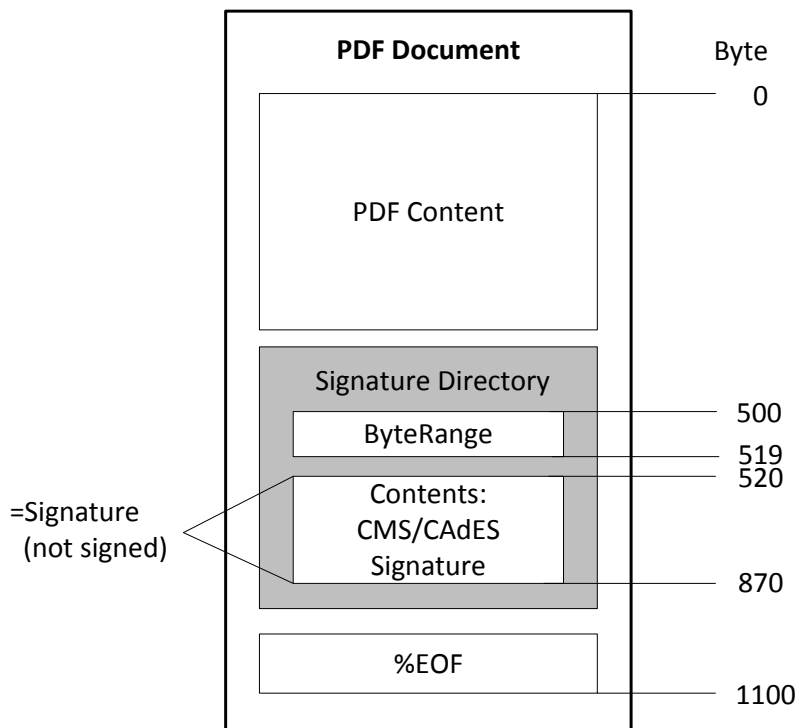


Figure 2.9: PDF signature

Chapter 3

The Need for Next-Generation Technologies and Applications



“Frustration, although quite painful at times, is a very positive and essential part of success.”

[Bo Bennett]

3.1 Introduction

Globalisation affects citizens and enterprise worldwide. Nowadays different people with different cultures interact and trade goods or services. Additionally, the mobility of citizens and enterprises increases and creates administrative burdens for them. The European Single Market¹ tries to reduce the barriers for those citizens and enterprises. This should ensure a free trade of goods, services and capital as well as a free movement of peoples². In particular in Europe with its high diversity of different countries and cultures, this is a great challenge.

This chapter is twofold. In Section 3.2 different European initiatives are summarised, which main objective was or still is to stimulate a consolidation of the European citizens and enterprises and thus creating a European Single Market. This section includes historic as well as actual political and strategic initiatives, which form the basis for this intended consolidation. In addition, European interoperability programmes are discussed, which have been taken place or are still active. Furthermore the European legal framework is outlined, whereas the focus is given on European Directives and Regulations, which influenced the thesis. Finally, this section concludes with a brief overview of the European Large Scale Pilot projects.

In Section 3.3 the need for next-generation technologies and applications in the area of e-Documents are discussed. First of all, this section highlights the interoperability issues for e-Document exchange and signature verification across borders. Following it is highlighted that the digital document sanitizing problem raises issues, which cannot be covered by conventional electronic signatures. All these issues raise the need for next-generation technologies. Furthermore, the Digital Agenda for Europe [European Commission, 2010a] and the e-Government Action Plan [European Commission, 2010d] raise issues, which cannot be solved by current applications. Hence, this section highlights the need for next-generation applications - especially in the area of open government data, identity management and public administration procedures. Finally, this section contains concrete action points, which are taken up by the remainder of the thesis.

3.2 European Initiatives

3.2.1 Overview

In the last centuries different European initiatives have been realised or are still in their realisation phase. Figure 3.1 illustrates the main historic and actual initiatives on a timeline. Thereby following groups of initiatives exist

Political and strategic initiatives: They comprise political and strategic commitments from the European Commission to achieve pan-European interoperability.

Interoperability programmes: To realise the objectives of the political and strategic initiatives, different interoperability programmes have been launched to develop appropriate concepts.

¹http://ec.europa.eu/internal_market/index_en.htm.

²Known as the EU's "four freedoms".

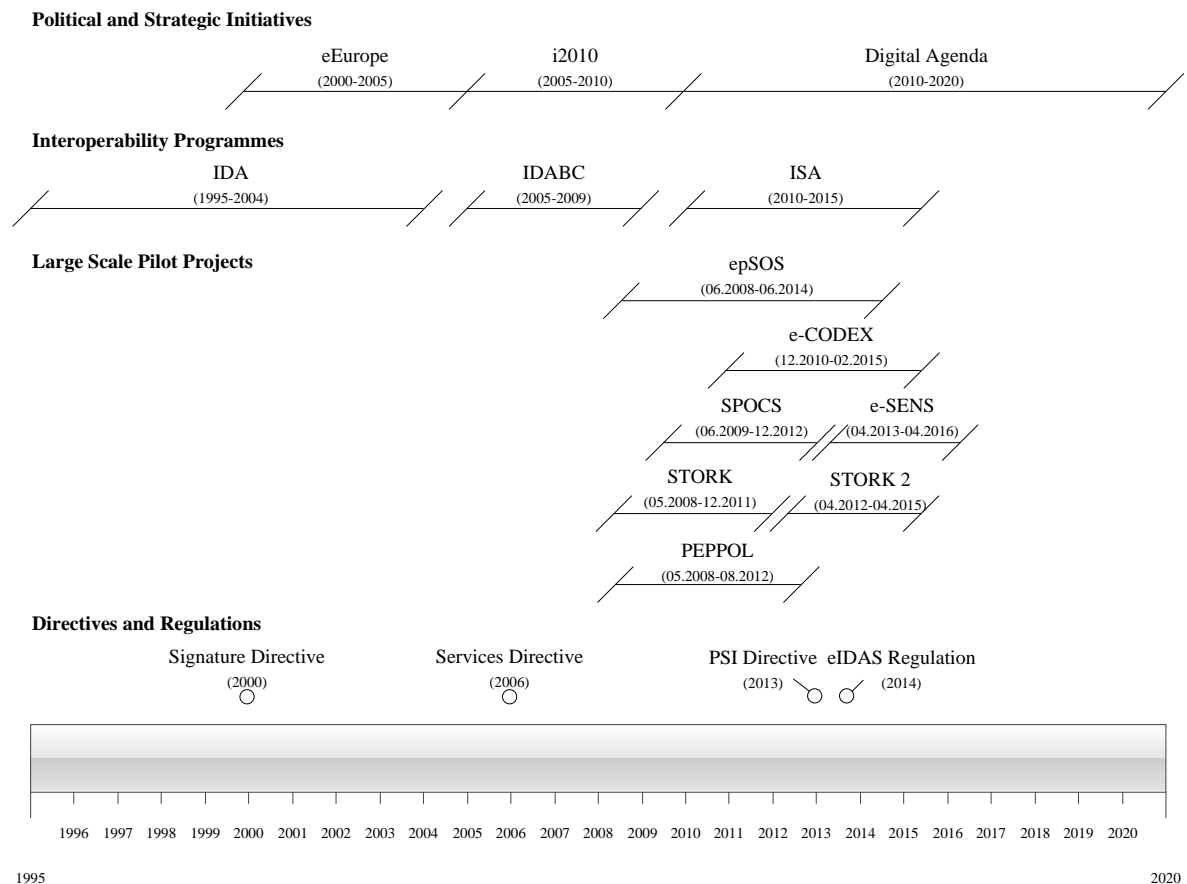


Figure 3.1: Timeline EU initiatives

Large scale pilot projects: This comprises so called Pilots Type A project, which have been launched by the European Commission. Main objective of these projects is to interconnect existing national infrastructures and thus create interoperability across borders.

Directives and Regulations: They comprise the appropriate legal framework which is needed to create a common single market³.

In the following subsections these initiatives are described in more detail.

3.2.2 Political and Strategic Initiatives

3.2.2.1 eEurope (2000-2005)

In March 2000 a special meeting was held by the European Council in Lisbon. Main objective of this meeting was to . . .

“ . . . agree a new strategic goal for the Union in order to strengthen employment, economic reform and social cohesion as part of a knowledge-based economy.” [Council, 2000]

³Remark: only Directives and Regulations, which are related to the thesis are quoted.

Based upon this meeting the initiative eEurope [European Commission, 2000] was launched which main intention is to

“... shift to a digital, knowledge-based economy, prompted by new goods and services, will be a powerful engine for growth, competitiveness and jobs. In addition, it will be capable of improving citizens’ quality of life and the environment.” [Council, 2000]

The initiative eEurope started in the year 2000 and lasted until 2005. It was accompanied with the eEurope 2002 Action Plan [Council of the European Union and European Commission, 2000] and its successor eEurope 2005 Action Plan [European Commission, 2002]. Main objectives of the eEurope initiative were⁴:

- *“Bring every citizen, home and school, every business and every administration into the digital age and online.”*
- *“Create a digitally literate Europe, supported by an entrepreneurial culture ready to finance and develop new ideas.”*
- *“Ensure that the whole process is socially inclusive, builds consumer trust and strengthens social cohesion.”*

3.2.2.2 i2010 (2005-2010)

The i2010 strategy bases upon the eEurope 2005 Action Plan and is the successor of the eEurope initiative. The strategy has been officially published in [European Commission, 2005] and has following priorities:

- *“the completion of a Single European Information Space which promotes an open and competitive internal market for information society and media;”*
- *“strengthening Innovation and Investment in ICT research to promote growth and more and better jobs;”*
- *“achieving an Inclusive European Information Society that promotes growth and jobs in a manner that is consistent with sustainable development and that prioritises better public services and quality of life.”* [European Commission, 2005]

In context of the i2010 initiative and the ICT research priority, the European Commission has launched two research programmes:

- Seventh Research Framework Programme (FP7)⁵
- Competitiveness and Innovation Programme (CIP)⁶

⁴According to http://europa.eu/legislation_summaries/information_society/strategies/124221_en.htm.

⁵http://ec.europa.eu/research/fp7/understanding/fp7inbrief/home_en.html.

⁶<http://ec.europa.eu/cip/>.

Thereby, the Competitiveness and Innovation Programme consists of three sub-programmes, whereas within the sub-programme “ICT-Policy Support Programme (ICT-PSP)” the Large Scale Pilot projects have been launched (cf. Section 3.2.5). In addition, in 2006, the i2010 e-Government Action Plan [European Commission, 2006b] started and lasted until 2010. The main objective of this action plan was to make public service more efficient and modern.

3.2.2.3 Digital Agenda for Europe

Europe 2020⁷ is the European Union’s growth strategy for the next ten years. This strategy comprises following main targets: employment, research & development, climate & energy sustainability, education and poverty & social exclusion. One of these flagships is the Digital Agenda for Europe, which aims...

“... to chart a course to maximise the social and economic potential of ICT, most notably the internet, a vital medium of economic and societal activity: for doing business, working, playing, communicating and expressing ourselves freely.” [European Commission, 2010a]

The Digital Agenda has identified the seven most significant problem areas, which hinder the growth of the European ICT area and thus aggravate a common digital single market. According to [European Commission, 2010a] these seven obstacles are:

- Fragmented digital markets
- Lack of interoperability
- Rising cybercrime and risk of low trust in networks
- Lack of investment in networks
- Insufficient research and innovation efforts
- Lack of digital literacy and skills
- Missed opportunities in addressing societal challenges

To address these obstacles the Digital Agenda defines following key actions:

A vibrant digital single market: The European digital market is widely fragmented and aggravates cross-border activities such as cross-border transactions and cross-border identification of citizens and enterprises. Hence, a high priority is given to facilitate cross-border transactions and comprises - among other activities - revisions of the EU Signature Directive [The Council of the European Union, 2000] and the PSI Directive [European Commission, 2003] (cf. Section 3.2.4).

⁷<http://ec.europa.eu/europe2020/>.

Interoperability and standards: Interoperability and open standards are seen as one of the key enabler for a digital single market. Therefore the EU standardisation policy will be reformed. In particular this concerns the European Interoperability Framework EIF (cf. Section 3.2.3.2) and the ISA programme (cf. Section 3.2.3.3).

Trust and security: Amongst other activities, the European Network and Information Security Agency (ENISA) will be modernised and a Computer Emergency Response Team (CERT) for EU institutions will be established to increase trust and security.

Fast and ultra-fast internet access: Fast internet access is seen as a key element for a growing economy. Hence, actions will be taken into account to increase the number of citizens and enterprise having access to high-speed broadband networks.

Research and innovation: Also research and innovation are considered to be a key enabler for a growing economy. Therefore, investments into ICT research and development should be significantly increased.

Enhancing digital literacy, skills and inclusion: To create a digital single market, a sufficient amount of employers with ICT skills are essential. Thus, the ICT competences must be increased as well as the mobility of these people.

ICT-enabled benefits for EU society: EU citizens and enterprise should profit from more ICT enabled processes and procedures. Amongst other activities, this concerns e-Government public services in particular. Hence, e-Government services should be fully interoperable, even for cross-border transactions.

Based upon the i2010 e-Government Action Plan a new e-Government Action Plan [European Commission, 2010d] has been published for the period 2011-2015. The main objective of this action plan is to support. . .

“... the transition from current eGovernment to a new generation of open, flexible and collaborative seamless eGovernment services at local, regional, national and European levels that will empower citizens and businesses.” [European Commission, 2010d]

3.2.3 Interoperability Programmes

3.2.3.1 IDA (1995-2004)

The Interchange of Data across Administrations (IDA) programme was launched in 1995 and ended 2004. Main objectives of this programme were⁸:

- *“to achieve a high degree of interoperability between the telematic networks in the Member States and between the Community and the Member States;”*
- *“to make such networks converge towards a common telematic interface between the Community and the Member States;”*

⁸According to http://europa.eu/legislation_summaries/information_society/strategies/124147a_en.htm.

- *“to achieve benefits for Member State administrations and the Community resulting in particular from the streamlining of operations, a reduction in maintenance, speeding up the implementation of new networks and the provision of safe and reliable data interchange;”*
- *“to extend the benefits of these networks to EU businesses and citizens;”*
- *“to promote the spread of best practice and encourage the development of innovative telematic solutions in administrations.”*

The IDA programme could not achieve all of its targets. Hence, a new interoperability programme was launched.

3.2.3.2 IDABC (2005-2009)

IDABC stands for Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens and is the successor of the IDA programme and lasted from 2005 until 2009. The main target of IDABC was to provide e-Government service for administrations, enterprises and citizens. In more detail, the programme aimed to⁹:

- *“enable the interchange of information between public administrations, as well as between such administrations and the Community institutions;”*
- *“facilitate the delivery of pan-European services to businesses and citizens taking account of their needs;”*
- *“achieve interoperability across different policy areas, notably on the basis of a European Interoperability Framework;”*
- *“promote the spread of good practice and encourage the development of innovative telematic solutions in public administrations.”*

A key target was to establish a European Interoperability Framework to increase the interoperability between e-Government services.

European Interoperability Framework (EIF) Among other achievements version 1.0 of the *European Interoperability Framework (EIF)* [European Commission, 2004] was issued. Meanwhile version 2.0 [European Commission, 2011b] has been published under the ISA programme (cf. Section 3.2.3.3). Thereby, EIF 2.0 defines four layers of interoperability (see also Figure 3.2):

Legal interoperability: This is needed as public administrations work within the national legal framework and different legal frameworks in different countries lead to incompatibilities.

Organisational interoperability: Means interoperability on the level of business processes, organisational relationships and change management.

Semantic interoperability: The exchanged information underlies different interpretation in different countries (due to language, culture, etc.). Thus precise meanings and formats must be ensured.

⁹According to http://europa.eu/legislation_summaries/information_society/strategies/124147b_en.htm.

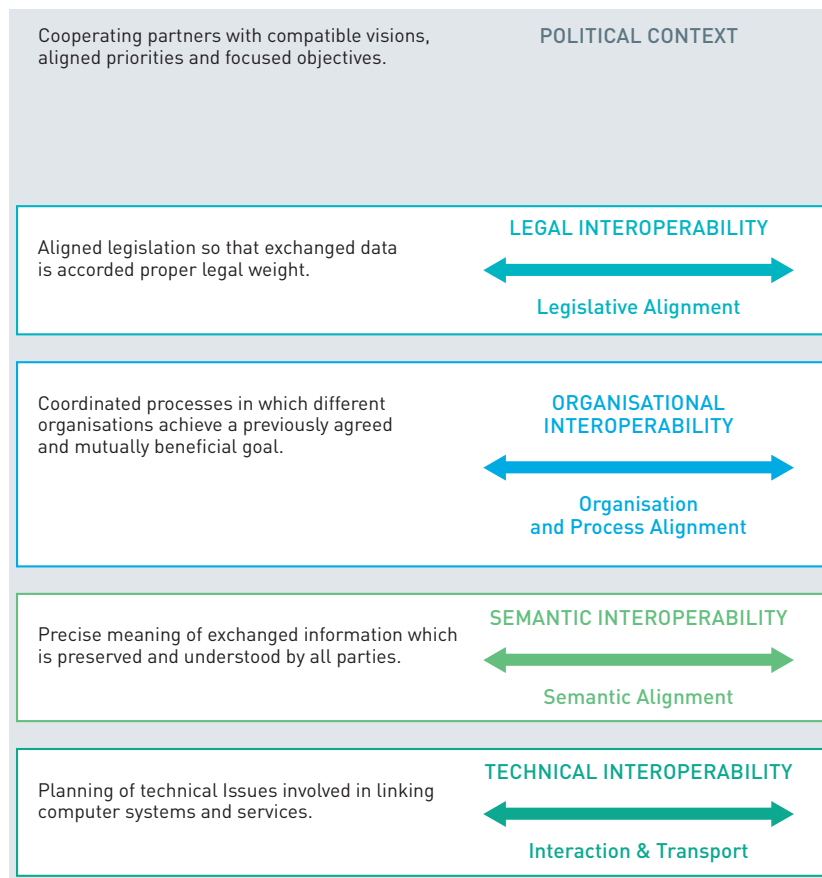


Figure 3.2: EIF 2.0: Interoperability layers [European Commission, 2011b]

Technical interoperability: Different national infrastructures use different technical specifications. Thus interoperability is needed, if these infrastructures are interconnected.

Furthermore, EIF 2.0 defines 12 main principles separated into three categories. These principles should be taken into account for developing and maintaining European public services.

3.2.3.3 ISA (2010-2015)

The new Interoperability Solutions for European Public Administrations (ISA) programme has been launched in 2010 and has replaced the IDABC programme. The main objectives of ISA are¹⁰:

“ISA supports and facilitates efficient and effective cross-border electronic collaboration between European public administrations. The programme enables the delivery of electronic public services and ensures the availability, interoperability, re-use and sharing of common solutions.”

¹⁰According to http://ec.europa.eu/isa/index_en.htm.

ISA has started numerous activities to address different interoperability issues. These activities address following clusters¹¹:

Trusted information exchange: This cluster deals with the secure transfer of data. Amongst others it takes into account the findings of different Large Scale Pilot projects and interoperability issues of electronic signatures.

Interoperability architecture: This cluster elaborates on the alignment of cross-border and cross-sector infrastructures to achieve interoperability between them.

Assessment of the ICT implications of new EU legislation: Assess and considers ICT implications in an early stage of new legislative rules to ensure a timely implementation of the legislation.

Accompanying measures: Sets up accompanying measures to raise awareness and ensure the recognition of interoperability as one of the key elements for building public services.

3.2.4 Directives and Regulations

3.2.4.1 Signature Directive

As described in Section 2.4.3¹² the EU Signature Directive [The Council of the European Union, 2000] forms the legal framework for electronic signatures within Europe. The main purpose of the Directive is...

“... to facilitate the use of electronic signatures and to contribute to their legal recognition.” [The Council of the European Union, 2000]

and to establish...

“... a legal framework for electronic signatures and certain certification-services in order to ensure the proper functioning of the internal market.” [The Council of the European Union, 2000]

This Directive had to be implemented by the Member States. For instance, in Austria this has been done by the Austrian Signature Law [Republik Österreich, 2010]. Meanwhile, the so called eIDAS Regulation (cf. Section 3.2.4.4) has been published and will replace the Signature Directive and their national implementations by 1st July 2016.

3.2.4.2 Services Directive

The EU Services Directive [European Commission, 2006a] has been announced in the year 2006. Main objective of this Directive is to reduce the barriers to establish and carry out services in foreign Member States. Therefore the Directive foresees the establishment of so called Points of Single Contact (PSC). These PSCs are responsible for handling all required processes needed to establish

¹¹According to http://ec.europa.eu/isa/actions/index_en.htm.

¹²Within this section a more detailed view on the defined characteristics of electronic signatures is given.

and carry out services across borders. That means a service provider, who wants to go abroad, is only required to contact the PSC and not the different competent authorities (CA) in the background. Thereby the main obligation is that all required procedures must be able by electronic means.

3.2.4.3 PSI Directive

The PSI (public sector information) Directive regulates the re-use of public sector information in Europe. Thereby, the first revision of the Directive [European Commission, 2003] had a very traditional and conservative view on this re-use. This has been changed by the new revision of the Directive, which has been published in the year 2013 [European Commission, 2013b]. Compared to the first revision the updated Directive comprises following main changes:

- The new revision bases upon open data and incorporates most of the open data principles (cf. Section 8.2.1).
- The scope of the Directive has been amended. Now also museum, archives and libraries fall under the Directive.

The main objective of the PSI Directive is to establish

“... a minimum set of rules governing the re-use ...” [European Commission, 2013b]

and to provide

“... the practical means of facilitating re-use of existing documents held by public sector bodies of the Member States.” [European Commission, 2013b]

The term re-use is well defined in the Directive and means

“the use by persons or legal entities of documents held by public sector bodies, for commercial or non-commercial purposes other than the initial purpose with the public task for which the documents were produced.” [European Commission, 2013b]

Furthermore the Directive defines:

- Requirements for the processing of requests on the re-use of information, which includes the definition of deadlines and the possible explanations of rejection.
- Public sector information should be available in an open and machine-readable format including appropriate meta data.
- The re-use can be charged, but these charges are limited and must be transparent for the users.
- Public sector information can be reused with or without a license.
- Definition of precautionary measures for non-discrimination and fair competition.

3.2.4.4 eIDAS Regulation

The Regulation on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market (eIDAS) has been approved by the European Parliament in March 2014. The main objective of this Regulation is . . .

“... to enhance trust in electronic transactions in the internal market by providing a common foundation for secure electronic interaction between businesses, citizens and public authorities, thereby increasing the effectiveness of public and private online services, electronic business and electronic commerce in the Union.” [European Commission, 2014b]

Thereby the Regulation is not only going to replace the EU Signature Directive and its national implementations, but also to enhance the scope. Hence, the Regulation defines following key elements (chapters):

Electronic identification: This chapter elaborates on electronic identification (eID) in the European Union. The main property is that there will be no central eID system in Europe. Instead the national eID systems are used, whereas these national systems can be notified by the Member State. Each notified system must be recognised by the other Member State if the assurance level is substantial or high¹³. That means this chapter defines the legal framework for electronic identities (for natural persons and legal entities).

Trust services: In this chapter qualified and non-qualified trust services are treated, whereas requirements, liability, supervision and accreditation for qualified trust services are well defined focusing on a cross-border verifiability. Following concrete trust services are defined (separated in different sections):

Electronic signatures: This section enhances and expands the *acquis* of the EU Signature Directive to be applicable in cross-border and cross-sector use cases.

Electronic seals: Electronic signatures, as defined in the Regulation, are applicable for natural persons only. Electronic seals provide very similar functionalities for legal entities.

Electronic time stamp: This section gives legal evidence for electronic time stamping services.

Electronic registered delivery service: This section defines requirements for qualified electronic registered delivery services and elaborates on the legal effect of electronic registered delivery services.

Website authentication: Within this section requirements for qualified certificates for website authentication are defined.

Electronic documents: This chapter defines legal effects on electronic documents and states:

“An electronic document shall not be denied legal effect and admissibility as evidence in legal proceedings solely on the grounds that it is in electronic form.” [European Commission, 2014b]

¹³The assurance levels are defined in Article 6 of the Regulation.

Finally, the eIDAS Regulation comprises some additional delegated and implementing acts, which will be essential for the implementation and application of the Regulation in the different Member States. In general, the eIDAS Regulation must be implemented by 1st July 2016 - except some defined articles and paragraphs. These exceptions mainly concern the European Commission, which is responsible for coordinating further technical details on the respective articles and paragraphs.

3.2.5 Large Scale Pilot Projects

3.2.5.1 Overview

Within the (meanwhile run out) Competitiveness and Innovation Framework Programme¹⁴ (CIP) different Large Scale Pilot projects have been launched. Main objective of these projects are to specify, implement and pilot ICT enabled cross-border services in following areas:

- Healthcare
- Procurement
- Mobility of citizens and enterprises
- Justice
- Public administration

All of these projects had or still have to prove their applicability in a piloting phase. In this phase the developed solution must be deployed in real life infrastructures of the participating countries. Thereby, main issue was or is not to develop and deploy a central system, but to establish an interoperability layer on top of the different national systems. Via this interoperability layer the national systems are interconnected. The following subsections briefly discuss the different LSP projects.

3.2.5.2 e-CODEX

The project e-CODEX¹⁵ (e-Justice Communication via Online Data Exchange) deals with interoperable e-Justice applications and services. Main objective is that EU citizens and enterprises have access to legal cases across borders and to improve the interoperability between legal authorities.

The increasing mobility of citizens and enterprises leads to more transnational proceedings and lawsuits. This requires the collaboration of different national judicial systems. In parallel, the use of ICT makes legal proceedings more transparent, efficient and economic. Simultaneously citizens, enterprises and public administrations profit from an easier access to legal cases.

Hence, e-CODEX works on an interoperable European e-Justice system. The planned solution must consider the juridical autonomy and subsidiarity. The national infrastructures and e-Services must be taken into account and must not be replaced by a central system. Instead e-CODEX is working on an interoperability layer to connect the different national solutions.

¹⁴http://ec.europa.eu/cip/index_en.htm.

¹⁵Website: <http://www.ecodex.eu/>, Project duration: December 2010 - February 2015.

3.2.5.3 epSOS

The LSP project epSOS¹⁶ (Smart Open Services for European Patients) works in the area of electronic healthcare and will last until end of June 2014.

Main objective of epSOS is to establish a service infrastructure to facilitate the communication system between the different national healthcare systems in Europe. Thus it works on the interoperability issues which occur by connecting these healthcare systems. Since April 2012 the developed epSOS systems are tested to prove their applicability. This enables European patients to use following cross-border services (if they are medicated in one of the piloting countries):

- Access to all important medical data (patient summary) about the medical treatment and therapy
- Usage of electronic prescription, i.e. usage of e-Prescription and e-Medication systems

3.2.5.4 PEPPOL

Together with STORK, PEPPOL¹⁷ (PanEuropean Public Procurement OnLine) was the first LSP project and was working in the e-Procurement area to facilitate procurement across borders.

Public sector entities are the biggest purchaser in Europe. Nevertheless, the exchange of electronic data between the public sector entities and the suppliers lacks compared to the private sector. Hence, the main objective of PEPPOL was to facilitate the procurement process especially across borders. Therefore PEPPOL created common standards for the electronic communication between enterprises and public institutions, responsible for the procurement of goods. Meanwhile the OpenPEPPOL association¹⁸ has taken over the responsibility of PEPPOL and continues the work on the established specifications.

3.2.5.5 SPOCS

The LSP project SPOCS¹⁹ (Simple Procedures Online for Crossborder Services) worked on the further development of the implementation of the EU Services Directive. Figure 3.3 illustrates the overall scenario of SPOCS in context to the EU Services Directive. In this scenario a service provider from Member State B (MS B) wants to open a business in another Member State A (MS A). Therefore, she contacts the PSC of MS A and applies for an application. By using content syndication PSC A and PCS B exchange information such as document equivalency. Then, PSC A forwards the application to the different competent authorities in the background. These CAs process the application and issue an official decision about the application. This decision is sent to PSC A, where the decision is delivered to the service provider by using an e-Delivery system across borders. That means, SPOCS worked on a second generation for PSCs enabling opening a business in a foreign Member State by using electronic means only.

¹⁶Website: <http://www.epsos.eu/>, Project duration: July 2008 - June 2014.

¹⁷Website: <http://www.peppol.eu>, Project duration: May 2008 - August 2012.

¹⁸http://www.peppol.eu/about_peppol.

¹⁹Website: <http://www.eu-spocs.eu/>, Project duration: June 2009 - December 2012.

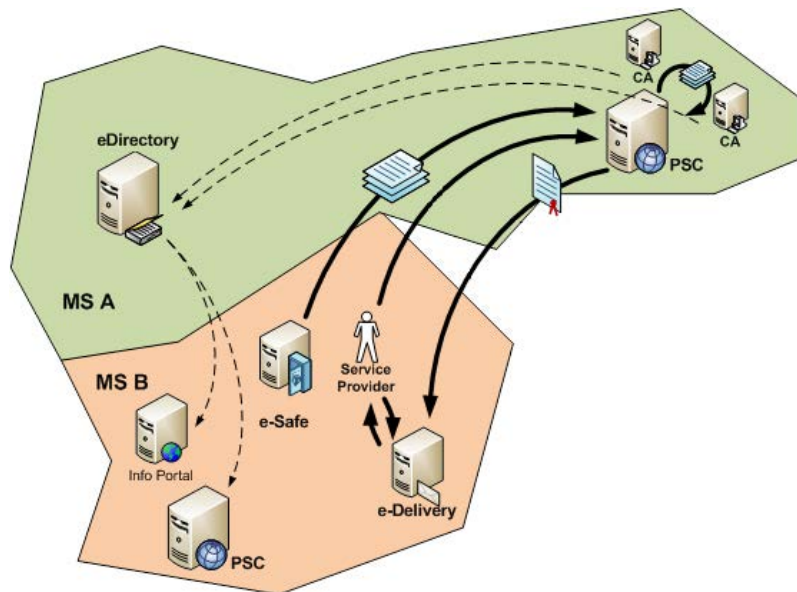


Figure 3.3: EU Services Directive and SPOCS [Rössler et al., 2011]

3.2.5.6 STORK and STORK 2.0

End of 2011 the LSP project STORK²⁰ (Secure idenTity acrOss boRders linKed) finished. Main objective of STORK was to establish an interoperable identity framework. This framework enables to European citizens to identity and authentication at a foreign service by using their national electronic identity (eID). The developed interoperability framework has been tested in several pilots to prove its applicability. These pilots were:

Pilot 1 - Cross-border authentication: This pilot dealt with the integration of cross-border authentication into different portals.

Pilot 2 - Saferchat: This pilot established an online portal for the safe communication between students. Thereby, only students between a certain age were allowed to log in a the portal.

Pilot 3 - eID student mobility: Within this pilot, foreign students have been enabled to authenticate at their guest university.

Pilot 4 - eID electronic delivery: This pilot dealt with the electronic delivery (e-Delivery) of electronic data across borders. Thereby, the interoperability layer has been set on top of the different existing national certified electronic mail (CEM) systems.

Pilot 5 - EU citizen change of address: In this pilot the scenario of changing the address across borders has been tested.

Pilot 6 - ECAS integration: Within this pilot the STORK framework has been added as authentication mechanism to the central European Commission Authentication System (ECAS).

²⁰Website: <https://www.eid-stork.eu>, Project duration: May 2008 - December 2011.

Since April 2012 the successor project STORK 2.0 has started²¹. The focus of STORK 2.0 lies on the identification and authentication of legal persons as well as on the power to representation (between legal and natural persons) across borders. That means STORK 2.0 extends the existing interoperability framework of STORK to include legal persons and electronic mandates solutions of the different national eID solutions.

3.2.5.7 E-SENS

The Large Scale Pilot project e-SENS²² (Electronic Simple European Networked Services) tries to merge the solutions of the previous LSP projects to create a European digital single market.

Despite the solutions of the previous LSP projects, still a lot of barriers exist, which hinders a cross-border usage of public administration services. Hence an increased bureaucracy effort still exists for European citizens and enterprises. Without appropriate interoperability measures between the different national public administrations it is not possible to achieve efficient public services across borders. Thus the main tasks of e-SENS are:

- Interconnect national public administration services
- Expansion of public administration services based upon a European standard infrastructure

The main objectives of this LSP are:

- Facilitate operating and opening a business in its own country or any other Member State by electronic means
- Improve the support of citizens, which go abroad for work or apprenticeship (or any other training)

²¹Website: <https://www.eid-stork2.eu/>; Project duration: April 2012-April 2015.

²²Website: <http://www.esens.eu/>, Project duration: April 2013 - April 2016.

3.3 Next-Generation Technologies and Applications

This section highlights the needs for next-generation technologies and applications in the area of e-Documents. It assesses the current situation and derives challenges and issues for the upcoming usage of e-Documents - especially in the sector of interoperability, security and efficient processing. These challenges and issues have been taken up by the present thesis. To facilitate this take up, key action points for the thesis have been defined based upon the identified challenges and issues. Thereby, each key action point summarises the concrete challenge and gives the concrete thesis chapter, where this challenge is treated in detail.

3.3.1 Next-Generation Technologies

3.3.1.1 Interoperable Document Exchange

Electronic documents are one of the main pillars in electronic communication. In particular, this applies for the exchange of information in e-Government based processes and procedures. Current frameworks for the exchange of data or e-Documents focus on the pure data exchange and/or do not take into account the needs for e-Government based processes. Examples for that are UN/EDIFACT²³ and ebXML²⁴. Approaches for the e-Government area exist, such as VCD²⁵ or EDIAKT²⁶. Nevertheless, these approaches are tailored to specific use cases and/or are focused on national infrastructures.

In place of all e-Government use cases, where e-Documents need to be exchanged between different entities across borders, Figure 3.4 shows where e-Documents are exchanged based upon the EU Services Directive. Thereby, e-Documents are used. . .

1. . . in the communication between service providers and PSCs
2. . . in the internal communication between PSCs and CAs
3. . . in the internal communication between different CAs
4. . . as content in e-Safe²⁷ applications
5. . . as content within e-Delivery applications

This list does not claim to be complete, but highlights the most important exchange paths in the use cases of the EU Services Directive. The list shows that there are many exchanges which have a cross-border nature. That means the affected e-Documents must be exchanged across borders. This leads to interoperability issues as there exists no appropriate framework, which enables a secure and automatic processing of e-Documents. That means there is the need for an interoperable framework

²³UN/EDIFACT stands for United Nations Electronic Data Interchange For Administration, Commerce and Transport and represents a cross-sectoral format for the exchange of data.

²⁴Represents an XML based standard for the exchange of data in business processes

²⁵The Virtual Company Dossier (VCD) represents a set of tools for the exchange of information in pan-European e-Procurement processes.

²⁶EDIAKT is an Austrian standard for the exchange and archiving of electronic records.

²⁷An electronic safe (e-Safe) is an application, which enables a citizen to store e-Documents in a secure manner online. Via this e-Safe needed e-Documents, such as a birth certificate, can be attached when submitting an application.

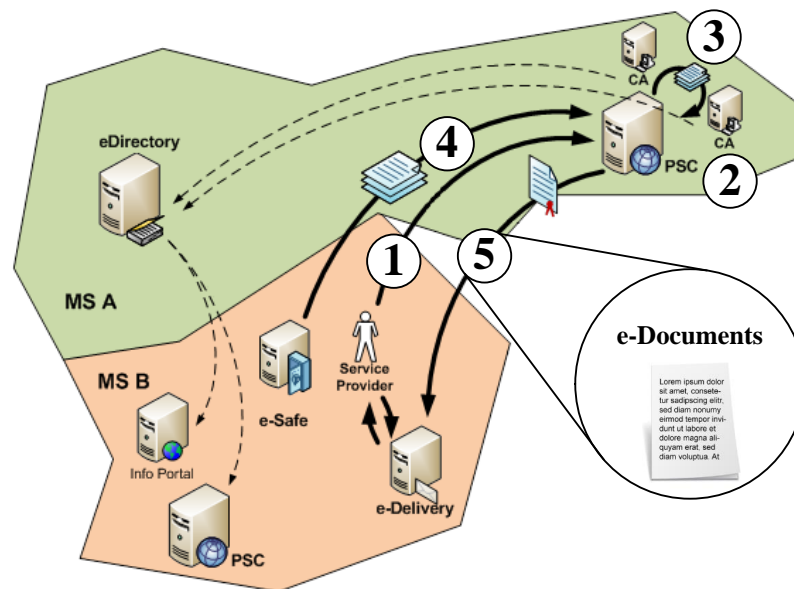


Figure 3.4: E-Document usage on the basis of the EU Services Directive and the LSP SPOCS

for exchanging e-Documents, especially in cross-border scenarios. Hence following key action point can be stated:

Key action point 1 *An interoperability framework for the exchange of e-Documents in (government based) cross-border scenarios is needed (see Chapter 4).*

3.3.1.2 Issues of Electronic Signatures

For electronic signatures also interoperability issues exist. These issues mainly concern the verification of the signature. According to Figure 3.5 interoperability issues exist in two areas:

Signature formats: For electronic signatures a variety of different signature formats exist. This variety reaches from open standards to proprietary formats. This leads to interoperability issues, as the verifiability of - especially proprietary formats - cannot be ensured. In particular in a cross-border context, a proprietary format, specified in Member State A, is usually not verifiable or even processable by other Member States. This issue has already been discovered by the European Commission. Hence, the Commission has published a Commission Decision [European Commission, 2014a]. This decision defines on the one hand reference signature formats (cf. Section 2.4.4) for processing of documents signed electronically by competent authorities under the EU Services Directive. These reference formats follow open and well established standards and thus ensure a verifiability and processability across borders. On the other hand, it defines, that if still a proprietary format is used by a Member State, an appropriate and open verification service must be deployed.

Certificate validation: Concerning the certificate validation the main issue is the verification if a signature is a qualified electronic signature and thus is equivalent to a handwritten signature (as

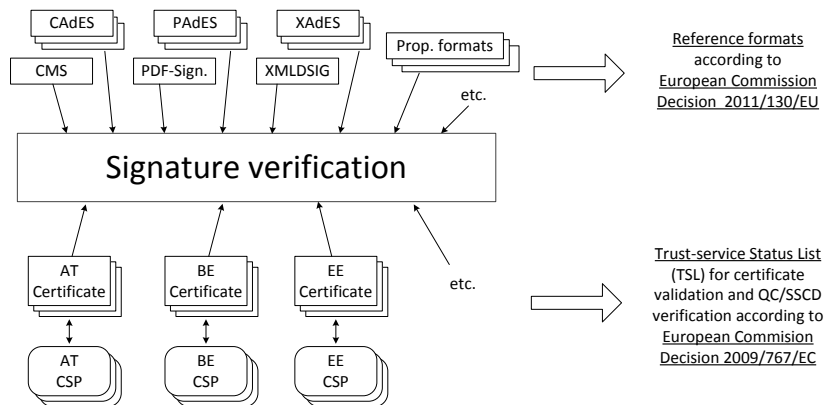


Figure 3.5: Interoperability issues electronic documents

the signatory can be uniquely identified). This is a vital property and prerequisite for many applications in the area of e-Documents²⁸ and identity management²⁹ (cf. Section 2.4.3). Thereby a qualified electronic signature must be an advanced electronic signature, which bases upon a qualified signer certificate (QC) and was created by using a secure-signature-creation device (SSCD). That means the verification of the QC and SSCD property is essential. Unfortunately, this verification creates interoperability issues as these properties depend on the national certification service providers (CSPs). The amount of CSPs in the EU Member States varies and causes issues when a timely verification of the QC and SSCD properties is needed. Hence, the European Commission has published a Commission Decision [European Commission, 2009a] on trusted-service status lists (TSLs). That means each Member State is obliged to publish a TSL, which includes all national CSPs, which are able to issue qualified certificates (cf. Section 4.5).

The first issue on signature formats has been solved by the Commission Decision on reference signature formats and the required availability of verification services for other signature formats. Concerning the second issues, the European Commission has published a tool, called SD-DSS³⁰, for the verification of electronic signature based upon trusted-service status lists. Nevertheless, the Austrian signature verification service MOA-SP³¹ lacks on this issue and does not support trusted-service status lists. This leads to following key action point:

Key action point 2 *An extension of the Austrian signature verification service is needed to support the use of TSL as well as the support of verifying the QC and SSCD property (see Chapter 4).*

3.3.1.3 Digital Document Sanitizing Problem

An important property of (conventional) electronic signatures is that any modification of the signed data can be immediately detected and leads to an invalid signature. Let us assume having a signed

²⁸For instance, only a contract signed with a qualified electronic signature unfolds its legal meaning.

²⁹For a high-level authentication of persons, identity management systems rely on a unique identification of the signatory.

³⁰<https://joinup.ec.europa.eu/software/sd-dss/>.

³¹<https://joinup.ec.europa.eu/software/moa-idspss>.

document. This document includes private and personal data, which must be censored before sending it to another party or even publish it anywhere. While censoring these private and personal data prevents from publishing non-disclosable data, the integrity of the censored data can no longer be verified. This is called the digital document sanitizing problem, first published by Miyazaki et al. [2003].

In literature, so called editable signatures have been published, which allow for (certain) modifications of the signed data, while retaining the verifiability of the applied signatures. Such modifications allow for more sophisticated applications even in the e-Government area (cf. Section 3.3.2). Nevertheless, the applicability of editable signature in the e-Government domain has never been assessed so far. This creates the following action point:

Key action point 3 *Assessment of editable signature schemes and implementation of an editable signature scheme, which is applicable in the e-Government context (see Chapters 5 and 6).*

3.3.1.4 Efficient Processing

Electronic documents are an important part of public administration procedures. Besides security, efficiency is one of the major requirements of public administration procedures according to Zefferer et al. [2014]. This importance of efficiency is also underpinned by the Digital Agenda for Europe:

“Europe’s public sector expenditure should be used to spur innovation while raising the efficiency and quality of public services.” [European Commission, 2010a]

Furthermore, also the European Interoperability Framework defines effectiveness and efficiency as one of their major principles:

“Solutions should serve businesses and citizens in the most effective and efficient way, providing the best value for taxpayers’ money.” [European Commission, 2011b]

As e-Documents are a major part of public administration procedures, a faster processing of e-Documents enables more efficient public administration procedures. Thereby, an efficiency increase can be achieved by a faster and more automatic processing of e-Documents. This leads to the following key action point:

Key action point 4 *More efficient and automatic processing of e-Documents is needed (see Chapter 7).*

3.3.2 Next-Generation Applications

3.3.2.1 Open Government Data

The public sector is holding a lot of information and data, which can be of great value for citizens and enterprises. This has also been recognised by the European Commission. The Digital Agenda for Europe highlights the need to public sector information and thus open government data:

“For example, governments can stimulate content markets by making public sector information available on transparent, effective, non discriminatory terms. This is an important source of potential growth of innovative online services. The re-use of these information resources has been partly harmonised, but additionally public bodies must be obliged to open up data resources for cross-border applications and services.” [European Commission, 2010a]

Public sector information is also one of the priorities of the e-Government Action Plan:

“The combination of new technologies, open specifications, innovative architectures and the availability of public sector information can deliver greater value to citizens with fewer resources.” [European Commission, 2010d]

Surprisingly, security in general and security aspects such as authenticity and integrity of the published data are hardly discussed so far. Nevertheless, these security aspects should be considered, as the use of forged data might for instance lead to resource claims. To ensure the trustworthiness of the provided data has benefits for the data provider³² and the data recipient³³. Hence - from a research perspective - it is interesting to evaluate and discuss the need for next-generation applications in this area. That means following key action can be formulated:

Key action point 5 *Next generation applications in the area of open government data and public sector information are needed, which consider security aspects such as the authenticity and integrity of the provided data (see Chapter 8).*

3.3.2.2 Identity Management

Identity management is a vital element of electronic communication as stated by the Digital Agenda for Europe:

“Electronic identity (eID) technologies and authentication services are essential for transactions on the internet both in the private and public sectors.” [European Commission, 2010a]

This importance is also highlighted in the e-Government Action Plan:

“EU-wide electronic identity systems are coming into existence, which will enable people to access public services electronically across the EU.” [European Commission, 2010d]

Many identity management systems for electronic identities are not user-centric in terms of, which data is revealed to the application a user wants to access. That means citizens are required to reveal their entire identity usually. However, due to privacy reasons, many citizens do not want to disclose their entire identity data. The STORK framework (cf. Section 3.2.5.6) and the German eID card Margraf [2011] allow such a selective disclosure of chosen identity attributes. Nevertheless, the Austrian

³²The data provider is able to prove that the data has not been altered.

³³The data recipient can trust on the validity and correctness of the provided data.

eID systems does not allow for such a selective disclosure. That means there is the need to include such a selective disclosure support to the Austrian eID systems, which leads to following key action point:

Key action point 6 *Next generation of an identity management system, enabling a selective disclosure of identity attributes and applicable to the Austrian eID system, is needed (see Chapter 9).*

3.3.2.3 Public Administration Procedures

Public administration procedures are one of the main pillars for e-Government. Unfortunately many public administration procedures still lack on efficiency due to administrative burdens as stated by the e-Government Action Plan:

“In practice however, many procedures and requirements make interactions with governments burdensome in terms of time and resources. Therefore simplification or elimination of administrative processes should be an important objective, as laid out in the Action Programme for reducing administrative burdens in the European Union.” [European Commission, 2010d]

This is also underpinned by the “Study on eGovernment and the Reduction of Administrative Burden” [European Commission, 2014c] and the final report on “The functioning and usability of the Points of Single Contact under the Services Directive” [European Commission, 2012]. For instance, this study state:

“Points of Single Contact have not yet led to a simplification in administration in terms of business establishment.” [European Commission, 2012]

and

“There is still a long way to go in order to move towards truly transactional eGovernment portals.” [European Commission, 2012]

Furthermore, this report has found out that more than 41% of the focus group had significant problems to complete the procedure. The importance of efficient public administration procedures is also recognised by the research community. For instance, Zefferer et al. [2014] have defined, based on the findings of J. R. Gil-Garcia [2007] and Altameem et al. [2006], efficiency as one of the major success factors for successful e-Government services. In addition, they also define security as such a success factor. The increase of both, efficiency and security, touches the technical, organisational and legal level. As the present thesis focuses on the technical aspects, following key action point can be formulated:

Key action point 7 *Technical approaches for efficient and secure public administration procedures across borders are needed (see Chapter 10).*

Part II

Next-Generation Technologies for Electronic Documents

Chapter 4

Electronic Documents Interoperability



“Success is the ability to go from failure to failure without losing your enthusiasm.”

[Winston Churchill]

4.1 Introduction

The Digital Agenda for Europa [European Commission, 2010b] aims to use ICT as key enabler for a digital single market. Here, interoperability and standards, security and trust as well as ICT-enabled benefits for EU society are major action points to bring benefits for the EU and the Member States. Thereby, e-Documents are one of the key factors of success. In particular, the importance of e-Documents in this area has been identified by the Large Scale Pilot SPOCS (cf. Section 3.2.5.5). As indicated by SPOCS [SPOCS Consortium, 2011] e-Documents and the exchange of e-Documents play a vital role for the completion of e-Government services:

“Exchanges of documents to be provided for the completion of procedures and formalities related to a service activity.”

Furthermore, according to SPOCS Consortium [2011], the authenticity of e-Documents is a cornerstone for future and secure e-Government services.

“... set standards on e-Documents and its authenticity that have the potential to influence future deployments of eGovernment services.”

As highlighted in Section 3.2.5.5 the LSP SPOCS aimed to build up the next generation of Points of Single Contact (PSC). These PSCs are defined by the EU Services Directive [European Commission, 2006a] and represent one stop shops for citizen to get in contact with the administration. Main objective was to enable the opening of a business via online means across borders. Obviously, a secure and authentic exchange of e-Documents is vital for such scenarios. Thereby the exchange and authenticity of e-Documents, are directly concerned with interoperability issues - especially in a cross-border context (cf. Section 3.3).

To foster the interoperability between all affected parties (PSCs and the competent authorities behind), the *first part* of this chapter presents an interoperability framework for cross-border exchange of e-Documents, which has been developed by the author of this thesis in the course of the LSP SPOCS. This includes the core specification of the framework, the implementation and the evaluation of the framework during the piloting phase of the LSP SPOCS.

The *second part* of this chapter elaborates in detail on the authenticity of e-Documents, which is achieved by applying electronic signatures usually. Here, the focus of the subchapter lies on the verification of electronic signatures, which raises interoperability issues in a cross-border context in particular. Thereby it is essential to know the status of the involved certification service providers (CSP), which are issuing qualified certificates. That means, to know if a CSP is accredited, under supervision or nothing of both¹. To facilitate the verification of the statuses of European CSPs, the European Commission specified that each Member State must establish a so called trust-service status list stating including all accredited and supervised CSPs in their country.

Therefore, the remainder of this chapter is structured as follows. In Section 4.2 the interoperable electronic document framework is presented. Section 4.3 gives details on the implementation of this framework. The implementation has been tested and evaluated in real life applications and services. This piloting phase is presented and evaluated in Section 4.4. Following Section 4.5 treats the verification of electronic signatures, in particular in cross-border scenarios. This includes the signature

¹To be accredited or under supervision is a prerequisite to get the permission to issue qualified certificates.

verification based on trust-service status lists, which are seen as enabler for cross-border signature verification. This section concludes with an evaluation and survey of the current status of the trust-service status lists implementation. Finally, Section 4.6 summarizes the findings of this chapter and draws conclusions.

4.2 Interoperable Electronic Document Framework

4.2.1 Overview and Motivation

This section presents the electronic document framework developed by the author in the course of work package 2 (“e-Document”) of the Large Scale Pilot SPOCS². Looking at the e-Government landscape in Europe, e-Documents may be of any format and may be issued by different entities, such as the public administration, private organizations or citizens. Currently most EU Member States use various document formats as shown by Rössler et al. [2011]. In general, e-Documents can be split into the following categories:

- Structured formats and technologies
- Unstructured formats and technologies
- Container formats and technologies

Structured e-Document formats are documents, whose content is structured according to a well-defined schema. Therefore structured e-Document formats are usually machine interpretable and so applicable for automated processing. The most known representative of this technology is the XML format [Bray et al., 2006]. In contrast to structured e-Document formats, the content of *unstructured e-Document formats* cannot be automatically processed. Such formats are mainly used for a visual representation of e-Documents. The most popular unstructured e-Document format is PDF [ISO/IEC, 2008]. Finally, *container formats* usually carry different types of data. A container format specifies how the data is stored. In general such formats are self-contained, i.e. all information and data needed for the processing of the document is stored in the container. Starting with MIME, as one of the first container formats, these formats grew more and more in popularity.

Additionally various authentication mechanisms exist (cf. Section 2.4). Here it is distinguished between mechanisms which are tightly bound to particular e-Document formats (e.g. PDF signatures [ISO/IEC, 2008]) and mechanisms which can be used with almost every e-Document format (e.g. XML signatures [Bartel et al., 2008]). Based on the presented survey, following main requirements for an interoperable container format can be identified. The container format...

- ... should introduce a multi-layered interoperable document container for cross-border exchange of e-Documents.
- ... should not be restricted to support only selected e-Document formats and technologies.

²The author of this thesis was leading work package 2 from July 2010 to December 2012 and was a main contributor to the findings of this work package. Credits go also to Thomas Rössler (work package leader from May 2009 to June 2010) and the other involved partners.

- ... should handle all e-Documents which are currently used and be opened for new formats and technologies.
- ... should support semantic interoperability.
- ... should support authenticity in addition to the authentication mechanism provided within the contained e-Documents.

As a result, a multi-layered interoperable electronic document container, the so called *Omnifarious Container for e-Documents*³ (short: OCD) has been developed. Its specification can be found in [Stranacher et al., 2011]. Although the OCD has been developed with respect to the EU Services Directive [European Commission, 2006a], the OCD is not limited to use cases of the Directive only. Due to its generic concept it supports any kind of electronic information exchange on the basis of electronic documents.

For the OCD a *logical* and a *physical structure* have been developed. These structures define the core elements of the OCD. In addition methods and processes, which can be applied to the core elements, have been specified. The following subsections will describe them in more detail.

4.2.2 Logical Structure

The logical structure of the OCD defines the different layers for carrying the appropriate data. Therefore an OCD is composed of three layers (as illustrated in Figure 4.1):

- Payload layer
- Metadata layer
- Authentication layer

4.2.2.1 Payload Layer

Within the payload layer all e-Documents, which should be transported via the OCD, can be stored. Thereby an OCD is able to hold any kind of electronic data in its payload layer, no matter if these documents are signed/unsigned or encrypted/not encrypted. This ensures that every existing document can be transferred.

4.2.2.2 Metadata Layer

To support the automatic processing of e-Documents, the metadata layer has been introduced. This layer gives a unified description of the e-Documents given in the payload and of the entire container. Here OCD introduced two levels of metadata:

- Metadata level 1: Payload description
- Metadata level 2: Container description

³Omnifarious means versatile or adaptable. Credits for inventing this fantastic acronym go to Thomas Rössler.

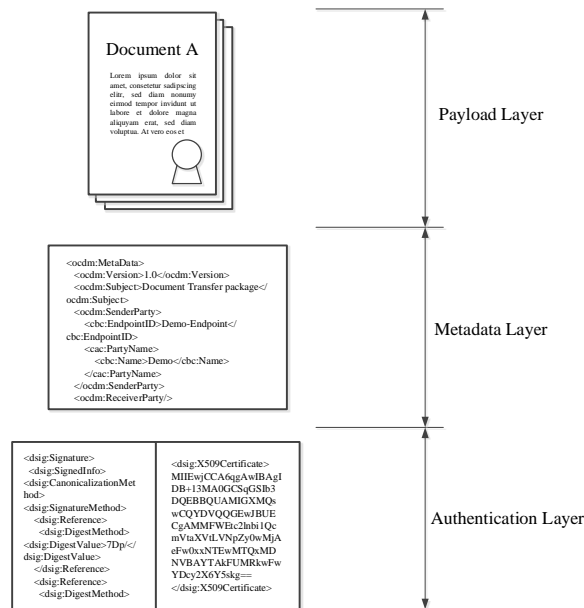


Figure 4.1: Multi-layered container format OCD

Metadata level 2 gives a description of the entire container, such as sender and receiver of the OCD. *Metadata level 1* gives a unified description of each payload document given in the OCD. Depending on the availability of metadata, a variety of different metadata can be added. In the best case, level 1 gives a unified description of the content of documents by, for instance, an identifier indicating the types of documents, a list of field identifiers indicating which information is given in the documents and, ideally, an extract of the values of the fields indicated by the field identifiers. Both metadata levels are organized in one single XML file which satisfies the specified XML schema given in [Stranacher et al., 2011].

4.2.2.3 Authentication Layer

Additionally to the authentication mechanisms provided by the payload documents, the entire OCD and all its affiliated elements can be optionally signed. This signature can be used to enable authentication, but has per se no legal meaning. Any legal meaning depends on the signatures of the payload documents. Depending on the physical structure of the OCD different format for this container signature are supported (see Section 4.2.3).

4.2.2.4 Visual Representation

Finally, a visual representation of the metadata and the authentication layers has been defined. Via XSL style sheet transformations the metadata and signature is transformed into a human readable visual representation as shown in Figure 4.2.

Metadata

Receiver party

Name	John Doe
------	----------

Metadata level 1 for Document ID-10

ID:	ID-10
Document version:	1.0
Description:	Example Document
Issue date:	2012-02-28 09:56:11
External document	URI: OCD/Payload/ID-10/document.doc Document hash: c27a43a5d42e1ca8f27d7c24bbc2ba97cdb1c63c632e3d5761082efef2we Hash algorithm(-s): http://www.w3.org/2001/04/xmlenc#sha256

OCD Signature


Signature value	A5f/NmUxjA9MbpeETO5pgrhypZMQo+y1tN15xgQ5B2aWvah+g=	
	Signatory	SPOCS TEST, INFOCERT SPA, IT
	Date/Time-UTC	2012-02-28T09:56:12
	Issuer	InfoCert Servizi di Certificazione, Ente Certificatore, INFOCERT SPA, IT
	Serial-No.	1367956
	Method	RSA-SHA256
	Note	This is a test signature

Figure 4.2: Visual representation of metadata and authentication layer

4.2.3 Physical Structure

The physical structure defines the physical implementation of the logical structure of OCD containers. Currently there are two physical structures defined:

- ZIP based OCD
- PDF based OCD

The *ZIP based OCD* is mainly based on the ETSI specification for Associated Signature Containers (ASiC) as specified in [ETSI, 2012]. Thereby, ASiC specifies

“[...] the use of container structures, to bind together a number of signed objects (e.g. documents, XML structured data, spreadsheet, multimedia content) with either detached advanced electronic signatures or timestamp tokens into one single digital container based on ZIP.”

Here, XAdES signatures, as defined in [ETSI, 2010b], for the authentication layer are used. The intention for the ZIP based OCDs is to be used mainly in the back office area, where no citizens are involved.

The *PDF based OCD* uses the mechanism “PDF with attachments” as defined in [ISO/IEC, 2008]. Here the visual representation of the OCD metadata serves as master PDF file. All other objects, such as payload documents or metadata file, are added as attachments to the master PDF file. For the authentication layer PAdES signatures, as defined in [ETSI, 2010a], are used. The intention for PDF based OCDs is to be used in all cases, where citizens are directly involved.

4.2.4 Methods

This subsection describes the main methods applicable to OCDs. These methods define operations on the core elements of OCDs and are needed to handle OCDs in real life scenarios. In general, following methods can be applied:

- Creation of OCDs
- Validation and verification of OCDs
- Extraction of OCDs

OCD Creation This method defines how an OCD is created. Inputs for this method are documents from different EU Member States (various document formats, signed and unsigned documents, etc.) and appropriate metadata (depending on the availability of these metadata). The output of this method is the signed or unsigned OCD container.

OCD Validation and Verification This method defines how an OCD is validated and its signatures (e.g. signatures of the payload documents and container signature) are verified. The validation comprises, for instance, checks if the OCD container structure is compliant to the specification, whereas the verification is responsible to verify the included signatures. Input for this method is an OCD and output is a validation and verification report.

OCD Extraction This method defines how the information contained in an OCD can be extracted. Input for this method is an OCD and outputs are the extracted information (payload document(s), metadata and authentication data).

The following section will give a brief overview about implemented open modules based on the above defined methods.

4.3 Implementation

4.3.1 Overview

Based on the defined methods software modules have been implemented. Each module has been implemented as open source module licensed under EUPL⁴ to ensure the take up of the modules by EU Member States. Additionally, all modules are available as Java API and SOAP Web-Service. The following software modules have been created:

- OCD creation Module
- OCD validation and verification Module
- OCD extraction Module

In the subsections below, these three software modules are described including the interface description and architecture.

⁴European Union Public Licence, <http://joinup.ec.europa.eu/software/page/eupl>

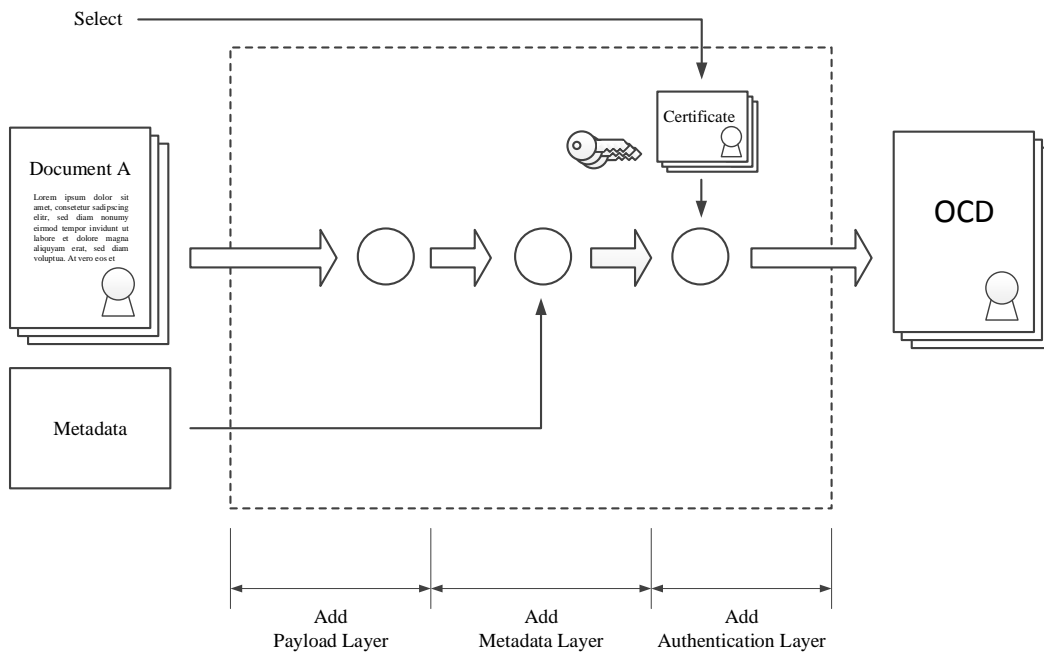


Figure 4.3: OCD creation module

4.3.2 OCD Creation

This module takes over the creation of OCDs according to the OCD specification. The module interface and the main building blocks are described below (see also Figure 4.3 and [Stranacher et al., 2012]).

Interface The input includes the documents which should be added to the OCD along with the appropriate metadata for the document itself and the entire container. In case a signed OCD should be created, a certain signature key could be selected out of some pre-configured keys⁵.

Architecture Figure 4.3 illustrates the architecture. The module creates an OCD step-by-step, which is reflected in the module architecture. In the first phase the payload layer is created. Next the metadata layer is created out of the information provided by the input interface and the payload layer. Optionally, the entire OCD is signed using the selected signature key.

4.3.3 OCD Validation and Verification

This module is responsible for the validation and verification of OCDs. The following subsections describe the module interface and the architecture (see also Figure 4.4 and [Buso et al., 2012]).

⁵The module configuration offers the possibility to configure different signature keys.

Interface Similar to the OCD Creation Module, the interface has been designed to simplify the usage. A required input is the OCD, which should be verified, and an optional verification time⁶. An XML based validation and verification report, which comprises results of all conducted validation and verification processes, serves as output.

Architecture Figure 4.4 shows the architecture. The module architecture shows the different validation and verification steps. First, a basic OCD validation takes place. This validation checks if the given OCD is compliant to the OCD specification. Following all signatures contained in the OCD (signature of the entire OCD and signatures of the payload documents) are verified. This comprises the core signature verification and the certificate validation as well. Here, the signature verification can take place in two ways:

1. Using the internal signature verification mechanism: This mechanism supports all signature formats as defined in the EC Decision on establishing minimum requirements for the cross-border processing of documents signed electronically [European Commission, 2014a].
2. Using external verification services: The module offers the possibility to connect to external verification services for conducting the signature verification. This can be used to support the verification of proprietary documents, which are not covered by the internal verification mechanism. A few EU Member States are using such proprietary document, such as Austria [Leitold et al., 2010] and Lithuania [Lithuanian Archives Department, 2009]. These Member States are obliged (due to the EC Decision 2013/662/EC [European Commission, 2013a]) to operate a public signature verification service supporting the proprietary format. These services can be used to connect to the OCD validation and verification module.

Second last step is the (optional) metadata verification, which gives the possibility to verify if the given OCD includes a certain set of metadata (cf. Chapter 7). Finally, a report generator collects all validation and verification results and generates a common - XML based - validation and verification report.

4.3.4 OCD Extraction

This module takes over the extraction of information out of given OCDs. The module interface and the main building blocks are described below (see also Figure 4.5 and [Stranacher et al., 2012]).

Interface The module requires an OCD as input only. Depending on the content of the OCD, the output comprises one or more payload documents and the metadata.

Architecture Figure 4.5 illustrates the architecture. First, a basic OCD validation - similar to the OCD Validation and Verification modules - takes place. This validation checks if the given OCD is compliant to the OCD specification. Following, the payload extractor extracts the container payload documents and the metadata extractor is responsible to extract the whole metadata.

⁶If no explicit verification time is given, the signing time given in the signature or the actual time (following the rules of the respective signature scheme) is used.

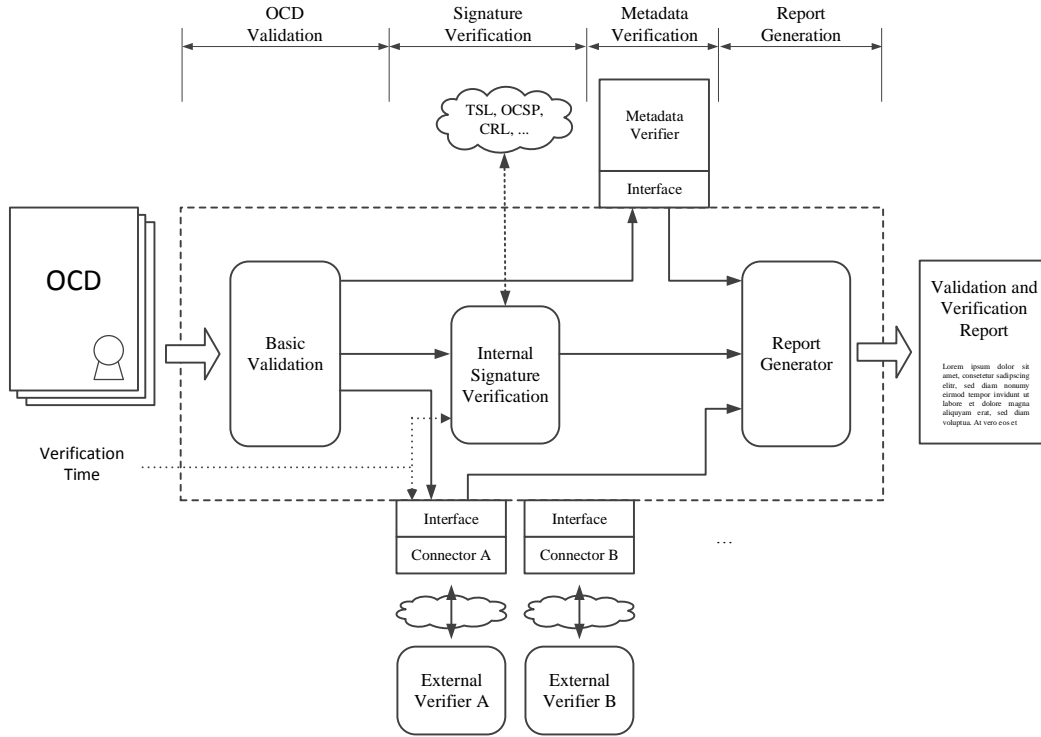


Figure 4.4: OCD validation and verification module

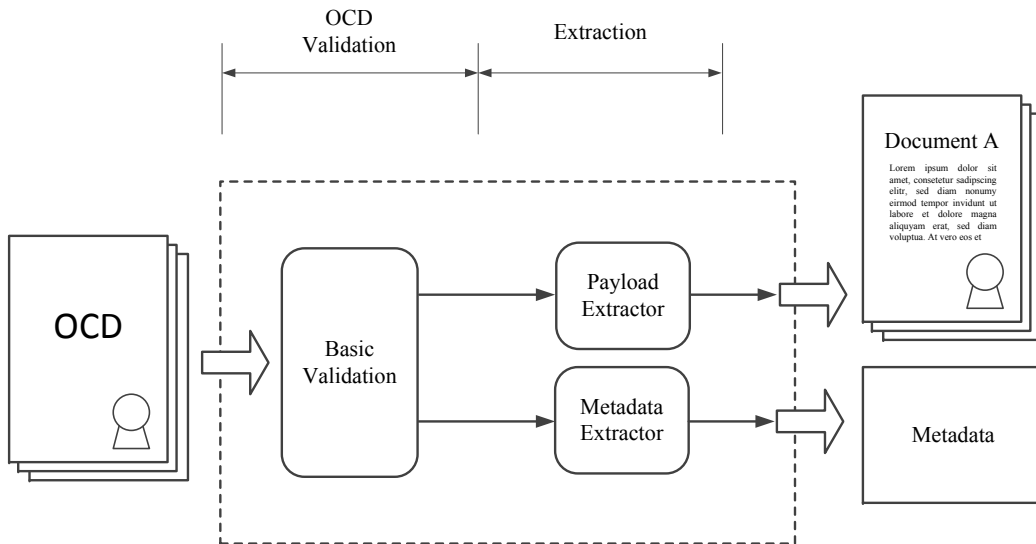


Figure 4.5: OCD extraction module

4.4 Deployment and Evaluation

During the piloting phase⁷ in the Large Scale Pilot SPOCS, the developed OCD modules have been deployed in real life environments at the involved Points of Single Contact. Here, service providers (such as travel agents, master builders or real estate agents) have been able to open a business in a foreign country from their home country using electronic means. All involved modules (including the modules from e-Delivery, e-Safe, etc.) have been evaluated. The overall result was positive as stated in the evaluation report [Fotiou et al., 2012]:

“[...] the functionality of SPOCS modules was rated positively.”

The remainder of this section summarizes the conducted evaluation of the OCD modules. Within this evaluation the specification, the implementation, the deployment of the modules in the piloting countries as well as sustainability issues have been validated. First of all, general results are given, which reflect the overall applicability, purpose and functionality of the OCD modules. Finally, a SWOT (Strengths, Weaknesses, Opportunities and Threats) analysis of the module is given.

4.4.0.1 General Results

Due to the low amount of real service providers⁸, the evaluation results mainly bases on the feedback given by Point of Single Contacts, Competent Authorities and IT service providers. The OCD modules have been evaluated positively overall. The provided functionalities meet the requirements of all piloting countries and non-piloting countries (where applicable). This positive result involves all phases of the OCD development, in particular the specification, implementation and deployment (including support for the piloting countries).

Figure 4.6 emphasizes the positive evaluation result of the OCD modules⁹. It shows the overall evaluation of all building blocks for the fitness for purpose. Here, the OCD modules have been highest rated concerning concept and specification as well as implementation. In addition the support is ranked on the second position.

4.4.0.2 SWOT Analysis

The detailed findings of the evaluation have been incorporated into a SWOT analysis. Figure 4.7 represents this SWOT analysis. The analysis gives details on the strengths and weaknesses of the OCD modules as well as opportunities and threats. Finally it contains strategies and measures which may help different stakeholders to increase the sustainability of the OCD modules.

⁷The piloting phase lasted from 1st July 2011 to 31st December 2012.

⁸According to Fotiou et al. [2012], the low amount of real service providers mainly based upon the generally low volume of cross-border transactions at Point of Single Contacts and the crisis in Europe.

⁹In the piloting context, the SPOCS OCD modules are called *Building Block e-Documents*.

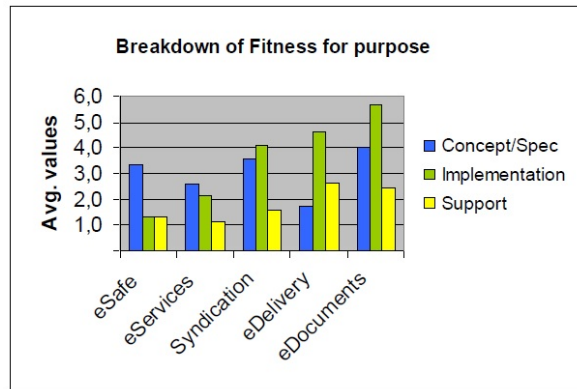


Figure 4.6: Overall evaluation of “fitness for purpose” [Fotiou et al., 2012]

	Strengths	Weaknesses
SWOT Analysis	<ul style="list-style-type: none"> (a) Given technical functionality fulfils the requirements of the piloting countries (b) Purpose of the OCD modules are clear and fits the needs of cross-border services in general (c) OCD modules are on a mature level and can be used in real life environments (d) Easy installation and configuration (e) Technical documentation contains all really needed information (f) OCD modules can be easily reused in other services apart of the EU Services Directive (g) The OCD modules enable smart, safe and trusted services (h) The OCD modules support semantic interoperability (metadata layer) 	<ul style="list-style-type: none"> (a) Missing standardization of parts of the OCD container, i.e. metadata layer, PDF based OCD container (b) Legal constraints and missing document equivalency. These weaknesses do not base on the OCD modules itself and concern the other deployed SPOCS modules too. (c) Broader support of different metadata in the metadata layer missing (d) Missing client-GUI-tool to create and verify OCD containers
Opportunities	<ul style="list-style-type: none"> (1) Further maintenance of the OCD modules in the LSP e-SENS to foster the sustainability (2) Deployment of the OCD modules in e-SENS scenarios and related (cross-border) services (3) Stimulate and promote the usage of the OCD modules in other Point of Single Contacts and Competent Authorities. This will help to increase the sustainability and to facilitate the establishment of a common e-Documents exchange framework 	<ul style="list-style-type: none"> (1) Push the standardisation activities in e-SENS to increase the sustainability of the OCD modules (2) Further development of the OCD modules in the LSP e-SENS to foster the applicability of the modules for services under the EU Services Directive as well as other services on a cross-border level (3) Establish a legal framework to facilitate document equivalency and to remove the barriers on the cross-border exchange of e-Documents
Threats	<ul style="list-style-type: none"> (1) Promotion of the successful usage of the OCD modules to stimulate other users and decision makers to rely on the modules (2) The OCD modules (and the other SPOCS modules as well) have proven that opening a business on cross-border level by basing on electronic means is possible. This enables do start discussion concerning the missing legal basis as well as further standardisation activities 	<ul style="list-style-type: none"> (1) Early start of further standardisation activities in e-SENS, to achieve common consent. This will allow to base upon fully standardized modules and thus increasing the sustainability of the modules. (2) The missing legal framework for document equivalency and cross-border exchange of e-Documents hinders the sustainability of the building block e-Documents (and the other SPOCS building blocks) currently. Therefore appropriate efforts on European level should be made to these obstacles.
	<ul style="list-style-type: none"> (a) Missing legal basis for cross-border exchange of e-Documents and document equivalency (b) OCD modules are not applicable for other parties, apart from SPOCS or the EU Services Directive (c) No consent on standardisation activities of the OCD modules 	

Figure 4.7: SWOT analysis of OCD modules [Stranacher, 2013]

4.5 Cross-border Signature Validation

4.5.1 Overview and Motivation

In 1999 the European Commission published the Directive on a Community framework for electronic signatures, better known as the Signature Directive [The Council of the European Union, 2000]. The Directive includes a definition of different levels of electronic signatures and their legal effects. In particular it defines that an electronic signature is legally equivalent to a handwritten signature (Article 5), if the electronic signature satisfies following requirements:

- The signature must be an advanced electronic signature
- The signature must base on a qualified certificate (QC)
- The signature must be created using a secure signature device (SSCD)

Such a signature is usually called *qualified signature*, even if the Signature Directive does not explicitly define this term. In many e-Government processes a qualified signature is a precondition for further processing. Especially in cross-border services¹⁰ the verification of qualified signatures (relying on a qualified certificate and secure signature creation device) becomes difficult, as qualified certificates can only be issued by certification service providers (CSPs), which are accredited or under supervision¹¹.

The verification of the status of a certain CSP (and so the QC and SSCD property of an electronic signature) is part of the certificate validation. Some Member States have such a single CSP¹², whereas other Member States have a quite high number of CSPs¹³. Obviously the status verification is getting very complex, especially for cross border verifications. This situation has been recognized by the European Commission. Hence, to facilitate the verification of the status of certification service provides, EU Member States are obliged to maintain a trusted list of certification service providers issuing qualified certificates [European Commission, 2013a]. Thus, this subchapter elaborates on electronic signature verification (focusing on the certificate validation) based upon such trusted lists.

The remainder of this subchapter is structured as follows. Section 4.5.2 gives the technical background information on the trusted lists, the so called trust-service status lists (TSL). In Section 4.5.3 the core implementation for handling TSLs is presented. Section 4.5.4 elaborates on the integration of the TSL core implementation into the Austrian open source module MOA-SP, which represents a widely used signature verification module. Finally, Section 4.5.5 evaluates this integration and gives a survey of the current status of the issued Member State TSLs in Europe.

4.5.2 Trusted Lists

Trusted lists are implemented by the ETSI standard on Trust-service Status List (TSL). To provide a single point of contact, the European Commission maintains and published a central trusted list (EU-

¹⁰For instance, when signing a form or application. It is also of special interest for the cross-border identification and authentication of persons. See EU large scale pilots STORK and STORK 2.0 (cf. Section 3.2.5.6).

¹¹Another interoperability issue are the variety of existing signature formats (cf. Section 2.4). However, this is not addressed in this thesis in detail.

¹²Such as Austria.

¹³For instance, Italy has currently 37 CSPs.

TSL), which holds references to the different Member State trusted lists (MS-TSL). Trust-service status lists have been specified by ETSI [ETSI, 2009]. The main objective is to publish information about the status of a trust-service provider in such a way...

“[...] that interested parties may determine whether a trust service is or was operating under the approval of any recognized scheme at either the time the service was provided, or the time at which a transaction reliant on that service took place.” [ETSI, 2009]

The trust-service provider status includes information about the provider including whether the provider is or was acting under the scheme of a certain scheme operator. Thereby, the TSL specification defines a scheme as

“[...] any organized process of supervision, monitoring, approval or such practices that are intended to apply oversight with the objective of ensuring adherence to specific criteria in order to maintain confidence in the services under the scope of the scheme.” [ETSI, 2009]

That means the status information contains the current status and historical status information. Historical status information is important to verify, for instance, if a certification service provider was accredited or supervised at the issuing time of a certain qualified certificate. In relation to electronic signatures, the EU-TSL holds status information about certification service providers, which are issuing qualified certificates under the scheme of a certain accredited or supervised body. The logical structure of a TSL consists of following four components:

- Information about the TSL and the scheme itself facilitating its identification
- Information about the trust service providers and whose services are within the scope of the scheme
- For each trust service provider: information about current status of each service operated by the provider (including the certificate which represents the particular service)
- For each service: information about historical status information

For authentication purposes the whole TSL is signed by the scheme operator. Additionally a TSL must be published in a machine processable form and shall be published in a human readable form [European Commission, 2013a].

4.5.3 Core TSL Implementation

4.5.3.1 Overview

This section contains a description of the core implementation to handle TSLs¹⁴ (TSL library). The library is used to download, parse and handle the particular trust-service status lists and is available as Java API. The following subsections give an overview about the architecture of the TSL library. In addition, detailed information about the main building blocks of the TSL library is given.

¹⁴Credits for implementing this core library go to Konrad Lanz.

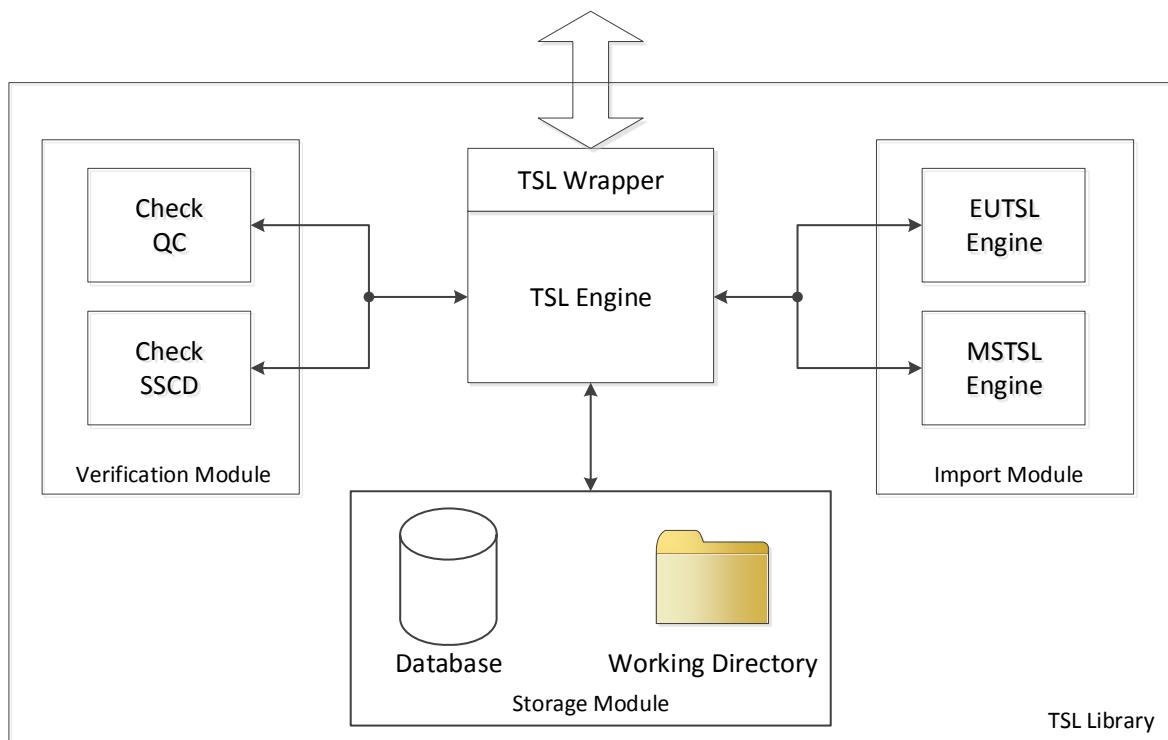


Figure 4.8: TSL library architecture

4.5.3.2 TSL Library Architecture

Figure 4.8 illustrates the architecture of the TSL library. The key component is the *TSL Engine*. It coordinates the TSL import and the certificate extension verifications, which are used to verify the QC and SSCD properties. These operations are implemented in submodules to achieve a modular architecture of the TSL library.

The entire functionality which is required to import and verify the particular TSL information's are implemented in the *Import Module*. The verified TSL information is stored in a structured format in a *Database*. Additionally, certificates (extracted from the TSL information) are stored in a local file system based *Working Directory*. All store and read functionalities are encapsulated in the *Storage Module*. The *Verification Module* implements functionalities to verify certificate extensions depending on the information, which is stored in the Storage Module. The benefit of this architecture is fast processing, due to the internal database and the separate certificate extension verifications (as they must be performed for each certificate validation during a signature verification process). In the following more details on the Import Module and the Verification Module are presented.

4.5.3.3 Import Module

A TSL import operation consists of several steps and therefore this operation is splitted into two separate modules. The two modules are the *EUTSL Engine*, which is used to download, verify and import the EU TSL information and the *MSTSL Engine* which process the different Member States

TSLs. Figure 4.9 illustrates the process flow of a TSL import operation. The following steps are performed to download and import the TSLs:

1. The TSL Engine initializes the database connection and setup the URL to EU trust-service status list.
2. Processing of the EU TSL:
 - (a) The EU-TSL is downloaded and stored in the local working directory.
 - (b) Afterwards, the TSL is validated (XML schema validation and verification of the applied signature).
 - (c) Finally, the pointers to the Member State TSLs are stored in the database and the signer certificates of the particular TSL are stored in the working directory.
3. Processing of the Member State TSLs. This step is repeated for each Member State TSL¹⁵.
 - (a) The MS-TSL is downloaded and stored in the local working directory.
 - (b) Afterwards, the MS-TSL is validated (XML schema validation and verification of the applied signature).
 - (c) Finally, the TSL information and all extracted certificates are stored.
4. In the last step, the import and validation errors are stored into the database and the database connection is closed.

4.5.3.4 Verification Module

The Verification Module is used to verify the QC or SSCD property of an electronic signature. This property check requires several steps and uses information from the local database. Figure 4.10 shows the process flow of a verification of these properties. The following steps are performed after a request to verify the QC oder SSCD property has been received.

1. Initialization of the database access.
2. The certificate chain is sorted beginning with the end-entity certificate.
3. Search for information in the TSL database for every certificate in the chain.
4. Select the critical extensions in the end-user certificate and the TSL information.
5. Compare these extensions according the TSL criteria-list element.
6. For every match extension, check if a QC or SSCD property exists.
7. Return the result.

¹⁵Two modes of operation exist to import the Member State TSLs. The MS-TSLs are either sequential or parallel (using multithreaded processes) processed.

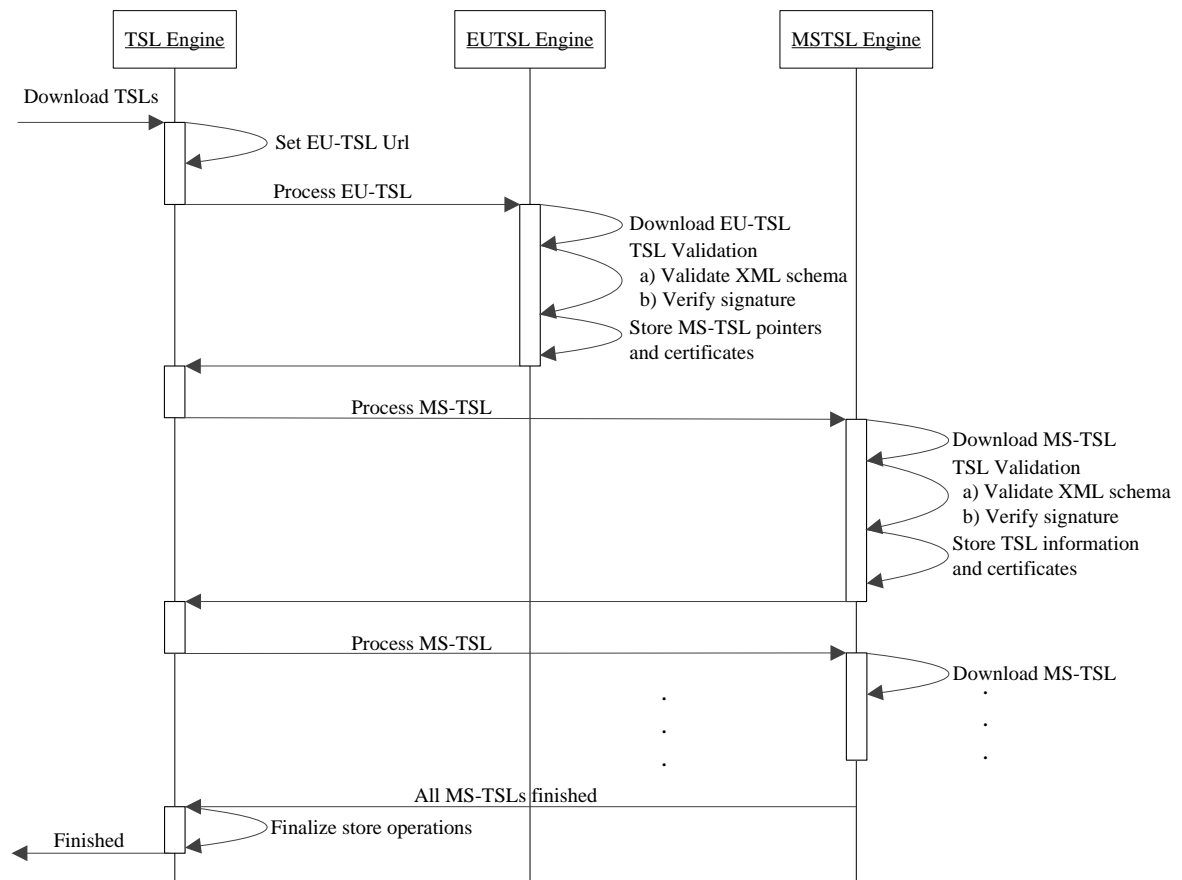


Figure 4.9: Process flow: TSL import module

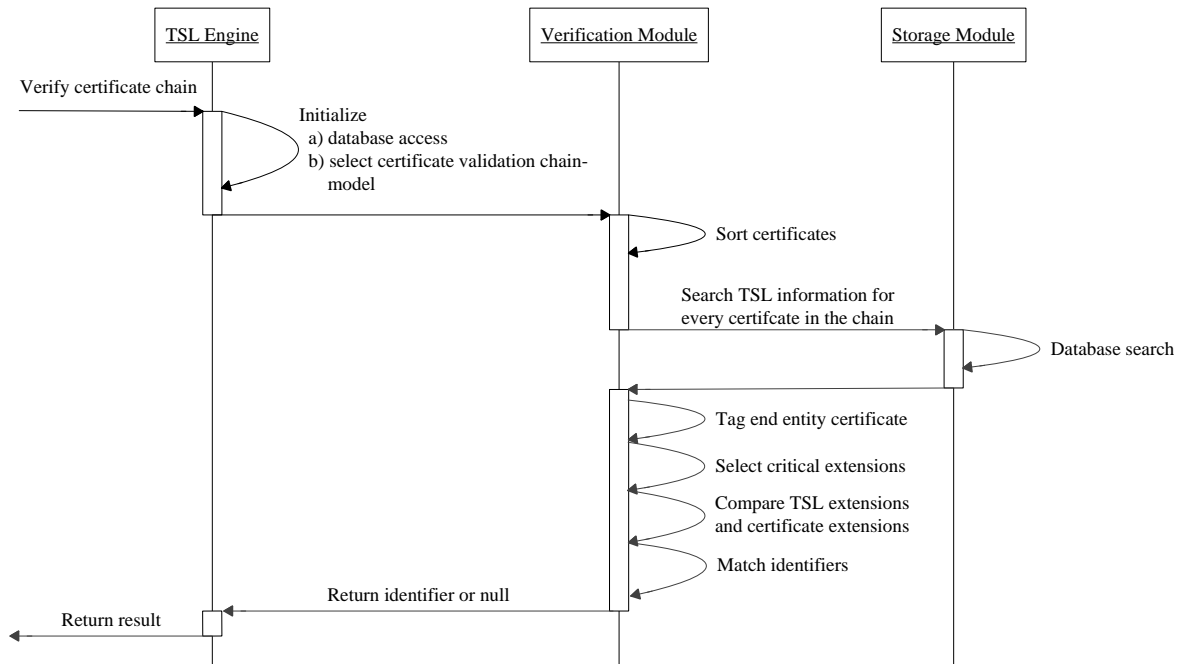


Figure 4.10: Process flow: TSL verification module

4.5.3.5 TSL Library Error Handling

Actually only a few TSLs are strictly compliant to the ETSI specification (cf. Section 4.5.5.2). Therefore, a schema-validation error-handling becomes necessary. During the TSL import operation, all validation errors are logged and stored in the database to obtain an overview of the current situation. The error handling can also be used to correct a subset of errors on the fly during the XML schema-validation process. If such a special defined error is found, a specific error-correction method is started to solve the fault and afterwards the strict XML schema validation is restarted. If a second error is found on the same place, then the TSL is rejected. The advantage of this solution is a combination of a strict XML schema validation and an adjusted XML schema error handling.

4.5.4 TSL Integration

4.5.4.1 Overview

This subsection elaborates on the integration of the TSL library into the Austrian open source software module MOA-SP¹⁶. In the following a brief overview about the existing functionality of MOA-SP, focusing on the certificate validation, is given. Next, concrete requirements are identified, which must be taken into consideration for the TSL integration. Then, the extended architecture and the modified process flow of the integrated TSL support are elaborated.

¹⁶MOA-SP supports the validation of XMLDSIG, XAdES, CMS and CAAdES signatures. More information can be found at: <https://joinup.ec.europa.eu/software/moa-idspss/description>.

4.5.4.2 Existing Functionality

Figure 4.11 shows the architecture of MOA-SP. According to the general signature verification process, MOA-SP performs the core signature-verification and the following certificate validation. During the core signature-verification MOA-SP verifies the cryptographic validity of the signature (*Core Signature Verification Unit*) based upon the supported signature formats. The certificate validation-process (*Certificate Validation Unit*) verifies if the signer certificate is valid and performs following validations:

- Validation of the signer certificate itself. That means to verify if the certificate is timely valid (e.g. has not expired) or is not revoked (e.g. due to a key compromise). For the latter verification MOA-SP is using a certificate revocation list (CRL) or the online certificate status protocol¹⁷.
- Build a certificate chain from the signer certificate up to a root certificate according to the PKIX specification [Cooper et al., 2008] and verify the validity of all certificates in this chain.
- Validation if a certificate of the chain matches a certificate in a defined *trustprofile*. Such trust-profiles are configured in the MOA-SP configuration and define a set of trusted certificates (=truststore). This last validation is very important to define which signer certificates are trusted.

Only if all validation steps have a positive result, the whole certificate validation is positive. Actually this trustprofile mechanism only bases on manually configured trusted certificates. In the following this mechanism is extended to support TSL.

4.5.4.3 Requirements

Basis for the requirement analysis have been the application operators and users of MOA-SP. Their needs and preconditions have been the leading factors for the analysis. Here it was essential to find a trade-off between the widespread functionality of the TSL library and a still easy configurable and useable MOA-SP module. Following main requirements have been identified:

Backward compatibility: MOA-SP is widely used¹⁸. Therefore it is of high importance to retain the existing functionalities, especially for all application operators and users which do not need TSL support. This means, that current configurations and user interfaces must still be usable after the TSL integration.

Minimal effort: The installation and configuration effort of the TSL functionality should be reduced to a minimum. At the same time, really needed configuration options should be easily modifiable.

Integration: The TSL support should fit well into the existing architecture of MOA-SP and should allow an easy integration into existing applications.

¹⁷At least one of these two revocation services is given in the certificate usually.

¹⁸See http://www.digitales.oesterreich.gv.at/site/cob__28748/5250/default.aspx.

4.5.4.4 Extended Architecture

Based upon the existing architecture of MOA-SP, the additional required functionalities for integration TSLs into MOA-SP have been added. Figure 4.11 illustrates the extended architecture and the extended trustprofile mechanism. The extended mechanism allows adding TSL support for trustprofiles (*TSL-enabled trustprofiles*). This means that TSLs can be used as an additional trust anchor¹⁹ during the certificate validation.

Due to the wide range of functions of the *TSL library*, a *TSL Wrapper*²⁰ has been developed following the KISS principle. This wrapper encapsulates the functionalities of the TSL library and provides specific methods, which are needed by MOA-SP. Thereby, the wrapper provides an easy applicable interface for other verification services, which intends to use the TSL library functionalities. These interface methods are:

Initialization: This method initializes the TSL library (creation and initialization of the TSL database, define a TSL working directory, etc.)

Download: In this method the national TSLs are downloaded, parsed and the information in the TSL library is updated.

Export: This method exports a set of CA certificates, whose CA issues qualified certificates and its certification service provider is under supervision or accredited.

QC and SSCD check: These last methods verify if the signer certificate is qualified and if the signer signature has been created by the use of a secure signature creation device.

4.5.4.5 Extended Process Flows

Based upon the extended architecture, the process flows have been changed in case of a configured TSL support. In the following the modified process flows during the server startup and the certificate validation process are explained.

Startup and TSL Unit Initialization During the server start MOA-SP performs different initializations based upon the MOA-SP configuration. The corresponding process flow (in case of TSL-enabled trustprofiles are configured) is illustrated in Figure 4.12 and consists of following process steps:

1. After the server startup MOA-SP performs the non-TSL specific initializations based upon the configuration.
2. Initialization of the TSL Unit, e.g. definition of database location, creation of the working directory, etc.
3. Update of the TSL-enabled trustprofiles:

¹⁹That means in addition to optional, manually, added trusted certificates.

²⁰Beside the thesis author's contribution, credits for implementing this wrapper go to Thomas Lenz as well.

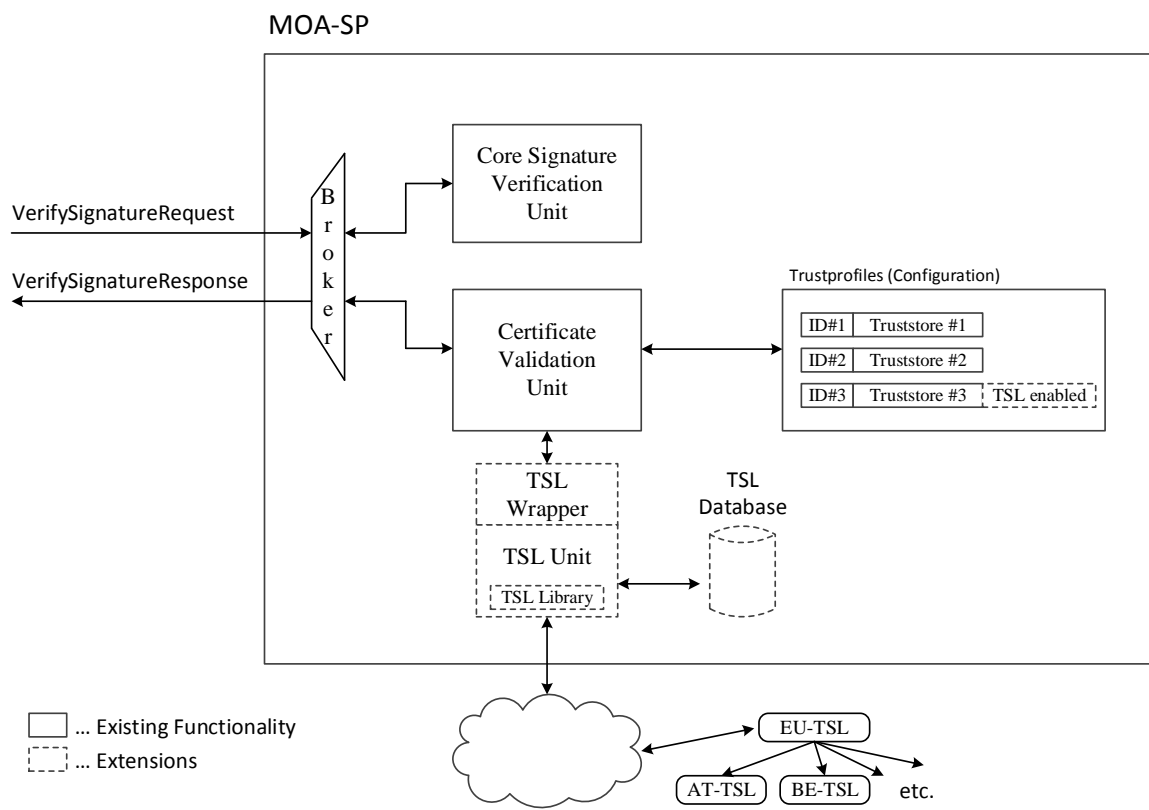


Figure 4.11: MOA-SP architecture

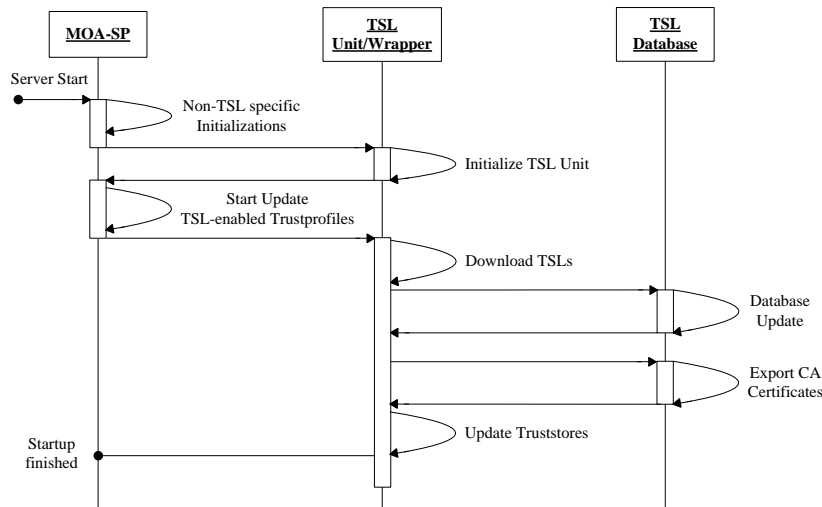


Figure 4.12: Process flow: startup and TSL unit initialization

- (a) Download of the actual national TSLs via the EU-TSL.
- (b) Parse these TSLs and update of the information in the database.
- (c) For each TSL-enabled trustprofile:
 - i. Export all CA certificates matching the properties defined in the MOA-SP configuration (i.e. CA certificate from all or only selected Member States).
 - ii. Update of the truststore, i.e. storing the exported certificates in the truststore.

In addition, the update of the TSL-enabled trustprofiles is executed on a regular basis. The update interval is adjustable via the configuration.

Signature Verification and TSL based Certificate Validation Figure 4.13 shows the process flow for verifying a signature based upon a TSL-based trustprofile. After receiving a request to verify a signature following process steps are performed (whereas the broker distributes the request to the different units):

1. MOA-SP performs the cryptographic signature verification according to the signature scheme rules.
2. The Certificate Validation Unit builds the certificate chain and verifies the validity of all certificates.
3. Via the TSL Unit it is verified if the signer certificate is qualified and if the signature has been created using a secure signature creation device.
4. Finally MOA-SP consolidates all verification and validation results and returns a verification response.

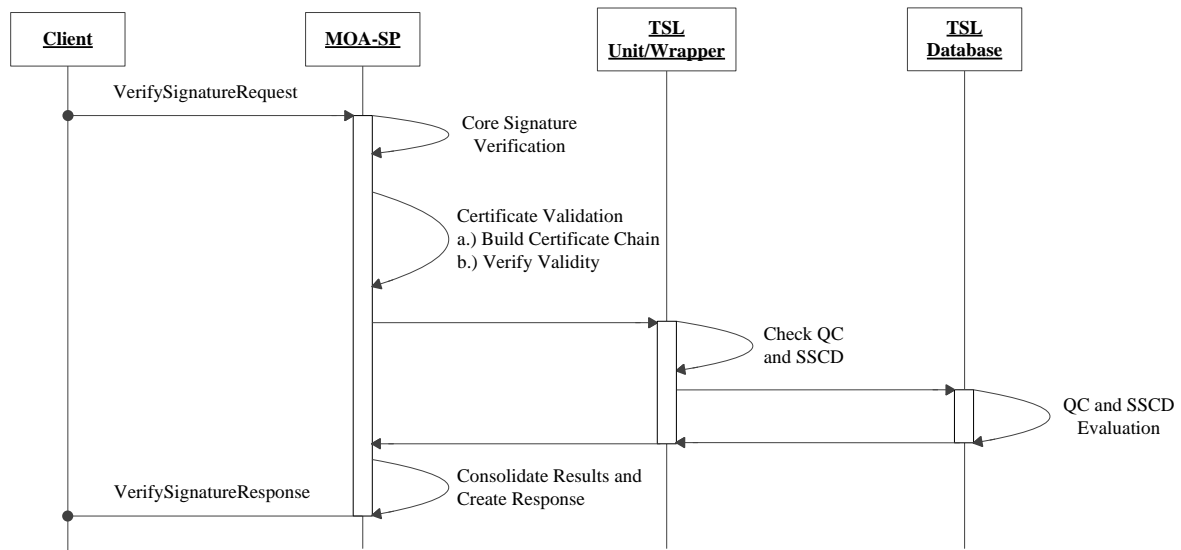


Figure 4.13: Process flow: signature verification and TSL based certificate validation

4.5.5 Evaluation and Survey

On the one side this section comprises the evaluation²¹ of the implementation in a real life environment. On the other side, issues concerning the distributed MS-TSLs itself are discussed.

4.5.5.1 Evaluation

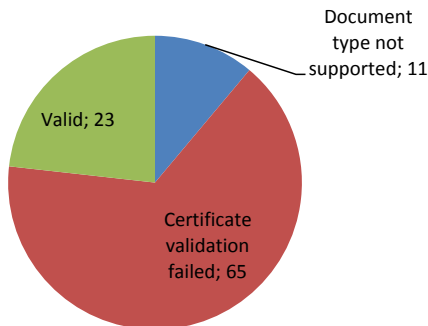
In Austria, a Web based signature-verification tool [Lenz et al., 2013b] has been developed which is in common use and enables the verification of different document and signature formats. This tool uses MOA-SP in the backend for the signature-verification. Thus, this tool has been chosen for the evaluation of the presented solution. For the evaluation an automatic test-framework, which is a part of the Web based signature-verification tool, is used to verify the electronic signatures of a wide range of signed documents. For the tests, a set of 99 different documents, which are signed with certificates from 15 European Member States, have been used. As basic principle for this evaluation, the same MOA-SP configuration, which is also in use in the productive application, has been used. Figure 4.14 (left) illustrates the certificate validation result, if no TSL information is used. Actually more the 60 per cent of the documents failed the verification process, because no valid certificate chain²² can be determined.

If the MOA-SP module is used with TSL support, then the certificate validation-result shows a clearly better result (see Figure 4.14 (right)). By using information from the TSLs, the verification results are much better, because valid certificate chains can be determined by using the additional information for the TSL. Thus, from a functional point of view TSLs are beneficial for certificate validation. Nevertheless, correct verification results strongly depend on valid trust-service status lists, which are issued by the respective Member States. However, there are still problems with valid Mem-

²¹Beside the thesis author's contribution, credits for conducting this evaluation go to Thomas Lenz as well.

²²Not valid in terms of no trusted anchor certificate has been found.

Without TSL Information



With TSL Information

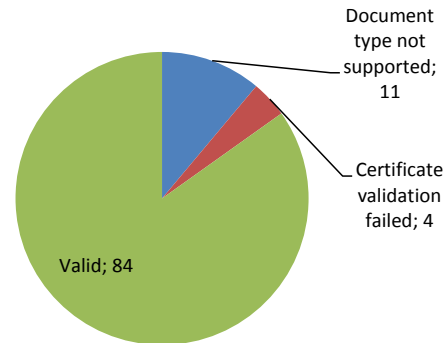


Figure 4.14: Results certificate validation (as at 04.06.2013)

ber States TSLs because actually more than three quarters of them have structural flaws. These flaws are discussed in the following subsection.

4.5.5.2 MS-TSL Usage Issues

Actual, only seven out of the distributed Member State TSLs have no structural flaws²³. To overcome these problems the error handling and correction functionality of the core TSL library can be used. Figure 4.15 illustrates the distribution of TSLs, which actually can be handled with the TSL library depending on the errors, which have been found. Actually, six Member State TSLs²⁴ are rejected, because they have serious schema issues, such as missing XML elements, or the signature of the TSL cannot be verified. In addition, 14 Member State TSLs are accepted after an error correction (cf. Section 4.5.3.5) is performed. The following enumeration illustrates the structural flaws which are able to be corrected:

- The XMLDSIG signature can be verified but uses an XML transformation which is not allowed, according to the TSL standard [ETSI, 2009]. This error can be corrected for 10 Member State TSLs²⁵.
- Deletion of some characters, like hyphens, which are included in front of or after XML elements. This error can be corrected in three Member State TSLs²⁶.
- Solve problems with an incorrectly used XML xsd:ID type. This error can be corrected in three Member State TSLs²⁷.

Nevertheless, another problem with the distributed MS-TSL exists. The TSL specification contains an element *NextUpdate*, which is specified as following:

²³AT, BE, CZ, FI, HU, LI, LU.

²⁴BG, DE, DK, EE, MT, RO.

²⁵CY, ES, FR, LT, LV, NL, PL, PT, SE, SI.

²⁶GR, IT, PL.

²⁷CY, LV, NL.

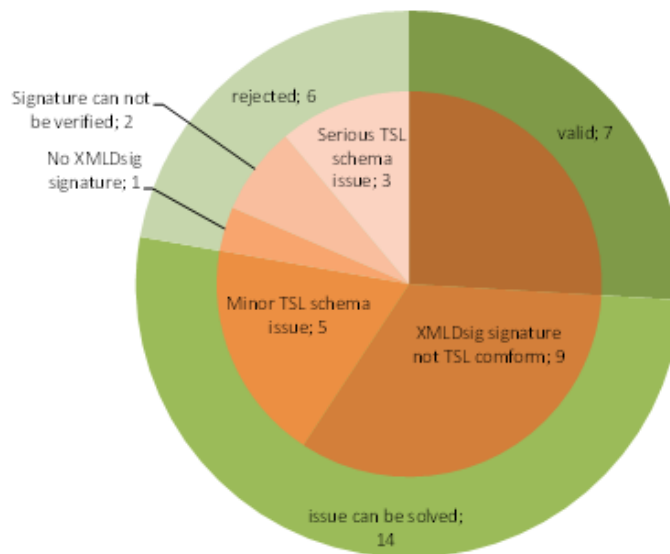


Figure 4.15: Distribution of TSLs in the implementation according to their validation flaws (as at 04.06.2013)

“[...] This REQUIRED field specifies the latest date and time by which the next TSL will be issued expressed as UTC time.” [ETSI, 2009]

Unfortunately, some MS-TSLs have been expired²⁸. As consequence, these TSLs cannot be seen as trustworthy, even if the rest of the TSL is valid (which is not the case for most of the expired TSLs anyhow). Hence, as the individual Member States are responsible for issuing their TSL, they must be taken into account to establish an appropriate governance process.

4.6 Summary and Conclusions

The main contributions in this chapter are the interoperable electronic document framework OCD for a secure exchange of e-Documents across borders and the cross-border signature verification via trust-service status lists. Both are important cornerstones for secure and interoperable e-Government services across borders.

The OCD has proven its applicability and adaptability through the 18 months lasting piloting phase in the LSP SPOCS. From a technical perspective all requirements of the (piloting) countries have been fulfilled and the quality OCD modules has been assessed very high. Nevertheless, a few action points for the future development of OCD exist. These action points have been handed over to the follower LSP e-SENS (cf. Section 3.2.5.7) as they must be treated on a higher level and do not concern the OCD only. First, although the OCD bases mostly on open standards, it would be a great benefit to standardize the OCD itself. Concrete suggestion for this standardization, which is seen as a key element for achieving sustainability, are given in the OCD final report [Stranacher, 2013]. The second action point concerns the equivalence of electronic documents. That means that there is still a missing

²⁸As at 26.02.2014 six MS-TSLs (DE, DK, FR, IE, LV, MT) are expired.

consent on which e-Documents from a Member State A are legally equivalent to an e-Document in Member State B. For instance, what is the equivalent Austrian document to a Greek certificate of qualification for an architect? It is even not clear if there exists such an equivalency for all use cases. This equivalency is an important issue concerning the interoperability of e-Documents. However, this has been out of scope of the OCD and has been handed over to the LSP e-SENS therefore.

The integration of trust-service status lists into the Austrian signature verification service MOA-SP has been positively evaluated. Nevertheless, the entire system is dependent from the quality of the issued MS-TSLs. Unfortunately, as shown in Section 4.5.5.2, the quality of the issued TSLs is widespread and only a small number of TSLs are fully correct. For some TSL, which have minor errors only, error correction mechanisms of the core TSL library are applicable. Nevertheless, some TSLs have major technical errors. The reason for that may base on the complexity of the TSL specification. In the first EC TSL-Decision [European Commission, 2009a] and their corrigenda [European Commission, 2009b], the EC references to the TSL specification only. This has been recognized by the EC as not sufficient. Hence, the first amendment of the Decision [European Commission, 2010c] adds some additional information on how to issue a MS-TSL. Unfortunately this additional information has been still not adequate. Therefore, the currently last amendment [European Commission, 2013a] gives a detailed common template for issuing the MS-TSL and references a new TSL specification [ETSI, 2013b], which is intended to replace the former TSL specification [ETSI, 2009]. This last decision shall apply from 1st February 2014, but is still not implemented in all Member States. Nevertheless, this hopefully will increase the quality of the issued MS-TSLs in the near futures. However, the entire EC decision amendment process is quite an indicator for the complexity of the TSL specification and their issues during the implementation in the Member States.

Furthermore, another non-technical problem exists. All Member States are responsible for issuing their national TSL. Unfortunately not every MS has a clear governance process, as six MS-TSLs are currently (as at 26.02.2014) expired. Here, corresponding measures must be introduced from EC side to obtain non-expired MS-TSLs.

Chapter 5

Examination and Assessment of Editable Signatures



“Education is what remains after one has forgotten what one has learned in school.”

[Albert Einstein]

5.1 Introduction

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 identify¹ the creator of the signature (authenticity) and is able to verify that the signed data has not been modified (data 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.

Conventional electronic signatures are currently the means of choice to assure trustworthiness in e-Business and e-Government applications and services. They rely on the fact that signed data, which are modified by any entity, immediately breaks the signature. That means that during a signature verification process these modifications (even if the signatory itself has made them) are detected and the verification result is negative. As detailed highlighted in Section 3, the progressing digitalisation as well as the need for interoperability and next-generation applications creates additional use cases. Hence, new challenges for electronic signatures arise. These challenges cannot efficiently be fulfilled by currently deployed conventional electronic signatures.

The common requirement for these new use cases can be summarised as the need to allow subsequent modifications of signed data, by still preserving the validity of the original signature. This is known as the digital document sanitizing problem, first published by Miyazaki et al. [2003] and treated in detail in Section 3. Obviously, such modifications cannot be fulfilled by conventional electronic signatures. Editable Signatures allow such subsequent modifications². In the last years a variety of editable signature schemes³ has been introduced in literature, but their capabilities to assure the integrity and authenticity in e-Business and e-Government use cases has not been assessed so far. This renders a concrete implementation of solutions based on editable signatures impossible. To overcome this issue, this chapter examines and assesses existing editable signature schemes. Hence, the main objectives of this chapter are:

- Presentation of the status quo of current developments in the sector of editable signatures.
- Definition of requirements for using editable signatures in e-Business and e-Government applications especially.
- Detailed examination and assessment of selected editable signature schemes.

Therefore, the remainder of this chapter is structured as follows. In Section 5.2 the status quo in the area of editable signatures is presented. This includes descriptions of the basic principles and the current fields of applications of the different editable signatures schemes. Section 5.3 derives concrete legal, organisational and technical requirements that have to be met by editable signature schemes when being applied to e-Government and e-Business applications. Potential and well selected

¹For a *unique* identification a legal basis must be given, such as the Austrian Citizen Card [Leitold and Posch, 2004; Leitold H., 2002]

²For an (informal) definition of an editable signature see following subsection 5.2

³In literature, the terms signature scheme and signature are often treated synonymously. However, a formal difference exists. A signature scheme defines the formal model and processing to create and verify a signature according to the schema, whereas a signature represents the implementation of a signature scheme.

candidates of editable signature schemes are examined and discussed in Section 5.4. In Section 5.5, the derived requirements are mapped to the examined editable signature schemes in order to assess the schemes' capabilities to meet the given requirements. Finally, Section 5.6 summarizes the findings of this chapter and draws conclusions.

5.2 Status Quo of Editable Signatures

Editable signatures exist about a decade and several different editable signature schemes have been published. However, the nomenclature and notations are inconsistent and differ slightly in literature. Hence, this subchapter gives some preliminary (informal) definitions. First of all, an editable signature scheme is defined as follows:

Definition 3 *An “editable signature scheme” is a signature scheme, which allows modifications of signed data, but preserves the authenticity and integrity of the unchanged data.*

Editable signature schemes can be categorised into three different schemes depending on the way how these modifications take place and depending on the parties, which are able to do these modifications. The first category - *redactable signatures* - has been invented by Johnson et al. [2002] and Steinfeld et al. [2001] in parallel. They represent the simplest editable signature schemes and are defined as follows:

Definition 4 *A “redactable signature scheme” enables any party to delete/exchange (parts of) signed data by a single character by still preserving the validity of the original signature.*

The typical use case for redactable signatures is anonymization, i.e. to redact or blacken text blocks from signed data. Furthermore a *redactor* is defined as follows:

Definition 5 *A “redactor” is a party, which conducts the redaction and exchanges certain message blocks in the signed data by a single character.*

Sanitizable signature schemes are the second category and represent a further development of redactable signature schemes. These schemes allow for more sophisticated modifications by another pre-defined party. Formally, sanitizable signature schemes are defined as:

Definition 6 *A “sanitizable signature scheme” is a signature schema, which enables designated parties to replace predetermined parts of original signed data by still preserving the validity of the original signature.*

That means the original signatory of a sanitizable signature is able to define which parts of a signed message are modifiable and which party or parties are allowed to do such modification. These delegated parties are defined as follows:

Definition 7 *A “sanitizer” (or “censor”) is a party, which receives rights (from the original signatory) to modify certain parts of signed data.*

The third and last category of editable signature schemes are *blank digital signature schemes*. Blank digital signatures are a novel scheme proposed by Hanser and Slamanig [2013]. According to the authors a blank digital signature scheme is defined as follows:

Definition 8 An “*blank digital signature scheme*” is a signature scheme, which allows an originator to define and sign a message template. This message template describes fixed parts of a message as well as several choices for exchangeable parts of a message. Then, a proxy is given the power to sign template instantiations of the template given by the originator by using some secret information. The resulting message signature can be publicly verified under the originator’s and the proxy’s signature verification keys. [Hanser and Slamanig, 2013]

In the following subchapters the basic principle of these three editable signature schemes and their fields of application are explained.

5.2.1 Basic Principles

5.2.1.1 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. Figure 5.1 illustrates the basic principle for the creation of a conventional signature and a redactable signature. First of all, to create a conventional signature, a message m is divided into several message blocks. For illustration, we assume a split into $m_1 \dots m_5$. For each of these message blocks a hash function H is applied, creating the hash values $h_1 \dots h_5$. 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.

To create a redactable signature, we assume to redact the message block “private” in Figure 5.1. Thus the redacted message block m_4^* contains a “*” character which blinds the message block “private”. Computing the hash value of the redacted message block m_4^* 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 verification process⁴. 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 nature of the hash function. Several redactable signatures schemes do exist, which all base on this basic principle of retaining the original hash values.

If a redacted message block is small or has a strong structure, e.g. only “Yes” or “No” is possible, the redacted message can be “reconstructed” by a simple computation of all hash values and comparison of these result with the given hash value. Therefore, for real implementations of redactable signature schemes, the hash function is replaced by so called commitments⁵ as shown by Steinfeld et al. [2001]. A further extension, as introduced by Johnson et al. [2002], is to use hash trees to reduce the number of randomizers, which are needed for the commitment.

⁴That means $H(m_4)$ instead of $H(m_4^*)$ is used for calculating $Hash_{TOTAL}$ during signature verification.

⁵Commitments 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.

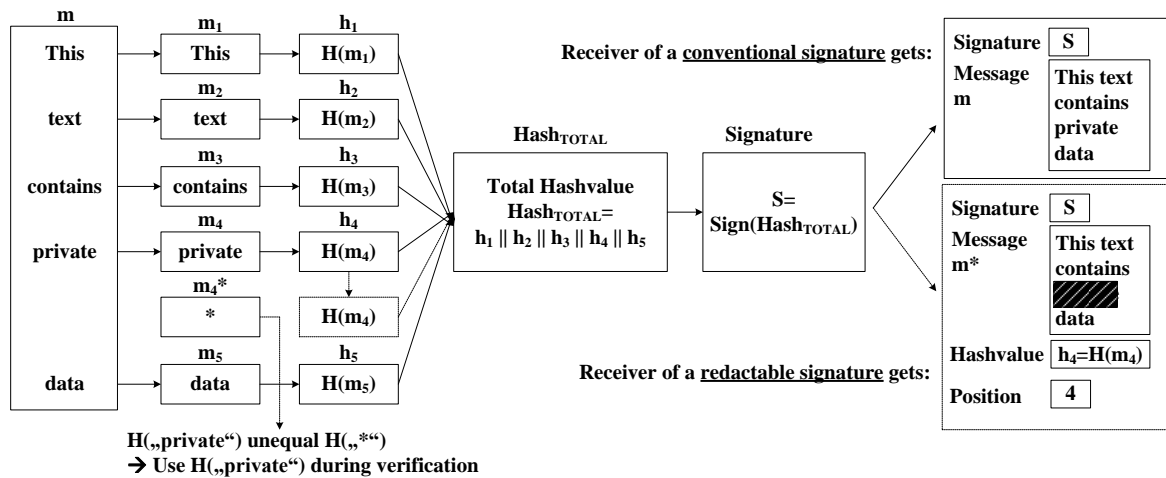


Figure 5.1: Basic principle of redactable signatures

Figure 5.2 illustrates the sequence diagram for the basic principle, starting from the redactable signature creation, the message redaction to the signature verification. In detail, the process consists of following steps:

- Step 1 - Signature creation:** The message m is created and splitted into the message blocks $m_1 \dots m_5$. Then the hash function H is applied to all message blocks, thus creating hash values $h_1 \dots h_5$. The total hash value $Hash_{TOTAL}$ is created by a concatenation of these hash values. Finally the signature S is created by signing $Hash_{TOTAL}$ using the signatory's private key. The signature S (including $Hash_{TOTAL}$) and the message m are sent to the redactor.
- Step 2 - Redaction:** The redactor receives the message m and redacts the word “private” by replacing it with another character, such as “*”. To retain the validity of the signature S the redactor must add information to the S . Therefore, the redactor stores the original hash value h_4 and the redaction position 4. Finally, the signature S , the redacted message m^* , the original hash value h_4 and the redaction position 4 are sent to the verifier.
- Step 3 - Signature verification:** The verifier builds the message blocks $m_1 \dots m_5$ out of the redacted message m^* . Then, she applies the hash function H to all message blocks except for m_4 (indicated by redaction position 4). In this case the received hash value h_4 is used. After that the hash values $h_1 \dots h_5$ are concatenated to the total hash value $Hash_{TOTAL}^*$. If this (re-calculated) hash value is equal to the hash value $Hash_{TOTAL}$ contained in the signature S , the message has not been altered (except for the redaction of message block m_4). Finally, the verifier verifies the signature itself by using the signatory's public key.

5.2.1.2 Sanitizable Signatures

A main property of redactable signature schemes is that they only allow blacken certain message blocks of a signed message. To allow also *replacements* of message blocks with other message blocks, Ateniese et al. [2005] introduced the concept of sanitizable signatures, which emerged from chameleon

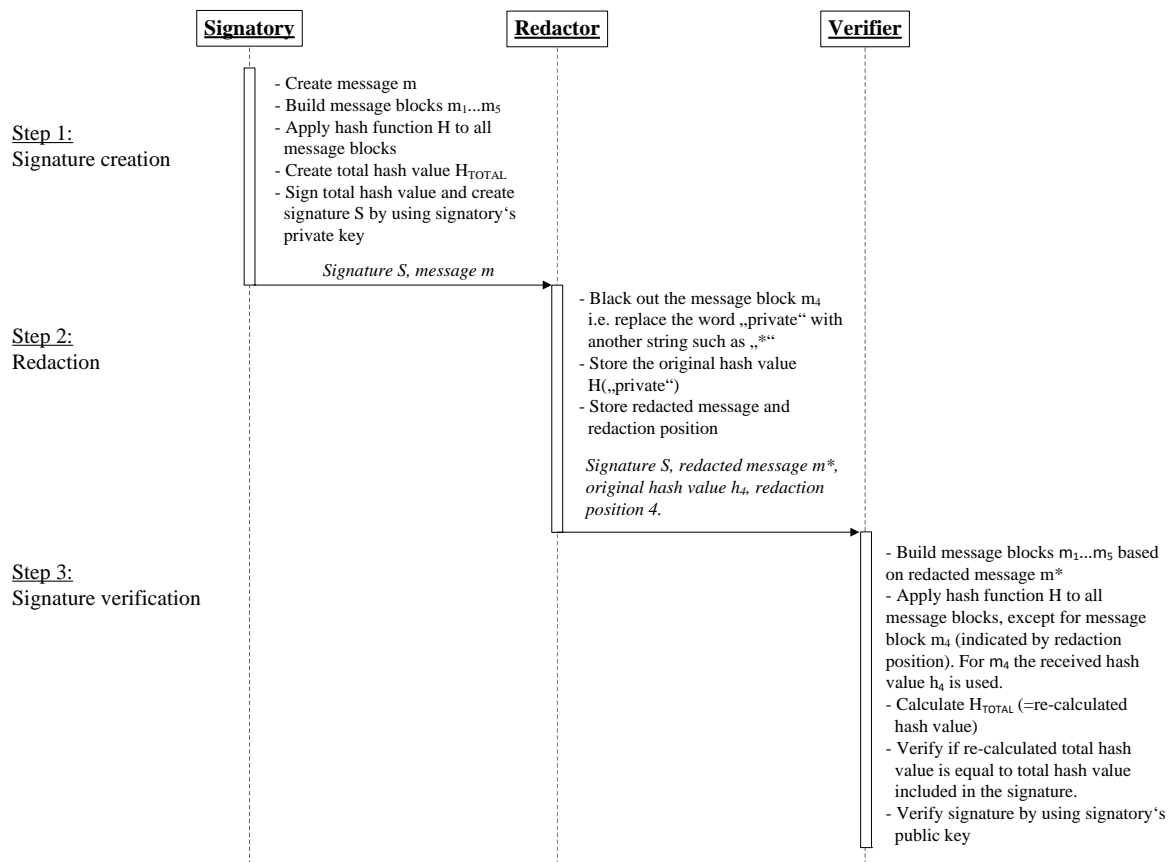


Figure 5.2: Sequence diagram of redactable signatures

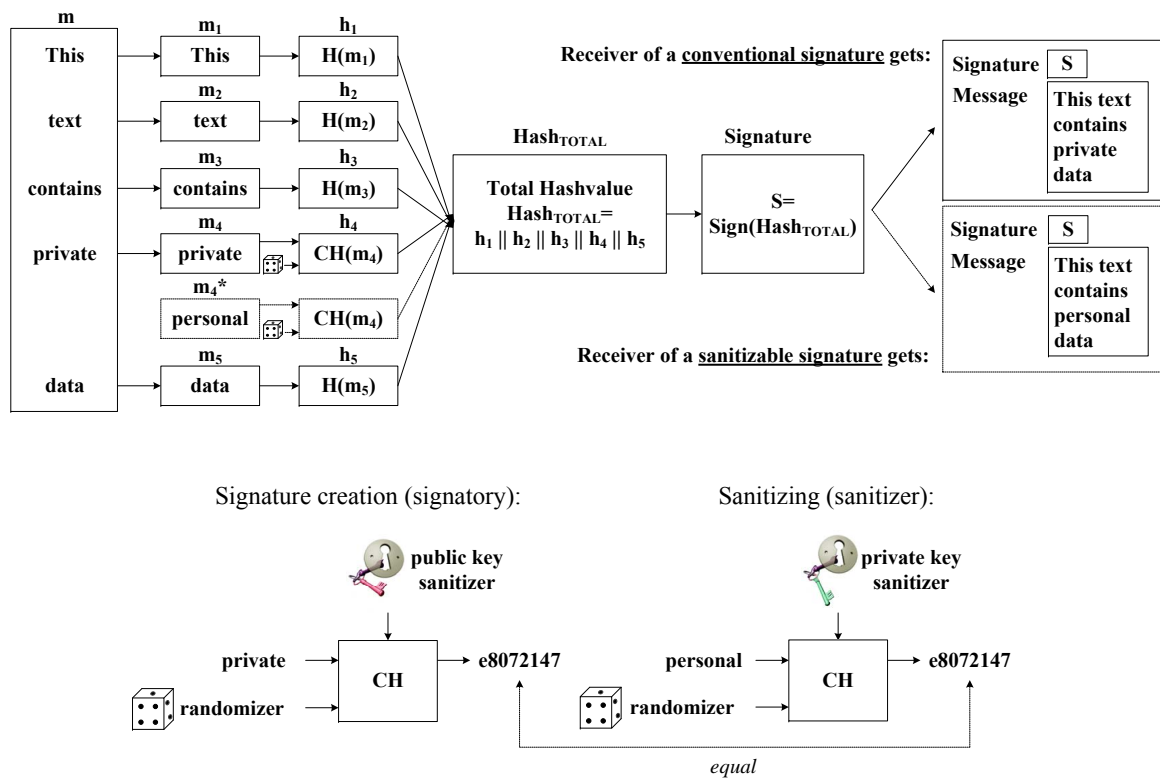


Figure 5.3: Basic principle of redactable signatures

signatures introduced by Krawczyk and Rabin [2000]. Sanitizable signatures use a chameleon hash function (also called as trapdoor commitment) such as [Ateniese and Medeiros, 2004] for message blocks that are sanitizable (see also Figure 5.3). Such chameleon hash functions are parameterized with the public key of the sanitizer⁶ and have randomizers as additional input. Thereby, the original signatory defines which concrete public key is used as parameter. Because of this parameterisation, the sanitizer is able to compute hash collisions by using her private key (=trapdoor information), which corresponds to the defined public key. This means the sanitizer is able to generate message blocks, which lead to the same hash value as for the sanitized message block. Obviously, only sanitizers, which are in possession of the corresponding private key are able to sanitize. Furthermore sanitizable signature schemes allow for a restriction of the message space of the message to be exchanged. That means the signatory can specify the message blocks which are used for the replacement. For instance, only the message blocks “car” or “house” can be inserted.

Figure 5.4 shows a sequence diagram of a sanitizable signature creation and verification. The entire process consists of following steps:

Step 1 - Signature creation: The message m is created and splitted into the message blocks $m_1 \dots m_5$. Then the hash function H is applied to all message blocks, which should be non-exchangeable. In our example this applies for the message blocks m_1 , m_2 , m_3 and m_5 (thus, creating the

⁶Or even with the public keys of several sanitizers depending on the functionalities of the used chameleon hash function and the concrete sanitizable signature scheme

hash values h_1, h_2, h_3 and h_5). For the message block m_4 , which should be exchangeable, the chameleon hash function CH is applied by using sanitizer's public key and produces the hash value h_4 . Afterwards the total hash value $Hash_{TOTAL}$ is created by a concatenation of hash values h_1 to h_5 . Finally the signature S is created by signing $Hash_{TOTAL}$ using the signatory's private key. The signature S (including $Hash_{TOTAL}$) and the message m are sent to the sanitizer.

Step 2 - Sanitizing: The sanitizer receives the message m and exchanges (sanitizes) the word "private" with the word "personal". To retain the original (chameleon) hash value h_4 , which is $CH("private")$, the sanitizer is able to compute collision by using her private key so that $CH("private")$ is equal to $CH("personal")$. Finally, the sanitizer sends the signature S and the sanitized message m^* to the verifier.

Step 3 - Signature verification: The verifier builds the message blocks $m_1 \dots m_5$ out of the sanitized message m^* . Then, she applies the hash function H to all message blocks except for m_4 . For message block m_4 she applies the chameleon hash CH (without the trapdoor information). These hash values h_1 to h_5 are concatenated to the re-calculated total hash value. After that the verifier verifies if this re-calculated hash value is equal to the hash value included in the signature. If these values are equal the message has not been altered or modifications have been allowed by the signatory (as it is the case in our example). Finally, the verifier verifies the signature itself by using the signatory's public key.

5.2.1.3 Blank Digital Signatures

Blank digital signatures are a novel scheme invented by Hanser and Slamanig [2013]. Figure 5.5 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 the originator gives a proxy the permission to create message instances. In the instantiation process the proxy selects certain choices of the exchangeable message parts. Finally, the proxy signs the message instance. This resulting signature can be publicly verified using the originator's and proxy'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. Thereby, the verifying party does not learn anything about the unused choice from the template.

Figure 5.6 shows the sequence diagram of an entire blank digital signature process. This process consists of following steps:

Step 1 - Template creation: The originator specifies an appropriate message template by defining fixed and exchangeable message blocks. These exchangeable message blocks contain several choices, which can be selected by the proxy.

Step 2 - Assignment of permissions: The originator gives a proxy⁷ the permission to create message instantiations of the defined template.

⁷Permissions can also be given to several proxies. For illustration, we assume that the permission is given a single proxy only.

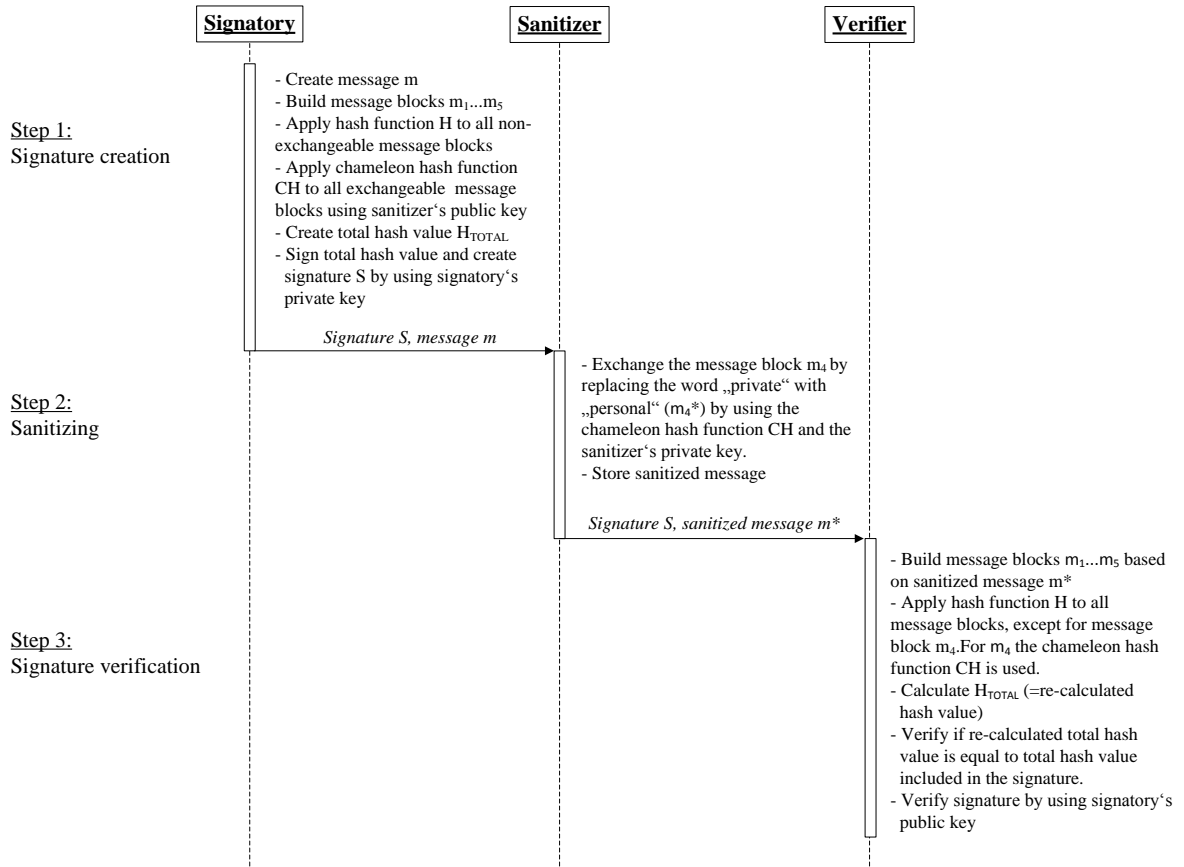


Figure 5.4: Sequence diagram of sanitizable signatures

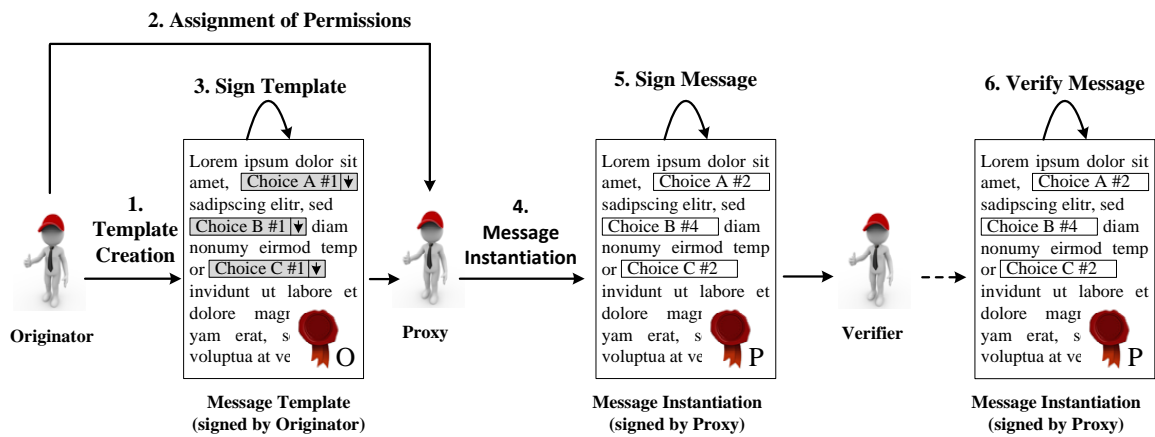


Figure 5.5: Basic principle of blank digital signatures

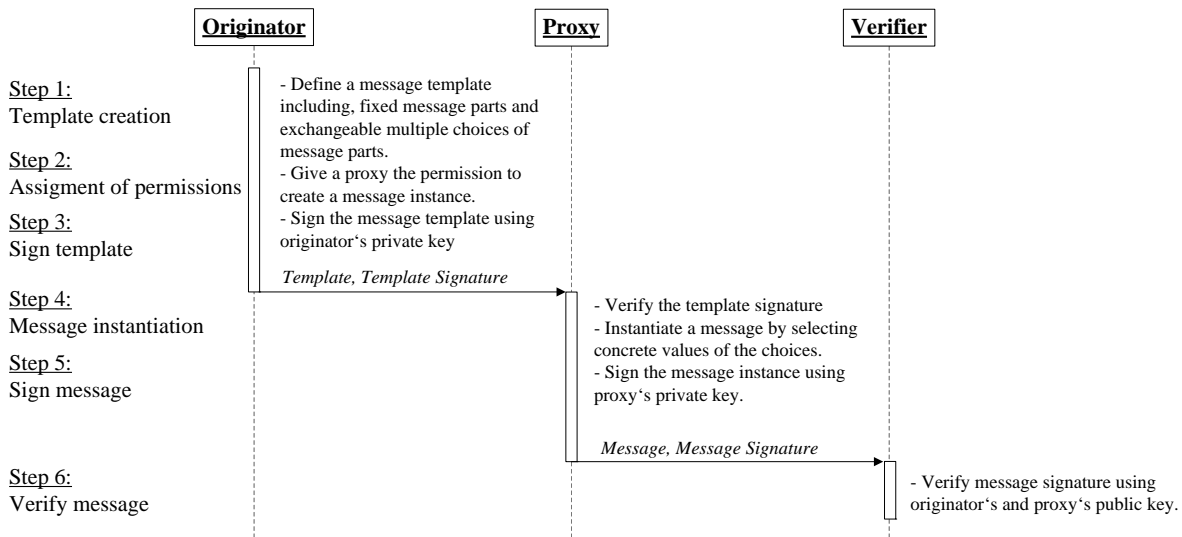


Figure 5.6: Sequence diagram of blank digital signatures

Step 3 - Sign template: The template is signed by the originator using the originator's private signature key. Finally, the originator sends the template and the template signature to the proxy.

Step 4 - Message instantiation: The proxy verifies the received template signature. Then the proxy creates a message instance by selecting concrete values of the available choices of the exchangeable message parts.

Step 5 - Sign message: The message instance is signed by the proxy using the proxy's private signature key. Then the proxy sends the message instance and the message signature to the verifier.

Step 6 - Message verification: The verifier is able to verify the message signature by using the originator's and proxy's public key.

Blank digital signature schemes can also be applied as signature scheme with similar capabilities as sanitizable signature schemes. That means the exchangeable parts of the message template can be interpreted as replacements. Nevertheless, some main differences exist:

- The originator does not commit to an instantiation, i.e. selecting concrete values for the exchangeable message blocks. In contrast, the signatory of a sanitizable signature signs a message ("commits to an instantiation"), but gives the sanitizer the possibility to replace defined message blocks.
- The unused choices of a blank digital signature message template are not revealed to the verification party, which is in contrast to sanitizable signatures. Here, the not selected replacement options may be visible to the verification party (depending on the concrete sanitizable signature scheme).
- The sanitized message of a sanitizable signature is not signed by the sanitizer (as part of the scheme protocol). In contrast, the message instantiation of a blank digital signature is signed by

the proxy. That means, to assure the authenticity of the sanitized message, the message must be signed by the sanitizer separately.

5.2.2 Fields of Application

Johnson et al. [2002] and Steinfeld et al. [2001] describe the fields of application for redactable signature schemes very generally. Here, the general use case is the redaction and censorship of signed records. This has been refined by Bauer et al. [2009], Brzuska et al. [2010a] and Slamanig and Rass [2010]. They describe different use cases for the application of redactable signatures in electronic healthcare applications. Thereby, redactable signatures are mainly used for anonymization of medical records. In particular, Slamanig and Rass [2010] elaborate on *achieving k-anonymity* as well as on *privacy preserving and unbiased opinions*. This covers the anonymization and redaction of medical records, so that patients cannot be uniquely identified anymore or other medical experts are able to give an unbiased diagnosis.

Sanitizable signatures provide more flexibility regarding modification of data in contrast to redactable signatures. Hence, sanitizable signatures have a wider field of application. Of course, they can also be used for anonymization of data records similar to the use cases of redactable signatures. In addition, Canard et al. [2008] elaborates on the usage of sanitizable signatures in the area of licenses for digital right management. Furthermore, Ateniese et al. [2005] shows use cases for sanitizable signatures for secure routing in modern routing protocols.

As blank digital signatures have similar capabilities as sanitizable signature, all fields of application described above also apply to blank digital signatures. Additionally, Hanser and Slamanig [2013] elaborate on the usage for partially blank signed contracts. In this use case, a person wants to sign a contract under certain conditions, e.g. the person is willing to buy a car for a predefined set of potential prices. This person can now delegate the purchasing to her attorney, by using a blank digital signature, defining the set of prices as exchangeable message blocks. Finally, the attorney is able to select a certain price, whereas the signature of the person remains valid.

5.3 Requirements

5.3.1 Legal Requirements

The EU Signature Directive [The Council of the European Union, 2000] does not differ between conventional signatures, editable signatures or any other signature type. Therefore the regulations and requirements, defined in the Directive, also apply for editable signatures. Therefore, following general legal requirements are defined:

Accountability: In case of a dispute the signatory must be able to prove that certain modifications have been done by a certain party. 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).

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 thus to identify the original signatory.

Qualified Electronic Signature: These additional requirements are not necessarily needed for all e-Business and e-Government use cases. An editable signature scheme may, optionally, meet also the requirements for qualified electronic signatures as defined by the Signature Directive.

5.3.2 Organisational Requirements

Beside legal requirements, there exist also some general requirements on organisational level. These requirements concern mainly the role of the modification-parties⁸ and the original signatory. So, following general organisational requirements are defined:

Definition and Revocation of Modification-Parties: Modification-parties should be easily definable by using existing systems (to avoid additional investments) and the signatory should also have the opportunity to revoke the modification permissions.

Non-Disclosure Agreement: Modification-parties must sign an appropriate confidentiality agreement. In particular, regarding the data protection, as these parties may 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 modification-parties (e.g. who is allowed to sign/redact, who is responsible in case of a dispute).

Service Level Agreement/Security Compliance: Modification-parties must ensure to redact or sanitize data within an appropriate time frame (especially for real time data). Furthermore, modification-parties must be compliant to current security regulations as they may operate on private and personal data.

5.3.3 Technical Requirements

On a technical level there exists also some requirements, which are tightly bound to the particular editable signature schemes. Therefore, following technical requirements have been identified:

Designated Modification-Parties: Designated modification-parties 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 data. Persons except the signatory and the designated modification-parties must not be able to sanitize or redact data. Any change of the data by unauthorized persons must be recognizable.

Privacy: The redactable or sanitized data as well as the original signature must not allow revealing the redacted or sanitized 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 defined by [Bartel et al., 2008].

⁸The term *modification-party* is used to denote a redactor, sanitizer or proxy.

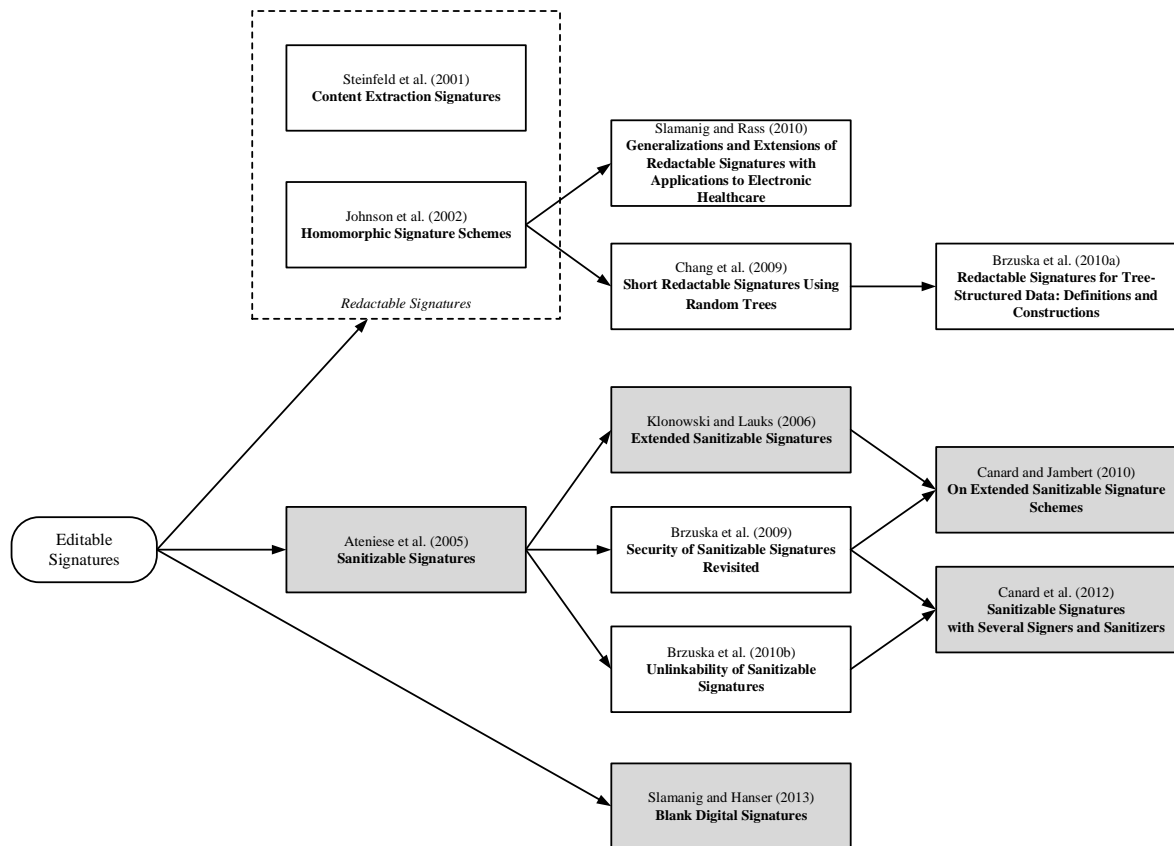


Figure 5.7: Overview about editable signature schemes

Compatibility: The signature scheme must be compatible to (at least one of) the reference signature formats defined in the European Commission Decision 2014/148/EC [European Commission, 2014a].

5.4 Examination

In the following, various editable signature schemes are examined. Figure 5.7 shows an overview of the most relevant⁹ 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 and e-Government services is to support the definition of designated modification-parties. Redactable signature schemes, such as Steinfeld et al. [2001] and Johnson et al. [2002], do not offer the definition of designated modification-parties. They allow any party to perform redactions. 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 modification-parties.

⁹Relevant in terms of citation rate and author's reputation (mainly based on h-index).

Thus, the following subsections examine selected editable signature schemes only. The selected signature schemes, which are marked grey in Figure 5.7, 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 this scheme is skipped from the analysis.
- Brzuska et al. [2010b] proposed an update of Ateniese et al. [2005] which does not permit creating a link between different signatures over the same original message. This functionality is not of interest for the e-Business and e-Government use cases, so this scheme is skipped too.

5.4.1 Sanitizable Signatures by Ateniese et al. [2005]

The basic principle of sanitizable signatures bases upon commitments, which in turn build upon hash functions. Ateniese et al. [2005] proposed the first scheme for sanitizable signatures, where a designated sanitizer is able to modify designated parts of a signed message. Based on using chameleon hash functions the sanitizer 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.

5.4.2 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 sanitizer 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 accumulators¹⁰ or bloom filters¹¹.

5.4.3 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.

¹⁰An accumulator is a one-way hash function which satisfies a quasi-commutative property. See Benaloh and Mare (1994) for details.

¹¹Bloom filters are data structures which allow to efficiently test whether an element is a member of a certain set or not. See Bloom (1970) for details.

5.4.4 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 signatories and multiple sanitizers. To support multiple signatories and sanitizers, the authors make use of group signatures¹². Their scheme also provides group anonymity. That means a signatory (resp. sanitizer) is anonymous for other entities, which are not in the group of signatories (resp. sanitizers).

5.4.5 Blank Digital Signatures by Hanser and Slamanig [2013]

Blank digital signatures, proposed by Hanser and Slamanig [2013], are a new signature scheme, which makes use of elliptic curve pairings¹³ and polynomial commitments¹⁴. 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 scheme¹⁵. This update includes a simplified construction of the signatures allowing significantly performance enhancements. Finally, this update incorporates full security proofs.

5.5 Assessment

5.5.1 Legal Assessment

In this section, editable signature schemes are assessed 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.

¹²Group signatures give a group of signatories signing rights.

¹³Pairings are bilinear mappings as defined by Silverman (1986).

¹⁴Conventional commitments applied to polynomial functions are called polynomial commitments (see Kate et al. (2010) for details).

¹⁵https://online.tugraz.at/tug_online/voe_main2.getvolltext?pCurrPk=69904

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 the best of the thesis author's knowledge - 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 sanitizers, 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 5.5.3).

5.5.2 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 e-Business and e-Government services. In fact, all organisational requirements identified in Section 5.3.2 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 sanitizers can be fulfilled on technical level as all of the proposed schemes rely on a public key infrastructure and hence on existing and well-established revocation mechanism. Thereby, the role of the sanitizer can be indicated by an additional object identifier (OID) in the certificate. 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.

5.5.3 Technical Assessment

The technical assessment concerning applicability to structured data and the signature format compliance to the European Commission Decision 2014/148/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 [Bartel et al., 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 subsections comprise the further technical assessment of the different editable signature schemes.

5.5.3.1 Assessment of Sanitizable Signatures by Ateniese et al. [2005]

Concerning designated sanitizers and designated parts 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 ...”

That means the requirement for designated modification-parties and designated parts is fulfilled. In addition the privacy requirement is also fulfilled as

“[...] the indistinguishability requirement provides for privacy.”

The authors also state that

“[...] accountability follows from the unforgeability requirement.”

Nevertheless, this has been proven as not true by Brzuska et al. [2009]:

“[...] our results are in contrast to the claim by Ateniese et al. [1] that, for example, accountability follows from the unforgeability requirement. Our results show that unforgeability follows from accountability whereas the other direction is not true.”

Hence, accountability is not provided by the Ateniese sanitizable signature scheme.

5.5.3.2 Assessment of Extended Sanitizable Signatures by Klonowski and Lauks [2006]

The extended sanitizable signature scheme of Klonowski and Lauks (2006) provides a designated modification-party 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.

5.5.3.3 Assessment of Extended Sanitizable Signature Schemes by Canard and Jambert [2010]

As this scheme strongly bases on Ateniese et al. [2005], it provides designated modification-parties as needed by the 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.”¹⁶

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

¹⁶Message parts which can be modified by a redactor are often called admissible blocks.

“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].

5.5.3.4 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 modification-parties 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.

5.5.3.5 Assessment of Blank Digital Signatures by Hanser and Slamanig [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 modification-parties requirement is fulfilled as

“Immutability guarantees that no malicious proxy can compute message templates or templates instantiations not intended by the signer.”

They even prove that their scheme supports the privacy requirement. Finally, the scheme fulfils the accountability requirement, as the originator/proxy signs the template/message instance with a conventional signature (which provides accountability in any case).

5.6 Summary and Conclusions

Table 5.1 summarizes the results of the legal and technical assessment. It shows that Ateniese et al. [2005] and Klonowski and Lauks [2006] are assessed to be not suitable for e-Business and e-Government applications. In contrast, the sanitizable signature schemes of Canard and Jambert [2010] and Canard et al. [2012] as well as blank digital signatures of Hanser and Slamanig [2013] meet all technical requirements. Hence these schemes are appropriate for the use in e-Business and e-Government applications. In addition, blank digital signatures fulfil the requirement on qualified electronic signature. Nevertheless, obstacles hindering an application of these schemes in real applications exist. Concrete implementations for these signature schemes do not exist yet or are not

compliant to the recommended advanced signature formats defined by the European Commission Decision 2014/148/EC.

To overcome these issues, the most promising editable signature scheme - blank digital signatures by Hanser and Slamanig [2013] - have been chosen to be extended, as a core-implementation is available [Derler, 2013]. Section 6 elaborates on these extensions in detail.

Table 5.1: Assessment summary (legal and technical) of examined editable signature schemes

Signature Scheme	Account-ability	AdES ^a	QES ^b	Des. MP ^c	Des. Parts	Privacy	Struc. data	2014/148/EC Compl.
Ateniese et al. [2005]	×	✓	×	✓	✓	✓	✓	≈
Canard and Jambert [2010]	✓	✓	×	✓	✓	✓	✓	≈
Canard et al. [2012]	✓	✓	×	✓	✓	✓	✓	≈
Klonowski and Lauks [2006]	≈ ^d	✓	×	✓	✓	✓	✓	≈
Hanser and Slamanig [2013]	✓	✓	✓	✓	✓	✓	✓	≈

✓ = applied/supported, × = not applied/supported and ≈ = partly applied/supported

^aAdES = Advanced Electronic Signature

^bQES = Qualified Electronic Signatures

^cDesignated modification-party

^dThis scheme supports accountability only for the version where accumulators are used. In case the bloom filter is used accountability is no achievable.

Chapter 6

An Advanced Editable Signature Scheme



“Science is organized knowledge.”

[Herbert Spencer]

6.1 Introduction

The quintessence of the digital document sanitizing problem [Miyazaki et al., 2003] is that conventional signatures immediately break if the signed data are modified. In Section 3.3.1.3 editable signatures are presented as a common approach to solve this problem. Editable signatures allow for (pre-defined) modifications of signed, but preserve the authenticity and integrity of the unchanged data. This enables more sophisticated next-generation applications in the area of open government data, identity management and public administration procedures (cf. Section 3.3.2 and Part III for details).

The previous Chapter 5 has evaluated and assessed different editable signature scheme. It has shown that only a few number of editable signature scheme are applicable in the e-Government domain. Namely, these are the sanitizable signature schemes of Canard and Jambert [2010] and Canard et al. [2012] as well as the blank digital signature scheme of Hanser and Slamang [2013] (cf. Section 5.5). For those positively assessed schemes core implementations usually exist. These core implementations focus on the cryptographic implementation. Hence, they lack on applicability in the e-Government domain, especially concerning the support of standardised interfaces or standardised signature and data formats.

To bridge this gap, the present chapter presents the implementation of an *advanced editable signature scheme*. As basis for this implementation, the core implementation of blank digital signatures (BDS) has been chosen, as this signature scheme has been additionally assessed to fulfil the requirements for qualified electronic signatures (cf. Section 5.5.1). Based upon the existing core implementation [Derler, 2013], an advanced editable signature scheme creating XML-based advanced electronic signatures (XAdES signatures) is implemented. This advanced signature scheme is applicable to any XML based content. This is beneficial as many e-Government based applications rely on XML based data and XAdES signatures are compliant to the signature reference formats defined by the European Commission Decision 2014/148/EC [European Commission, 2014a].

The remainder of this chapter is structured as follows. Section 6.2 defines requirements for the implementation of an advanced editable signature scheme. In Section 6.3 the existing core implementation of BDS is described in detail. Additionally, it includes an analysis of the parameters and values used in the BDS core implementation. Following Section 6.4 introduces the architecture for the advanced editable signature scheme based upon the BDS core implementation. Section 6.5 elaborates on the concrete implementation of this architecture. Finally, Section 6.6 evaluates this implementation against the defined requirements and draws conclusions.

6.2 Requirements

This section defines general requirements for the implementation of an advanced editable signature scheme as considered to be necessary by the thesis author. They must be applicable to any advanced editable signature scheme, when used in e-Government applications and services. Hence, following requirements are defined:

Applicable to standard data formats: The advanced editable signature scheme must be applicable to standardised data formats. Depending on the use case, the scheme may be applicable to a single data format such as XML [Bray et al., 2006] or PDF [ISO/IEC, 2008].

Standardised signature format: The signatures created by the advanced editable signature scheme must be compliant to a standardised signature format. When deployed in the e-Government domain, the signature format must be one of the reference formats defined by the European Commission Decision 2014/148/EC [European Commission, 2014a].

Complexity and integration effort: Editable signature schemes are usually far more complex as conventional signature schemes. Nevertheless, an advanced editable signature scheme should hide the complexity to the user as far as possible. In addition, in the e-Government domain various applications and service are already deployed. Although advanced editable signatures target on next generation applications, existing infrastructures and applications should be taken into account to reduce the integration effort when advanced editable signatures are deployed.

Rely on existing core implementations: Existing core implementations of editable signature schemes are usually well assessed by the research community. Especially from a security perspective the signing features of these core implementations can be considered to be secure. Hence, it is reasonable that advanced editable signature schemes should rely on such core implementations.

6.3 Existing BDS Core Implementation

On the one hand BDS has been the best assessed editable signature in Chapter 5, on the other hand it is also the most flexible editable signature scheme. Therefore BDS has been chosen for implementing an advanced editable signature scheme. In the following subsection the process flow of the BDS core implementation is described. Then the BDS template and message format is presented. Finally, it is evaluated how this implementation can be used for an XAdES-based implementation of an advanced editable signature scheme. This mainly comprises an analysis of the use parameters and values in the BDS process flow.

The entire BDS process consists of following entities: (a) a trusted third party (TTP) for attesting keys and issuing certificates, (b) an originator, who creates signed templates, (c) a proxy, who is assigned by the originator to create a signed message instances based upon the templates and (d) any entity, which receives the signed message and is able to verify it.

6.3.1 Process Flow

Figure 6.1 illustrates the process flow of the core implementation. The process flow bases upon the basic principle presented in Section 5.2.1.3 and consists of following steps:

Step 0: First of all the key pairs for the conventional digital signature scheme (DSS) are generated and attested by the TTP - for the originator and the proxy as well.

Step 1: Afterwards, by using the security parameter κ^1 and the maximum template size t , the public parameters pp for BDS are created by using function K . These parameters are made publicly available for all involved entities.

¹This security parameter also contains the system parameters *sysparams*.

- Step 2:** The originator defines an appropriate template T (see Section 6.3.2 for the template format) and signs it by using function S and following parameters: her private signing key sk_{Orig}^{DSS} , the proxy's public verification key pk_{Proxy}^{DSS} , the template T and the public parameters pp . This creates the template signature σ_T and a private template-dependent key sk_{Proxy}^T . Both of them are sent to the proxy.
- Step 3:** The proxy verifies the template signature σ_T by using the function V_T and additionally using the template T , the originator's public verification key pk_{Orig}^{DSS} , the private template-dependent key sk_{Proxy}^T , her public verification key pk_{Proxy}^{DSS} and the public parameters pp .
- Step 4:** If the template signature verification was successful, the proxy creates the message M out of the template T . Then she creates the message signature σ_M by using function I with following parameters: the template T , the message M , the template signature σ_T , the private template-dependent key sk_{Proxy}^T , her private signing key sk_{Proxy}^{DSS} and the public parameters pp . Finally, message M and message signature σ_M can be passed to any entity.
- Step 5:** By using the message M , the message signature σ_M , the public verification keys of originator and proxy (pk_{Orig}^{DSS} and pk_{Proxy}^{DSS}) and the public parameters p , any entity is able to verify the message signatures via the function V_M . In case the verification is positive, the message M is a valid instance of the template T .

6.3.2 BDS Template and Message Format

The BDS core implementation uses a specific template and message format, which is tightly bound to the implementation itself. Hence an analysis of this format is needed. According to [Derler, 2013], Figure 6.2 and Figure 6.3 illustrate the template and the message format². The template format allows several message elements within one `templateentry` element. In contrast, the message format allows only one message element, which represents the selected choice. In addition, the message element contains an attribute `type` with following three options: (a) `blank` to denote any message blocks, which can be inserted by the proxy (but limited to the value given in the `length` attribute), (b) `exch` denotes a message block, which can be exchanged by the proxy (possible message blocks are given in the `text` elements) and (c) `fix` to denote fixed (non-modifiable) message blocks whereas the fixed message block is given in the `text` element. Obviously, this format is applicable to any unstructured text based message, but lacks on XML based data.

Finally, as last element, the `signature` element represents the template signature in the template format and the message signature in the message format. Obviously the format of this `signature` element is proprietary and does not follow any standard or specification.

6.3.3 BDS Parameter Analysis

During a BDS core process different parameters are exchanged between the affected parties. Here, the exchanges between originator and proxy as well as between proxy and any entity are of special

²“?” means at most once, “+” means at least once and “*” means any number of times.

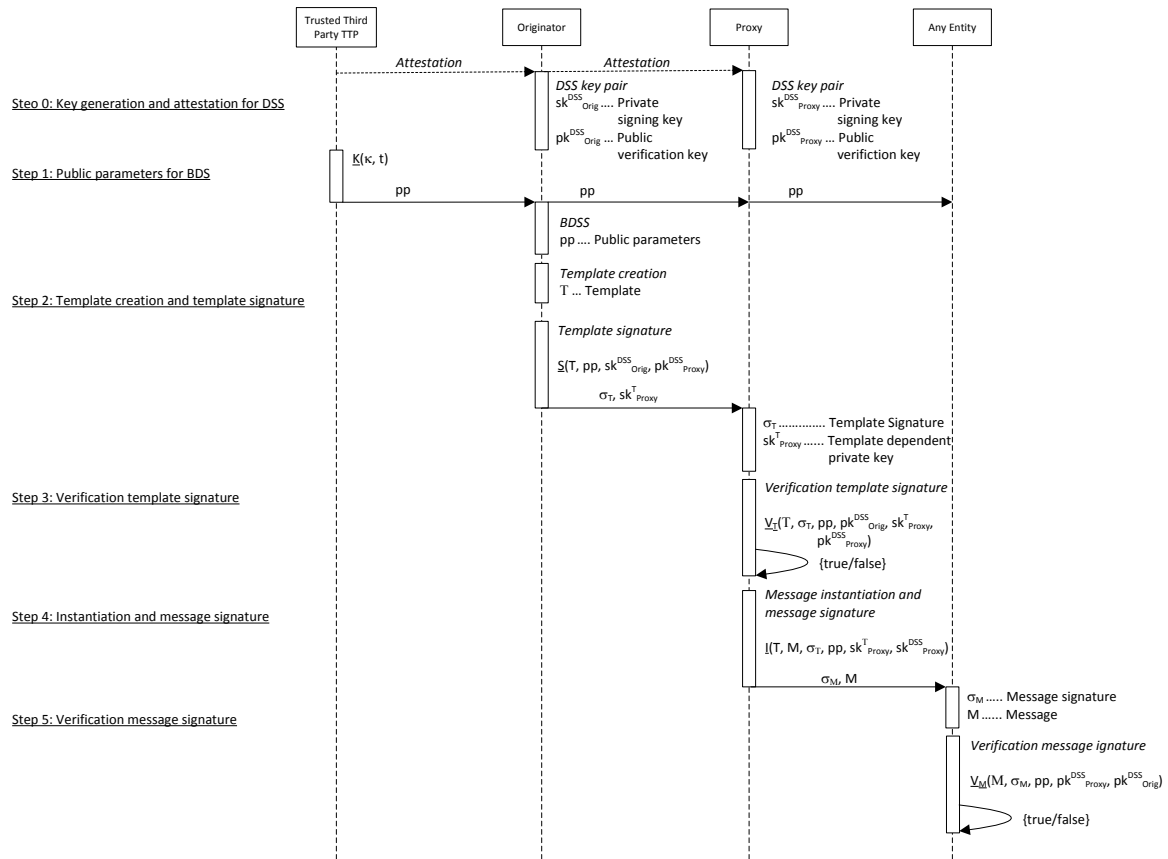


Figure 6.1: Process flow: BDS core implementation

```

1 <template id="...">
2   (<templateentry>
3     (<message type="blank||exch||fix" length="[Integer]">
4       (<text>[String]</text>)?
5     </message>)*
6   </templateentry>)+
7   (<signature>
8     <signaturevalue>[Base64 encoded string]</signaturevalue>
9     (<keyId>[String]</keyId>)?
10    (<ttpcert>[Base64 encoded string]</ttpcert>)?
11    (<originatorcert>[Base64 encoded string]</originatorcert>)?
12    (<proxycert>[Base64 encoded string]</proxycert>)?
13  </signature>)?
14 </template>

```

Figure 6.2: BDS template format [Derler, 2013]

```

1 <instance id="...">
2   (<message type="blank|exch|fix" length="[Integer]">
3     <text>[String]</text>
4   </message>)+
5   (<signature>
6     <signaturevalue>[Base64 encoded string]</signaturevalue>
7     (<keyId>[String]</keyId>)?
8     (<ttpcert>[Base64 encoded string]</ttpcert>)?
9     (<originatorcert>[Base64 encoded string]</originatorcert>)?
10    (<proxycert>[Base64 encoded string]</proxycert>)?
11  </signature>)?
12 </instance>

```

Figure 6.3: BDS message format [Derler, 2013]

interest. During these exchanges the respective signatures³ are sent from one party to the other one. The analysis of all other exchanged parameters is vital too as they are needed for the verification of the signatures. Hence, Table 6.1 lists all exchanged parameters for both exchanges.

Table 6.1: BDS parameter analysis

Originator \Leftrightarrow Proxy	Proxy \Leftrightarrow Any Entity	Meaning
$sysparams$	$sysparams$	BDS system parameters
pp	pp	Public BDS parameters
σ_T		Template signature value
	σ_M	Message signature value
sk^T_{Proxy}		Private template-dependent key
pk^{DSS}_{Orig}	pk^{DSS}_{Proxy}	Public verification keys of originator and proxy
T		Template
	M	Message

6.4 Architecture

In this section, the architecture of the BDS based advanced editable signature scheme is presented. First of all, to fulfil the defined requirements the architecture must consider two main extensions to the core implementation. These extensions are:

Template/message mapping: As indicated in the previous section, the template/message format of

³That means the template signature is exchanged between originator and proxy. The message signature is exchanged between proxy and any entity.

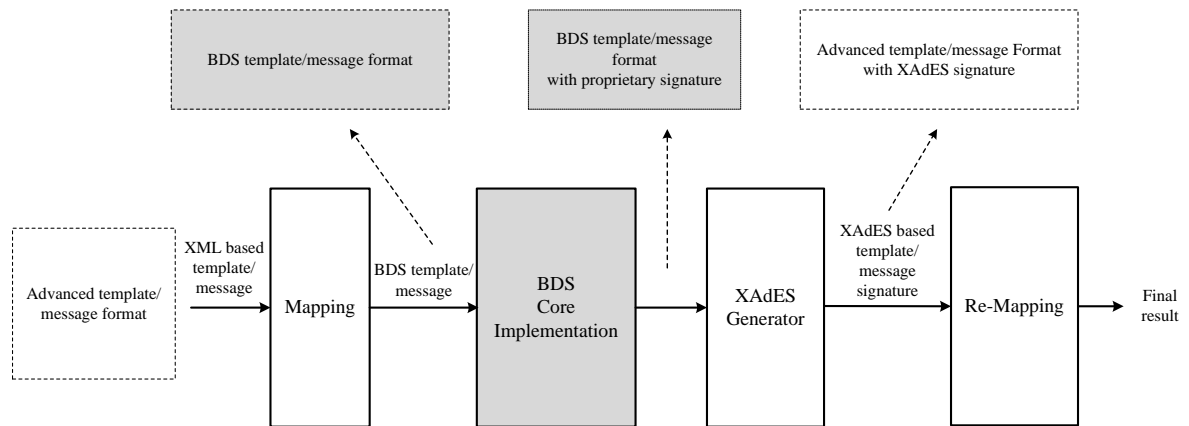


Figure 6.4: Architecture of the advanced editable signature scheme

the BDS core implementation cannot directly re-used for XML based data. To be applicable for XML data a mapping must be introduced to map from the XML data to the template/message format and vice versa. Therefore an appropriate advanced template/message format must be specified and a re-mapping mechanism must be provided.

XAdES based signatures: The BDS core implementation uses a proprietary format for the template and the message signatures, which is not compliant to any standard or specification. To be applicable for real life applications, this format must be changed to an approved standard. In the e-Government context the signature format XAdES for advanced electronic signatures is the best choice (cf. Section 5.5), as the implementation is intended to be used for XML based data.

Figure 6.4 shows the architecture of the advanced editable signature scheme including the needed extensions. Central point is the BDS core implementation. First of all an appropriate advanced template or message must be defined based upon the XML data to be signed. Via a mapping component this advanced template or message format (see Section 6.5.1 for details) is mapped to the BDS core implementation format. Then the core implementation creates the template or message signature and produces the BDS output format including the signature in a proprietary signature format. Then the XAdES generator transforms this proprietary format into the XAdES based signature format. Finally, by using the re-mapping mechanism the advanced template or message format, including the XAdES signature, is optionally transformed to the original inputted XML format.

6.5 Implementation

This section elaborates on the implementation of the architecture⁴. The focus is given on the extensions of the core implementation, namely the template and message mapping as well as the transformation of the proprietary signature format to an XAdES based signature.

⁴Credits for the implementation go also to Christian Maierhofer.

6.5.1 Template and Message Mapping

To be applicable for XML based data an advanced template and message format must be introduced. This format must support the different types (*fix*, *exch* and *blank*) of message blocks. Listing 6.1 gives an example of this advanced format. Here an element `bdssType` is specified, which value can be set to `exch` or `blank` indicating if the following message block can be exchanged or any other message block can be inserted by the proxy. For instance, within the element `as:GivenName` the `bdssType` is set to `exch` and the following `value` elements define the message blocks which can be used by the proxy. Analogous this is the same case for the `as:IdNumber` element. For the element `as:FamilyName` the type `blank` is defined. That means the `value` element specifies the initial message block (which is `Doe`), but the proxy is able to exchange this message block with any other block. Finally, the element `as:DateOfBirth` does not contain a `bdssType`, which means that this message block is fixed and cannot be modified by the proxy.

The setup of the `as:GivenName` shows also how the advanced editable scheme can be used to redact messages. As shown in Figure 6.5 the given name is defined as exchangeable, whereas the first `value` element defines the default value, which is `John`. Based upon this template the proxy is able to redact the given name by selecting the second choice, which is a `*`. Thus the proxy has redacted the given name.

Listing 6.1: Final XSL transformed result

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <as:Assertion xmlns:as="urn:as">
3   <as:Person>
4     <as:Name>
5       <as:GivenName>
6         <bdssType>exch</bdssType>
7         <value>John</value>
8         <value>*</value>
9       </as:GivenName>
10      <as:FamilyName>
11        <bdssType>blank</bdssType>
12        <value>Doe</value>
13      </as:FamilyName>
14    </as:Name>
15    <as:DateOfBirth>11.08.1984</as:DateOfBirth>
16    <as:IdNumber>
17      <bdssType>exch</bdssType>
18      <value>123456</value>
19      <value>*</value>
20    </as:IdNumber>
21  </as:Person>
22 </as:Assertion>

```

The presented mapping may be re-mapped to the original XML format at the end. Therefore XSL stylesheet transformations are used. These transformations are also covered by the signature and enable the receiver of the signature to rebuild the original format. Section 6.5.2.2 gives more details on the signature processing. In addition, Appendix A contains such XSL stylesheet transformations within the exemplary signatures and gives an example of the re-mapped original XML format representing the final result.

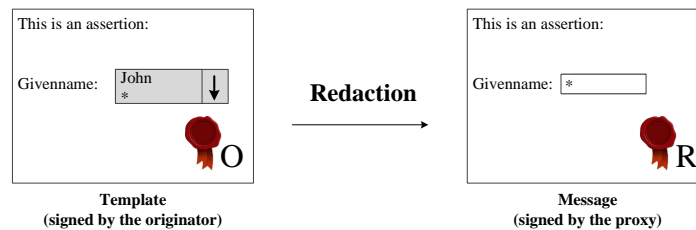


Figure 6.5: Example: Advanced editable signature used for redaction

6.5.2 XAdES based Signature

To transform the proprietary BDS signature format of the XAdES format following aspects must be considered:

- Which BDS parameters must be included in the XAdES signature format to enable the verification of the signature?
- How can these parameters mapped into the XAdES format?
- Which differences of the XAdES and BDS processing must be taken into account?

The following subsections elaborate on the parameter mapping and the signature processing.

6.5.2.1 XAdES Parameter Mapping

Table 6.2 shows the parameters of the BDS core implementation, separated for the template and message signature, and how they are mapped. Here the parameters are encoded within the appropriate certificate or the XAdES scheme. To summarize the parameter analysis, all BDS parameters fit into the XAdES structure or can be encoded in the certificates. In the following section the signature processing is described in detail.

6.5.2.2 Signature Processing

The conventional XAdES processing and the processing needed for the advanced editable signature scheme distinctly differs. Figure 6.6 illustrates the typical `SignedInfo` element of the advanced editable signature scheme basing upon BDS. Besides a canonicalization method, which defines a normalisation of the XML data, the signature method is specified as `http://www.iaik.tugraz.at/-bdss#ECDSAwithSHA256`. This specified URL indicates that the present signature is not a conventional XAdES signature, but a BDS based signature. Hence, it indicates that the processing differs. Furthermore the `SignedInfo` element contains following references:

- Reference to the `dsig:Object` element containing the template or message (depending if the signature represents a template or instance signature)
- Reference to the signed XAdES properties as defined by the XAdES specification [ETSI, 2010b].

Table 6.2: BDS parameter analysis

Template signature	Message signature	Mapping
$sysparams$	$sysparams$	BDS system parameters are encoded as an extension in the originator and proxy certificate.
pp	pp	Public BDS parameters are encoded as an extension in the originator and proxy certificate.
σ_T		Template signature value is encoded in the element <code>dsig:SignatureValue</code> of the template signature ^a .
	σ_M	Message signature value is encoded in the element <code>dsig:SignatureValue</code> of the message signature.
sk^T_{Proxy}		Private template-dependent key is encoded in the template signature value σ_T .
pk^{DSS}_{Orig}	pk^{DSS}_{Proxy}	Public verification keys of originator and proxy are encoded in the originator or proxy certificate. The mapping of the certificate depends whether it is a template or message signature. The signer certificate is mapped to the element <code>dsig:KeyInfo/dsig:X509Data-/dsig:X509Certificate^b</code> and the other certificate ^c is mapped as additional XAdES property to the element <code>xades:UnsignedSignature-Properties/xades:CertificateValues-/xades:OtherCertificate-/dsig:X509Certificate^d</code> .
T		Template is mapped to a <code>dsig:Object</code> of the template signature (in case of an enveloping signature) or represents the root element, which itself includes the template signature (in case of an enveloped signature).
	M	Message is mapped to a <code>dsig:Object</code> of the message signature (in case of an enveloping signature) or represents the root element, which itself includes the message signature (in case of an enveloped signature).

^a`dsig` denotes the namespace for XMLDSIG [Bartel et al., 2008].^bAdditionally, some signer certificate values are stored as signed XAdES properties according to the XAdES specification [ETSI, 2010b].^cFor clarity: In case of a template signature the originator certificate is the signer certificate. In contrast, for a message signature the proxy certificate is the signer certificate. See also the example signatures given in Appendix A.^d`xades` denotes the namespace for XAdES [ETSI, 2010b]

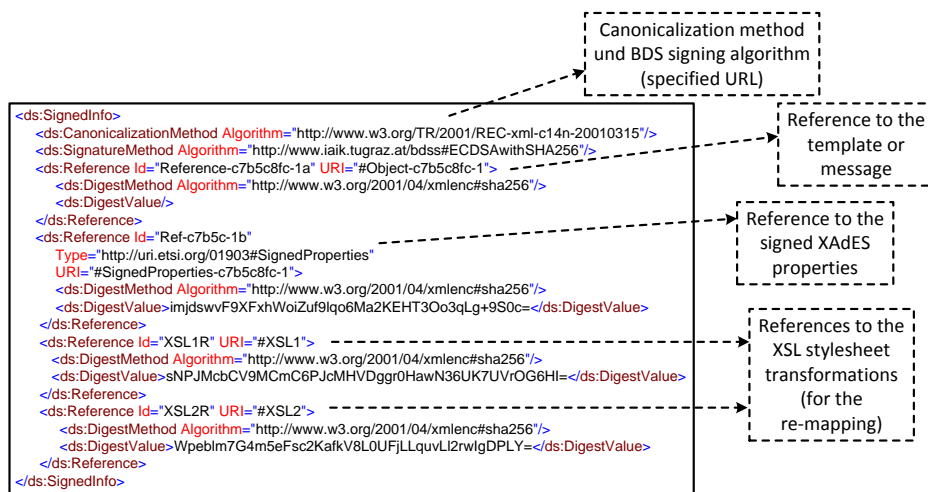


Figure 6.6: SignedInfo element of a template and message signature

- References to the XSL stylesheet transformations used for the re-mapping of the template or message format to the original XML data format.

For conventional XAdES signature this `SignedInfo` element is signed. This is not the case for the BDS based signature. Therefore the digest values of the XAdES properties and the XSL stylesheet transformations must be added to the template as fixed values to ensure that they are also signed. Hence, the authenticity and integrity of the XAdES properties and the XLS stylesheet transformations are ensured.

6.6 Evaluation and Conclusions

The presented implementation has shown that advanced editable signatures can fulfil the requirements given by the e-Government domain. That means advanced editable signatures can be deployed in e-Government services, ensuring the authenticity and integrity of sanitized data. In detail, Table 6.3 compares the defined requirements and indicates how these requirements have been fulfilled by the presented advanced editable signature scheme.

The presented advanced editable signature format is - to the best of the thesis author's knowledge - the only existing editable signature implementation, which is applicable to applications and services in the e-Government domain. In Part III of this thesis the introduced implementation is one of the cornerstones for achieving next generation applications for e-Documents.

Table 6.3: Evaluation result against the identified requirements

Requirement	Fulfilled through
Applicable to standard data formats	The presented implementation supports the standard format XML. Thus it is applicable to any XML-based data format.
Standardised signature format	The advanced editable signature scheme implementation bases upon the advanced electronic signature format XAdES. All required data and parameters can be mapped into the XAdES scheme. Nevertheless, the processing for the XAdES-based BDS signature must be changed, as the BDS processing is clearly different from the conventional XAdES processing. That means the presented solution is compliant to the XAdES scheme, but the processing itself is not fully compatible.
Complexity and integration effort	The architecture and the implementation have been designed to hide the complexity of the BDS scheme. Additionally, a further focus has been set to reduce the integration effort into existing infrastructures. This is also manifested through the support of the standard data format XML and the XAdES based signature.
Rely on existing core implementations	The presented implementation fully relies on the existing BDS core implementation. Appropriate mappings have been defined to map between the BDS template/message format and the advanced template/message format allowing to be applicable to XML based data.

Chapter 7

Electronic Document Processing



“Science may never come up with a better office communication system than the coffee break.”

[Earl Wilson]

7.1 Introduction

Electronic documents are one of the main pillars in electronic information exchange. Hence, the processing of e-Documents is essential for electronic communication. In particular, this applies for public administration procedures. As underpinned by the “Study on eGovernment and the Reduction of Administrative Burden” [European Commission, 2014c] and the final report on “The functioning and usability of the Points of Single Contact under the Services Directive” [European Commission, 2012], existing public administration procedures lack on efficiency. Here, the processing of e-Documents is a bottleneck, as incomplete or wrong data usually requires manual interaction to proceed with the procedure. This manual interaction renders a simplification of administrative procedures impossible and prevents efficiency increase.

To avoid manual interaction and thus to efficiently process e-Documents, data validation and data extraction find a way out. Thereby, data validation enables to verify if all needed data is available and must not be manually requested in a later stage of the process. For sure, data validation mechanism exists, but they allow simple validations only. Nevertheless, for complex processes and procedures more sophisticated approaches for data validation are needed. Such an approach, which enables complex data validations, is presented in this chapter.

In addition, data extraction enables the extraction of data out of available e-Documents. This can be used to automatically complete missing data. That means, using appropriate data validation and data extraction mechanisms enable an efficient processing of e-Documents by eliminating the need for manual interactions. Hence, this chapter also presents a comprehensive approach for the extraction of data out of available e-Documents.

Both presented approaches are applicable to meta data as well as document data¹. Thereby, meta data represents data about the data, which means for instance the issuer or creator of an electronic document. Meta data is usually available in a structured form [DLM Forum, 2011; DCMI, 2012]. In contrast, document data represents the real “payload” data. In case of a birth certificate, the name and date of birth of the person represent (parts of) the document data.

The remainder of this chapter is structured as follows. Section 7.2 identifies requirements for the architecture and implementation of comprehensive data validation and data extraction units. In Section 7.3 the architectures of these units are introduced. This includes the general architecture description and sequence diagrams showing the process flow. Section 7.4 presents the implementation of this architecture based upon XML structured data. Finally, Section 7.5 evaluates the approach and draws conclusions.

7.2 Requirements

This section defines general requirements for the efficient processing of e-Documents. These requirements base upon the assumption that an infrastructure for processing e-Documents already exists and that the prevention of manual interaction is required for efficient processing. The requirements must be fulfilled by the architectures and implementations. Following requirements are defined by the thesis author:

¹In general, this chapter uses the term data to denote both - meta and document data. If it must be distinguished between meta and document data it is written explicitly.

Integration effort and existing infrastructures: The integration effort must be minimized. In addition, the data validation and data extraction units must be easy integrable into existing infrastructures.

Modularity: Both, architecture and implementation must be modular and rely on well specified interfaces. This ensures that modules are easy exchangeable and that new modules are straightforward integrable if they base upon the specified interfaces.

Adaptability: The units must be easy adaptable to be applicable for different use cases. This should be ensured via an appropriate configuration unit, which allows for different adaptations and adjustments.

Automatic processing: Architecture and implementation must be designed to allow automatic processing without the need of additional manual interactions.

7.3 Architectures

7.3.1 Data Validation

Data validation in context of public administration procedures has only been slightly discussed so far. Figure 7.1 illustrates the architecture of the data validation unit. Inputs are a *profileId* for selecting a pre-configured validation profile and the *data* to be validated. Output is the respective *validation result*. The architecture consists of following core components:

Configuration: The configuration defines validation *Profiles*, whereas each profile has a unique identifier (`=profileId`). Each profile includes details about the validation process and consists of required and optional information. First, it contains information about the structure which the data has to follow. Second, it optionally contains further information about the content of the data. Such content information contains a path and a value, or a regular expression. The value or regular expression defines which content a data field must have and the path defines the location of these data field in the entire data. For illustration, suppose we have a profile for a birth certificate. The structural information may specify that the data must contain the name of the mother, father, and child. Optional content information may be that the child's name must be John Doe. Concrete validation profiles are defined in this configuration.

Structure Validation: This unit takes the validation request (including the `profileId` and the `data`) as input and is responsible for the structural validation. It consists of the *Structure Broker*, which gets the required structural information from the configuration (using the `profileId`) and triggers the *Parser/Validator*. The parser parses and validates the data against the structural information. Finally, the parser produces the structural validation result, which is sent to the *Result Generator*. In case the validation was successful and the content validation is activated in the configuration, the parsed data and the `profileId` are sent to the content validation unit.

Content Validation: The (optional) content validation takes the `profileId` and the parsed data as input. The *Content Broker* obtains the content information from the configuration and hands

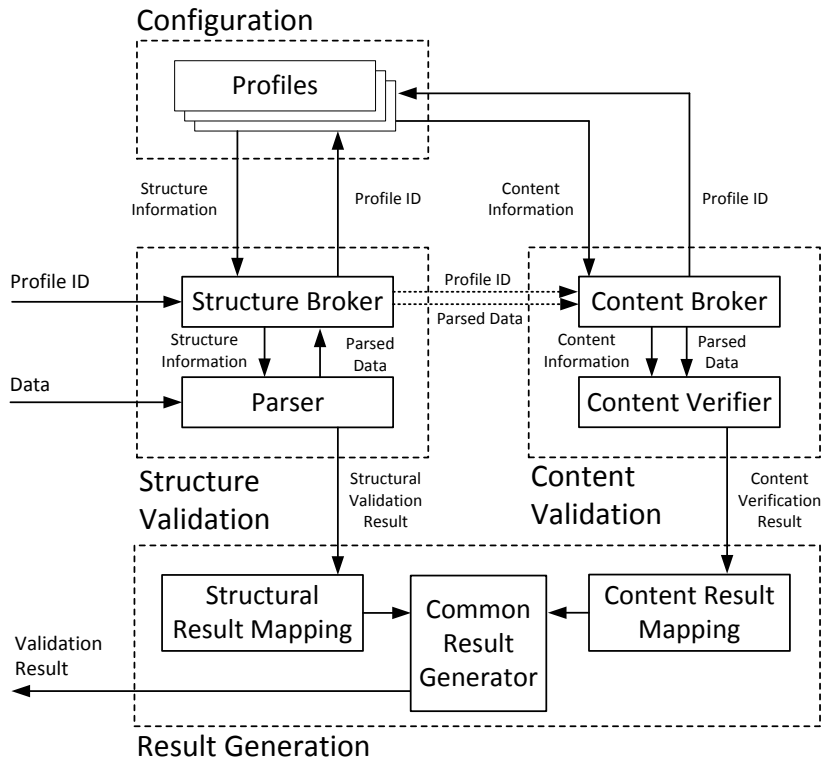


Figure 7.1: Architecture: Data validation

them over to the *Content Verifier*. For each content information, the path is evaluated to get the concrete data field. The content of these data field is verified against the given value or regular expression and produces a content verification result. Finally, all content verification results are sent to the result generator.

Result Generation: The result generator collects the structural validation results (*Structural Result Mapping*) and content verification results (*Content Result Mapping*). These parser and verifier specific results are mapped to common result codes, which are finally composed to a common validation result in the *Common Result Generator*.

Figure 7.2 shows the sequence diagram for the data validation unit. According to this diagram the process flow consists of²:

1. First of all, appropriate validation profiles must be configured by the operator or administrator of the data validation unit.
2. The *Authority* sends a data validation request to the *Structure Validation*. This request includes the *data* to be verified and a *profileId* selecting a certain validation profile.
3. The *Structure Validation* verifies the request and extracts the *profileId*.

²It is assumed that an authority operates and uses the unit.

4. Based upon the *profileId*, the information from the selected validation profile is retrieved from the *Configuration*.
5. The data is then parsed and validated against the structural information contained in the validation profile.
6. The result of this structural validation is sent to the *Result Generator*.
7. In case a content validation is configured and the data has been successfully parsed, the parsed data and the *profileId* are forwarded to the *Content Validation*.
8. The *Content Validation* retrieves the content information from the validation profile, by using the *profileId*.
9. For each configured content validation, the path information is evaluated to select the data field to be verified and verifies this data field against the given value or regular expression from the validation profile.
10. The content verification result is forwarded to the *Result Generator*.
11. The *Result Generator* generates a common validation result by mapping the individual results to common result codes.
12. Finally, the *Result Generator* returns this validation result to the requesting *Authority*.

7.3.2 Data Extraction

Figure 7.3 shows the architecture of the data extraction unit. Inputs are a `profileId` for selecting a pre-configured extraction profile and the `documents`, which contain data to be extracted. Output is the respective extracted data. The architecture consists of following core components:

Configuration: The configuration specifies information about the supported document formats (*Formats*) and extraction profiles (*Profiles*). The extraction profiles contain general and specific information about the extraction facilities of the supported document formats. In addition, it includes concrete data fields that are intended to extract. Here, simple template matching mechanisms [Huang et al., 2006] or more sophisticated approaches such as ontologies can be used (*Template/ontology information*). Extraction profiles and supported document formats are defined in the configuration. This is usually done by or on behalf of the authority, which wants to query the data validation unit.

Format Detection: This unit gets a data extraction request (including the `profileId` and the `document` to be extracted). The *Format Analysis* unit verifies the request and receives the supported document formats from the configuration. Then it triggers the *Hierarchic Assessment* unit to assess the document format. The result of the assessment (format information) is then sent to the central *Broker* including the document itself and the `profileId`.

Broker: The main objective of the central broker is to create an extraction request for the specific document. This request includes the document itself as well as the general and format-specific

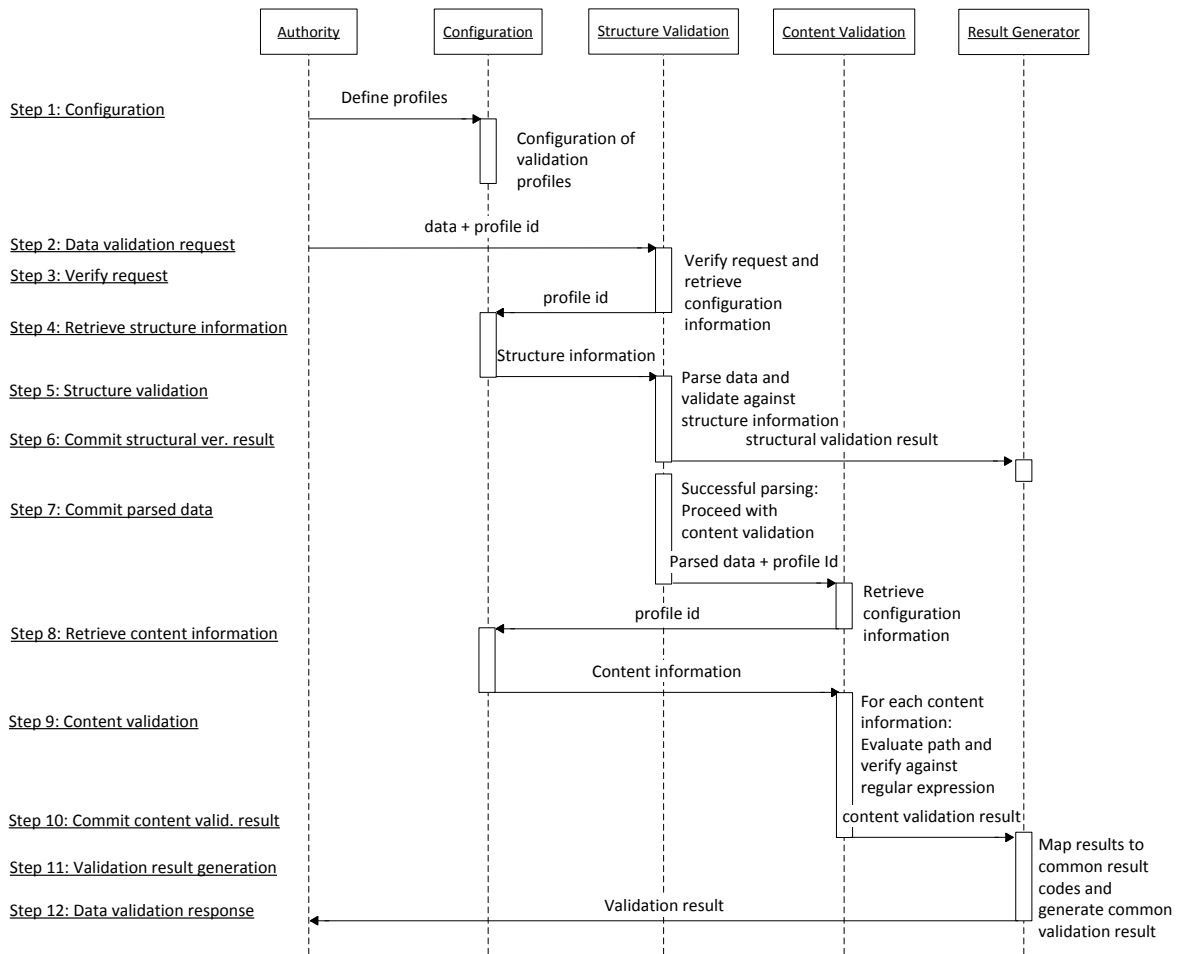


Figure 7.2: Sequence diagram: Data validation

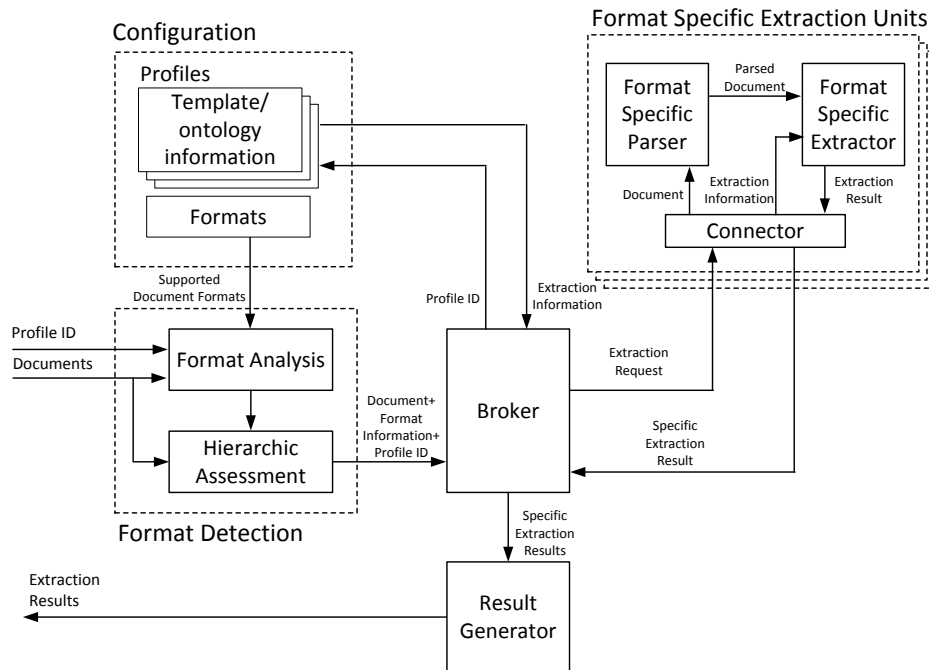


Figure 7.3: Architecture: Data extraction

extraction information. Thereby, the extraction information is gathered from the configuration. The request is sent to the *Format Specific Extraction* unit via a common *Connector* interface. After the broker has received the specific extraction result from the extraction unit, the broker forwards the result to the *Result Generator*.

Format specific Extraction Units: These units are responsible for data extraction out of the document. For each supported document format such an extraction unit exists. Each of these units must implement the common connector interface. In general, such a unit consists of a *Format Specific Parser* to parse the document and a *Format Specific Extractor*. The extractor extracts the requested data and returns a specific extraction result to the central *Broker*.

Result Generator: The result generator collects the extraction results and creates a common extraction result.

Figure 7.4 illustrates the process flow of the data extraction unit by means of a sequence diagram. According to this diagram the process flow consists of³:

1. Initially, appropriate extraction profiles and supported formats must be configured by the operator or administrator of the data extraction unit.
2. The *Authority* sends a data extraction request to the *Format Detection*. This request includes the *document*, which is used for the extraction, and a *profileId* selecting a certain extraction profile.
3. The *Format Detection* verifies the request and extracts the *profileId*.

³It is assumed that an authority operates and uses the unit.

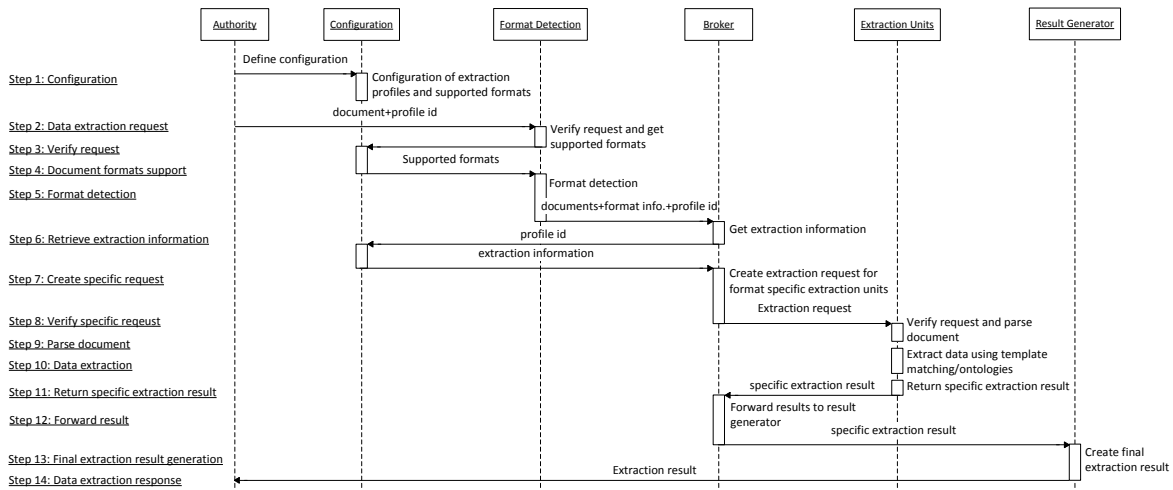


Figure 7.4: Sequence diagram: Data extraction

4. Supported formats are retrieved from the *Configuration* as MIME types.
5. The *Format Detection* analyses the format of the received document and verifies if the assessed format is supported. If it is supported, the assessed format, the document and the profileId are then forwarded to the *Broker*.
6. The *Broker* retrieves the extraction profile information from the *Configuration* by using the *profileId* from the request.
7. The *Broker* creates an extraction request for the format specific extraction and forwards the request, via the common *Connector* to the appropriate *Extraction Unit*.
8. The *Extraction Unit* verifies if the request follows the defined rules.
9. Then the document is parsed using the document-specific parser.
10. The data is extracted by using a template matching mechanism or ontology information, retrieved from the *Configuration*.
11. The format specific extraction result is returned, again via the common *Connector*, to the *Broker*.
12. The *Broker* forwards the result to the *Result Generator*.
13. The *Result Generator* generates a common extraction result by mapping the individual results to common result codes.
14. Finally, the *Result Generator* returns this extraction result to the requesting *Authority*.

7.4 Implementation

7.4.1 Overview

Based upon the presented architectures, units for the data validation and data extraction have been implemented as SOAP Web-Service. These modules represent a proof of concept Java implementation based upon XML. The XML basis has been chosen as e-Government services related to e-Documents often rely on XML based data. Hence an XML-based implementation seems to be a reasonable decision. The following subsections elaborate on the implementation of the data validation and data extraction in detail.

7.4.2 Data Validation

The implementation of the data validation bases upon the presented architecture. It uses XML schemes [W3C, 2012] for the structural validation and XPath⁴ evaluation for the content validation.

Input for the data validation is a data validation request, which is illustrated in Figure 7.5. The request consists of following elements:

SingleDataValidation: This element can occur several times, but at least once. It represents a single data validation. To support bulk requests, several single data validation elements can be included in one request.

Id (Attribute): Each `SingleDataValidation` element contains an optional `Id` attribute, which is used to indicate the appropriate result in the data validation response. If no `Id` attribute is given, the results are given according to the order of the `SingleDataValidation` elements.

Content: This element represents the content which should be validated. Thereby the content can be given directly as XML content via the element `XMLContent` or as Base64 encoded value in the element `Base64Content`.

ProfileId: Within this element the `Id` of the validation profile given in the configuration is selected.

Output of the implementation is a data validation response, which is shown in Figure 7.6 and consists of `SingleDataValidationResponse` and/or `ErrorResponse` elements:

SingleDataValidationResponse: This element contains the validation result for each `SingleDataValidation` given in the request and consists of following child elements and attributes:

Id (Attribute): Gives the `Id` value from the `SingleDataValidation` element in the request (if such an attribute has been specified).

StructureValidationResult: This element represents the validation result for the structural validation. In case of a successful structural validation, the child element `Result` occurs

⁴XPath is a W3C recommendation for addressing parts of an XML document and is specified in [Clark and DeRose, 1999].

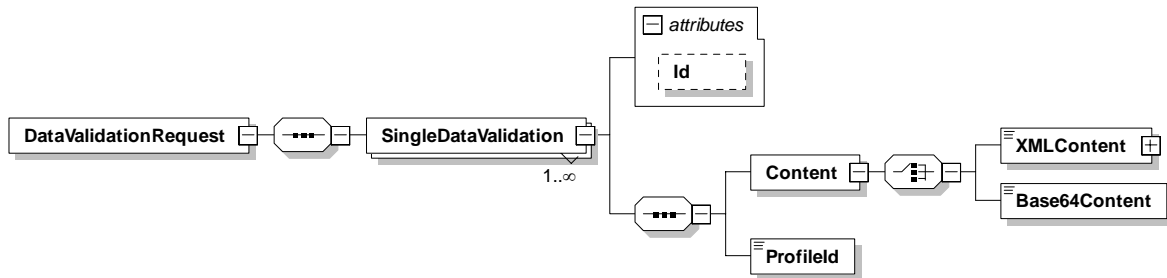


Figure 7.5: Data validation request

only once, indicating a successful validation. Otherwise, for each detected structural validation error, a `Result` element gives information about the error. Thereby the `Result` element consists of following elements:

Code: Includes a unique integer code, whereas code “0” indicates a successful validation. All other values indicate a specific validation error.

Info: Optionally includes a textual description of the code.

XPath: In case of a validation error, this element includes the XPath to the erroneous content.

ContentValidationResult: This element gives the content validation result, in case a content validation is specified in the configuration. Similar to the structural validation result, the content validation result consists of a `Result` element. As defined above, this element gives appropriate `Code`, `Info` and `XPath` element indicating a successful or erroneous content validation.

ErrorResponse: In case of any other error (such as the given `profileId` in the request, is not specified in the configuration) an error response is generated, which consists of following elements:

Code: Gives an integer value representing the error.

Info: Optionally includes a text message describing the error.

Figure 7.7 shows the implementation, which consists of following elements:

Configuration: The configuration bases upon an XML schema. This schema is shown in Figure 7.8 and consists of validation profile definitions. These profile definitions contain following elements:

profileId (Attribute): This attribute represents the `profileId`, which is used in the data validation request to select a specific validation profile.

StructureValidation: This element contains the information for the structural validation. The structural validation is performed by validating the input data against the configured XML schema, which is given in the element `SchemaLocation`. This element refers to a locally given XML schema file.

ContentValidation: This element represents the optional content validation information. For each content to be verified, a `Contents` element is given and includes following child elements.

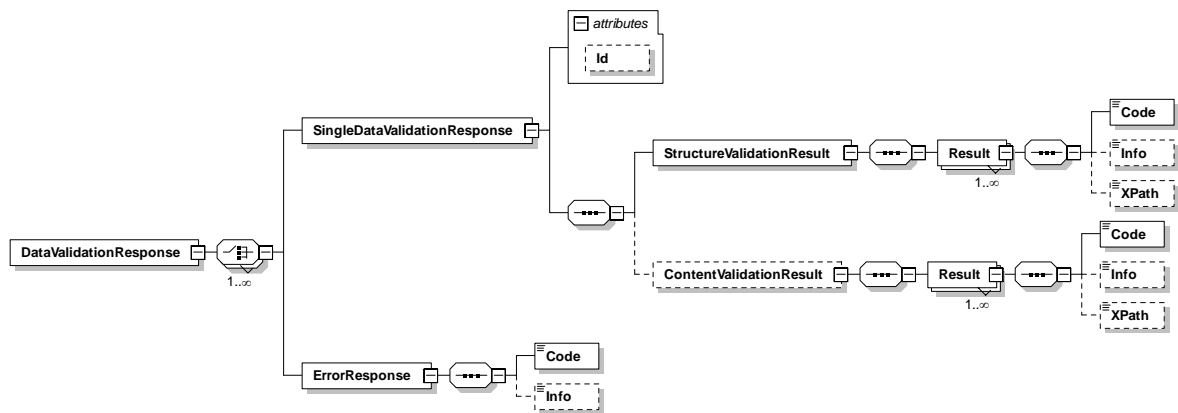


Figure 7.6: Data validation response

XPath: Gives the XPath to the specific XML content element, which is evaluated. Thereby, the value of this content element is either evaluated against the value given in the `Value` element or the regular expression given in the `RegExp` element.

Value: Gives a concrete value.

RegExp: Indicates a regular expression according to the XML schema specification [W3C, 2012].

Structure Validation: For parsing the input data against the pre-configured XML schema, a *SAX-Parser*⁵ is used. This parser is configured and started by the *Structure Broker*. After parsing the input data the parser returns the validation result to the broker. This result either indicates that the parsing was successful or gives detailed information about the parsing error. Finally, this SAX-Parser specific result is forwarded to the *SAXParser Result Mapping*.

Content Validation: For the content validation the configured XPaths are evaluated to get the addressed elements. Thereby, the *Content Broker* configures and starts the *XPath Evaluation*. The values of these addressed elements are then verified against the configured value or regular expression. The results of these verifications are then forwarded to the *XPath Evaluation Result Mapping*.

Result Generation: The *SAXParser Result Mapping* and *XPath Evaluation Result Mapping* map the SAXParser-specific/XPath-specific results to a common validation result. Finally, the *Common Result Generator* creates data validation response, which is returned.

7.4.3 Data Extraction

The presented architecture for the data extraction has been implemented on a proof of concept basis. Hence this implementation enables the extraction of data out of PDF and Microsoft Word⁶ documents,

⁵<http://www.saxproject.org/>.

⁶That means Microsoft Word documents in the Office Open XML Format [ECMA, 2012]. This format is denoted as DOCX.

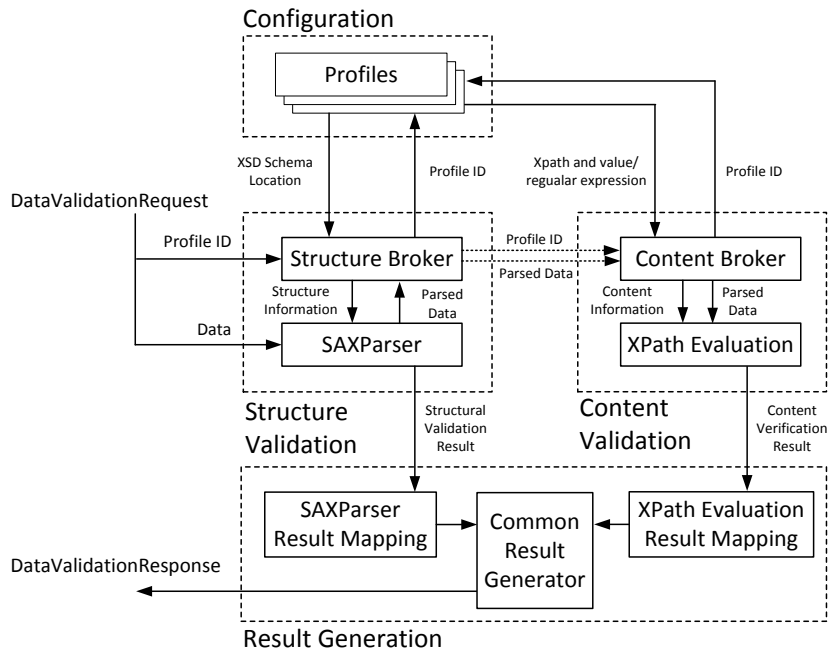


Figure 7.7: Data validation implementation

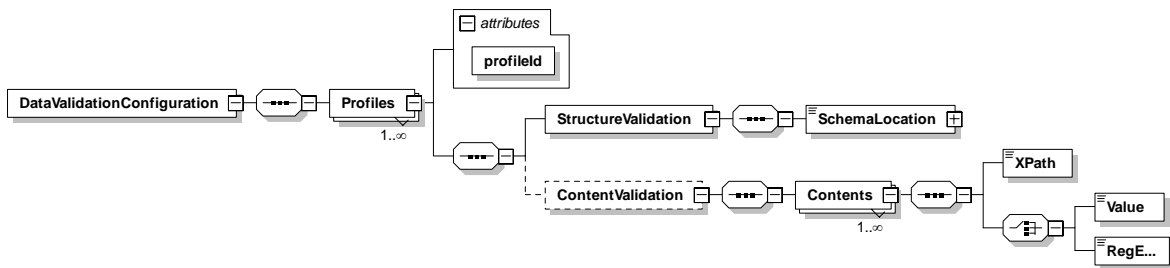


Figure 7.8: Data validation configuration

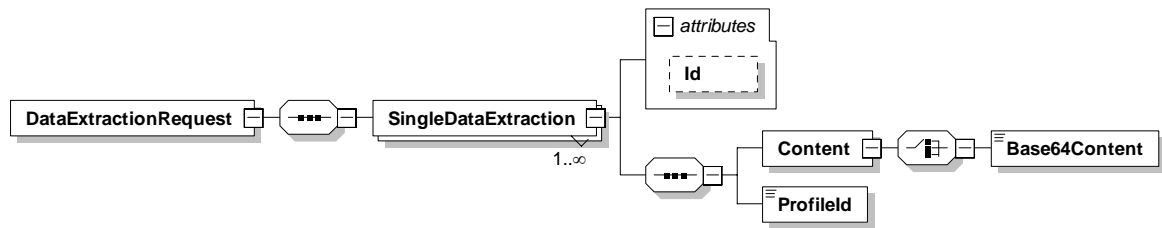


Figure 7.9: Data extraction request

as these formats are widely used, and returns the extraction result as XML content. As specified by the architecture, all required extraction information is given via pre-configured extraction information in the configuration. The extraction itself is done via a template matching mechanisms similar to the approach presented by Huang et al. [2006].

Input for the data extraction is a data extraction request, which is illustrated in Figure 7.9. The request consists of following elements:

SingleDataExtraction: This element can occur several times, but at least once. It represents a single data extraction. To support bulk request, several single data extraction elements can be included in one request.

Id (Attribute): Each `SingleDataExtraction` element contains an optional `Id` attribute, which is used to indicate the appropriate result in the data extraction response. If no `Id` attribute is given, the results are given according to the order of the `SingleDataExtraction` elements.

Content: This element represents the document which should be validated. Thereby the content is given as Base64 encoded value in the element `Base64Content`.

ProfileId: Within this element the `Id` of the extraction profile given in the configuration is selected.

Output of the implementation is a data extraction response, which is shown in Figure 7.10 and consists of `SingleDataExtractionResponse` and/or `ErrorResponse` elements:

SingleDataExtractionResponse: This element contains the extraction results for each `SingleDataExtraction` given in the request and consists of following child elements and attributes:

Id (Attribute): Gives the `Id` value from the `SingleDataExtraction` element in the request (if such an attribute has been specified).

ExtractionResults: This element represents the different extraction results for the configured extraction information in the requested extraction profile. Thereby, this element consists of following elements and attributes:

ExtractionId (Attribute): Represents the `Id` given in the configuration to indicate the appropriate extraction information. For instance, the `Id` “DocumentIssuer” may represent the extraction information, which is needed to extract the issuer of the given document.

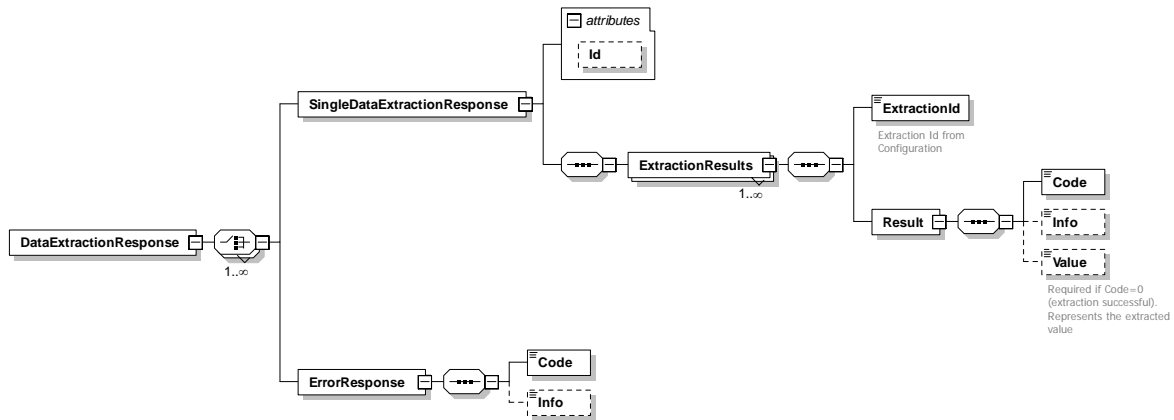


Figure 7.10: Data extraction response

Code: Includes a unique integer code, whereas code “0” indicates a successful extraction. All other values indicate a specific extraction error.

Info: Optionally includes a textual description of the code.

Value: In case of a successful extraction (Code “0”), this element contains the extracted content.

ErrorResponse: In case of any other error (such as the given profileId in the request, is not specified in the configuration) an error response is generated, which consists of following elements:

Code: Gives an integer value representing the error.

Info: Optionally includes a text message describing the error.

Figure 7.11 illustrates the implementation, which consists of following elements:

Configuration: The configuration bases upon XML for which an XML schema has been created. This configuration schema is shown in Figure 7.12 and consists of extraction profile definitions and supported document formats. It contains following elements:

Profiles: This element represents the extraction profiles and can occur once or more. For each profile any number of different extraction information can be given. Hence, this element consists of:

profileId (Attribute): This attribute represents the `profileId`, which is used in the data extraction request to select a specific extraction profile.

Extraction: This element represents a single extraction information. It consists of a required `ExtractionId` Attribute, to indicate a unique Id for the extraction, and an optional element `Template`. In general it is distinguished between meta data to be extracted and document data to be extracted. For the meta data extraction a specified extraction Id must be given and the supported extraction units must support the extraction of this specific meta data⁷. In contrast the document data extraction bases

⁷For instance, the specified Id “DocumentIssuer” means the meta data issuer from the given document. Thereby the implementation defines standard meta data Ids and their interpretation.

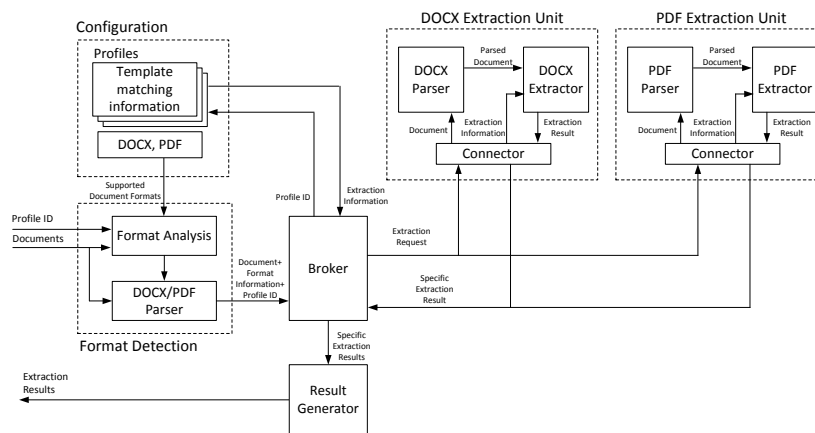


Figure 7.11: Data extraction implementation

upon a template matching mechanism. Hence, an element `Template` must be given, which consists of a choice of following elements:

Value: Gives a concrete value for the template match.

RegExp: Indicates a regular expression for the template match according to the XML schema specification [W3C, 2012].

SupportedFormats: This element lists `MimeType` child elements, indicating the supported document formats (by using the corresponding MIME types) for the data extraction.

Format Detection: This module simply takes the given document and tries to parse the document with the PDF and the DOCX parser. Depending on the output a PDF, DOCX or a not supported format has been detected.

Broker: Based upon the format detection result the broker forwards to the result generator (in case of a not supported format) or creates a format specific extraction request out of the extraction information retrieved from the configuration.

DOCX/PDF Extraction Unit: For the data extraction, the document is parsed in a first stage by the format-specific parser. Depending on the extraction information the meta data is extracted from the document and the document data is extracted by using the specified templates. For the extraction of PDF documents the Apache library PDFBox⁸ is used. For extracting content of Microsoft Word (DOCX) documents the Apache library POI⁹ is used.

Result Generation: Finally, this unit maps the specific results into a common extraction result and returns the data extraction response.

⁸PDFBox is an open source Java PDF library and available on <http://pdfbox.apache.org/>.

⁹POI is an open source Java API for Microsoft Documents and is available on <http://poi.apache.org/>.

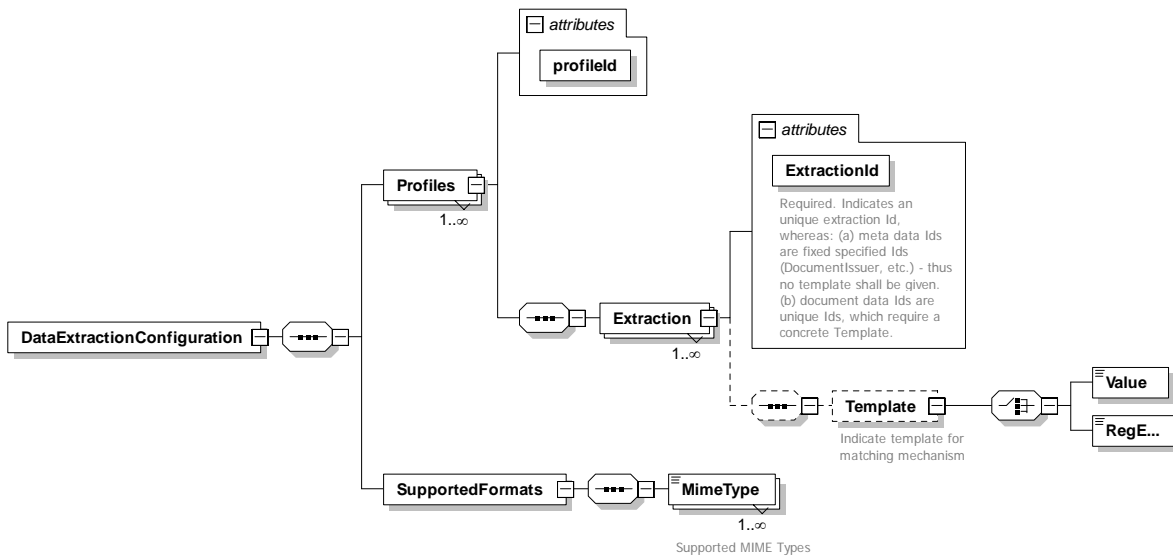


Figure 7.12: Data extraction configuration

7.5 Evaluation and Conclusions

The implementations have shown that the architectures are implementable and applicable. Both, the data validation and data extraction have been tested with various documents and data - mainly via available national and international test documents and data. The implemented XML-based data validation has been evaluated to be fit for purpose. The main reasons for that are that the mainly needed functionalities (a) parsing against a given XML schema (for the structural validation) and (b) evaluating XPath expressions (for the content validation) are already available by various mature tools and libraries. Hence, the implementation has shown that data validation for XML-based data is easy achievable. Of course, data validation of non-XML based or even unstructured data seems to be more challenging and requires additional effort.

The implemented data extraction enables the extraction of data out of PDF and Microsoft Word documents. This has been tested and evaluated with various available national and international test documents. Thereby, the success factor of the extraction results strongly depends on how “structured” the content of the documents is. For instance, a date of birth certificate follows a more general structure (by indicating the name, date of birth, parent’s names, etc.) compared to a medical or third party insurance certificate, which structure usually relies on the respective issuing party. That means, governmental based documents usually have a greater extraction success factor (due to their inherent structure) compared to documents issued by private parties. Another aspect concerning the data extraction is that currently many electronic documents are still available on paper or in the best case as scanned copy only [European Commission, 2014c]. Both cases prevent an automatic data extraction. Thus, Section 10 introduces an additional manual interaction backup, which does not eliminated the need for a manual interaction, but tries to the reduce the effort for this interaction.

To conclude, the proposed data validation and data extraction approaches have shown that they are able to fulfil the identified requirements from Section 7.2 with the indicated limitations. Table 7.1

compares these requirements and shows how they have been fulfilled by the presented architectures and implementations.

Table 7.1: Evaluation result against the identified requirements

Requirement	Fulfilled through
Integration effort and existing infrastructures	Both, the data validation and data extraction unit, are available as SOAP Web-Service. Hence, they are easy integrable into existing infrastructures, which us service-oriented architecture. Also other existing infrastructures can profit from this Web-Service approach as a Web-Service client can be easily set up on different platforms using different programming languages.
Modularity	Both units follow a modular approach, whereas each module targets on a specific functionality. These modules are combined with clear interfaces and common connectors to allow an easy exchange of modules as well as an easy adding of further modules.
Adaptability	Both units are adaptable to different use cases as (most of) the data validation and extraction is controlled by the configuration and does not require changes in the implementation. In particular, the data validation is controlled via structural and content information given in the validation profiles. Similarly, the extraction functionalities are controlled via extraction profiles indicating the template matching information to extract the needed data. Only the support of additional document formats requires to implement additional modules, which follow the given interface.
Automatic processing	After setting up the configuration, the data validation and data extraction are ready to run. The XML-based data validation has shown its fully applicability, the applicability of the implemented data extraction depends on the given documents as indicated above.

Finally, the findings of this chapter will be incorporated into the next-generation public administration procedures, which are discussed in detail in Chapter 10.

Part III

Next-Generation Applications for Electronic Documents

Chapter 8

Next-Generation Applications for Open Government Data



“Small opportunities are often the beginnings of great enterprises.”

[Demosthenes]

8.1 Introduction

As highlighted in Chapter 3 the need for next-generation applications is an essential implication of the Digital Agenda for Europa [European Commission, 2010b]. Here, one of the key topics is *open data*. In the last years open data has emerged and has significantly influenced the IT sector. The general idea behind open data is that data should be freely available for everyone to be used and republished. 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 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, via the platform [data.gv.at](http://www.data.gv.at)¹ various applications that make use of OGD provided by the government are already available for citizens ranging from various mobile smartphone apps to complex applications for desktop computers.

The importance of OGD has been highlighted by the European Commission too. In the year 2013 the European Commission has published an amendment of the Directive on the re-use of public sector information, short PSI Directive [European Commission, 2013b]. This amendment makes a great leap forward to open data as the first revision of this Directive has been published in 2003 [European Commission, 2003], far away the open data initiative has emerged, and thus had a very conventional sight on the provision of public sector data. 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 [Open Government Working Group, 2007]. 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 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 and open government data based solutions. The use of forged data might for instance lead to resource claims. In such cases, the provider of data should be able to proof that originally provided data has been altered. Additionally, assuring the data integrity and authenticity for open government data has clear benefits for the consumer (=OGD recipient) too. The recipient is able to trust the validity and correctness of the provided data. Current solutions based on public sector data and open government data usually do not support this feature. Hence, the remainder of this chapter presents next-generation applications enabling *Trusted Open Government Data (Trusted OGD)*.

Trusted OGD bases upon electronic signatures, which are the means of choice to achieve data integrity and authenticity. Thereby the data to be published is signed by the OGD provider. The signed data is then published and can be consumed by the OGD recipient. By positively validating the signature over the data, the recipient is can rely on unmodified and authentic data. The remainder of this chapter presents this Trusted OGD approach in detail and is structured as follows. First of all, the basic requirements of OGD and the PSI Directive are compared in Section 8.2. In addition this section identifies additional security requirements for achieving authenticity and integrity for open government data. Section 8.3 introduces the concept for Trusted OGD, which fulfils these additional security

¹<http://www.data.gv.at/>

requirements. Based upon this concept, architectures for the server- and client-side implementations are presented in Section 8.4. This includes a definition of architectural requirements and an analysis of popular OGD data formats. In particular, this analysis evaluates the signing capabilities of these OGD data formats, as applying an electronic signature is an essential step in the Trusted OGD approach. Then, Section 8.5 elaborates on the server- and client-side implementations. Finally, Section 8.6 contains an evaluation of the architecture and implementation against the identified requirements and draws conclusions.

8.2 Common Requirements

8.2.1 OGD vs. PSI Directive

OGD and the PSI Directive [European Commission, 2013b] 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 [Open Government Working Group, 2007] 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 documents² held by public sector bodies of the EU Member States.

In general, provision of government data in the 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:

Completeness: The 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.” [European Commission, 2013b]

Primary source: The 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.” [Open Government Working Group, 2007]

The PSI Directive does not explicitly provide any guidance for a primary source of data. It can be assumed that data provided by a public sector body fulfil this requirement.

²The 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)” [European Commission, 2013b]. For the following considerations only electronically available data, which come under the Directive, are considered.

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 is stated that

“public sector bodies should make the documents available in a time-frame that allows their full economical potential to be exploited.” [European Commission, 2013b]

Accessibility: Public data must be made available barrier-free for a 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. Article 3 of the PSI Directive states that

“Where possible, documents shall be made available through electronic means.” [European Commission, 2013b]

Machine processable: 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 in any pre-existing format or language, and, where possible and appropriate, in open and machine-readable format together with their metadata.” [European Commission, 2013b]

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. PSI data is not necessarily free of charge, as the Directive states:

“Where charges are made for the re-use of documents, those charges shall be limited to the marginal costs incurred for their reproduction, provision and dissemination.” [European Commission, 2013b]

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. The PSI Directive states that

“Both the format and the metadata should, in so far as possible, comply with formal open standards.” [European Commission, 2013b]

License: Open Government Data does not contain any requirements concerning licenses³. While in contrast the re-use of PSI imposes no strict guidelines. The Directive (Article 8) proposes that:

³Open Government Working Group [2007] state that OGD must be license-free, but they define license in the proper sense. They define license-free as data that “is not subject to any copyright, patent, trademark or trade secret regulation.” [Open Government Working Group, 2007].

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 processable	Data should be in automatically processable formats.	Data should be provided in a machine-readable format (where possible and appropriate)
Data Access	An anonymous access for anybody at any time should be provided.	PSI data is not necessarily free of charge, but charges are limited and must be published before.
Non-Proprietary	Data formats should base upon open standards to ensure the long-term readability.	Data should be, as far as possible, comply with open standards.
License	No information about license in the proper sense.	No strict guidelines defined. Data may be provided under designated and non-discriminatory conditions.

Figure 8.1: Overview OGD and PSI Directive requirements

“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.” [European Commission, 2013b]

and

“In some cases the re-use of documents will take place without a licence being agreed.” [European Commission, 2013b]

Figure 8.1 summarizes the different requirements of public sector data and compares their impact on OGD and PSI.

8.2.2 Security Requirements

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 a compliance with appropriate security requirements is strongly recommended. Hence, the previously defined requirements are considered to be incomplete and hence insufficient (for use cases where the authenticity and integrity of the published data is important). Therefore, the general principles are extended by the following two requirements in order to appropriately consider security aspects:

Authenticity and integrity: The authenticity and integrity of 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).

Authenticity and integrity for redacted government data: As defined in the previous section personal data must not be published as OGD or be provided as PSI 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.

8.3 Trusted Open Government Data - Concept

The objective of the presented concept is to ensure authenticity and integrity for OGD including the possibility to anonymize or redact (parts of) these data. To fulfil these requirements, the proposed concept integrates conventional and editable signature schemes. In the following details of this concept are discussed and it is presented how providers as well as recipients of OGD benefit from this approach. By using electronic signatures for public sector data, two general use cases can be distinguished. Depending on the use case, the presented concept makes use of different schemes for electronic signatures. In the following, the two general use cases covered by the concept are presented in detail.

8.3.1 Use Case 1 - Ensuring Authenticity and Integrity for OGD

In this scenario it is shown how a provider of OGD is able to provide authentic and integrity-protected data. Providing such secured data has following advantages:

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

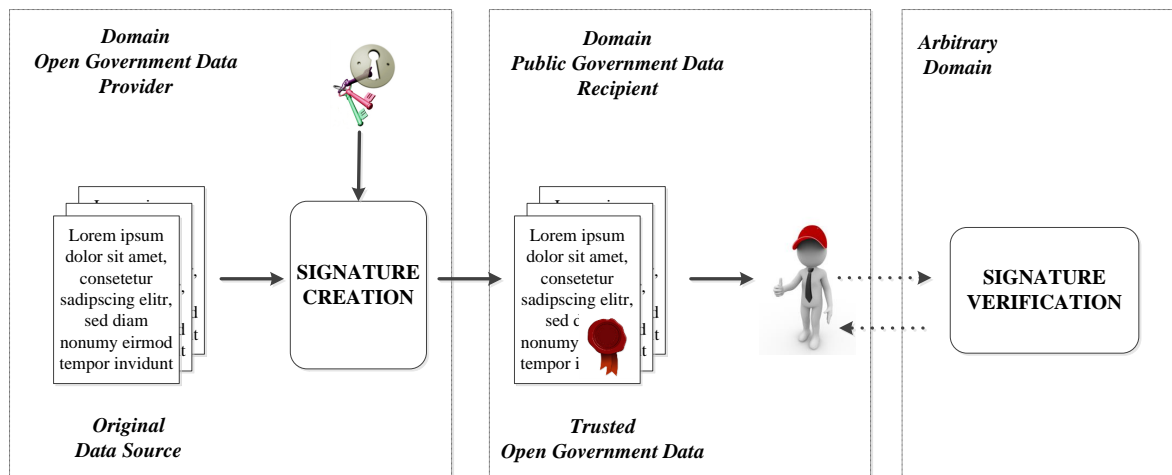


Figure 8.2: Use case 1 - Ensuring authenticity and integrity for OGD

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.

Authenticity of the data provider: The recipient of the OGD is able to reliably determine the identity of the data provider. This leverages the trust in the reliability and trustworthiness of the provided data.

The means of choice for implementing authentic and integrity-protected public sector data are conventional signature schemes. Figure 8.2 illustrates the basic approach. The original data source is located in the domain of the OGD provider. These data is signed with the private signature key of the provider. Depending on the data format, different signature formats are possible (cf. Section 8.4.2). Afterwards, the signed data is provided or published through appropriate communication channels as Trusted OGD.

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.

8.3.2 Use Case 2 - Authenticity and Integrity for Redacted OGD

This use case covers all applications, in which the original data set contains private or any other worthy of protection 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. An exemplary application may be signed minutes of a local council meeting, whereas the public has interest on these minutes. To publish them as open government data the need of redacting parts of these minutes due to official secret may needed. Hence, there is a need to anonymize or to redact the original private or other 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 OGD, the anonymized or redacted data must be signed

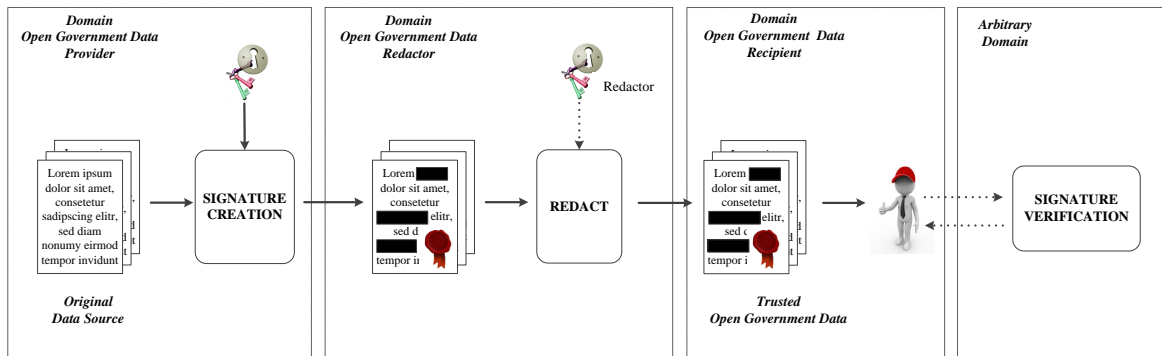


Figure 8.3: Use case 1 - Ensuring authenticity and integrity for OGD

again. For some applications, this is however not practical nor a 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 editable signatures produce a relief (cf. Chapter 5 and Chapter 6).

Figure 8.3 shows the basic principle of Trusted OGD based on editable signatures. The provider of the OGD uses its private key to create an editable signature. The redactor anonymizes or redacts the data and updates the editable signature. For this purpose, the redactor must use her private key. After this, the editable 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.

8.4 Trusted Open Government Data - Architectures

The architecture for a Trusted OGD comprises a server- and a client-side. The server-side is responsible for creating the trusted data, i.e. to apply the electronic signature, and is deployed at the OGD provider. On the client-side (i.e. the OGD recipient) these data are consumed and includes the verification of the electronic signature and visualization of the data. Following subsections give details on the server- and client-side architecture. Before that, requirements for Trusted OGD architectures are identified and signing capabilities of popular OGD data formats are analysed.

8.4.1 Requirements

On the way to a real implementation of Trusted OGD, appropriate requirements must be identified. The identified requirements base upon the assumption that an infrastructure for publishing OGD already exists. Hence, following requirements have been identified and must be fulfilled by the general architecture to achieve an adoption of the Trusted OGD approach:

Modularity and adaptability: The IT infrastructure is subject to constant changes. This must be taken into account by the architecture. Hence, the architecture must follow a modular and adaptable approach. That means main functionalities must be encapsulated in different modules.

Additional modules, especially modules for creating new signature formats and treating new data formats, must be easily connectable.

Minimal effort integration: In most cases an infrastructure for publishing OGD exist. For investment protection and to break down the barriers for adoption and take up, the architecture must be integrable into existing systems on minimal effort basis. In addition, the configuration must be easy and usable.

Interoperability: Interoperable services are one of the major objectives of the EU Digital Agenda for Europe especially across borders. This of particular relevance for signature and data formats. Hence, the architecture must be designed according to the requirements of the European Interoperability Framework (EIF, cf. Section 3.2.3.2) and must take into account current decisions in the area of signature and data formats.

Automatic processing: The architecture must be designed to allow automatic processing. Manual interaction should be eliminated.

8.4.2 Data Formats

Electronically signing the data to be published is the corner element of the Trusted OGD approach. To apply an electronic signature on OGD, it is necessary to analyse OGD data formats on their signing capabilities. In particular, this means which signature format can be applied to the respective OGD data format. Hence, this section analysis different OGD data formats.

In the OGD domain a lot of different data formats are deployed. These formats range from simple text-based formats via formats for structured data to container formats. Some formats are envisaged for a general deployment (e.g. CSV) whereas others have designated fields of application (e.g. geographical data formats such as GML). Structured data fulfil one the main principles of open data, which is automatic processing. In contrast, unstructured data are usually used to visualize data. For both types container formats are used partly. Container formats consist of several files, which are encapsulated into one container. Thus, the container is self-contained. Following subsections elaborate on the most common and most popular⁴ data formats used in open government data applications⁵. In particular, the signing capabilities of these formats are examined.

8.4.2.1 CSV

CSV means comma-separated values and is a text-based format specified in Shafranovich [2005]. It enables to store data in a simple structured way. Thus, a data set is separated by a special separator (usually a line break) and data fields are separated by another delimiter (comma, semicolon, etc.). Information about the data is not specified (e.g. no standards about the time or date format are given). CSV is a text-based format. Thus it can be signed using CAdES or XAdES. Depending on the use case the one or other format may be better applicable.

⁴Basis for this decision was a data format analysis of different open data platforms such as data.gv.at, offenedaten.de and data.gov.uk.

⁵Interfaces, which are sometimes (mistakenly) named as data format, are not treated. This applies for: WMS (Web Map Service), WFS (Web Feature Service), RSS (Really Simple Syndication) and WMTS (Web Map Tile Service).

8.4.2.2 XML

XML is a widespread description language and means eXtensible Markup Language. It is a W3C recommendation [W3C, 2008], but has evolved to a de-facto standard. XML is text-based, but enables to define complex structures of various data by using so called tags. To specify a concrete structure of a certain data set an XML or DTD schema can be defined. In case a data set follows the rules of this schema, the data are called valid. For XML-based data XAdES signatures are well suited. Basically, these data can also be signed using a CAdES signature, but is not reasonable and is not done in practice usually.

8.4.2.3 KML

KML is the abbreviation for Keyhole Markup Language and is an XML-based description language for geodata. It bases on Google Earth and Google Maps. KML is very powerful and is not only able to store geographic information (such as points, lines, images, etc.). Additionally, it enables to store information about the current traffic at certain regions for instance. The easiest use case is to set markers at certain locations. More complex use cases are for example detailed maps and models to mark weather and earthquake activities. KML is specified in [Open Geospatial Consortium Inc., 2008] and is enhanced by Google. KML bases on XML. Hence, XAdES signatures are well suited to sign KML data. For the compressed format KMZ also CAdES signature are applicable.

8.4.2.4 GML

As well as KML, GML is a description language for geodata GML means Geography Markup Language and is specified in [Open Geospatial Consortium Inc., 2012]. It bases on XML and enables a standardised encoding of geographic information. That means GML is used to describe objects (streets, buildings, bridge, etc.) and their properties. For instance, GML can be used to identify all registration districts in a city, by defining a closed polygon. As GML bases on XML, XAdES signatures are again well suited. Basically, a GML file can be signed using a CAdES signature. Nevertheless, this approach is practically not useful.

8.4.2.5 SHP

SHP means Shapefile and has been specified by the Environmental Systems Research Institute [Environmental Systems Research Institute, 1998]. It is a vector data format for geodata. SHP has evolved as a de-facto standard in the GIS⁶ area and represents rivers, lakes or bridges as vector functions (points, lines, polygons, etc.). A SHP file consists of several files. Thus, SHP is a typical representative of a container format. As a consequence, CAdES signatures are applicable to SHP data. Using a Base64 encoding, GML can be signed using a XAdES signature, but this is practical not reasonable.

8.4.2.6 SVG

SVG is a W3C recommendation [Dahlström et al., 2011] and means Scalable Vector Graphics. It is used to describe two-dimensional vector graphics. SVG bases upon XML, but can also be stored

⁶Geographic information system.

in a compressed form. It consists of three fundamental element types: vector graphics (built via graphic primitives), embedded raster graphics and text elements. XAdES signatures are well suited for the XML-based SVG format. In addition, CAdES signatures are also applicable for the compressed format.

8.4.2.7 PDF

PDF is a well-established standard and means Portable Document Format. Since version 1.7 PDF is an ISO standard. The main objective of PDF is the representation and visualization of electronic contents. Hence, PDF aggravates automatic processing usually. However, PDF is very popular and widespread, even in the OGD community. Obviously, PAdES signatures are well suited for signing PDF data. Basically, PDF data can be Base64 encoded and signed with XAdES, but this approach is neither practical nor reasonable.

8.4.2.8 ZIP

The ZIP data format has two main objectives. On the one hand it is used as container format to create a single, self-contained file and on the other hand it enables a compression of the included data. ZIP emerged to a widespread standard. Again, ZIP data can be signed using a Base64 encoding and XAdES signatures. However, CAdES signatures are far more applicable to ZIP data.

8.4.2.9 Summary Signing Capabilities

Table 8.1 summarizes the results of the OGD format examination. For each data format a reasonable and practical signature format can be applied. Depending on the data format, XAdES and CAdES signature are usually the best choice. However, PAdES signatures are applicable to PDF data only.

Table 8.1: Summary signing capabilities

Format	CAdES	XAdES	PAdES
CSV	✓	✓	×
XML	×	✓	×
KML/KMZ	✓	✓	×
GML	×	✓	×
SHP	✓	×	×
SVG	✓	✓	×
PDF	×	×	✓
ZIP	✓	×	×

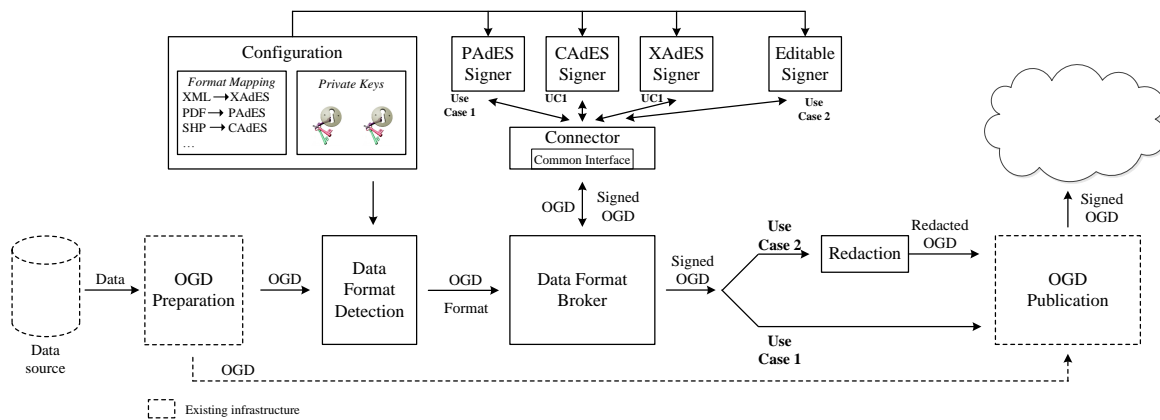


Figure 8.4: Server-side architecture for Trusted OGD

8.4.3 Server-side Architecture

Figure 8.4 shows the server-side architecture and how it fits into an existing infrastructure⁷. The existing infrastructure has been generalized and consists of following components:

Data source: Represents the original data source, which is the basis for creating OGD.

OGD preparation: This unit generates OGD out of the original data source.

OGD publication: The generated OGD is sent to this unit, which takes over the publication.

8.4.3.1 Use Case 1 - Ensuring Authenticity and Integrity for OGD

This use case refers to the use case presented in Section 8.3.1. The server-side architecture is shown in Figure 8.4 and it is shown how the existing infrastructure is extended to create Trusted OGD. Thereby, architecture has a modular design. In the following the extended modules, their interaction and the workflow are described:

Configuration: This module contains the configuration, which has two major components. First, a format mapping maps an OGD format to a signature format as elaborated in Section 8.4.2. Second, private signature keys are specified for creating the signatures. These keys can be software keys, hardware keys (via hardware security modules) or any other key source.

Data Format Detection: Based upon the input format and the characteristics of the OGD formats, this unit detects the OGD format and forwards this information to the data format broker.

Data Format Broker: Based upon the configured format mapping, this unit selects the appropriate signature format for the given OGD format and forwards the data to the connector.

⁷Existing infrastructure means all elements of an infrastructure, which already exists for publishing open government data.

Connector: The connector provides a common interface between the data format broker and the different signers.

Signer: Currently three different signer modules are defined - according to the findings in Section 8.4.2. Each signer takes the private key from the configuration and signs the given data according to the signature format specification. Via the connector also additional signers can be added, which implement the common interface.

8.4.3.2 Use Case 2 - Authenticity and Integrity for Redacted OGD

This use case refers to the use case presented in Section 8.3.2. Compared to the first use case, the architecture for use case 2 differs in the signature creation and the additional redaction of data. In detail, following components differ for use case 2:

Editable Signer: Instead of applying a conventional signature, an editable signature is created over the given data (for details see Chapter 5 and Chapter 6).

Redaction: In this additional process step, the signed data is redacted. This means, all data that is not suitable to be published (e.g. private and personal data) is redacted. The redacted data is then forwarded to the OGD publication component.

8.4.4 Client-side Architecture

Figure 8.5 illustrates the client-side architecture, whereas the architecture is the same for both use cases. It represents a general architecture, which can be implemented as smartphone app, web application or any other application type. The involved modules and the workflow are:

Data retrieval: This module retrieves the signed OGD from a publication source and forwards the data to the signature verification broker.

Signature verification broker: The broker is a module, which starts the signature verification process and evaluates the verification result. Depending on the concrete implementation an internal - self implemented - signature verification service can be used or the signature verification is done via an external verification service. After the verification, the broker forwards to the app engine.

Internal and external signature verification: These modules take over the signature verification. Either an internal verification is implemented for verifying the needed signature format or an external service is used. External services are beneficial as they usually support different signature formats and have a Web-Service interface, which is quite easier to implement as a complex signature verification service. Especially apps with minor resources only (e.g. smartphone app) can profit.

App Engine: This engine takes over the presentation and visualization of the given data in case the signature has been successfully verified. Otherwise an appropriate message should be displayed.

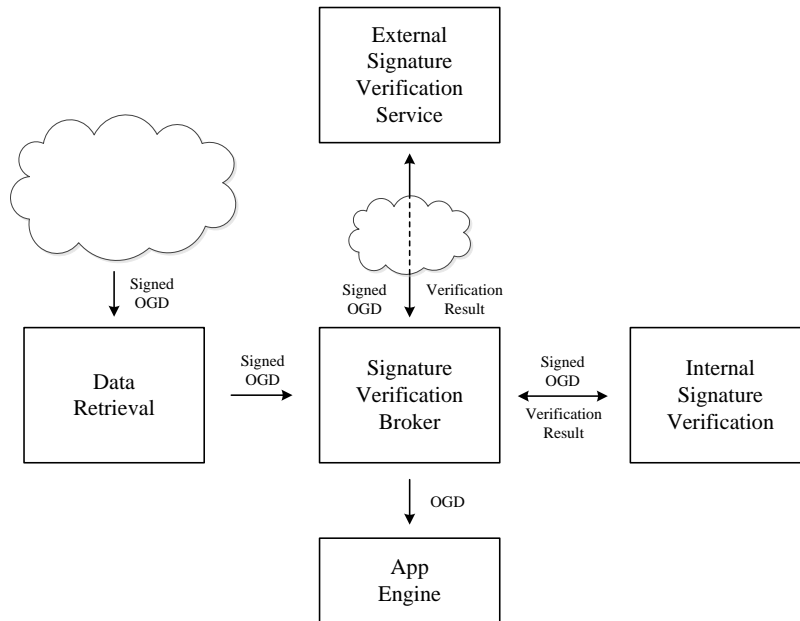


Figure 8.5: Client-side architecture for Trusted OGD

8.5 Trusted Open Government Data - Implementations

To evaluate the proposed architectures, a server-side Web-Service for publishing OGD in a secure and reliable manner has been implemented. In addition, a client-side smartphone application makes use of these data and includes usable functionalities for verifying the validity of the received data.

8.5.1 Server-side Implementation

8.5.1.1 Use Case 1 - Ensuring Authenticity and Integrity for OGD

Figure 8.6 shows the (proof of concept) server-side implementation. As data source the Austrian OGD portal data.gv.at has been chosen. This portal currently hosts more than 1200 different data sets. These data sets are used as input for the data format detection. The data format detection bases upon the format detection engine presented in [Zefferer et al., 2011] and assess the incoming data. Based upon a hierarchic assessment the data format of the incoming data is evaluated.

Based upon the result of this assessment and the configuration the appropriate signature format, as shown in Section 8.4.2, is chosen. Via the data format broker, the data to be signed is forwarded to the signing facility. This signing facility is implemented via the Austrian open source module MOA-SS⁸ for creating advanced electronic signatures. Currently CADES and XAdES signature are supported by this module. Hence, OGD formats mapping to these signature formats are supported by the implementation actually. The signed data is then returned to the data format broker. Finally, the signed OGD is forwarded to the publishing unit, where it is published on an Apache Web-server.

⁸<https://joinup.ec.europa.eu/software/moa-idspss/description>

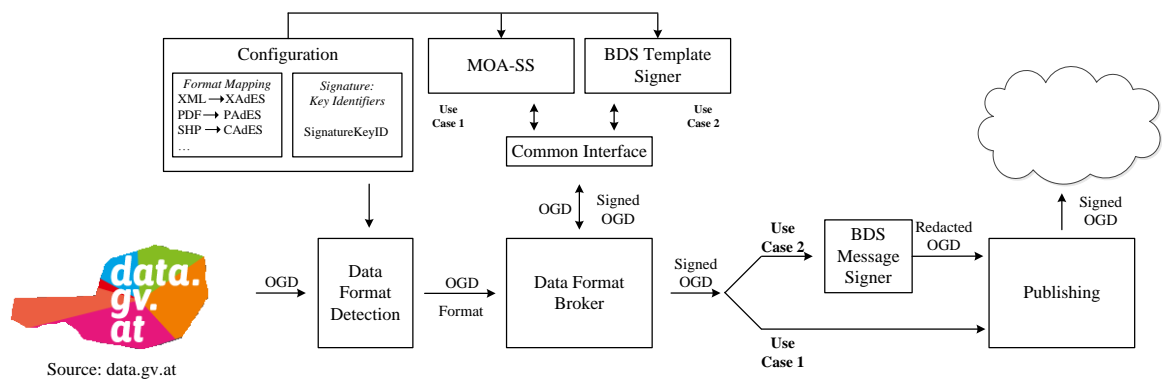


Figure 8.6: Server-side implementation for Trusted OGD

8.5.1.2 Use Case 2 - Authenticity and Integrity for Redacted OGD

For the second use case the signing functionality has been exchanged and the redaction step added. For the signing functionality the editable signature scheme blank digital signatures is used. Based upon the implementation of this scheme (cf. Chapter 6), a BDS template is created, whereas the defined proxy is able to redact, i.e. to replace certain text parts with a “*” character. That means the template creator defines the corresponding text parts as exchangeable and defines a “*” as only possible replacement character. Due to this BDS template, a BDS instance can be created, which represents the redacted data. Thereby the proxy selects the “*” out of the exchangeable text parts. Finally, these redacted data is sent to the publishing component.

8.5.2 Client-side Implementation

On the client-side an Android app has been implemented. Thereby, the differences between both use cases are marginal for the client side. This app bases on the OGD set “NEXTBIKE NÖ Fahrradverleihsystem”⁹, which represents a bike rental service and containing data about the location of bike rental stations. These data has been signed via the server-side implementation. For the first use case these data has been signed using an XAdES signature and for use case 2 an appropriate BDS template has been created¹⁰.

Due to limited sources on Android platforms, external signature verification services are used. For use case 1 the service of the Austrian Regulatory Authority for Broadcasting and Telecommunications RTR¹¹ and its Web-service are used (cf. [Lenz et al., 2013b,a]). For the second use case a self-deployed service, which is able to verify BDS instantiations, is used. As the main purpose of the app is to show the handling of signed OGD, the remaining functionalities are limited. Hence, in case the signature verification is positive, the app lists all rental station and enables to open the location in Google maps (for use case 1). Figure 8.7 illustrates two screenshots of the app, which show the positive verification result and a list of the rental stations.

⁹<http://www.data.gv.at/datensatz/?id=96d176fb-dfd4-49de-91fc-b4997ab353ba>

¹⁰For illustration the exact location of the bike rental stations have been redacted.

¹¹<https://pruefung.signatur.rtr.at/>

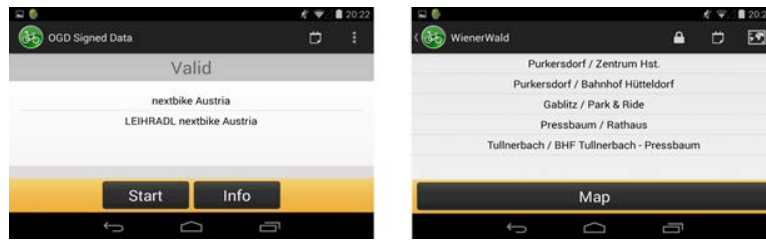


Figure 8.7: Screenshot Android app

8.6 Evaluation and Conclusions

The implementations have shown that a Trusted OGD is realizable - even for redacted data. The first use case seems to be a reasonable and easy viable approach for all data, which benefit from an authentic and trustworthiness publication. The server-side implementation of the first use case been successfully applied to most of the data published on the platform data.gv.at. Thereby, the appropriate signature format has been chosen by the implementation and thus has shown that for the examined data format (CSV, XML, KML/KMZ, GML, SHP, SVG, PDF and ZIP) an appropriate signature (CAAdES, XAdES, PAdES) can be applied. Issues on some - mainly XML-based - data have been spotted, which prevented from successful signing using the XAdES signature formats. These issues mainly concern irregularities between the provided data and the XML schema representing the XML format¹² and so are not an issue of the XAdES signature itself. Furthermore, the client-side implementation has shown that the signature verification of the received data is easy implementable by using an external signature verification service, supporting the respective signature format. Thereby, the used external signature verification service is available as SOAP Web-Service and is integrated by using an appropriate Web-Service client. As a Web-Service client can be easily implemented on different client platforms using different programming languages, the implementation of the client-side Trust OGD is easy achievable.

The server- and client-side implementation of the second use case rely on the XML-based implementation of the editable signature given in Section 6. Hence, the second use case implementation is applicable to XML-based data only. The server-side implementation has been applied to all available XML data formats published on the platform data.gv.at¹³ to test their functionality. Thus, it has shown that the approach for a Trusted OGD is basically also applicable to the redacted data. Nevertheless, an extension to support also other OGD data formats may be favorable. However, this requires - beside the changes of the Trusted OGD implementation - also the availability of another editable signature supporting these formats.

The presented implementations have shown that the identified requirements from Section 8.4.1 can be fulfilled with the indicated limitations. Table 8.2 compares these requirements and shows how they have been fulfilled by the presented architecture and implementation.

To conclude, Trusted OGD appears reasonable for all use cases where authenticity and integrity of the published data is needed. Here, the presented solution provides means to achieve this. Especially the solution for the first use case enables an easy and minimal effort means to achieve Trusted OGD.

¹²In concrete terms: the provide data could not be successfully parsed against the given XML schema, which is required in case an XML scheme is given [ETSI, 2010b].

¹³For instance the implementation has been tested on the data set of the "NEXTBIKE NÖ Fahrradverleihsystem" or the "Lawinenlagebericht Tirol"

Table 8.2: Evaluation result against the identified requirements

Requirement	Fulfilled through
Modularity and adaptability	The server- and client-side architecture follow a fully modular approach. Clear interfaces and common connectors allow for an easy exchange of certain modules as well as an easy extension of functionalities (e.g. adding of additional signer modules in the server-side architecture).
Minimal effort integration	Minimal effort integration is mainly an issue for the server-side architecture and implementation. Here, the minimal effort is achieved by decoupling the existing infrastructure from the added functionalities. Furthermore these added functionalities are equipped with generalized interfaces which allow integration into different existing systems. In addition, the server-side configuration is balanced between usability and configurable features. For sure, a concrete application in a real-life publication infrastructure has not been done yet and may raise unregarded issues.
Interoperability	Interoperability issues mainly concern OGD data formats and the signature formats used, whereas OGD data formats are beyond the sphere of influence of the presented solution. Nevertheless, it is recommended to use OGD data formats which follow the OGD principles. On the signature formats aspect, the presented solution makes use of the reference signature formats defined in the European Commission Decision 2014/148/EC European Commission [2014a]. Thus interoperability is achieved. Additionally, the presented design is fully compliant to the requirements of the European Interoperability Framework.
Automatic processing	On the server-side Trusted OGD can be easily and automatically achieved by the implementations, independently from the used OGD data format. On the client-side the automatic-processing capability depends on the used OGD data format. For sure, the signature verification can be done automatically, but the further processing of the data in the app engine strongly depends on the processing capability of the OGD data format. Again, it is recommended to use OGD data formats which follow the OGD principles.

Chapter 9

Next-Generation Applications for Identity Management



“Bureaucracy is a giant mechanism operated by pygmies.”

[Honore de Balzac]

9.1 Introduction

Similar to open data (cf. Chapter 8), the need for next generation applications in the domain of identity management are implied from the Digital Agenda for Europa (cf. Chapter 3). Still from the beginning identity management has been a cornerstone for e-Government processes and administrative procedures, as the identity of the citizen must be assured uniquely. Hence, most countries have set up an appropriate identity management assuring a unique identity, so called *eIDs*. Examples for such national eID solutions are the Austrian Citizen Card [Leitold and Posch, 2004] or the German eID card [Margraf, 2011]. Details on this individual eID solutions can be found in [Modinis, 2006; European Commission, 2009c; Siddhartha, 2008]. For these national eID solutions the identity attributes (identifying the citizen) are usually approved by an official *registration authority* and thus providing qualified and authentic identity data. If a citizen wants to log in at an online application, deployed at a *service provider*, the citizen must perform the identification and authentication processes at a (trusted) *identity provider*. This identity provider is then able to transfer the identity data in a structured form, according to the supported identity protocols such as SAML [OASIS Security Services TC, 2005] or OpenID [OpenId Cons., 2007], to the service provider. Based on the received identity data, the service provider is able to grant or deny access to the online application. Here, usually the entire set of identity data (e.g. name, date of birth, unique identifier, etc.) are revealed to the service provider.

For privacy reasons, many users do not want to reveal their entire identity data set and in some cases this is even not required. This need for privacy increases in case the identity provider is centrally deployed. This has in turn other clear benefits for - especially - small service providers, which do not have the appropriate or enough resources to deploy an identity provider in their own domain. Therefore, the present chapter proposes a novel *identity management model for national eID solutions*, which enables a *selective disclosure of identity data* to the identity and service provider. Here, the user itself is in full control, which identity data is sent to the identity and service provider in succession. Hence, the presented model brings together:

- Increased privacy for a central deployed identity provider
- Selective disclosure of identity attributes
- Applicable to national eID solutions and thus relying on qualified and authentic identity data
- User-centered as the user is in full control which data is sent to the identity and service provider

To demonstrate the applicability of this model, the Austrian eID system is adopted to this model. The remainder of this chapter is structured as follows. In Section 9.2 the status quo in the area of selective disclosure techniques is presented. This includes needed definitions and a brief description of selective disclosure approaches. The following Section 9.3 identifies concrete requirements to implement a user-centered eID model for selective disclosure. These requirements must be fulfilled by an identity management model enabling user-centered selective disclosure. Section 9.4 presents the proposed model based upon editable signatures, whereas the process flow is separated into a registration phase and a identification and authentication phase. Thereby, the registration process has to be conducted only once, whereas the latter must be performed for each access to a protected resource. Next Section 9.5 gives an overview about the existing Austrian eID system and explains the main pillars of this system. In Section 9.6 the model is applied to the Austrian eID system by using blank

digital signatures (cf. Chapters 5 and 6). This also includes a detailed description of the process flows. Finally, Section 9.7 evaluates the model and implementation against the defined requirements and draws conclusions.

9.2 Selective Disclosure

Selective disclosure is a well discussed topic in the research area [Lei and Feng, 2011; Tews and Jacobs, 2009; Sultana et al., 2013; Vullers and Alpar, 2013], especially to protect users' privacy. Before giving an informal definition of selective disclosure, the term identity data is defined as follows:

Definition 9 *“Identity data” are data, which intention is to (uniquely) identify a person.*

Based on that, selective disclosure is defined as follows:

Definition 10 *“Selective disclosure” (in the area of identity managements) means, that out of a set of available identity data, only a subset of these data is revealed to another entity.*

In the following different approaches for achieving privacy (and partly supporting) selective disclosure are briefly discussed. Thereby, the approaches are divided into approaches by using a trusted entity and using a semi-trusted entity. These entities are defined as follows:

Definition 11 *A “trusted entity” is an entity, which acts fully correct. That means all tasks and procedures are performed correctly as well as the processed and stored data is secured so that no data is revealed unintentionally to another entity.*

Definition 12 *A “semi-trusted entity” acts honest but curious, which means that the entity performs all tasks correctly, but stored or processed data may be leaked.*

Trusted entity approaches are using a trusted identity provider. Examples for this approach are the new German eID card [Margraf, 2011] and STORK (cf. Section 3.2.5.6). Whereas the first approach is tailored to the German eID system, latter provides interoperability models to achieve a cross-border identity management. Thereby, one interoperability model of STORK introduces a central gateway for each country. This gateway connects - on the one side - to the national eID infrastructure and - on the other side - transfers the identity data to the other national gateways across borders. As the user identifies and authenticates on her national gateway, the national gateway usually gets her full identity (depending on the national infrastructure). In the next step the national gateway is able to release only a subset of these identity data to other gateways. Nevertheless, the full identity data is revealed to the national gateway in this approach.

Typical suspects of semi-trusted behavior are cloud providers. For cloud providers privacy and data protection issues arise as they are able to inspect data if the data is not encrypted. When moving the identity provider into a public cloud, which is called Identity as a Service (IDaaS), these issues gain importance. Nuñez et al. [2012] propose an approach to bypass these issues based upon an extension of the OpenID protocol. Thereby, the OpenID provider is deployed in the public cloud, whereas the identity data are encrypted by using a re-encryption scheme. Hence, the OpenID provider is not able

to inspect the identity data. However, this approach does not support selective disclosure. Other typical approaches for semi-trusted scenarios including selective disclosure support are anonymous credential systems such as Idemix [Camenisch and Lysyanskaya, 2001] and U-Prove [Brands, 2000]. Unfortunately these systems lack on practicability as the underlying cryptographic technologies are quite complex and computational expensive to be implemented on smart cards [Bichsel et al., 2009] or other client based software. Additionally, anonymous credential systems require significant changes in the existing infrastructure, if such systems are adopted.

9.3 Requirements

In this section requirements for a user-centered identity management model enabling selective disclosure are defined. Pre-requisite is that the identity provider is (centrally) deployed in a trusted environment. Following concrete requirements must be fulfilled by the model:

Qualified and authentic identity data: The identity data belonging to a certain user must be verified and asserted by a trusted authority. Therefore the user must register at this authority once, which guarantees high quality of the identity data. The issued identity data must be verifiable by any party, even if only a subset of the identity data is revealed.

Integration effort and complexity: The model and its implementation must be integrable into existing infrastructures without significant changes.

User-centered: The user must have full control about her identity data and she is solely in control which identity attributes are disclosed to other entities.

Selective disclosure: The user must be able to reveal only a subset of her identity data to the identity and service provider.

Privacy: The service provider must not be able to learn anything about the undisclosed identity attributes.

Open standards: Wherever possible open standards and specifications must be used.

9.4 The Model

Figure 9.1 shows the proposed model for a user-centered identity management enabling selective disclosure. This model consists of following entities:

User: The user wants to access a resource or service, which is deployed at a certain service provider. To access this resource or service the user has to reveal selected attributes of her identity data. These identity data are issued by the registration authority.

Registration authority (RA): The registration authority acts as trusted third party and is responsible for issuing qualified and authentic identity data to the user. Hence, the user is required to register at the registration authority initially.

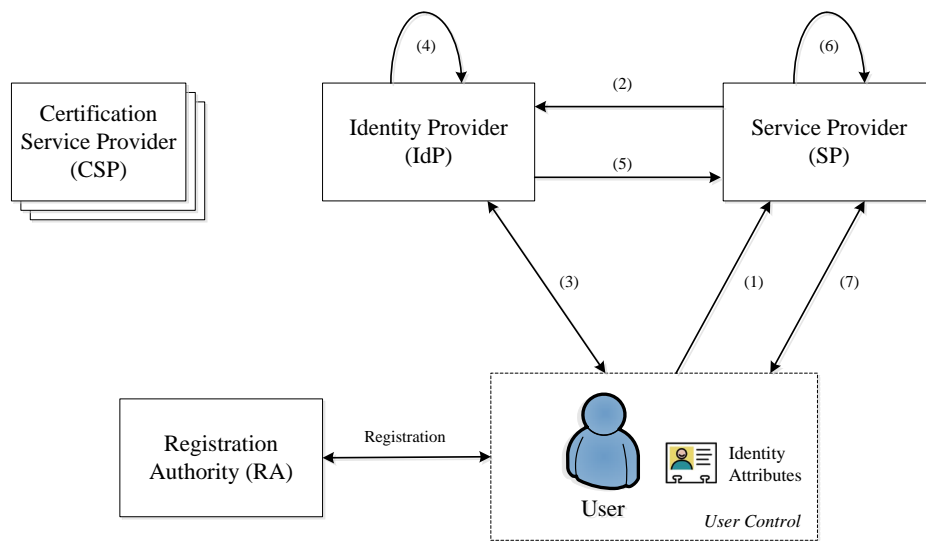


Figure 9.1: A user-centric and selective disclosure enabling model for eIDs

Service provider (SP): The service provider has resources or services deployed, the user wants to access. For gaining access to this resources a qualified identification and authentication is required.

Identity provider (IdP): The identity provider is deployed at the service provider or any other trusted (central) environment and provides functionalities for a qualified identification and authentication of users. By using this identity provider, the service provider gets the asserted identity and authentication data via standardised protocols such as SAML or OpenID.

Certification Service Providers (CSPs): Accredited or supervised certification service providers¹ certify the needed key material and issue appropriate digital certificates.

In the following, details on the registration process and the identification and authentication process when applying this model are given. The registration process has to be conducted only once, whereas the latter must be performed for each access to a protected resource.

9.4.1 Registration Process

In the proposed model, the registration authority is responsible for the issuance of the qualified and authentic identity data. Here, the data provisioning is carried out by an appropriate registration process. This process takes place between the user and the registration authority. As the details of this registration process are dependent from the eID approach, details of this process are out of scope of this model². Nevertheless, the identity data are digitally signed by the trusted registration authority by using an editable signature scheme. Applying such a signature has two functions. First, the signature of the registration authority assures the data is authentic and the integrity of the data is guaranteed. Second, based upon the functionalities of editable signature scheme, the authenticity and integrity can

¹Accredited or supervised according to the EU Signature Directive [The Council of the European Union, 2000].

²An exemplary registration process may need the personal appearance of the user at the office of the registration authority.

be assured even if only a subset of the identity data is revealed. Fore sure, the authentication itself depends on the needed authentication level of the service provider. Furthermore, the identity data will be stored in such a way that the user has full control over it. For storing the identity data two general possibilities exists:

Client approach: The identity data can be stored on the client-side, that means for instance on a secure token, such as the smart card of the user.

Server approach: The identity data can also be stored on the server-side via a trusted attribute provider. For this approach the attribute provider must deploy appropriate means to assure that only the respective user has control about her identity data (e.g. by using mobile TAN technologies).

Independent from the chosen approach, the identity data are anyhow issued by the trusted registration authority in a qualified and authentic manner.

9.4.2 Identification and Authentication Process

The identification and authentication process is shown in Figure 9.1. For better illustration, it is assumed that the identity provider is centrally deployed and not in the domain of the service provider. Basically, the identity provider has two main functionalities:

- Verification of the received identity data. That means verification of the applied signature of the registration authority.
- Structuring and transferring the identity data to the service provider by using an open and standardised interface, such as SAML or OpenID.

In the following the identification and authentication process is described stepwise:

1. The user wants to access a resource at the service provider. This resource is protected and requires authentication.
2. The authentication is carried out by the identity provider. Hence, the user is forwarded to the identity provider.
3. The user redacts all identity attribute, she does not want to be disclosed to the service provider (and even the identity provider). The redacted data is then sent to the identity provider. Thereby, the needs of the authentication level of the service provider and the redacted identity data must be aligned. That means, if a service provider requires at least the name of the user to enable the login, the user must reveals her name, otherwise the service provider denies the access.
4. The identity provider verifies the editable signature over the redacted identity data. Additionally, the identity provider structures the identity data according to an open and standardised identity protocol. To ensure the authenticity and integrity of this structure, the identity provider signs it.
5. The identity provider transfers the signed structure to the service provider according to the used identity protocol.

6. The service provider verifies the signature of the identity provider.
7. Based on the received identity attributes the service provider either grants or denies access.

In the following, Section 9.6 applies the presented model to the Austrian eID system to show its applicability. Previously, the Austrian eID system is described briefly in the following Section 9.5.

9.5 The Austrian eID System

Unique citizen identification and secure authentication in Austria is based on the Austrian citizen card³ [Leitold H., 2002], the official eID in Austria. Unique identification is based on a unique number, the so-called sourcePIN, which is wrapped in a special XML data structure, the so-called Identity Link (IDL), and stored on the citizen card. Thereby, the sourcePIN is derived from the Central Register of Residences (CRR) number. As shown in Listing 9.1⁴ the Identity Link includes the citizen's sourcePIN, first name, last name, date of birth, and a public key. This public key belongs to the corresponding private key, which is stored on the citizen card and is used to create qualified electronic signatures. To ensure authenticity and integrity of the Identity Link, it is digitally signed by the trusted SourcePIN Register Authority (SRA). Subsequently the Identity Link is denoted as $IDL = ((A_1, a_1), \dots, (A_m, a_m))$ being a sequence of identity attribute name A_i and value a_i pairs.

Listing 9.1: Identity Link

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:dsig="http://www.w3
   .org/2000/09/xmldsig#" xmlns:ecdsa="http://www.w3.org/2001/04/xmldsig-more#" xmlns:pr="
   http://reference.e-government.gv.at/namespace/persondata/20020228#" xmlns:si="http://www.
   w3.org/2001/XMLSchema-instance" AssertionID="szo.bmi.gv.at-AssertionID1376408486094522"
   IssueInstant="2013-08-13T17:41:26+01:00" Issuer="http://portal.bmi.gv.at/ref/szo/issuer"
   MajorVersion="1" MinorVersion="0">
3 <saml:AttributeStatement>
4 <saml:Subject>
5 <saml:SubjectConfirmation>
6 <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</
   saml:ConfirmationMethod>
7 <saml:SubjectConfirmationData>
8 <pr:Person si:type="pr:PhysicalPersonType">
9 <pr:Identification>
10 <pr:Value>Gq03dPrgcHsx3G01ZDH6SQ==</pr:Value>
11 <pr:Type>urn:publicid:gv.at:baseid</pr:Type>
12 </pr:Identification>
13 <pr:Name>
14 <pr:GivenName>Max</pr:GivenName>
15 <pr:FamilyName primary="undefined">Mustermann</pr:FamilyName>
16 </pr:Name>
17 <pr:DateOfBirth>1965-03-24</pr:DateOfBirth>
18 </pr:Person>
19 </saml:SubjectConfirmationData>
20 </saml:SubjectConfirmation>
21 </saml:Subject>
22 <saml:Attribute AttributeName="CitizenPublicKey" AttributeNamespace="urn:publicid:gv.
   at:namespaces:identitylink:1.2">
23 <saml:AttributeValue>

```

³Currently, the Austrian citizen card is implemented as client-side approach using smart cards and as server-side approach involving the citizen's mobile phone.

⁴The full example identity link is given in Appendix B.

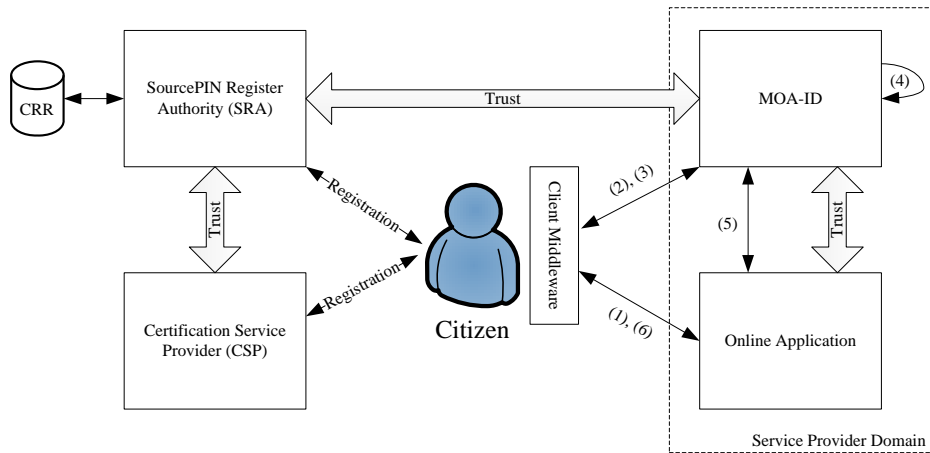


Figure 9.2: The Austrian eID system

```

24 <ecdsa:ECDSAKeyValue>
25 <ecdsa:DomainParameters>
26 <ecdsa:NamedCurve URN="urn:oid:1.2.840.10045.3.1.7"/>
27 </ecdsa:DomainParameters>
28 <ecdsa:PublicKey>
29 <ecdsa:X Value="
    21548781512348463624665512931810002987761006733116410810317945950596" si:type
    ="ecdsa:PrimeFieldElemType"/>
30 <ecdsa:Y Value="
    32548183375132839567164482875181817864226552443687973267521109533636" si:type
    ="ecdsa:PrimeFieldElemType"/>
31 </ecdsa:PublicKey>
32 </ecdsa:ECDSAKeyValue>
33 </saml:AttributeValue>
34 </saml:Attribute>
35 <saml:Attribute AttributeName="CitizenPublicKey" AttributeNamespace="urn:publicid:gv.
    at:namespaces:identitylink:1.2">
36 ...
37 </saml:Attribute>
38 </saml:AttributeStatement>
39 <dsig:Signature>...</dsig:Signature>
40 </saml:Assertion>

```

To preserve citizens' privacy, it is prevented by law (according to the Austrian e-Government Act [Republic of Austria, 2004]) to directly use the sourcePIN for identification at online applications. Therefore, the Austrian eID system implements a sector-specific identification model using domain-specific pseudonyms. These so called sector-specific PINs (ssPINs) are uniquely derived from the sourcePIN (by using a SHA1 hash function) and ensure citizen unlinkability across multiple sectors.

In the following the registration and authentication process in the Austrian eID System is briefly described. In addition, Figure 9.2 illustrates the involved entities and their interactions.

9.5.1 Registration Process

In order to activate an Austrian citizen card, citizens must prove their identity to the SRA. Finally, the SRA creates the sourcePIN and the Identity Link. These data and the qualified certificate, issued by

an accredited certification service provider (CSP), is stored on the citizen card. More precisely, for this process the SRA relies on cryptographic key material provided by the accredited CSP⁵.

9.5.2 Identification and Authentication Process

For facilitating the identification and authentication process using the Austrian citizen card at online applications, service providers usually rely on the open source module MOA-ID⁶. On the one side, this module manages the identification and authentication process with the citizen and, on the other side, provides citizen's identity data in a structured format to the online application. According to Figure 9.2, the identification and authentication process involves the following steps:

1. The citizen wants to access a protected resource, which requires citizen card authentication. The online application starts the authentication process and triggers MOA-ID.
2. First, MOA-ID reads the Identity Link from the citizen card through the client middleware and verifies its signature. This corresponds to the identification process.
3. Second, MOA-ID requests the citizen to create a qualified electronic signature (cf. Section 2.4) for authentication. The qualified electronic signature is verified by MOA-ID involving appropriate certificate revocation mechanisms (CRL, OCSP) provided by the CSP.
4. MOA-ID gets the sourcePIN according to the domain the service provider is assigned to and thus creates a sector-specific PIN (ssPIN).
5. MOA-ID assembles a special data structure which includes the ssPIN and additional personal data of the citizen such as first name, last name, and date of birth. The assembled data structure, called *assertion*, follows the specification of SAML [OASIS Security Services TC, 2005] and is transmitted to the online application.
6. Based on the data received, the online application is able to provide the protected resource to the citizen.

9.6 Application to the Austrian eID System

The current deployment approach of the Austrian eID system foresees a local deployment of MOA-ID within each service provider's domain. This local deployment is often an obstacle for many, especially small, service provider, which do not have the appropriate or enough resource for that. Hence, a central deployment of MOA-ID is advantageous for such service provider. Even if a central MOA-ID is deployed in a trusted environment, users may have privacy objections. Hence, this section applies the user-centered eID model enabling selective disclosure to the Austrian eID system. In the presented realisation, the implementation⁷ makes use of the advanced editable signature scheme (cf. Section 6) basing upon blank digital signatures [Hanser and Slamanig, 2013]. In principle, also other signature schemes, enabling the redaction of data by designated redactors may be used. Nevertheless,

⁵In Austria, the only existing accredited CSP is the company A-Trust (<http://www.a-trust.at/>).

⁶<https://joinup.ec.europa.eu/software/moa-idspss>

⁷Credits for the concrete implementation go to Christian Maierhofer.

The developed signature scheme is used to sign the Identity Link, which enables the citizen to redact specific identity attributes out of the Identity Link⁸. By using this technology, the realisation is easy integrable into the existing infrastructure and requires minimal changes on the service provider's side.

In the following, the registration as well as the identification and authentication process is discussed in detail.

9.6.1 Registration Process

From the user's and service provider's perspective the registration process does not change compared to the existing system. The required changes for applying our new model affect the creation of the data to be stored on the citizen card only and has to be done by the SRA as it is the case in the current approach. Figure 9.3 shows the sequence diagram of the registration process. In the following, it is assumed that the digital signature scheme (DSS) secret keys⁹ are generated by the respective entities and certified by a CSP. The entire registration consists of the subsequent steps:

1. Key generation of the respective entities.
2. The CSP generates the public parameter pp^{BDS} for the BDS scheme and publishes it.
3. The CSP certifies the public signature verification key for the SRA ($pk_{\text{SRA}}^{\text{DSS}}$), the citizen ($pk_{\text{Citizen}}^{\text{DSS}}$), and MOA-ID ($pk_{\text{MOA-ID}}^{\text{DSS}}$). Thus, appropriate certificates are generated. Additionally, these certificates contain the public parameter pp^{BDS} as a certificate extension (cf. Section 6.5.2.1).
4. The SRA creates a modified Identity Link IDL^* based upon the original IDL attributes. This IDL^* includes the attributes of IDL (e.g., name, date of birth, etc.) and *all* domain-specific pseudonyms (ssPINs) for all public sectors. Hence, $\text{IDL}^* = ((A_1, a_1), \dots, (A_k, a_k))$ is a sequence containing the original attributes and additionally the added ssPINs, whereas A represent the attribute name and a the attribute value.
5. Based on this IDL^* , a template T is generated, which defines the value pairs (A_i, a_i) to be redactable by the citizen. This template is then signed by the SRA using BDS.S . Appendix B gives an example of such an IDL^* . The template signing process outputs the template signature σ_T and the template dependent private key $sk_{\text{Citizen}}^{\text{BDS}, T}$ for the citizen (i.e., only the citizen holding this key is able to redact data and to create signed message instances).
6. In the last registration step, the following data are stored on the corresponding citizen card: the Template T representing the IDL^* , the Template signature σ_T and the Template dependent private key $sk_{\text{Citizen}}^{\text{BDS}, T}$.

⁸The SRA represents the originator and the citizen the proxy in terms of blank digital signatures.

⁹This keys represent the keys of a conventional digital signature scheme.

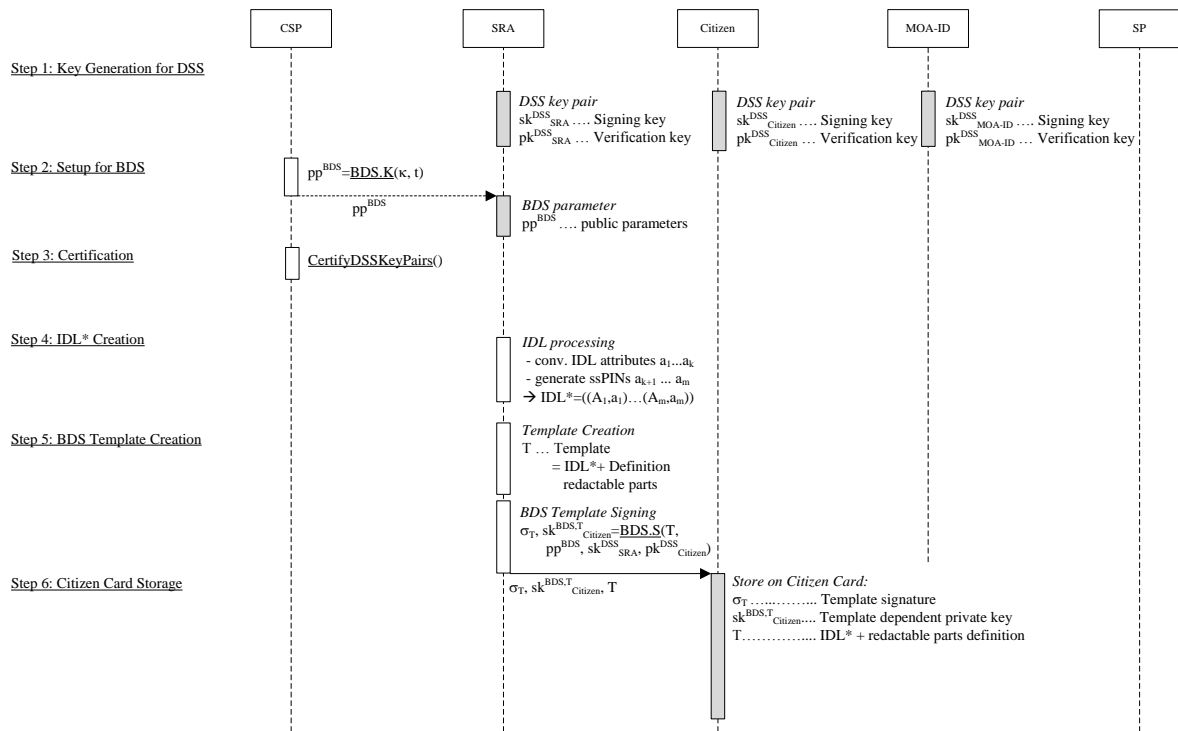


Figure 9.3: Sequence diagram of registration process

9.6.2 Identification and Authentication Process

Similar to the registration process, the identification and authentication process is designed to require minimal changes to the existing infrastructure. The main changes affect the (centrally deployed) identity provider MOA-ID and the client middleware. On the service provider’s side no essential extension must be made. The sequence diagram of the identification and authentication process is illustrated in Figure 9.4. The entire procedure consists of the following steps:

1. The citizen wants to access an application deployed and running at a service provider.
2. The service provider redirects the citizen to MOA-ID to request authentication. The authentication request holds the information in which sector the service provider operates.
3. MOA-ID sends a request to the citizen to get the citizen’s identity data and signature.
4. The citizen reads the template, holding IDL*. The verification of the template signature σ_T is optional, as it is assumed that IDL* stored on the citizen card is honestly computed by the SRA.
5. Due to data protection regulations, following redactions must be made: The sourcePIN and all pre-generated ssPINS not representing the given sector must be redacted out of IDL*; i.e., only the corresponding sector stays visible. In addition to these legally required redactions, the citizen is able to redact more identity attributes out of IDL*, which the citizen does not want to be sent to the service provider. For instance, the citizen may redact the name, but the date of birth is still available.

6. The message M is generated. This message includes the redacted IDL* and following additional information (to be compliant with the current implementation of MOA-ID):
 - Current date and time
 - Application data (e.g., application name, country in which the application is deployed, etc.)
 - Technical parameters (e.g., URL of the application, corresponding sector of the application, etc.)

This message is instantiated and signed by the citizen using the private key $sk_{\text{Citizen}}^{\text{DSS}}$ and the template dependent private key $sk_{\text{Citizen}}^{\text{BDS}, \text{T}}$. This outputs the message signature σ_M .

7. The citizen returns the identity data, consisting of the message signature σ_M and the message M , to MOA-ID.
8. MOA-ID verifies the message signature. In case this verification is positive, the message is authentic and a valid instance of the template as defined by the SRA.
9. MOA-ID creates an assertion Assert holding the available identity attributes and signs it using its private key $sk_{\text{MOA-ID}}^{\text{DSS}}$.
10. MOA-ID transmits this signed assertion (Assert and σ_A) to the service provider.
11. The service provider verifies the assertion signature and proceeds if the assertion is valid.
12. Depending on the available identity data of the citizen and the corresponding access rights, the service provider is able to grant or deny access.

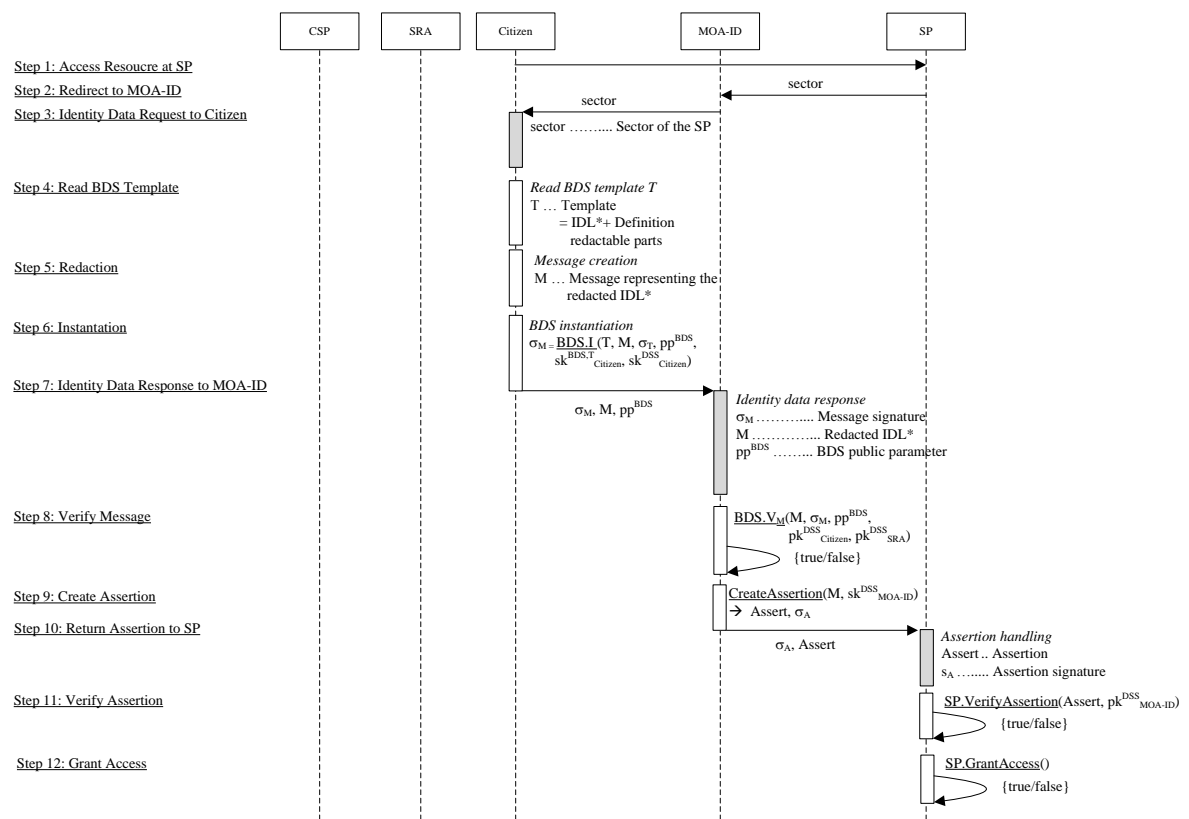


Figure 9.4: Sequence diagram of identification and authentication processes

9.7 Evaluation and Conclusions

The implementation has shown that the model is applicable to the existing infrastructure of the Austrian eID system. Table 9.1 compares the defined requirements (cf. Section 9.3) with the proposed model and the implementation. The present model has been designed according to the well established concept of identity provider and service provider. It well fits into existing infrastructure, which bases upon the deployment of the identity provider in a trusted environment.

The presented application to the Austrian eID system has been implemented on a prototype-basis. In concrete terms a demo SourcePIN Register Authority has been set up to issue appropriate identity links. All other changes has been implemented in the real life applications, such as the Austrian identity provider MOA-ID and the client middleware. Hence, to apply the presented model in real life and in large scale, the SourcePIN Register Authority must be involved, which may be a next step for the presented approach.

In case the identity provider should be deployed in a semi-trusted environment, such as a public cloud, additional privacy issues arise. In such a scenario, the identity provider acts honest but curious. An extended model which eliminates these privacy issues is presented in Zwattendorfer [2014]. This model uses proxy re-encryption whereas the registration authority encrypts the identity data for the user. Then the user is able to generate a re-encryption key for the identity provider, who is then able to re-encrypt the identity data for the service provider, whereas the identity provider only operates on encrypted data.

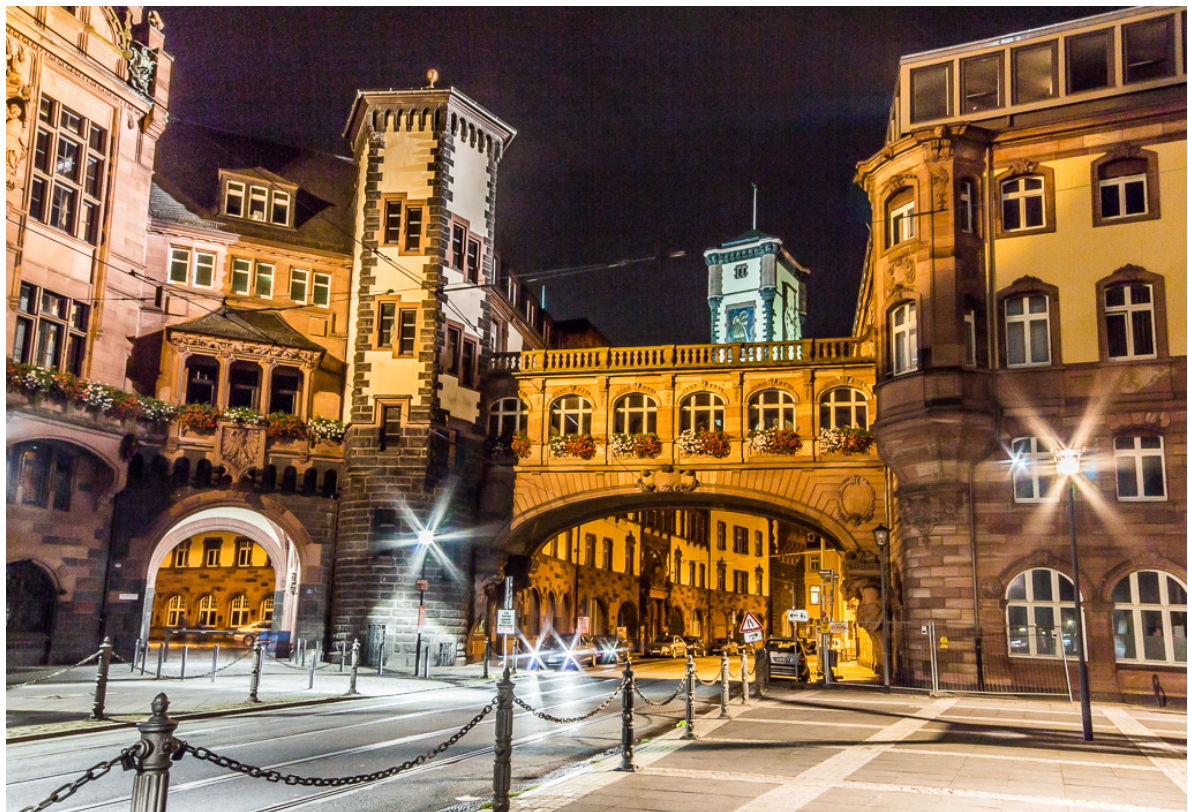
To conclude, the identity management model has been successfully applied to the Austrian eID system. That means, that the model may also be applicable to other eID systems, depending on the concrete system.

Table 9.1: Evaluation result against the defined requirements

Requirement	Fulfilled through
Qualified and authentic identity data	This requirement is fulfilled by using a trusted registration authority, which is responsible for the correct and high quality data provisioning. In the concrete application to the Austrian eID system, this trusted registration authority is represented by the SourcePIN Register Authority, which is responsible to issuing the Austrian identity link. Thus, also the implementation fulfils this requirement.
Integration effort and complexity	The model and implementation use existing identity protocols, which already support the transfer of the needed data out-of-the-box. The implementation fully integrates the existing Austrian eID infrastructure. The required changes mainly affects the centrally deployed SRA or centrally developed elements such as the identity provider MOA-ID. Especially, changes on the service providers' side are minimal and allow for an easy integration of the presented approach.
User-centered	The entire model is user-centered as the user specifies which data she wants to disclose to the identity provider and service provider. In the implementation of the client middleware the user is able to select the attributes from the identity link, which are revealed to the identity and service provider. Thus, the use is in full control, which data is disclosed.
Selective Disclosure	The identity attribute can be separately disclosed by the user. Here, the authenticity and integrity of the disclosed data remains due the usage of editable signatures. Thereby, the implementation bases on editable signature scheme presented in Section 6, which is applied to the identity link.
Privacy	The identity and service provider only gets the identity attributes, which are disclosed by the user. For sure, in this model the identity provider must act fully trusted as the identity provider is able to read the disclosed identity attributes.
Open standards	The model and the implementation makes use of open standards such as SAML. Furthermore, the implementation fully bases on the Austrian eID system, which in turn bases on open international standards or at least open national standards (such as the specification for the identity link).

Chapter 10

Next-Generation Public Administration Procedures



“Bureaucracy defends the status quo long past the time when the quo has lost its status.”

[Laurence J. Peter]

10.1 Introduction

According to EuroStat¹, 44% of all individuals [Eurostat, 2013b] and 87% of all enterprises [Eurostat, 2013a] in the euro area have interacted with public authorities using the Internet in the year 2012. So, electronic public administration procedures have evolved as an important element of e-Administration - on national as well as on international level. They also have proven their time- and cost-savings potential compared to paper-based procedures. Nevertheless, administrative electronic procedures still lack on efficiency and effectiveness, authenticity and integrity of the processed data, and on interoperability. These issues are underpinned by two European studies and the e-Government research community. In 2012 the Directorate General for Internal Market and Services² has published a Points of Single Contact (PSC) study. The establishment of PSCs in the EU Member States is one of the main requirements of the EU Services Directive [European Commission, 2006a] and provides one-stop-shops for service providers. This studies provide an assessment of the deployment of Points of Single Contacts in the EU Member States and reveals different gaps of the existing PSCs.

The second study, carried out by the Directorate General for Communications Networks, Content & Technology³ has published a study on e-Government and administrative burdens. The findings of this study are comparable to the findings of the PSC study. Also the research community is active in this area and define efficiency, interoperability and security as major success factors [Zefferer et al., 2014; J. R. Gil-Garcia, 2007; Altameem et al., 2006].

To bypass the mentioned issues, this chapter presents a next-generation flexible process model and a modular architecture for dedicated public administration procedures on national level and across borders in particular. The presented approach bases upon the PSC concept from the EU Services Directive and applies it also to use cases out of the Services Directive. Furthermore it increases the efficiency and effectiveness of public administration processes by using a pre-processing unit for processing e-Documents, which bases on the findings of Chapter 7. Finally, it ensures interoperability and security (in particular authenticity and integrity of the exchanged data) by applying the e-Document framework OCD (cf. Section 4.2), editable signatures (cf. Chapters 5 and 6) and the e-Delivery framework out of the SPOCS project (cf. Section 3.2.5.5).

The remainder of this chapter is structured as follows. Section 10.2 elaborates in detail on the issues and challenges of existing public administrative procedures. Based on that, Section 10.3 identifies concrete requirements for a next-generation of public administration procedures ensuring efficiency and effectiveness, authenticity and integrity, and interoperability. Section 10.4 presents the general architecture and defines the process model. To evaluate the architecture and process model, Section 10.5 presents an implementation based upon a concrete life event. Finally, Section 10.6 evaluates the architecture and the process model as well as the implementation against the identified requirements and draws conclusions.

¹<http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/>.

²DG MARKET, http://ec.europa.eu/dgs/internal_market/index_en.htm.

³DG CONNECT, <http://ec.europa.eu/dgs/connect/en/content/dg-connect>.

10.2 Issues and Challenges

Although electronic public administration procedures are already implemented in the field, they still are not perfect yet. Issues and challenges of public administration procedures have been assessed by the European Commission and the research community. Whereas the research community has published several papers the European Commission has conducted two studies.

In 2012 DG MARKET has published a *study on functioning and usability of the Points of Single Contact* under the Services Directive. This study provides

“[...] a preliminary assessment of the implementation of the Points of Single Contact in the Member States...” [European Commission, 2012]

Main findings of this study are that the establishment of PSCs has increased the interoperability, but still gaps have been identified, that

“[...] need to be overcome in order to unleash the full potential of the Points of Single Contact.” [European Commission, 2012]

Amongst other, these gaps are:

1. Simplification of administrative procedures is still not reached
2. Administrative procedures still base on conventional paper-based processes
3. 41% of the examined focus group had difficulties to complete the procedure

In addition, this study published a list of policy recommendations for PSCs and policy makers. A major set of recommendations focuses on administrative simplifications and completion of electronic procedures across border.

Furthermore, DG CONNECT has published a second *study on eGovernment and the Reduction of Administrative Burden* [European Commission, 2014c]. This study states:

“Simplification and personalization strategies involve making interactions between government and user as simple (and therefore as easy, quick, efficient and effective) as possible for users, which clearly reduces their administrative burden” [European Commission, 2014c]

Additionally, the study identified cross-border services as a challenge on European level:

“... provide citizens with equivalent cross-border services as is happening for enterprises under the EU Services Directive;” [European Commission, 2014c]

The need for administrative simplifications and reduced administrative burdens is also underpinned by the research community. For instance, Zefferer et al. [2014] have defined, based on the findings of J. R. Gil-Garcia [2007] and Altameem et al. [2006], efficiency as major success factors for successful e-Government services. Furthermore, they define security as

“... *crucial for most governmental and administrative procedures*” [Zefferer et al., 2014]

Following the findings of the studies and the research community, challenges for current public administration procedures can be divided into three categories:

1. Efficiency
2. Interoperability across borders
3. Security

To fully face these challenges, activities on a technical, organisational and legal level must be taken into account. As the present thesis focuses on the technical aspects, the remainder of this chapter considers technical activities in particular.

Concerning *efficiency*, the main reasons for delayed application processing are missing data and provisioning of non-machine-readable data. In the first case, application data and supplementary documents are usually sent through a web form from the PSC to the CA (or even to several CAs). There, the responsible official processes the application and may notice that data or documents are missing or are incomplete. At this point, a manual interaction with the applicant is needed, which delays the application and thus reduces efficiency of the whole procedure. For sure, CAs or the PSC have implemented basic data validation processes. Nevertheless, current validation implementations are very simple and do not support complex validations. Hence, there is need to establish sophisticated data validation processes.

The second challenge for current public administration procedures is *interoperability*. Cross-border interoperability focuses on interoperable services between different countries. Main issues are the data exchange and the delivery between countries. This concerns mainly the data exchange between PSC and CAs, between different CAs, as well as the delivery of the CA's decision to the applicant. Hence, elaborating appropriate data exchange and delivery mechanisms are the main challenges for interoperability.

The last challenge is *security* and concerns mainly the authenticity and integrity of the exchanged data especially in a cross-border context. Thereby, electronic signatures are the means of choice to ensure the trustworthiness of exchanged data. Hence, an appropriate usage of electronic signatures in public administration procedures must be ensured.

10.3 Requirements

Based upon the findings and discussed challenges in the previous subsection, requirements for a next-generation of public administration procedures have been identified and considered to be necessary by the thesis author. Thereby, following requirements - focusing on the technical level - have been identified:

Interoperability and cross-border services: Public administration procedures must rely on the established PSCs. Although PSCs need to be established in the context of the EU Services Directive only, an extension of the area of application into other domains seems reasonable as the concept of one-stop shops is widespread on the e-Government domain. In addition, findings of the EU Large Scale Pilot projects should be taken into account, as they already provide solutions to achieve interoperability.

Modularity and adaptability: Software components modelling public administration procedures electronically must be designed in an easy extensible and adaptable way in order to react quickly on changed needs. That means, needed functionalities must follow a modular and adaptable approach to ensure easy integration and extension of existing facilities.

Clear and usable interfaces: The communication between affected parties and involved modules and components must base on clear and usable interfaces. Wherever possible standardised interfaces should be used to achieve sustainable solutions.

Integration into existing infrastructures: PSCs are already deployed in different countries. That means, (basic) infrastructures exist. To facilitate the take up and to protect the past investment costs, the extended public administration procedure must be easily and smoothly integrable into existing infrastructures.

Automatic processing: A public administration process must ensure automatic processing. This means that manual interactions between applicants and administrations as well as between administrations amongst each other must be avoided as much as possible. Therefore, comprehensive data validation and data extraction facilities are needed, which are able to minimize the need for manual interactions.

Authenticity and integrity: Appropriate means to ensure the authenticity and integrity of the public administration procedure must be met.

Scalability: The public administration is concerned with high usage of their provided services and applications. Hence, appropriate means to ensure scalability of public administration procedures must be met.

Following section elaborates on the proposed general architecture and process flow for efficient and interoperable public administration processes, which aims to meet the identified requirements.

10.4 General Architecture and Process Model

10.4.1 Overview

To face the identified challenges and to fulfil the defined requirements, following basic action have been set:

- A pre-processing unit increases the efficiency of public administration procedures by ensuring mostly automatic processing through avoiding manual interactions based upon data validation and data extraction facilities.
- On the one hand interoperability is ensured by applying the PSC/CA model of the European Services Directive also to other use cases. On the other hand incorporating following findings of large scale pilot projects increases the interoperability too:
 - The OCD container (cf. Chapter 4), developed by the LSP SPOCS, is used as exchange format between the affected parties.
 - The e-Delivery solution of the LSP SPOCS (cf. Section 3.2.5.5) is used to enable a (cross-border) delivery of the final decision to the citizen.
- Authentication and integrity is ensured by the use of electronic signatures (as part of the OCD container). Depending on the use case, conventional or editable signatures are used.

For the general architecture and the process model two different use cases have been defined. The main difference between these use cases is the deployment location of the pre-processing unit. In the *first use case* the pre-processing unit is deployed in the domain of the PSC. This approach is beneficial if only one or a small number of PSC are deployed in a country. In case of several deployed PSCs the pre-processing unit must be setup up by each PSC, which may be adverse concerning the configuration effort. Therefore a common usage of the pre-processing unit seems to be beneficial. In addition, the deployment in the PSC domain is usually bad scalable, which may cause performance problems due to the potentially high loads of the pre-processing unit.

To overcome these issues, the *second use case* foresees a deployment of the pre-processing unit in the cloud. Via the cloud the pre-processing unit is accessible by several PSCs. The main advantages of this approach are:

Scalability: The cloud deployment enables a scalability of the pre-processing unit and is able to manage high loads of the unit.

Configuration effort: The configuration of the pre-processing unit has to be done only once. Potential issues through different configurations are eliminated.

Cost savings: No in-house infrastructure is needed for the pre-processing unit.

The following subsections discuss both use cases in more detail.

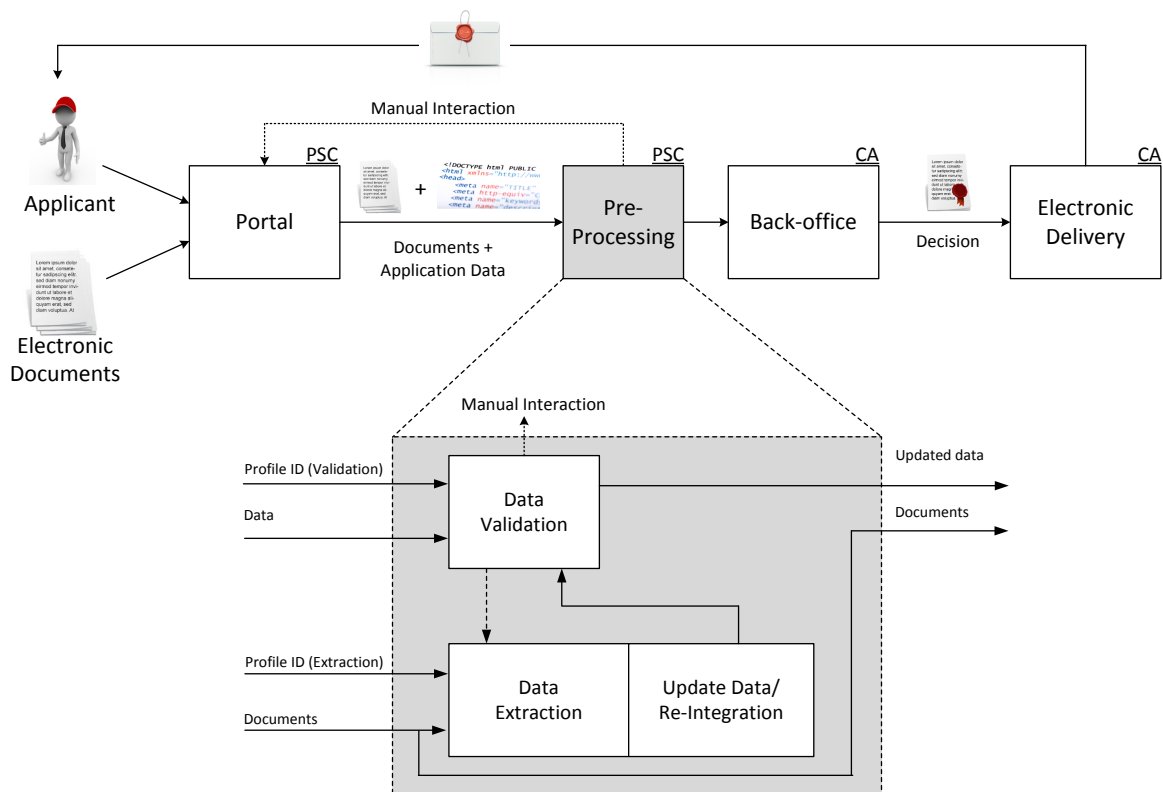


Figure 10.1: Architecture: Use case 1

10.4.2 Use Case 1: Non-Cloud Deployment

10.4.2.1 General Process Architecture

Figure 10.1 shows the general process architecture, which bases upon the common administration procedure structure (cf. Section 2.3). This structure has been amended by a pre-processing unit, whereas the other components (portal, back-office and electronic delivery) are nearly untouched. The pre-processing unit is deployed in the domain of the PSC and acts between the request for application (at the PSC) and the processing in the back-office (at the CA). It pre-processes the given data to ensure that no incomplete data are sent to the CA, which may cause delays and additional work on the CA's side.

The pre-processing unit strongly bases on the findings of Chapter 7. The main purpose is to avoid a manual interaction with the applicant at a later process stage and thus concerning the affected CAs. In case the pre-processing unit fails, a manual interaction backup exists. However, this interaction takes place on PSC side and thus is still more effective than manual interactions at the CAs.

The pre-processing unit consists of the following subunits: a comprehensive data validation unit, a data extraction unit, and a data update (data re-integration) unit. Following subsection elaborates on the process flow incorporating the pre-processing unit.

10.4.2.2 Process Model

Figure 10.2 illustrates the general process model for increasing efficiency and interoperability of public administration procedures. To ensure interoperability for cross-border applications, it incorporates the findings of the large-scale pilot projects SPOCS and STORK. The process model consists of following actors:

- a) The applicant, who wants to apply for an application
- b) The PSC as contact point for the applicant
- c) The CA responsible for processing the application

The detailed process flow is:

1. The applicant wants to apply for an application at the PSC.
2. Therefore, she fills in the corresponding forms and attaches the required documents.
3. Before sending the data to the CA, the PSC hands over these data to the pre-processing unit. Then a data validation process is executed to validate the application data and the attached documents. This validation must be tailored to the specific application to ensure a reasonable result⁴. This customization has to be configured initially. Of course, this customization causes additional work in the setup phase, but amortises over time because of having more automatic processes and thus reduced manual interactions. If the data validation was successful (i.e. all needed data are available): see step 6, otherwise see step 4.
4. The data extraction tries to extract the missing information from the available application data and documents. Thereby, the unit extracts or tries to extract the needed data. As for the data validation, also the data extraction mechanism must be tailored to the specific application to ensure a reasonable result⁵. The extracted data are added (re-integrated) to the present application data.
5. The entire data are then re-validated. In case re-validation was successful: see step 6. Otherwise a manual interaction is required. First, the official tries to manually extract the needed information from the available documents, which may happen by presence of a scanned document, which is not automatically extractable. If no appropriate data can be extracted, as for instance the needed document is missing, the applicant must be informed. Thereby, the applicant is requested to add or update the missing or wrong data.
6. The PSC creates an OCD container based upon the application data and attached documents⁶. This is done to ensure interoperability especially for cross-border applications, e.g. an applicant from country A applies for an application in country B. This generated OCD container is then handed over to the CA.

⁴That means the validation must validate if all data and documents are available, which are needed for the further processing at the CA. Depending on the complexity and kind of the application, the validation requires a structural validation (certain data fields are available) and an additional content validation (certain data fields must have certain values).

⁵That means the data extraction should extract all data which are needed. Depending on the complexity, various document formats must be supported. As for the data validation, the additional configuration effort amortises over time.

⁶Besides the application data, the attached documents are added to the payload layer of the OCD container.

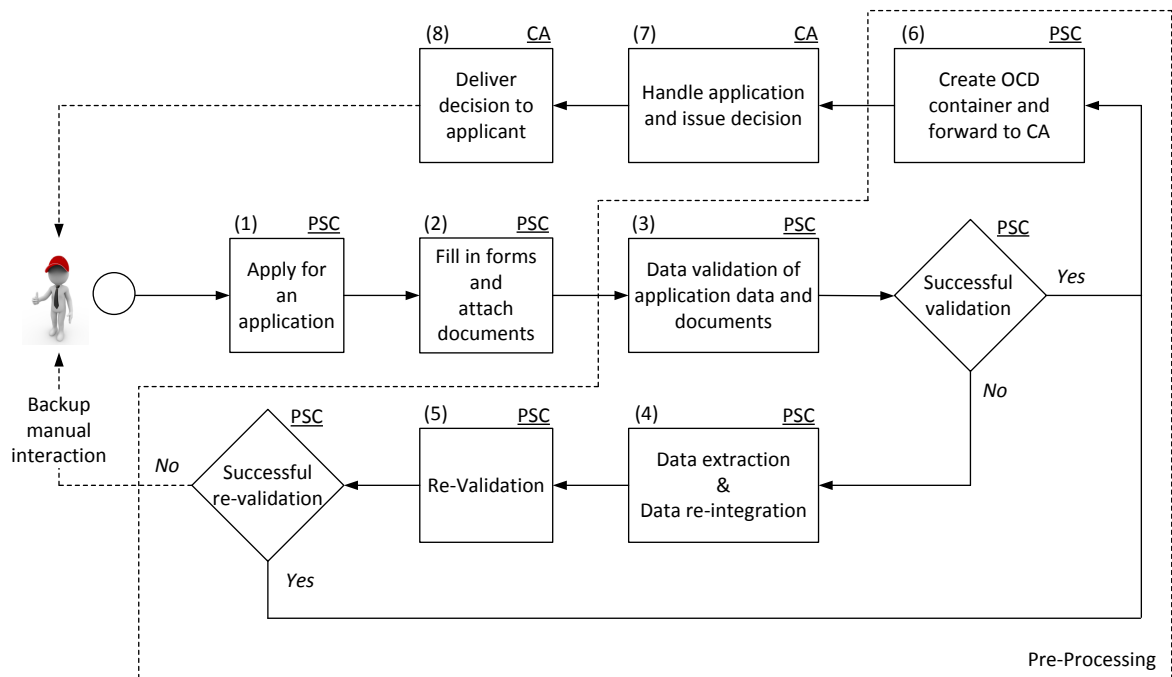


Figure 10.2: Process model: Use case 1

7. The CA receives the OCD container and automatically verifies the container using the OCD validation and verification module (cf. Section 4.3.3). This validation and verification includes also the signature verification of all contained signed e-Documents. Then the CA handles the application, whereas the data and document processing is easy and less time-consuming as all needed data are already available. After processing the application, the CA issues a decision.
8. This decision is delivered to the applicant using e-Delivery. Thereby, the national delivery system is used (in case of a national application) or by using the e-Delivery framework of the LSP SPOCS for a cross-border delivery of the decision.

10.4.3 Use Case 2: Cloud Deployment

10.4.3.1 General Process Architecture

Figure 10.3 shows the general architecture for the second use case. It consists of two independent PSCs and independent CAs. The pre-processing unit is deployed in the cloud and is shared by both PSCs. The functionality of the pre-processing unit is the same as for the first use case. The difference is that the unit is deployed in the cloud and thus does not need to be deployed by each PSC. As the documents and the application data must be forwarded to the pre-processing unit, they are wrapped into an OCD container beforehand. This container is signed by an editable signature, as parts of the contained data may be modified or update depending on the outcome of the validation and extraction unit. However, the use of this editable signature ensure that only data are modifiable which are specified by the PSC. For sure, data protection and privacy regulations must be taken into account for this deployment. That

means, to satisfy these regulations the pre-processing unit must be deployed in a private cloud, as other cloud models, such as the public cloud have clear privacy issues [Zwattendorfer et al., 2013b].

10.4.3.2 Process Model

Figure 10.4 illustrates the general process model for the second use case. The process flow bases upon the process flow of the first use case but takes the mentioned differences into account. The model consists of an additional actor, which is the cloud provider, where the pre-processing unit is deployed. The detailed process flow is⁷:

1. The applicant wants to apply for an application at the PSC.
2. Therefore, she fills in the corresponding forms and attaches the required documents.
3. The PSC creates an OCD container including the application data as well as the given documents. Then, the PSC signs the OCD container using an editable signature and defines which of the signed data can be modified. Additionally, the PSC assigns permission to the pre-processing unit. Hence, the pre-processing unit is able to update signed data, by retaining the authenticity and integrity of the PSC signature.
4. The pre-processing unit, deployed in the cloud, executes a data validation process to validate the application data and the attached documents. As for the first use case this validation must be tailored to the specific application to ensure a reasonable result. If the data validation was successful (i.e. all needed data are available): see step 8, otherwise see step 5.
5. The data extraction tries to extract the missing information from the available application data and documents. Thereby, the unit extracts or tries to extract the needed data. Again, the data extraction mechanism must be tailored to the specific application to ensure a reasonable result.
6. Based upon the extraction results the OCD container is updated with the extracted data. Due to the editable signature, the pre-processing unit is able to update the OCD container without invalidating the signature.
7. The entire data are then re-validated. In case re-validation was successful: see step 8. Otherwise a manual interaction is required. First, the official tries to manually extract the needed information from the available documents, which may happen by presence of a scanned document, which is not automatically extractable. If no appropriate data can be extracted, as for instance the needed document is missing, the applicant must be informed. Thereby, the applicant is requested to add or updated the missing or wrong data.
8. The PSC forwards the updated OCD container to the CA.
9. The CA receives the OCD container and automatically verifies the container using the OCD validation and verification module (cf. Section 4.3.3). This validation and verification includes also the signature verification of all contained signed e-Documents. Then the CA handles the application, whereas the data and document processing is easy and less time-consuming as all needed data are already available. After processing the application, the CA issues a decision.

⁷To hold the description self-contained, some process steps are repeated from the first use case even if they have not changed.

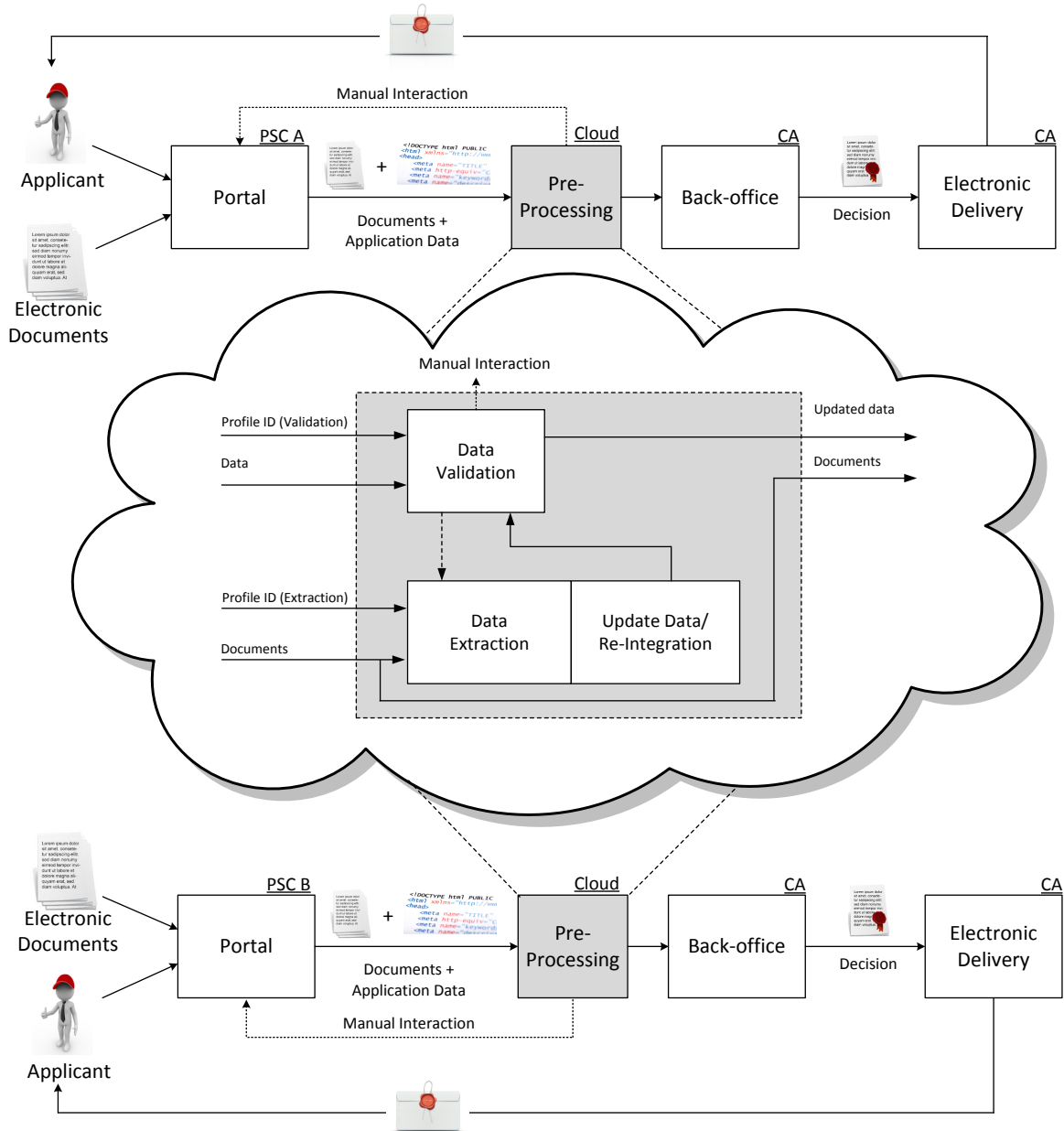


Figure 10.3: Architecture: Use case 2

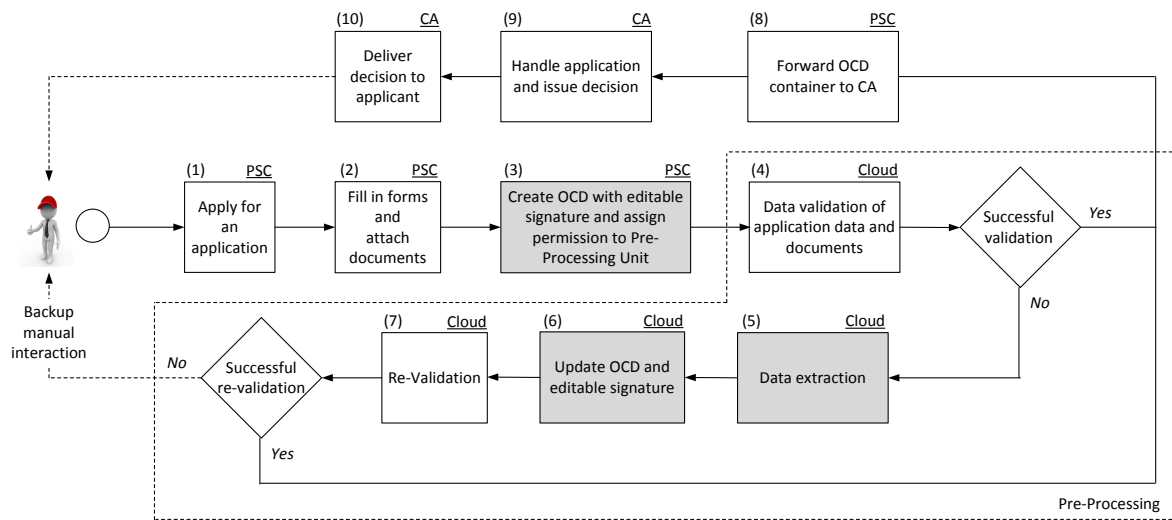


Figure 10.4: Process model: Use case 2

10. This decision is delivered to the applicant using e-Delivery. Thereby, the national delivery system is used (in case of a national application) or by using the e-Delivery framework of the LSP SPOCS for a cross-border delivery of the decision.

10.5 Implementation

The presented architecture has been implemented on a proof of concept prototype. Thereby a concrete use case has been implemented. This use case is that a foreign citizen wants to establish a ski school in Austria. According to the PSC Styria⁸ following documents are necessary as proof for her personal and businesslike prerequisites⁹:

- Proof of citizenship (Signed XML based document; contains name, date of birth and nationality of the citizen)
- Criminal record certificate (Signed XML based document; contains name and date of birth of the citizen)
- Medical certificate (Scanned PDF document; contains the needed medical information)
- Proof of qualification (Scanned PDF document; contains data to proof the citizen's qualification)
- Third party insurance certificate (PDF document; contains data about the third party insurance)

Following components have been implemented as shown in Figure 10.5:

⁸<http://www.eap.steiermark.gv.at/cms/beitrag/11201233/48164989/>.

⁹This list also contains the exemplary availability (in terms of document format, electronic signature and included basic data) of these documents to demonstrate the applicability of the pre-processing unit.

Demo PSC: A demo PSC has been implemented consisting of a simple portal. This portal provides a web form enabling to request an application for opening a ski school in Austria. The web form enables the applicant to fill in the required application data and to upload the needed e-Documents. After the application data and e-Documents have been successfully processed in the pre-processing unit, the demo PSC creates an OCD container based upon the findings of Section 4.2. This OCD container is then forwarded to the demo CA.

Pre-Processing Unit: A pre-processing unit has been implemented based upon the findings of Chapter 7. Thereby the validation unit, the extraction unit and the re-integration unit are interconnected accordingly. The data re-integration receives the specific extraction result and the application data as input. It re-integrates (updates) the application data with the extracted information and returns these updated application data. Furthermore, following configurations steps must be done for the data validation and data extraction unit:

Data validation: For all needed documents (proof of citizenship, etc.) appropriate validation profiles have been defined. For instance, the validation profile for the proof of citizenship verifies if the document structure contains a name, date of birth and citizenship.

Hint: Due to a missing international common structure of the needed documents, appropriate XML based structures of the proof of citizenship and criminal record certificate - based upon the Austrian structure of these documents - have been defined to demonstrate the applicability of the validation unit.

Data extraction: For the remaining documents¹⁰ appropriate extraction profiles including the corresponding template matching strings have been defined.

Hint: The third party insurance certificate is provided as PDF file, which enables the extraction of the needed data and therefore shows the applicability of the extraction unit. To demonstrate the backup manual interaction the medical certificate and the proof of qualification are available as scanned PDF file. Thus the needed data cannot be extracted automatically and the responsible person at the PSC tries to manually extract the needed data from the given documents.

Demo CA: A demo CA, which is responsible for processing the request is implemented. This demo CA receives the OCD container from the demo PSC and verifies the container. This OCD container contains all needed application data and electronic documents. Based on that the CA issues a decision. As the applicability of e-Delivery across borders has already been proven by Tauber [2012], this final step is omitted. Instead, the decision is sent back to the demo PSC, where the applicant is able to inspect it.

¹⁰That means the medical certificate, proof of qualification and third party insurance certificate.

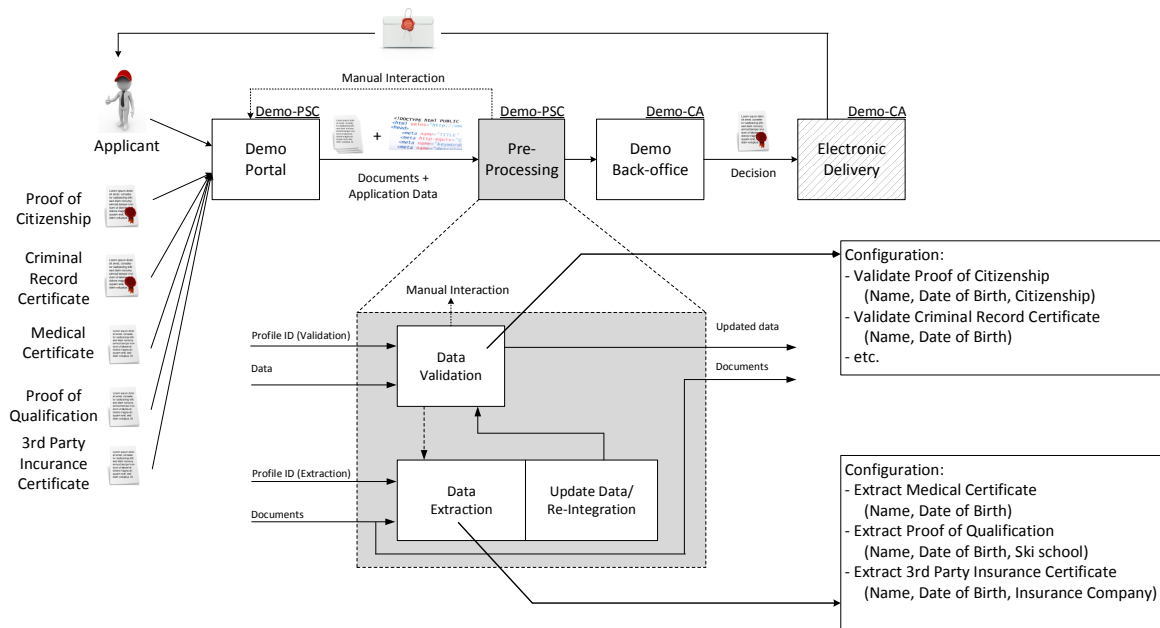


Figure 10.5: Implementation next-generation public administration procedures

10.6 Evaluation and Conclusions

The implementation has shown that the proposed architecture and model for a next-generation of public administration procedures is applicable and is able to fulfil the identified requirements with the following given limitations. The model has been implemented for a concrete use case, which was to establish a ski school in Austria. Thereby a demo process has been set up according to the rules given by the PSC Styria. The implementation of the presented approach direct at the PSC Styria may be a next step for the presented approach. Nevertheless, the implementation has proven that the model and implementation are basically applicable, but still some drawbacks exist.

The implementation has shown, that there is an additional effort to configure and customize the pre-processing unit. However, this additional effort has to be done only once (for each process) and thus amortises over time thanks to having more automatic processes and reduced manual interactions. Concerning the reduction of needed manual interaction, the result of the evaluation is similar to the evaluation given in Chapter 7. The XML-based data validation of the pre-processing works well and successful on XML data, whereas the quality of the data extraction depends on the structure of the given documents. Due to the still high amount of paper based documents (usually available as scanned copy) [European Commission, 2014c] an additional manual interaction backup has been introduced as an automatic extraction out of a scanned copy is yet not possible. In this backup mechanism, the official tries to manually extract the needed information from the available documents, which is still a benefit compared to a manual interaction back to the citizen. Only after, the official is not able to extract the needed data (i.e. the document is missing), the citizen must be informed.

The presented approach provides appropriate technical means to face the identified issues on efficiency, interoperability and security. Nevertheless, challenges on the semantic, organisational and legal level still exists. One of these challenges is the mutual recognition of e-Documents in the dif-

ferent Member States, which is also known as the document equivalence problem. This means for instance, if an Austrian criminal record certificate is recognised in another Member States and vice versa. This issue has also been addressed by the large scale pilot SPOCS [Stasis et al., 2012]. Additionally, the other main challenge is the recognition of the different national e-Delivery systems. Whereas, the recognition of different e-Delivery solutions seems to be solved by the upcoming eIDAS Regulation [European Commission, 2014b], the challenges concerning e-Documents still exist. The reason for that is, that the eIDAS Regulation states only:

“An electronic document shall not be denied legal effect and admissibility as evidence in legal proceedings solely on the grounds that it is in electronic form.” [European Commission, 2014b]

For a concrete mutual recognition of e-Documents, this seems to be insufficient and is seen by the thesis author as a missed opportunity for achieving document equivalency within Europe even if the implementing acts of the regulation may be more precise. Hence, additional activities must be set by the European Commission to face this issue.

Finally, Table 10.1 compares the identified requirements from Section 10.3 and shows how they have been fulfilled by the proposed approach with the given limitations.

Table 10.1: Evaluation result against the identified requirements

Requirement	Fulfilled through
Interoperability and cross-border services	Interoperability is achieved by several means. On the one hand the concept of PSCs as one-stop-shop is applied also to other use cases out of the EU Services Directive. On the other side findings of the large scale pilot projects have been incorporated. In particular, the interoperable e-Document framework OCD (cf. Section 4.2) and the e-Delivery solution of the LSP SPOCS are used.
Modularity and adaptability	The architecture of the entire process and the pre-processing unit is fully modular (see also Section 7.5). Through this modular approach, certain modules can be easily exchanged or the functionality can be extended by adding further modules.
Clear and usable interfaces	The developed architecture and process model have been developed following the KISS principle. That means all interface are designed for easy usage and hide the complexity of the operations behind. Wherever possible open standards and specifications have been used.
Integration into existing infrastructures	The entire architecture takes into account the existing infrastructure - in particular the already deployed PSCs and CAs in the back-office and the general e-Government process architecture given in Section 2.3. The pre-processing unit is easy integrable thanks to the easy interface.
Automatic processing	The decrease or - in the best case - removal of the need of manual interaction is achieved by the use of the pre-processing unit, which validates the data and optionally tries to update the needed data. As already indicated limitations concerning the availability of extractable electronic document exists. Nevertheless, an additional manual interaction backup exists, where an official tries to extract the need data in a first stage and only if this is not possible a manual interaction back to the citizen is needed.
Authenticity and Integrity	Authenticity and integrity is ensured by the use of electronic signatures. Wherever e-Documents and data are exchanged an OCD container is used. This container is signed by a conventional signature or an editable signature depending on the use case.
Scalability	The pre-processing unit is concerned with a high usage including load peaks - especially if the unit is used by several PSCs. Therefore, the second use case foresees a deployment of the pre-processing unit in the cloud. Hence, this deployment fully benefits from the scalability and elastic advantages of a cloud deployment. For sure, data protection and privacy regulations must be taken into account for this use case. Thus, a deployment within a private cloud seems to be favorable.

Chapter 11

Summary and Conclusions



“Work expands (so as) to fill the time available for its completion.”

[Cyril Northcote Parkinson]

11.1 Summary

The basis for this thesis has been given by various sources. As indicated in Figure 11.1 these sources have been various European legal Regulations and Directives and other European initiatives like the Digital Agenda for Europe or the e-Government action plan. An important additional source have been the e-Government and IT security research community focusing on electronic signatures. Together they formed the need for next-generation technologies and next-generation technologies for e-Documents. To create concrete actions for the thesis, seven major key action points have been identified in Chapter 3. These key action points concluded the first part of the thesis, which elaborated in the beginning about the transition from traditional to electronic administrative services and an overview about various European initiatives.

In Part II next-generation technologies for e-Documents have been developed. In the Chapters 4 to 7 different core technologies have been developed. These core technologies have been brought together to develop next-generation applications for e-Documents, which are presented in Part III of the thesis. Table 11.1 gives the concrete thesis results compared to the identified key actions points.

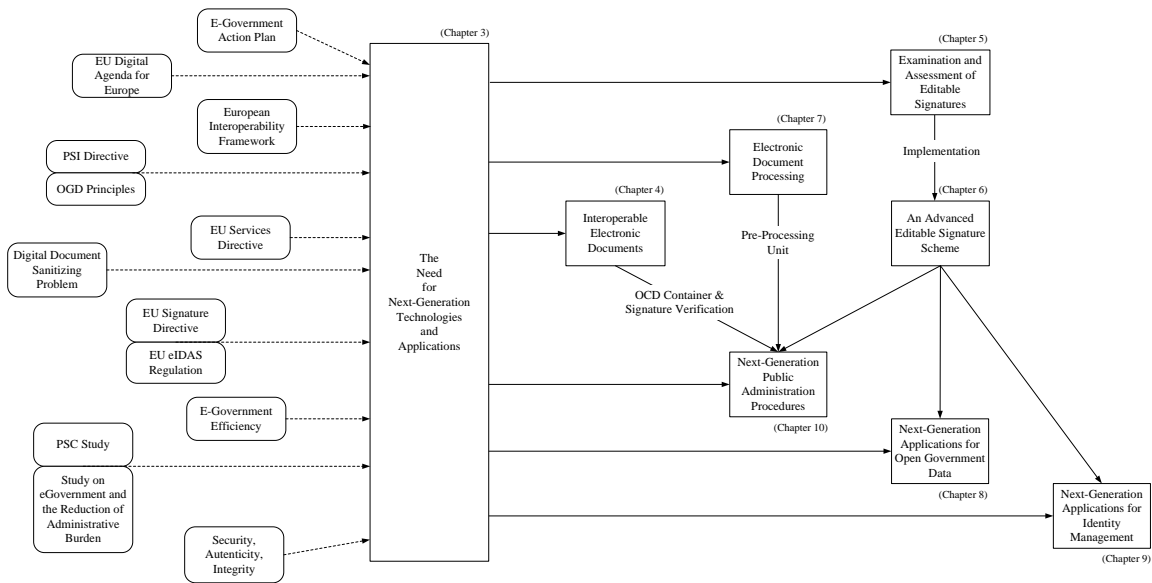


Figure 11.1: Thesis summary

Table 11.1: Key action points vs. thesis results

Key action point	Thesis result
#1 An interoperability framework for the exchange of e-Documents in (government based) cross-border scenarios is needed	<p>In Chapter 4 an interoperable e-Document framework - called OCD (Omnifarious Container for e-Documents) - has been developed. This framework enables the secure exchange of electronic documents across borders. Although developed with focus on the use cases of the EU Services Directive, the framework is applicable for each type of data exchange based upon e-Documents both on national and international level.</p> <p>The OCD has proven its applicability in a 18 month lasting piloting phase in the large scale pilot project SPOCS, whereby the implemented software modules have been deployed in real life environments. From a technical point of view the OCD has met all requirements of the piloting countries. Additionally the modules have been assessed being on a mature level and ready to be deployed in other real life environments outside of the project SPOCS.</p>
#2 An extension of the Austrian signature verification service is needed to support the use of TSL as well as the support of verifying the QC and SSCD property	<p>Chapter 4 presents a cross-border signature verification via trust-service status lists based upon the Austrian module MOA-SP. The presented implementation supports the certificate validation based upon TSLs. Thus it enables - on the one side - the certificate validation up to a trusted certificate and - on the other side - the verification if the signer certificate is a qualified certificate (QC) and if the signature has been created using a secure signature creation device (SSCD).</p> <p>All of these validations and verifications are important pillars for the cross-border verification of electronic signatures. Especially, the verification of the QC and SSCD properties enable the recognition of a qualified electronic signatures. Qualified electronic signatures are legally equivalent to handwritten signatures as they allow a unique identification of the signatory, which is a vital prerequisite for many applications in the area of e-Documents and identity management.</p>

Table 11.1: Key action points vs. thesis results

Key action point	Thesis result
#3 Assessment of editable signature schemes and implementation of an editable signature scheme, which is applicable in the e-Government context	<p>Editable signatures allow modifications of signed data, but simultaneously preserve the authenticity and integrity of the unchanged data. This is an important property for publishing signed data, which need to be censored or sanitized beforehand due to privacy or legal regulations protecting private or personal data (this is well known as the digital document sanitizing problem). In Chapter 5 different selected editable signature schemes have been evaluated and assessed focusing on the applicability in the e-Government domain. This assessment bore a list of three suitable editable signature schemes.</p> <p>Nevertheless obstacles hindering an application of these schemes in real applications exist. Hence, in Chapter 6 a concrete implementation of an editable signature scheme has been presented. This implementation bases upon the existing core implementation of the editable signature scheme “blank digital signature”. The presented implementation bases upon XML and the advanced electronic signature standard XAdES. Furthermore, the implementation is applicable in the e-Government domain and serves as basis for the next-generation applications presented in the remainder of the thesis.</p>
#4 More efficient and automatic processing of e-Documents is needed	<p>Efficiency is a main pillar for successful e-Government processes and procedures. This has been underpinned by different studies from the European Commission and also from the e-Government research community. Thereby, the processing of e-Documents is a bottleneck, as incomplete or wrong data usually requires manual interaction to proceed with the procedure. This manual interaction renders a simplification of administrative procedures impossible and prevents efficiency increase.</p> <p>To face this challenge, in Chapter 7 concepts and software modules have been develop to improve the processing of e-Documents. First a validation module provides comprehensive and sophisticated validation possibilities of e-Documents and other data. The second module provides extraction facilities which enable the extraction of data out of available e-Documents. Both developed modules serve as a basis for the pre-processing unit, which is a main finding for key action point #7.</p>

Table 11.1: Key action points vs. thesis results

Key action point	Thesis result
<p>#5 Next generation applications in the area of open government data and public sector information are needed, which consider security aspects such as the authenticity and integrity of the provided data</p>	<p>Open Government Data is currently one of the most discussed topics in the e-Government area. Surprisingly, security aspects such as authenticity and integrity of the published data has not been assessed so far. Therefore, Chapter 8 presents next-generation applications enabling Trusted Open Government Data (Trusted OGD). Trusted Open Government Data enables an authentic and trustworthiness publication of OGD by means of electronic signatures. Thereby, the data to be published is signed by the OGD provider before publishing. The signed OGD can then be verified by the OGD recipient. In case of a successful verification the recipient can fully trust the received data. The presented approach makes use of conventional electronic signatures as well as editable signatures enabling the publication of even redacted data.</p>
<p>#6 Next generation of an identity management system, enabling a selective disclosure of identity attributes and applicable to the Austrian eID system, is needed</p>	<p>Identity management is one of the main pillars of e-Government and most countries have set up different eID solutions. Additionally, most solutions require to reveal the entire identity data to the identity provider and/or the service provider. For privacy reasons, many users do not want to reveal their entire identity data set and in some cases this is even not required. Therefore, Chapter 9 developed a novel approach for a user-centered identity management model for national eID solutions, which enables a selective disclosure of identity data. The approach bases upon editable signature and has been adopted to the Austrian eID system.</p>

Table 11.1: Key action points vs. thesis results

Key action point	Thesis result
#7 Technical approaches for efficient and secure public administration procedures across borders are needed	<p>Efficiency, security and interoperability are the main success factors for successful public administration procedures as defined by different European studies and the e-Government research community. The common implication of them is that existing public administration procedures need to improve the efficiency, security (in terms of authenticity and integrity of the exchanged data) and even interoperability.</p> <p>Therefore, Chapter 10 presents a next-generation flexible process model and a modular architecture of public administration procedures on national level and across borders in particular. The approach takes into account the findings of the large scale pilot projects SPOCS and STORK. Furthermore it bases upon the EU Services Directive, but extends the usage also for use case out of the Services Directive. To increase the efficiency, the model makes use of a pre-processing unit, which intention is to minimize or - in the best case - eliminate the need of manual interactions. Thereby the approach makes use of the different core technologies developed in Part II of the thesis.</p>

11.2 Conclusions

The developed concepts and methods in this thesis have proven their applicability through appropriate implementations. Thereby, the implementations of the core technologies served as a basis for the implementations of the next-generation applications. For each main finding, representing the solution for the identified key actions point, individual conclusions can be given. Hence, Table 11.2 presents this individual conclusions separated for each main findings of the thesis.

Table 11.2: Individual conclusions of the main findings

Main finding	Individual conclusions
A) Interoperability framework OCD	<p>The framework OCD has been assessed positively. Nevertheless a few issues and challenges for the future of the OCD exist. One of the major challenges concerns the standardisation of the OCD as this is a key element for achieving sustainability. Although the OCD bases mostly on open standards and specifications, the OCD itself is not yet standardised. Concrete suggestions for the standardisation are given in the OCD final report [Stranacher, 2013] which comprise the standardisation of the meta data layer and the PDF based implementation of the OCD container.</p> <p>Another main challenge affects the mutual recognition of e-Documents (=e-Document equivalency). This challenge must be treated on a higher level and concerns both legal and semantical aspects. The mutual recognition of e-Documents is not only an issue of this finding, but also from the main finding on efficient and secure public administration procedures (cf. main finding #7).</p>
B) Austrian signature verification service with TSL support	<p>The integration of trust-service status lists into the Austrian signature verification service MOA-SP has been positively evaluated. Nevertheless, the entire system is dependent from the quality of the issued Member State TSLs. Unfortunately, the quality of this issued TSLs is very widespread and only a small number of TSLs are fully correct. The reason for that seems to lie in the complexity of the TSL specification and EC TSL-Decisions.</p> <p>The upcoming eIDAS Regulation may open the door for improving the current situation. The regulation states that “<i>Each Member State shall establish, maintain and publish trusted lists with information related to the qualified trust service providers for which it is competent together with information related to the qualified trust services provided by them.</i>” [European Commission, 2014b]. This statement is quite general. Nevertheless, the European Commission is obliged to define appropriate technical specifications and formats for trusted lists by means of implementing acts. The implementation of these acts will determine if the current situation can be improved.</p>
C) Editable signature scheme for e-Government	<p>The presented editable signature scheme has proven its applicability in the e-Government domain. Nevertheless, further developments may be beneficial. The current implementation bases upon Java. It may be advantageous to port this implementation also to other platforms. Another aspect may be the development of an editable signature scheme based upon PDF and the advanced electronic signature standard PAdES.</p>

Table 11.2: Individual conclusions of the main findings

Main finding	Individual conclusions
D) Efficient e-Document processing	The prototypical implementations of the data validation and data extraction modules have shown that they are applicable in the e-Government domain. Nevertheless, for the deployment of these modules in real life environments the implementation must be fostered to be on a mature level. Additionally, the functionalities of the modules may be extended. The implemented data validation module currently supports the validation of XML based data only. This may be extended to other data formats in the future. Finally, the implemented data extraction module currently enables the data extraction out of PDF and Word documents. This may also be extended to further data formats.
E) Trusted Open Government Data	The applicability of the Trusted OGD approach has been proven by proof-of-concept implementations for the server-side and client-side component. Thereby, the server-side is responsible for publishing the data and the client-side application makes use of the published and signed data. The server-side implementation can be used by OGD provider through an easy integration into the existing infrastructure. Nevertheless, the implementation may be fostered to be on a mature level and ready to be deployed in real life environments. The client-side implementation may serve as best practice for Trusted OGD applications.
F) User-centered identity management enabling selective disclosure	The presented identity management model and its implementation for the Austrian eID system have proven their applicability. The implementation already takes into account the specifics of the Austrian eID system. Nevertheless, for a concrete and large scale implementation some challenges still exist. For instance one of these challenges is that the model requires changes of the identity link, which is issued by the SourcePIN Register Authority. Another challenge is to implement the needed functionalities for the editable signatures on or via smartcards representing an Austria Citizen Card.

Table 11.2: Individual conclusions of the main findings

Main finding	Individual conclusions
G) Efficient and secure public administration procedures	<p>The proposed model has been implemented on a proof-of-concept basis and has shown its applicability. Thus it represents an approach for next-generation of public administration procedures on a technical level. For sure, as also indicated by the different European studies and the research community, challenges on the semantic, organisational and legal level still exists.</p> <p>The main challenges for e-Documents is the mutual recognition of e-Documents in the different Member States as already indicated above. That means that there is still a missing consent on which e-Documents from a Member State A are legally equivalent to an e-Document in Member State B. For instance, what is the equivalent Austrian document to a Greek certificate of qualification for an architect? It is even not clear if there exists such an equivalency for all use cases. Article 37 of the upcoming eIDAS regulation regulates the legal effects of electronic documents. Unfortunately, this article consists of a single and general statement, which does not contain any details. From the thesis author's point of view this is a missed opportunity for achieving document equivalency within Europe. Nevertheless, the large scale pilot e-SENS may provide an appropriate proposal for the mutual recognition of e-Documents.</p>

Part IV

Appendices

Appendix A

Advanced Editable Signature Examples

A.1 Template Signature (enveloping)

Listing A.1: Template signature (enveloping)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ds:Signature Id="signature-1-1" xsi:schemaLocation="http://www.w3.org/2000/09/xmldsig#
   xmldsig-core-schema.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://
   www.w3.org/2001/XMLSchema-instance" xmlns:bdss="urn:bdss">
3   <ds:SignedInfo>
4     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
5     <ds:SignatureMethod Algorithm="http://www.iaik.tugraz.at/bdss#ECDSAwithSHA256"/>
6     <ds:Reference Id="Reference-c7b5c8fc-1a" URI="#Object-c7b5c8fc-1">
7       <ds:Transforms>
8         <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
9           <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect">id("
             Object-c7b5c8fc-1")/node()/</XPath>
10          </ds:Transform>
11        </ds:Transforms>
12        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
13        <ds:DigestValue/>
14      </ds:Reference>
15      <ds:Reference Id="Reference-c7b5c8fc-1b" Type="http://uri.etsi.org/01903#
        SignedProperties" URI="#SignedProperties-c7b5c8fc-1">
16        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
17        <ds:DigestValue>imjdswwF9XFxhWoiZuf9lqo6Ma2KEHT3Oo3qLg+9S0c=</ds:DigestValue>
18      </ds:Reference>
19      <ds:Reference Id="XSL1R" URI="#XSL1">
20        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
21        <ds:DigestValue>sNPJMcbCV9McmC6PJcMHVDggr0HawN36UK7UVrOG6HI=</ds:DigestValue>
22      </ds:Reference>
23      <ds:Reference Id="XSL2R" URI="#XSL2">
24        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
25        <ds:DigestValue>Wpeblm7G4m5eFsc2KafkV8L0UFjLLquvLl2rwIgdPLY=</ds:DigestValue>
26      </ds:Reference>
27    </ds:SignedInfo>
28    <ds:SignatureValue>BQAAAakDAAABaQIAAAAEdGVzdAAAAAEWAAAAIQCEzHph69nPOQ/lhnUagsDyb/iG
29    EGCx2YHD00/Z0W0wwAAACZxAAAAIQJ+Pgk37gPZ8GDe2WlBucvd30DYev7LqPZW
30    P38NABnHBgAAABkWAAAAFMfhPg0IeY56mNfxZnzfIemYIrfAAAA63IAAAAhAlbR
31    zF27q5iRdNzk22DwCqK9tXj4sqRy4TUMu9GJDB1AAAnGIAAABJYQAAACA9AFYS
32    YchrEi2OuN3LeKwJUKtdrLP3kD1480VtjjRheQAAACBMM2ZYrnadhDJ98p5uiOAx
33    6JFyYU0AHGMocHwnKwycMwAAAEphAAAAIH790EsdVozySuYXcCCT8XVYSHZfpvT6
34    458nqrW+DxakAAAAIQCjvxdg/soN7XqY8PadNcdW5b1AWYDXqSiLL7/j9FKn7AAA
35    ACECWHqqsjXI105bUyS9MdTk4hW+MdD6USPiwX3Alyfz1scAAAA3MDUCGQCoxmZY
```

```

36 VGB60H1Cn9B0CbFj3JnfZ8ufjUICGEw+VNF1wBDfw9xZzgLCyztMT3gpC7mu5QAA
37 AEsQAAAAHxmooa7vRkpW99CEaIjCjVBEL9VnzEkwMfi8gxTGx4kAAAAjCAAAAB5h
38 0VUWD4XnLORz0PhVeG/N+kOFgNY9kCHJPsSBLPM=</ds:SignatureValue>
39 <ds:KeyInfo>
40   <ds:X509Data>   <ds:X509Certificate>
41     MIIBcjCCASgCJGM0OWViNjY0LTZjMmQtNGIwZS04YWQxLTcyYzFlZDZAYMmY3YzAL
42     BgcqhkjOPQIBBQAwQzELMAkGA1UEBhMCQVQxEDAoBgNVBAoTB1RVIEdyYXoxDTAL
43     BgNVBAsTBELBSUsxZzARBgNVBAMTCm9yaWdpbmF0b3IwHhcNMTIwMzA0MTU0ODI2
44     WhcNMTQwMzA0MTU0ODI2WjBDMQswCQYDVQQGEwJBVDEQMA4GA1UEChMHVFUgR3Jh
45     eJENMAsGA1UECxMESUFSZsETMBEGA1UEAxMKb3JpZ2luYXRvcjBjBMBMGByqGSM49
46     AgEGCCqGSM49AwEBAzIABH1qSrz0FvMG7TuxVmG1T6pEJi+mP2+QO+fe9VxG6q8r
47     2fx98A8hy8gYKw3pfqRGBzALBgcqhkjOPQIBBQADNwAwNAIYLHhulrvuaGhXb172
48     FayNvzUXqtIzTjKAahdYwwpJVlrb8o3g9k84miMf78VP4La4Pk=</ds:X509Certificate>
49   </ds:X509Data>
50 </ds:KeyInfo>
51 <ds:Object Id="Object-c7b5c8fc-1">
52   <template id="test">
53     <templateentry>
54       <message type="fix" length="0">
55         <text>&lt;as:Assertion xmlns:as="urn:as" />&gt;</text>
56       </message>
57     </templateentry>
58     <templateentry>
59       <message type="fix" length="0">
60         <text>&lt;as:Person>&gt;</text>
61       </message>
62     </templateentry>
63     <templateentry>
64       <message type="fix" length="0">
65         <text>&lt;as:Name>&gt;</text>
66       </message>
67     </templateentry>
68     <templateentry>
69       <message type="fix" length="0">
70         <text>&lt;as:GivenName>&gt;</text>
71       </message>
72     </templateentry>
73     <templateentry>
74       <message type="exch">
75         <text>&lt;value>Max</value>&lt;/value>&gt;</text>
76       </message>
77     <templateentry>
78       <message type="exch">
79         <text>&lt;value>*&lt;/value>&lt;/value>&gt;</text>
80       </message>
81     </templateentry>
82     <templateentry>
83       <message type="fix" length="0">
84         <text>&lt;/as:GivenName>&gt;</text>
85       </message>
86     </templateentry>
87     <templateentry>
88       <message type="fix" length="0">
89         <text>&lt;as:FamilyName>&gt;</text>
90       </message>
91     </templateentry>
92     <templateentry>
93       <message type="blank" length="100">
94         <text>&lt;value>Mustermann</value>&lt;/value>&gt;</text>
95       </message>
96     </templateentry>
97     <templateentry>
98       <message type="fix" length="0">
99         <text>&lt;/as:FamilyName>&gt;</text>
100      </message>

```

```

99     </templateentry>
100    <templateentry>
101      <message type="fix" length="0">
102        <text>&lt;/as:Name&gt;</text>
103      </message>
104    </templateentry>
105    <templateentry>
106      <message type="fix" length="0">
107        <text>&lt;/as:DateOfBirth&gt;</text>
108      </message>
109    </templateentry>
110    <templateentry>
111      <message type="fix" length="0">
112        <text>&lt;/value&gt;11.08.1984&lt;/value&gt;</text>
113      </message>
114    </templateentry>
115    <templateentry>
116      <message type="fix" length="0">
117        <text>&lt;/as:DateOfBirth&gt;</text>
118      </message>
119    </templateentry>
120    <templateentry>
121      <message type="fix" length="0">
122        <text>&lt;/as:IdNumber&gt;</text>
123      </message>
124    </templateentry>
125    <templateentry>
126      <message type="exch">
127        <text>&lt;/value&gt;123456&lt;/value&gt;</text>
128      </message>
129      <message type="exch">
130        <text>&lt;/value&gt;*&lt;/value&gt;</text>
131      </message>
132    </templateentry>
133    <templateentry>
134      <message type="fix" length="0">
135        <text>&lt;/as:IdNumber&gt;</text>
136      </message>
137    </templateentry>
138    <templateentry>
139      <message type="fix" length="0">
140        <text>&lt;/as:Person&gt;</text>
141      </message>
142    </templateentry>
143    <templateentry>
144      <message type="fix" length="0">
145        <text>&lt;/as:Assertion&gt;</text>
146      </message>
147    </templateentry>
148    <templateentry>
149      <message type="fix" length="0">
150        <text>&lt;/xadesDigestTemplate&gt;&lt;/ds:DigestMethod Algorithm=&quot;http://www.w3
          .org/2001/04/xmlenc#sha256&quot;/&gt;&lt;/ds:DigestValue&gt;
          imjdswwF9XFxhWoiZuf9lqo6Ma2KEHT3Oo3qLg+9S0c=&lt;/ds:DigestValue&gt;&lt;/
          xadesDigestTemplate&gt;</text>
151      </message>
152    </templateentry>
153    <templateentry>
154      <message type="blank" length="5000">
155        <text/>
156      </message>
157    </templateentry>
158    <templateentry>
159      <message type="fix" length="0">

```

```

160     <text>&lt;XSL1Digest&gt;&lt;ds:DigestMethod Algorithm=&quot;http://www.w3.org
      /2001/04/xmlenc#sha256&quot;/&gt;&lt;ds:DigestValue&gt;
      sNPJMcbCV9MCmC6PJcMHVDggr0HawN36UK7UVrOG6HI=&lt;/ds:DigestValue&gt;&lt;/
      XSL1Digest&gt;</text>
161   </message>
162 </templateentry>
163 <templateentry>
164   <message type="fix" length="0">
165     <text>&lt;XSL2Digest&gt;&lt;ds:DigestMethod Algorithm=&quot;http://www.w3.org
      /2001/04/xmlenc#sha256&quot;/&gt;&lt;ds:DigestValue&gt;
      Wpeblm7G4m5eFsc2KafkV8L0UFjLLquvLl2rwIgdPLY=&lt;/ds:DigestValue&gt;&lt;/
      XSL2Digest&gt;</text>
166   </message>
167 </templateentry>
168 </template>
169 </ds:Object>
170 <ds:Object Id="Object-c7b5c8fc-2">
171   <xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Target="#"
      signature-1-1">
172     <xades:SignedProperties Id="SignedProperties-c7b5c8fc-1">
173       <xades:SignedSignatureProperties>
174         <xades:SigningTime>2013-11-13T10:24:47Z</xades:SigningTime>
175         <xades:SigningCertificate>
176           <xades:Cert>
177             <xades:CertDigest>
178               <xades:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
179               <xades:DigestValue>UwQcXi5kqUS3l22iPE35RMcdQB010bo13FILwB9C7HA=</
                  xades:DigestValue>
180             </xades:CertDigest>
181             <xades:IssuerSerial>
182               <ds:X509IssuerName>CN=originator,OU=IAIK,O=TU Graz,C=AT</ds:X509IssuerName>
183               <ds:X509SerialNumber>19272052632667013612905612204028238895
184               0573981001023456762351981992857059753962749966179</ds:X509SerialNumber>
185             </xades:IssuerSerial>
186           </xades:Cert>
187           </xades:SigningCertificate>
188           <xades:SignaturePolicyIdentifier>
189             <xades:SignaturePolicyImplied/>
190           </xades:SignaturePolicyIdentifier>
191         </xades:SignedSignatureProperties>
192         <xades:SignedDataObjectProperties>
193           <xades:DataObjectFormat ObjectReference="#"Reference-c7b5c8fc-1a">
194             <xades:MimeType>text/xml</xades:MimeType>
195           </xades:DataObjectFormat>
196         </xades:SignedDataObjectProperties>
197       </xades:SignedProperties>
198       <xades:UnsignedProperties Id="UnsignedProperties-c7b5c8fc-1">
199         <xades:UnsignedSignatureProperties>
200           <xades:CertificateValues>
201             <xades:OtherCertificate type="proxy">
202               <ds:X509Certificate>
                MIIBaTCCAR4CJGEzYjcyZDExLTEzNGQtNDJjYy04MTM5LWY4NTE0MGM5MzM3ZTAL
203 BgcqhkjOPQIBBQAwPjELMAkGA1UEBhMCQVQxEDAOBgNVBAoTB1RVIEdyYXoxDTAL
204 BgNVBAsTBElBSUsxZjAMBgNVBAMTBXByb3h5MB4XDTEyMDMwNDE1NDgyN1oXDTE0
205 MDMwNDE1NDgyN1owPjELMAkGA1UEBhMCQVQxEDAOBgNVBAoTB1RVIEdyYXoxDTAL
206 BgNVBAsTBElBSUsxZjAMBgNVBAMTBXByb3h5MEkwEwYHKoZIzj0CAQYIKoZIzj0D
207 AQEDMgAEHsc7i/LieOyFuAKfAj161kTeFeJHXwRuB0sVwMnKgaD/xG93zY70HiZW
208 bo5FzNanMAsGByqGSM49AgEFAAM4ADA1AhhLyhtEKLHy0EHPvCjI939pPJnIC72c
209 AH8CGQC7GRgd9XmCITQSPanKZHHLHpa/+AOhpOE=</ds:X509Certificate>
210             </xades:OtherCertificate>
211           </xades:CertificateValues>
212         </xades:UnsignedSignatureProperties>
213       </xades:UnsignedProperties>
214     </xades:QualifyingProperties>

```

```

215 </ds:Object>
216 <ds:Object Id="XSL1">
217   <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ds=
      "http://www.w3.org/2000/09/xmldsig#">
218     <xsl:template match="/">
219       <xsl:for-each select="//message/messageentry">
220         <xsl:if test="not(contains(current(),'xades')) and not(contains(current(),'
          XSL1Digest')) and not(contains(current(),'XSL2Digest'))">
221           <xsl:value-of select="current()" disable-output-escaping="yes"/>
222         </xsl:if>
223       </xsl:for-each>
224     </xsl:template>
225   </xsl:stylesheet>
226 </ds:Object>
227 <ds:Object Id="XSL2">
228   <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
229     <xsl:template match="@*|node()">
230       <xsl:copy>
231         <xsl:apply-templates select="@*|node()" />
232       </xsl:copy>
233     </xsl:template>
234     <xsl:template match="value">
235       <xsl:value-of select="."/>
236     </xsl:template>
237   </xsl:stylesheet>
238 </ds:Object>
239 </ds:Signature>

```

A.2 Message Signature (enveloping)

Listing A.2: Message signature (enveloping)

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ds:Signature Id="signature-1-1" xsi:schemaLocation="http://www.w3.org/2000/09/xmldsig#
  xmldsig-core-schema.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://
  www.w3.org/2001/XMLSchema-instance" xmlns:bdss="urn:bdss">
3   <ds:SignedInfo>
4     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
5     <ds:SignatureMethod Algorithm="http://www.iaik.tugraz.at/bdss#ECDSAwithSHA256"/>
6     <ds:Reference Id="Reference-c7b5c8fc-1a" URI="#Object-c7b5c8fc-1">
7       <ds:Transforms>
8         <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
9           <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect">id("
            Object-c7b5c8fc-1")/node()/XPath>
10          </ds:Transform>
11        </ds:Transforms>
12        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
13        <ds:DigestValue/>
14      </ds:Reference>
15      <ds:Reference Id="Reference-c7b5c8fc-1b" Type="http://uri.etsi.org/01903#
        SignedProperties" URI="#SignedProperties-c7b5c8fc-1">
16        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
17        <ds:DigestValue>9eYIbbVadV0n2Tt59UPKncjuf5v4vm8OidWldfE7iAU=</ds:DigestValue>
18      </ds:Reference>
19      <ds:Reference Id="XSL1R" URI="#XSL1">
20        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
21        <ds:DigestValue>sNPJMcbCV9McmC6PJcMHVDggr0HawN36UK7UVrOG6HI=</ds:DigestValue>
22      </ds:Reference>
23      <ds:Reference Id="XSL2R" URI="#XSL2">
24        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

```

```

25     <ds:DigestValue>Wpeblm7G4m5eFsc2KafkV8L0UFjLLquvLl2rwIgdPLY=</ds:DigestValue>
26   </ds:Reference>
27 </ds:SignedInfo>
28   <ds:SignatureValue>BgAAAY8EAAAAM2IAAABJYQAAACB3Kpp2u7yQdB2wToM7ydpWpq8+dgHk00RKppsL
29 fJQQLAAAACBkiVoOmUSOHdznMFFLGH3Ui5gOKXWmB4Wpze4Bh+NoKQAAAElhAAAA
30 ICJ/ffnIco9YRyp1C6MKH48oq2kZpragZ++kTQtnTC06AAAAIH6GtqV48i7XWObW
31 mcncyJR02c96zQGdKCy74BrygitcAAAA63IAAAAhAyZrBKJmHxYgESS4xtltKhxy
32 p6Mwytd6sJuMaaGMKougAAAAngIAAABJYQAAACA0mKxplBpmaYKS0xmRVXELVD0p
33 nUmXvEs8Xy77T5HBvgAAACB30z+jt2UWiTbjCalox/FMRFOD0q/Y1QsCX6GIhpLk
34 TAAAAEphAAAAIHM8eQeJV08LU511C4tAjRzzGxzgWLBp+niv/ho8KeHbAAAAIQCO
35 dEWZ+gzMmYEQJCN1Tqmj52tWV/aRadfkMlVyb+6u6gAAACECA/86CVbWfRysuGIJ
36 hJJ+S5UzdZcWlaxjOyDssDb2+N8AAAA3MDUCGQDpImXh+6OJ3snUivLfJrlBhvpO
37 UW9h4zYCGFLw39GQDh3yHZGAexzOmxPv1YeWpPJKfwAAAakDAAABaQIAAAAEdGVz
38 dAAAAEWAAAAIQCEzHph69nPOQ/lhnUagsDyb/iGEGCx2YHDO00/Z0WOWAAACZx
39 AAAAIQJ+Pgk37gPZ8GDe2WlBucvd3ODYev7LqPZWP38NABnHBgAAABkWAAAAFMfh
40 Pg0IeY56mNfxZnzfIemYIrfTAAAA63IAAAAhAlbrZf27q5iRdNzk22DwcQqK9tXj
41 4sqRy4TUMu9GJDB1AAAANGIAAABJYQAAACA9AFYSYchrEi2OuN3LeKwJUKtdrLP3
42 kD1480VtjjRheQAAACBMM2ZYrnadhDj98p5uiOAx6JFYU0AHGMocHwnKwycMwAA
43 AEphAAAAIH790EsdVozySuYXcCCT8XVYSHzfpvt6458nqrW+DxakAAAAIQCjvxdg
44 /soN7XqY8PadNcdW5b1AWYDXqSiLL7/j9Fkn7AAAACECWHgqsjXl105bUyS9MdTk
45 4hW+MdD6USPiwX3Alfyfz1sAAAA3MDUCGQCoxmZyVGB60H1Cn9B0CbFj3JnfZ8uf
46 jUICGEw+VNF1wBDfw9xZzgLcyztMT3gpC7mu5Q==</ds:SignatureValue>
47   <ds:KeyInfo>
48     <ds:X509Data>
49       <ds:X509Certificate>MIIBATCCAR4CJGEzYjcyZDExLTeZNGQtNDJjYy04MTM5LWY4NTE0MG5MzZTAL
50 BgcqhkjOPQIBBQAwPjELMAKGA1UEBhMCQVQxEDA0BgNVBAoTB1RVIEdyYXoxDTAL
51 BgNVBA8TBELBSUsxDjAMBgNVBAMTBXByb3h5MmB4XDTEyMDMwNDE1NDgyN1oXDTE0
52 MDMwNDE1NDgyN1owPjELMAKGA1UEBhMCQVQxEDA0BgNVBAoTB1RVIEdyYXoxDTAL
53 BgNVBA8TBELBSUsxDjAMBgNVBAMTBXByb3h5MmB4XDTEyMDMwNDE1NDgyN1oXDTE0
54 AQEDMgAEHsc7i/LieOyFuAKfAj161kTeFeJHXwRuB0sVwMnKgAD/xG93zY70HiZW
55 bo5FzNanMASGByqGSM49AgEFAAM4ADA1AhhLyhtEKLHy0EHPvCjI939pPjNc7c
56 AH8CGQC7GRgd9XmCItQSPankZHH1Hpa/+AOHpOE=</ds:X509Certificate>
57     </ds:X509Data>
58   </ds:KeyInfo>
59   <ds:Object Id="Object-c7b5c8fc-1">
60     <message id="test">
61       <messageentry type="fix" length="0">
62         <text>&lt;as:Assertion xmlns:as="urn:as" />&gt;</text>
63       </messageentry>
64       <messageentry type="fix" length="0">
65         <text>&lt;as:Person />&gt;</text>
66       </messageentry>
67       <messageentry type="fix" length="0">
68         <text>&lt;as:Name />&gt;</text>
69       </messageentry>
70       <messageentry type="fix" length="0">
71         <text>&lt;as:GivenName />&gt;</text>
72       </messageentry>
73       <messageentry type="exch">
74         <text>&lt;value />&gt;&lt;/value />&gt;</text>
75       </messageentry>
76       <messageentry type="fix" length="0">
77         <text>&lt;/as:GivenName />&gt;</text>
78       </messageentry>
79       <messageentry type="fix" length="0">
80         <text>&lt;as:FamilyName />&gt;</text>
81       </messageentry>
82       <messageentry type="blank" length="100">
83         <text>&lt;value />&gt;&lt;/value />&gt;</text>
84       </messageentry>
85       <messageentry type="fix" length="0">
86         <text>&lt;/as:FamilyName />&gt;</text>
87       </messageentry>
88       <messageentry type="fix" length="0">

```



```

89     <text>&lt;/as:Name&gt;</text>
90 </messageentry>
91 <messageentry type="fix" length="0">
92     <text>&lt;/as:DateOfBirth&gt;</text>
93 </messageentry>
94 <messageentry type="fix" length="0">
95     <text>&lt;/value&gt;11.08.1984&lt;/value&gt;</text>
96 </messageentry>
97 <messageentry type="fix" length="0">
98     <text>&lt;/as:DateOfBirth&gt;</text>
99 </messageentry>
100 <messageentry type="fix" length="0">
101     <text>&lt;/as:IdNumber&gt;</text>
102 </messageentry>
103 <messageentry type="exch">
104     <text>&lt;/value&gt;*&lt;/value&gt;</text>
105 </messageentry>
106 <messageentry type="fix" length="0">
107     <text>&lt;/as:IdNumber&gt;</text>
108 </messageentry>
109 <messageentry type="fix" length="0">
110     <text>&lt;/as:Person&gt;</text>
111 </messageentry>
112 <messageentry type="fix" length="0">
113     <text>&lt;/as:Assertion&gt;</text>
114 </messageentry>
115 <messageentry type="fix" length="0">
116     <text>&lt;/xadesDigestTemplate&gt;&lt;/ds:DigestMethod Algorithm=&quot;http://www.w3.
      org/2001/04/xmlenc#sha256&quot;/&gt;&lt;/ds:DigestValue&gt;
      imjdswwF9XFxhWoiZuf9lqo6Ma2KEHT3Oo3qLg+9S0c=&lt;/ds:DigestValue&gt;&lt;/
      xadesDigestTemplate&gt;</text>
117 </messageentry>
118 <messageentry type="blank" length="5000">
119     <text>&lt;/xades:SignedProperties Id=&quot;SignedProperties-c7b5c8fc-1&quot;&gt;&lt;/
      xades:SignedSignatureProperties&gt;&lt;/xades:SigningTime&gt;2013-11-13T10:24:49Z&
      lt;/xades:SigningTime&gt;&lt;/xades:SigningCertificate&gt;&lt;/xades:Cert&gt;&lt;/
      xades:CertDigest&gt;&lt;/xades:DigestMethod Algorithm=&quot;http://www.w3.org
      /2001/04/xmlenc#sha256&quot;/&gt;&lt;/xades:DigestValue&gt;
      gtVO3eSovYaU2l0ZokIRcuG3X7hRHVxFLgKv50m2HI=&lt;/xades:DigestValue&gt;&lt;/
      xades:CertDigest&gt;&lt;/xades:IssuerSerial&gt;&lt;/ds:X509IssuerName&gt;CN=proxy,
      OU=IAIK,O=TU Graz,C=AT&lt;/ds:X509IssuerName&gt;&lt;/ds:X509SerialNumber&gt
      ;18882880999756117483064540647715700545514847411348484536
120 0852062343150228966058260379493&lt;/ds:X509SerialNumber&gt;&lt;/xades:IssuerSerial&gt;&lt;/
      xades:Cert&gt;&lt;/xades:SigningCertificate&gt;&lt;/xades:SignaturePolicyIdentifier&gt;&lt;/
      xades:SignaturePolicyImplied/&gt;&lt;/xades:SignaturePolicyIdentifier&gt;&lt;/
      xades:SignedSignatureProperties&gt;&lt;/xades:SignedDataObjectProperties&gt;&lt;/
      xades:DataObjectFormat ObjectReference=&quot;#Reference-c7b5c8fc-1a&quot;&gt;&lt;/
      xades:MimeType&gt;text/xml&lt;/xades:MimeType&gt;&lt;/xades:DataObjectFormat&gt;&lt;/
      xades:SignedDataObjectProperties&gt;&lt;/xades:SignedProperties&gt;</text>
121 </messageentry>
122 <messageentry type="fix" length="0">
123     <text>&lt;/XSL1Digest&gt;&lt;/ds:DigestMethod Algorithm=&quot;http://www.w3.org
      /2001/04/xmlenc#sha256&quot;/&gt;&lt;/ds:DigestValue&gt;
      sNPJMcbCV9McmC6PJcMHVDggr0HawN36UK7UVrOG6HI=&lt;/ds:DigestValue&gt;&lt;/
      XSL1Digest&gt;</text>
124 </messageentry>
125 <messageentry type="fix" length="0">
126     <text>&lt;/XSL2Digest&gt;&lt;/ds:DigestMethod Algorithm=&quot;http://www.w3.org
      /2001/04/xmlenc#sha256&quot;/&gt;&lt;/ds:DigestValue&gt;
      Wpeblm7G4m5eFsc2KafkV8L0UFjLLquvLl2rwIgdPLY=&lt;/ds:DigestValue&gt;&lt;/
      XSL2Digest&gt;</text>
127 </messageentry>
128 </message>
129 </ds:Object>

```

```

130 <ds:Object Id="Object-c7b5c8fc-2">
131   <xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Target="#"
      signature-1-1">
132     <xades:SignedProperties Id="SignedProperties-c7b5c8fc-1">
133       <xades:SignedSignatureProperties>
134         <xades:SigningTime>2013-11-13T10:24:49Z</xades:SigningTime>
135         <xades:SigningCertificate>
136           <xades:Cert>
137             <xades:CertDigest>
138               <xades:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
139               <xades:DigestValue>gtVO3eSovYaU2l0ZokIRcuG3X7hRHVVxFLgKv50m2HI=</
                  xades:DigestValue>
140             </xades:CertDigest>
141             <xades:IssuerSerial>
142               <ds:X509IssuerName>CN=proxy,OU=IAIK,O=TU Graz,C=AT</ds:X509IssuerName>
143               <ds:X509SerialNumber>188828809997561174830645406477157005455148474
144                 113484845360852062343150228966058260379493</ds:X509SerialNumber>
145             </xades:IssuerSerial>
146           </xades:Cert>
147         </xades:SigningCertificate>
148         <xades:SignaturePolicyIdentifier>
149           <xades:SignaturePolicyImplied/>
150         </xades:SignaturePolicyIdentifier>
151       </xades:SignedSignatureProperties>
152       <xades:SignedDataObjectProperties>
153         <xades:DataObjectFormat ObjectReference="#"Reference-c7b5c8fc-1a">
154           <xades:MimeType>text/xml</xades:MimeType>
155         </xades:DataObjectFormat>
156       </xades:SignedDataObjectProperties>
157     </xades:SignedProperties>
158     <xades:UnsignedProperties Id="UnsignedProperties-c7b5c8fc-1">
159       <xades:UnsignedSignatureProperties>
160         <xades:CertificateValues>
161           <xades:OtherCertificate type="originator">
162             <ds:X509Certificate>
163               MIIBcjCCASgCJGM00WViNjY0LTZjMmQtNGIwZS04YWQxLTcyYzFlZDAyMmY3YzAL
164               BgcqhkjOPQIBBQAwQzELMAkGA1UEBhMCQVQxEDAoBgNVBAoTB1RVEEYXoxDTAL
165               BgNVBAsTBElBSUsxEzARBgNVBAMTCm9yaWdpbmF0b3IwHhcNMTIwMzA0MTU0ODI2
166               WhcNMTQwMzA0MTU0ODI2WjBDMQswCQYDVQQGEwJBVDEQMA4GA1UEChMHVFUgR3Jh
167               eJENMAsGALUECxMESUFJszETMBEgAlUEAxMKb3JpZ2luYXRvcjBjBjBjBjBjBjBjBjBj
168               AgEGCCqGSM49AwEBAzIABHlqSrz0FvMG7TuxVmG1T6pEJi+mP2+QO+fe9VxG6q8r
169               2fx98A8hy8gYKw3pfgRGBzALBgcqhkjOPQIBBQADNwAwNAIYLHhulrvuaGhXb172
170               FayNvzUXqtIzTjKAAhhdYwvpJv1rB8o3g9k84miMf78VP4La4Pk=</ds:X509Certificate>
171           </xades:OtherCertificate>
172         </xades:CertificateValues>
173       </xades:UnsignedSignatureProperties>
174     </xades:UnsignedProperties>
175   </xades:QualifyingProperties>
176 </ds:Object>
177 <ds:Object Id="XSL1">
178   <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:ds=
      "http://www.w3.org/2000/09/xmldsig#">
179     <xsl:template match="/">
180       <xsl:for-each select="//message/messageentry">
181         <xsl:if test="not(contains(current(),'xades')) and not(contains(current(),'
              XSL1Digest')) and not(contains(current(),'XSL2Digest'))">
182           <xsl:value-of select="current()" disable-output-escaping="yes"/>
183         </xsl:if>
184       </xsl:for-each>
185     </xsl:template>
186   </xsl:stylesheet>
187 </ds:Object>
188 <ds:Object Id="XSL2">
189   <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

```

```
189     <xsl:template match="@*|node() ">
190       <xsl:copy>
191         <xsl:apply-templates select="@*|node() "/>
192       </xsl:copy>
193     </xsl:template>
194     <xsl:template match="value">
195       <xsl:value-of select="."/>
196     </xsl:template>
197   </xsl:stylesheet>
198 </ds:Object>
199 </ds:Signature>
```

A.3 Final XSL Transformed Result

Listing A.3: Final XSL transformed result

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <as:Assertion xmlns:as="urn:as">
3   <as:Person>
4     <as:Name>
5       <as:GivenName>*</as:GivenName>
6       <as:FamilyName>*</as:FamilyName>
7     </as:Name>
8     <as:DateOfBirth>11.08.1984</as:DateOfBirth>
9     <as:IdNumber>*</as:IdNumber>
10   </as:Person>
11 </as:Assertion>
```


Appendix B

Identity Management Example

B.1 Example Identity Link

Listing B.1: Example identity link

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:dsig="http://www.w3
   .org/2000/09/xmldsig#" xmlns:ecdsa="http://www.w3.org/2001/04/xmldsig-more#" xmlns:pr="
   http://reference.e-government.gv.at/namespace/persondata/20020228#" xmlns:si="http://www.
   w3.org/2001/XMLSchema-instance" AssertionID="szo.bmi.gv.at-AssertionID1376408486094522"
   IssueInstant="2013-08-13T17:41:26+01:00" Issuer="http://portal.bmi.gv.at/ref/szo/issuer"
   MajorVersion="1" MinorVersion="0">
3 <saml:AttributeStatement>
4   <saml:Subject>
5     <saml:SubjectConfirmation>
6       <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</
         saml:ConfirmationMethod>
7       <saml:SubjectConfirmationData>
8         <pr:Person si:type="pr:PhysicalPersonType">
9           <pr:Identification>
10            <pr:Value>Gq03dPrgcHsx3G01ZDH6SQ==</pr:Value>
11            <pr:Type>urn:publicid:gv.at:baseid</pr:Type>
12          </pr:Identification>
13          <pr:Name>
14            <pr:GivenName>Max</pr:GivenName>
15            <pr:FamilyName primary="undefined">Mustermann</pr:FamilyName>
16          </pr:Name>
17          <pr:DateOfBirth>1965-03-24</pr:DateOfBirth>
18        </pr:Person>
19      </saml:SubjectConfirmationData>
20    </saml:SubjectConfirmation>
21  </saml:Subject>
22  <saml:Attribute AttributeName="CitizenPublicKey" AttributeNamespace="urn:publicid:gv.
     at:namespaces:identitylink:1.2">
23    <saml:AttributeValue>
24      <ecdsa:ECDSAKeyValue>
25        <ecdsa:DomainParameters>
26          <ecdsa:NamedCurve URN="urn:oid:1.2.840.10045.3.1.7"/>
27        </ecdsa:DomainParameters>
28        <ecdsa:PublicKey>
29          <ecdsa:X Value="
            215487815123484636246655129318100029877610067331164103666990893810317945950596
            " si:type="ecdsa:PrimeFieldElemType"/>
```

```

30         <ecdsa:Y Value="
31             32548183375132839567164482875181817864226552443687973143696540267521109533636
32             " si:type="ecdsa:PrimeFieldElemType"/>
33         </ecdsa:PublicKey>
34     </ecdsa:ECDSAKeyValue>
35 </saml:AttributeValue>
36 </saml:Attribute>
37 <saml:Attribute AttributeName="CitizenPublicKey" AttributeNamespace="urn:publicid:gv.
38     at:namespaces:identitylink:1.2">
39     <saml:AttributeValue>
40     <dsig:RSAKeyValue>
41     <dsig:Modulus>zb54fuGZwEvlNEmEGZSSJNT6bDwk6ONn/xjtjK5mnZljCQiWeolJ/
42     c5hFGkuou9xGB3MjGEBTnp
43     XsvNE8afJniKQnGE2CjRNhXKDRGld3bPtsBF4YUVzqJKg6cn/hU4scUBk1qp/4LA5oUE5aEvvMMw
44     bPGA7Cods5PrM/f/6l3h/qp6hBGRwQvV5rqmZL3WJf0sPZFursYjW/gBgEZj48n6uIGfrYTEweET
45     jjAbNJ9TSZtWbhff4IFnfEfnqtN</dsig:Modulus>
46     <dsig:Exponent>AQCB</dsig:Exponent>
47     </dsig:RSAKeyValue>
48 </saml:AttributeValue>
49 </saml:Attribute>
50 </saml:AttributeStatement>
51 <dsig:Signature>
52     <dsig:SignedInfo>
53     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
54     <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
55     <dsig:Reference URI="">
56     <dsig:Transforms>
57     <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
58     <dsig:XPath>not (ancestor-or-self::pr:Identification)</dsig:XPath>
59     </dsig:Transform>
60     <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
61     </dsig:Transforms>
62     <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
63     <dsig:DigestValue>eVjZE9Vgafz7CfRKbUmQc1Q7VWw=</dsig:DigestValue>
64 </dsig:Reference>
65 <dsig:Reference Type="http://www.w3.org/2000/09/xmldsig#Manifest" URI="#manifest">
66     <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
67     <dsig:DigestValue>NVdTfsoQnwrRqCHqf4hWWD8I24Y=</dsig:DigestValue>
68 </dsig:Reference>
69 </dsig:SignedInfo>
70 <dsig:SignatureValue>
71     EswDIDte26wNzyOWWUfkrZq3XRSbMgUBblTs jh4Sz9wfDZNQ1CbJy2J7XfDLqxeEpJ4BHJHxEP90++
72     Doj0TdbXyNPLF3R2TRuhVB3XtTY5WsQscstqngjs4D7E5P+PRONAI50RNF'PB0aqhqsx1bblj kptMKP3BR/
73     MSuIRxfA9X4QZIEUswouYo
74     dQ0Fgmz1BH4xUJlNb7exaRQiB29B8FiKU735Rjsl4AqIa93NMxUzmnsXTwCtGwy6l2JvSpX/8
75     fQWVw9pbP0glTGXT9S3GqNBzvSuC7
76     FNuh/ZQXK5OgRyCjGj1UT57vfA4zKJOve7MwCI5Zao9ih+6M9S1IFKvg==
77 </dsig:SignatureValue>
78 <dsig:KeyInfo>
79     <dsig:X509Data>
80     <dsig:X509Certificate>MIIF3TCCBMWgAwIBAgIDByniMA0GCSqGSIb3DQEBBQUAMIGfMQswCQYDVQQG
81     EwJBVDFIMEYGA1UECgw/QS1UcnVzdCBH2XMuIGYuIFNpY2h1cmhlaXRzc3lzdGVtZSBpbSB1bGVrdHlueIE
82     RhdGVudmVya2VociBhbWJIMSiWIAyDVQQQLDBlhlXNpZ24tY29ycG9yYXRlLWxpZ2h0LTAYMSiWIAyDVQQD
83     DBlhlXNpZ24tY29ycG9yYXRlLWxpZ2h0LTAYMB4XDTEwMDcyODE1MDcyODE1MDcyODE1MDcyODE1MDcyODE1
84     OXNpZ24tY29ycG9yYXRlLWxpZ2h0LTAYMjEwMDcyODE1MDcyODE1MDcyODE1MDcyODE1MDcyODE1MDcyODE1
85     VS/1r5sWcra9Hhdm7w5Gtx/2ukyDX0kdkxawkhP4EQEzi/SI+Fugn+WqgQ1nAdlxbx/dcBw5w1h9b3lmuw
86     Uf4z3ooQWUD2DgA/kKd1KejNR43mLUsmvSzevPxT9zs78pOR1OacB7IszTVJPXeOEaaNZHnnB/UeO3g8LE
87     V/30kXcUgcMkbIiiaBH11171Pq0COj9kqjXoe70rRjLY5i3KwOpa6TMCAwEAAOAQAgcwggIDMBMGAlUdIw
88     QMMAQAcEkcWDP6A0DMH8GCCsGAQUFBwEBBHMwTANBggrBgEFBQcwAYYbaHR0cDovL29jc3AuYS10cnVz

```

```

86      dC5hdC9vY3NwMEYGCCsGAQUFBzACHjpodHRwOi8vd3d3LmEt dHJ1c3QuYXQvY2VydHMvYS1zaWduLWNvcn
87      BvcnF0ZS1saWdodC0wMmEuY3J0MFQGA1UdIARNMESwSQYgKigAEQESMD8wPQYIKwYBBQUHAgEWMWh0dHA6
88      Ly93d3cuYS10cnVzdC5hdC9kb2NzL2NwL2Etc2lmbi1BbXRzc2lnbmF0dXIwZ4G4A1UdHwSBljCBkzCBkK
89      CBjaCBioaBh2xkYXA6Ly9sZGFwLmEt dHJ1c3QuYXQvY3U9YS1zaWduLWNvcnBvcnF0ZS1saWdodC0wMixv
90      PUEtVHJ1c3QsYz1BVD9jZlZlZ0aWZpY2F0ZXJldm9jYXRpb25saXNOP2Jhc2U/b2JqZWN0Y2xhc3M9ZWlkQ2
91      VydGhmaWNhdGlvbkF1dGhvcml0eTARBgNVHQ4ECgQITAgOnhr0tbowDgYDVR0PAQH/BAQDAgSwMCAgA1Ud
92      EQQZMBEhFWlhcmN1cy5oaWxkQGRzay5ndi5hdDAJBgNVHRMEAjaAMA4GByooAAoBBwEEAwEB/zAUBgcqKA
93      AKAQEBAkMB0JTQi1EU0swDQYJKoZIhvcNAQEFBQADggEBAHTklvPCH/bJSOLIPbLUEkSguFHsektSZ8V
94      r22x/Yv7EzsxoQrJIiz2mQ2gQqFuExdWYxvsowjiSbiis9iUf1c0zscvDS3mIZxGs4M89XHs jHnIyb+Fuw
95      namw65QrFvM1tNB1ZMjxJ3x+YmHLHdtT3BEBcr3/NCRHd2S0HoBspNz9HVgJaZy111R7poKBvnAc4gli+Q
96      TvyVb00PtKxR9Lw/9ABInX/lpZpxqrPy7Ib2OP8z6dd3WHmIsCiSHUaj0Dxwvln6fYJjhxZ141SnbovlCL
97      YtrsZLXoi9ljIqX4x00PwMI2RfNc9cXxTRrRS6rEOvX7Ppvg
98      XiDXhp592Yyp4=</dsig:X509Certificate>
99      </dsig:X509Data>
100     </dsig:KeyInfo>
101     <dsig:Object>
102       <dsig:Manifest Id="manifest">
103         <dsig:Reference URI="">
104           <dsig:Transforms>
105             <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
106               <dsig:XPath>not(ancestor-or-self::dsig:Signature)</dsig:XPath>
107             </dsig:Transform>
108           </dsig:Transforms>
109           <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
110           <dsig:DigestValue>l3202ZiTWGc3kF2+BJ8t733653U=</dsig:DigestValue>
111         </dsig:Reference>
112       </dsig:Manifest>
113     </dsig:Object>
114   </dsig:Signature>
115 </saml:Assertion>

```

B.2 Example Identity Link*

Listing B.2: Example identity link* (Template)

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <template id="test">
3  <templateentry><message type="fix" length="0">
4  <text>&lt;saml:Assertion AssertionID=&quot;sizr.bmi.gv.at-AssertionID13815189324521476&quot;
      IssueInstant=&quot;2013-10-11T21:15:32+01:00&quot; Issuer=&quot;http://portal.bmi.gv.at/
      ref/sizr/issuer&quot; MajorVersion=&quot;1&quot; MinorVersion=&quot;0&quot; xmlns:dsig=&
      quot;http://www.w3.org/2000/09/xmldsig#&quot; xmlns:ecdsa=&quot;http://www.w3.org
      /2001/04/xmldsig-more#&quot; xmlns:pr=&quot;http://reference.e-government.gv.at/namespaces
      /persondata/20020228#&quot; xmlns:saml=&quot;urn:oasis:names:tc:SAML:1.0:assertion&quot;
      xmlns:si=&quot;http://www.w3.org/2001/XMLSchema-instance&quot;&gt;&lt;/text>
5  </message>
6  </templateentry>
7  <templateentry><message type="fix" length="0">
8  <text>&lt;saml:AttributeStatement&gt;&lt;/text>
9  </message>
10 </templateentry>
11 <templateentry><message type="fix" length="0">
12 <text>&lt;saml:Subject&gt;&lt;/text>
13 </message>
14 </templateentry>
15 <templateentry><message type="fix" length="0">
16 <text>&lt;saml:SubjectConfirmation&gt;&lt;/text>
17 </message>
18 </templateentry>
19 <templateentry><message type="fix" length="0">

```

```

20 <text>&lt;saml:ConfirmationMethod&gt;</text>
21 </message>
22 </templateentry>
23 <templateentry><message type="fix" length="0">
24 <text>&lt;value&gt;urn:oasis:names:tc:SAML:1.0:cm:sender-vouches&lt;/value&gt;</text>
25 </message>
26 </templateentry>
27 <templateentry><message type="fix" length="0">
28 <text>&lt;/saml:ConfirmationMethod&gt;</text>
29 </message>
30 </templateentry>
31 <templateentry><message type="fix" length="0">
32 <text>&lt;saml:SubjectConfirmationData&gt;</text>
33 </message>
34 </templateentry>
35 <templateentry><message type="fix" length="0">
36 <text>&lt;pr:Person si:type="&quot;pr:PhysicalPersonType&quot;&gt;</text>
37 </message>
38 </templateentry>
39 <templateentry><message type="fix" length="0">
40 <text>&lt;pr:Identification&gt;</text>
41 </message>
42 </templateentry>
43 <templateentry><message type="fix" length="0">
44 <text>&lt;pr:Value&gt;</text>
45 </message>
46 </templateentry>
47 <templateentry><message type="exch">
48 <text>&lt;value&gt;Qq03dPrgcHsx3G01KSH6SQ==&lt;/value&gt;</text>
49 </message>
50 <message type="exch">
51 <text>&lt;value&gt;*&lt;/value&gt;</text>
52 </message>
53 </templateentry>
54 <templateentry><message type="fix" length="0">
55 <text>&lt;/pr:Value&gt;</text>
56 </message>
57 </templateentry>
58 <templateentry><message type="fix" length="0">
59 <text>&lt;pr:Type&gt;</text>
60 </message>
61 </templateentry>
62 <templateentry><message type="fix" length="0">
63 <text>&lt;value&gt;urn:publicid:gv.at:baseid&lt;/value&gt;</text>
64 </message>
65 </templateentry>
66 <templateentry><message type="fix" length="0">
67 <text>&lt;/pr:Type&gt;</text>
68 </message>
69 </templateentry>
70 <templateentry><message type="fix" length="0">
71 <text>&lt;/pr:Identification&gt;</text>
72 </message>
73 </templateentry>
74 <templateentry><message type="fix" length="0">
75 <text>&lt;pr:Name&gt;</text>
76 </message>
77 </templateentry>
78 <templateentry><message type="fix" length="0">
79 <text>&lt;pr:GivenName&gt;</text>
80 </message>
81 </templateentry>
82 <templateentry><message type="exch">
83 <text>&lt;value&gt;Max&lt;/value&gt;</text>

```



```
84 </message>
85 <message type="exch">
86 <text>&lt;value&gt;*&lt;/value&gt;</text>
87 </message>
88 </templateentry>
89 <templateentry><message type="fix" length="0">
90 <text>&lt;/pr:GivenName&gt;</text>
91 </message>
92 </templateentry>
93 <templateentry><message type="fix" length="0">
94 <text>&lt;/pr:FamilyName primary=&quot;undefined&quot;&gt;</text>
95 </message>
96 </templateentry>
97 <templateentry><message type="exch">
98 <text>&lt;value&gt;Mustermann&lt;/value&gt;</text>
99 </message>
100 <message type="exch">
101 <text>&lt;value&gt;*&lt;/value&gt;</text>
102 </message>
103 </templateentry>
104 <templateentry><message type="fix" length="0">
105 <text>&lt;/pr:FamilyName&gt;</text>
106 </message>
107 </templateentry>
108 <templateentry><message type="fix" length="0">
109 <text>&lt;/pr:Name&gt;</text>
110 </message>
111 </templateentry>
112 <templateentry><message type="fix" length="0">
113 <text>&lt;/pr:DateOfBirth&gt;</text>
114 </message>
115 </templateentry>
116 <templateentry><message type="exch">
117 <text>&lt;value&gt;1984-10-15&lt;/value&gt;</text>
118 </message>
119 <message type="exch">
120 <text>&lt;value&gt;*&lt;/value&gt;</text>
121 </message>
122 </templateentry>
123 <templateentry><message type="fix" length="0">
124 <text>&lt;/pr:DateOfBirth&gt;</text>
125 </message>
126 </templateentry>
127 <templateentry><message type="fix" length="0">
128 <text>&lt;/pr:Person&gt;</text>
129 </message>
130 </templateentry>
131 <templateentry><message type="fix" length="0">
132 <text>&lt;/saml:SubjectConfirmationData&gt;</text>
133 </message>
134 </templateentry>
135 <templateentry><message type="fix" length="0">
136 <text>&lt;/saml:SubjectConfirmation&gt;</text>
137 </message>
138 </templateentry>
139 <templateentry><message type="fix" length="0">
140 <text>&lt;/saml:Subject&gt;</text>
141 </message>
142 </templateentry>
143 <templateentry><message type="fix" length="0">
144 <text>&lt;saml:Attribute AttributeName=&quot;ssPINST&quot; AttributeNamespace=&quot;
    urn:publicid:gv.at:namespaces:identitylink:1.2&quot;&gt;</text>
145 </message>
146 </templateentry>
```

```

147 <templateentry><message type="fix" length="0">
148 <text>&lt;saml:AttributeValue&gt;</text>
149 </message>
150 </templateentry>
151 <templateentry><message type="exch">
152 <text>&lt;value&gt;uUevHSXjSG0rBnL3b6x4M/4TKbE=&lt;/value&gt;</text>
153 </message>
154 <message type="exch">
155 <text>&lt;value&gt;*&lt;/value&gt;</text>
156 </message>
157 </templateentry>
158 <templateentry><message type="fix" length="0">
159 <text>&lt;/saml:AttributeValue&gt;</text>
160 </message>
161 </templateentry>
162 <templateentry><message type="fix" length="0">
163 <text>&lt;/saml:Attribute&gt;</text>
164 </message>
165 </templateentry>
166 <templateentry><message type="fix" length="0">
167 <text>&lt;saml:Attribute AttributeName="ssPINBW" AttributeNamespace="
    urn:publicid:gv.at:namespaces:identitylink:1.2" &gt;</text>
168 </message>
169 </templateentry>
170 <templateentry><message type="fix" length="0">
171 <text>&lt;saml:AttributeValue&gt;</text>
172 </message>
173 </templateentry>
174 <templateentry><message type="exch">
175 <text>&lt;value&gt;mVZjLm46PDFaz7fCMzh9CU2Tf8=&lt;/value&gt;</text>
176 </message>
177 <message type="exch">
178 <text>&lt;value&gt;*&lt;/value&gt;</text>
179 </message>
180 </templateentry>
181 <templateentry><message type="fix" length="0">
182 <text>&lt;/saml:AttributeValue&gt;</text>
183 </message>
184 </templateentry>
185 <templateentry><message type="fix" length="0">
186 <text>&lt;/saml:Attribute&gt;</text>
187 </message>
188 </templateentry>
189 <templateentry><message type="fix" length="0">
190 <text>&lt;saml:Attribute AttributeName="ssPINPRIVAT" AttributeNamespace="
    urn:publicid:gv.at:namespaces:identitylink:1.2" &gt;</text>
191 </message>
192 </templateentry>
193 <templateentry><message type="fix" length="0">
194 <text>&lt;saml:AttributeValue&gt;</text>
195 </message>
196 </templateentry>
197 <templateentry><message type="blank" length="1000">
198 <text>&lt;value&gt;*&lt;/value&gt;</text>
199 </message>
200 </templateentry>
201 <templateentry><message type="fix" length="0">
202 <text>&lt;/saml:AttributeValue&gt;</text>
203 </message>
204 </templateentry>
205 <templateentry><message type="fix" length="0">
206 <text>&lt;/saml:Attribute&gt;</text>
207 </message>
208 </templateentry>

```

```

209 <templateentry><message type="fix" length="0">
210 <text>&lt;saml:Attribute AttributeName=&quot;OAInfos&quot; AttributeNamespace=&quot;
    urn:publicid:gv.at:namespaces:identitylink:1.2&quot;&gt;</text>
211 </message>
212 </templateentry>
213 <templateentry><message type="fix" length="0">
214 <text>&lt;saml:AttributeValue&gt;</text>
215 </message>
216 </templateentry>
217 <templateentry><message type="blank" length="1000">
218 <text>&lt;value&gt;*&lt;/value&gt;</text>
219 </message>
220 </templateentry>
221 <templateentry><message type="fix" length="0">
222 <text>&lt;/saml:AttributeValue&gt;</text>
223 </message>
224 </templateentry>
225 <templateentry><message type="fix" length="0">
226 <text>&lt;/saml:Attribute&gt;</text>
227 </message>
228 </templateentry>
229 <templateentry><message type="fix" length="0">
230 <text>&lt;/saml:AttributeStatement&gt;</text>
231 </message>
232 </templateentry>
233 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
234   <ds:SignedInfo>
235     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
236     <ds:SignatureMethod Algorithm="http://www.iaik.tugraz.at/bdss#ECDSAwithSHA256" />
237     <ds:Reference URI="">
238       <ds:Transforms>
239         <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
240       </ds:Transforms>
241       <ds:DigestMethod Algorithm="http://www.iaik.tugraz.at/bdss#ECDSAwithSHA256" />
242       <ds:DigestValue></ds:DigestValue>
243     </ds:Reference>
244     <ds:Reference Id="Reference-c7b5c8fc-1b" Type="http://uri.etsi.org/01903#
      SignedProperties" URI="#SignedProperties-c7b5c8fc-1">
245       <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
246       <ds:DigestValue>mGA3el/aT9Bvyj7jCCWl7gvvIJ4lQxorWufZPafW4P4=</ds:DigestValue>
247     </ds:Reference>
248   </ds:SignedInfo>
249   <ds:SignatureValue>AwAAAWgCAAAABHRLc3QAAAABoAAAAACEAwL8NDFdlotb234nHDKTDL5QR6MgBEQ8i
250 YMSMwdj/vAIAAAAmcQAAACECiq22NBIGkW0DUeaCb4gZeOmEQu5DS9D1jJJj5K4Y
251 YmYAAAZFGAAABTH4T4NCHmOepjX8WZ83yHpmCK37QAAAOPYAAAAIQOs3yCrk6dE
252 uopCrJPIYoG+wBivMspTAKv5HCKhins+qgAAAjtiAAAAASWEAAAAgUD/lQPXPsuwr
253 NOHQDtdta4qH/aiHskQeExfLx4cg0QAAAAGasInoYQZP9c65DAaZPHa+UeLLCAI
254 ePHaDNYgNm/srmlAAABJYQAAACAdp4CDCQ/+UcCQcY/c6kKpGt+neezlar3rR8Kh
255 OJ6EBAAAACBjy9XRjEtK7lmu0wuDeYk4G3fp4ALBatdG7qGciAmy0wAAACEDqf9p
256 Ykjz5KgLXI9I/bKgWRBTKUpNSzmqfDYfjdZ9FDQAAAA4MDYCGQC1Tz1v5KLonzf3
257 W75ZhtRfE/x9Djns5kACGQDlmeqqXujJmRRrgUEmUINMxKWNsMoJOSY=</ds:SignatureValue>
258   <ds:KeyInfo>
259     <ds:X509Data>
260       <ds:X509Certificate>MIIBcJCCASgCJGM0OWVinjY0LTZjMmQtNGIwZS04YWQxLTcyYzFlZDAyMmY3YzAL
261 BgcqhkjOPQIBBQAwQzELMAkGA1UEBhMCQVQxEDAOBgNVBAoTB1RVIEEduYXoxDTAL
262 BgNVBAStBE1BSUsxEzARBgNVBAMTCm9yaWdpbmF0b3IwHhcNMTIwMzA0MTU0ODI2
263 WhcNMTQwMzA0MTU0ODI2WjBDMQswCQYDVQQGEwJBVDEQMA4GA1UEEChMHVUgR3Jh
264 eJENMAsGA1UECjMESUFSZSFTMBEGA1UEEAxMKb3JpZ21uYXRvcjBjBjBMBGByqGSM49
265 AgEGCCqGSM49AwEBAzIABHlqSrZ0FvMG7TuxVmG1T6pEJi+mP2+QO+fe9VxG6q8r
266 2fx98A8hy8gYKw3pfqRGBzALBgcqhkjOPQIBBQADNwAwNAIYLHhulrvuaGhXb172
267 FayNvzUXqtIzTjKAAhhdYwppJVlrb8o3g9k84miMf78VP4La4Pk=</ds:X509Certificate>
268     </ds:X509Data>
269   </ds:KeyInfo>
270 </ds:Object Id="Object-c7b5c8fc-2">

```

```

271 <xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Target="#"
272 signature-1-1">
273 <xades:SignedProperties Id="SignedProperties-c7b5c8fc-1">
274 <xades:SignedSignatureProperties>
275 <xades:SigningTime>2014-03-14T10:25:53Z</xades:SigningTime>
276 <xades:SigningCertificate>
277 <xades:Cert>
278 <xades:CertDigest>
279 <xades:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
280 <xades:DigestValue>UwQCXi5kqUS3l22iPE35RMCdQB010bo13FILwB9C7HA=</
281 xades:DigestValue>
282 </xades:CertDigest>
283 <xades:IssuerSerial>
284 <ds:X509IssuerName>CN=originator,OU=IAIK,O=TU Graz,C=AT</ds:X509IssuerName>
285 <ds:X509SerialNumber>1927205263266701361290561220402823889505739810010234567
286 62351981992857059753962749966179</ds:X509SerialNumber>
287 </xades:IssuerSerial>
288 </xades:Cert>
289 </xades:SigningCertificate>
290 <xades:SignaturePolicyIdentifier>
291 <xades:SignaturePolicyImplied/>
292 </xades:SignaturePolicyIdentifier>
293 </xades:SignedSignatureProperties>
294 <xades:SignedDataObjectProperties>
295 <xades:DataObjectFormat ObjectReference="#Reference-c7b5c8fc-1a">
296 <xades:MimeType>text/xml</xades:MimeType>
297 </xades:DataObjectFormat>
298 </xades:SignedDataObjectProperties>
299 </xades:SignedProperties>
300 <xades:UnsignedProperties Id="UnsignedProperties-c7b5c8fc-1">
301 <xades:UnsignedSignatureProperties>
302 <xades:CertificateValues>
303 <xades:OtherCertificate type="proxy">
304 <ds:X509Certificate>
305 MIIBATCCAR4CJGEzYjcyZDEzLWVzNGQtdjYyO4MTM5LWY4NTE0MGM5MzZTAL
306 BgcqhkjOPQIBBQAwPjELMAkGA1UEBhMCQVQxEDAOBgNVBAMoTB1RVIEdyYXoxDTAL
307 BgNVBAsTBELBSUsxDjAMBgNVBAMTBXByb3h5MjB4XDTEyMDMwNDE1NDgyN1oXDTE0
308 MDMwNDE1NDgyN1owPjELMAkGA1UEBhMCQVQxEDAOBgNVBAMoTB1RVIEdyYXoxDTAL
309 BgNVBAsTBELBSUsxDjAMBgNVBAMTBXByb3h5MjB4XDTEyMDMwNDE1NDgyN1oXDTE0
310 AQEDMGAEHsc7i/LieOyFuAKfaj16lkTeFeJHXwRuB0sVwMnKgaD/xG93zY70HiZW
311 bo5FzNanMAsGByqGSM49AgEFAAM4ADA1AhhLyhtEKLHy0EHPvcjI939pPjNjIC72c
312 AH8CGQC7GRgd9XmCItQSPanKZHHlHpa/+AOHpOE=</ds:X509Certificate>
313 </xades:OtherCertificate>
314 </xades:CertificateValues>
315 </xades:UnsignedSignatureProperties>
316 </xades:UnsignedProperties>
317 </xades:QualifyingProperties>
318 </ds:Object>
319 </ds:Signature>
320 <templateentry>
321 <message type="fix" length="0">
322 <text>&lt;/saml:Assertion&gt;</text>
323 </message>
324 </templateentry>
325 <templateentry>
326 <message type="fix" length="0">
327 <text>&lt;xadesDigestTemplate&gt;&lt;ds:DigestMethod Algorithm="http://www.w3.org
328 /2001/04/xmlenc#sha256" /&gt;&lt;ds:DigestValue&gt;mGA3e1/
aT9Bvyj7jCCWl7gvvIJ4lQxorWufZPafW4P4=&lt;/ds:DigestValue&gt;&lt;/xadesDigestTemplate&gt;<
/text>
</message>
</templateentry>
<templateentry>
<message type="blank" length="5000">

```

```
329 <text></text>  
330 </message>  
331 </templateentry>  
332 </template>
```


Appendix C

Publications

This appendix overviews the publications of the thesis author. Overall the thesis author has 33 publications whereas 13 publications are directly related to the thesis. Following section C.1 gives all thesis related publications - separated into publications in conferences proceedings and publications in journals. Additionally, Section C.2 gives all other publication for the sake of completeness.

C.1 Thesis related Publications

C.1.1 Journals

Table C.1: Publications in journals

Publication	Author ^a	Thesis Chapter
[Tauber et al., 2012]	C	Chapter 3
[Stranacher et al., 2013d]	F	Chapter 5
[Stranacher et al., 2013a]	F	Chapters 5 and 8

^aF means first author and C means co-authors

C.1.2 Conference Proceedings

Table C.2: Publications at conferences

Publication	Author ^a	Thesis Chapter
[Stranacher and Kawecki, 2012]	F	Chapter 4
[Stranacher and Zwattendorfer, 2012]	F	Chapter 4
[Stranacher et al., 2012]	F	Chapters 5 and 8
[Stranacher and Zwattendorfer, 2013]	F	Chapters 5, 8 and 9
[Stranacher et al., 2013b]	F	Chapters 5 and 8
[Stranacher et al., 2013g]	F	Chapters 7 and 10
[Stranacher et al., 2013e]	F	Chapter 4
[Stranacher et al., 2013c]	F	Chapter 5
[Stranacher and Zwattendorfer, 2014]	F	Chapters 7 and 10
[Stranacher et al., 2014]	F	Chapter 8

^aF means first author and C means co-authors

C.2 Other Publications

Table C.3: Other publications

Publication	Author ^a
[Stranacher et al., 2009]	F
[Stranacher and Zwattendorfer, 2009]	F
[Stranacher, 2010]	F
[Zefferer et al., 2012]	C
[Zwattendorfer et al., 2012b]	C
[Zwattendorfer et al., 2012a]	C
[Krnjic et al., 2013]	C
[Zwattendorfer et al., 2013a]	C
[Lenz et al., 2013b]	C
[Tauber et al., 2013]	C
[Zwattendorfer et al., 2013b]	C
[Zwattendorfer et al., 2014b]	C
[Posch et al., 2012]	C
[Lenz et al., 2013a]	C
[Stranacher et al., 2013f]	C
[Zefferer et al., 2014]	C
[Stranacher et al., 2008]	F
[Tauber et al., 2011]	C
[Lenz et al., 2014]	C
[Slamanig et al., 2014]	C
[Zwattendorfer et al., 2014a]	C

^aF means first author and C means co-authors

Bibliography

- Altameem et al. [2006]. *Critical success factors of e-Government: A proposed model for e-Government implementation*. In *Innovations in Information Technology*, pages 1–5. (Cited on pages 43, 160 and 161.)
- Ateniese, Giuseppe, Daniel H Chou, Breno De Medeiros, and Gene Tsudik [2005]. *Sanitizable Signatures*. in *European Symposium on Research in Computer Security ESORICS*, 3679, pages 159–177. (Cited on pages 77, 83, 86, 88, 89, 90 and 91.)
- Ateniese, Giuseppe and Breno De Medeiros [2004]. *On the Key Exposure Problem in Chameleon Hashes*. *Proceedings of the Fourth Conference on Security in Communication Networks (SCN 2004)*, 3352, pages 1–16. (Cited on page 79.)
- Bartel, Mark, John Boyer, Barb Fox, Brian LaMacchia, and Ed Simon [2008]. *XML Signature Syntax and Processing (Second Edition)*, *W3C Recommendation*. (Cited on pages 19, 49, 84, 88 and 102.)
- Bauer, David, Douglas M. Blough, and Apurva Mohan [2009]. *Redactable signatures on data with dependencies and their application to personal health records*. *Proceedings of the 8th ACM workshop on Privacy in the electronic society - WPES '09*, pages 91–100. doi:10.1145/1655188.1655201. <http://portal.acm.org/citation.cfm?doid=1655188.1655201>. (Cited on page 83.)
- Bichsel, Patrik, Jan Camenisch, Thomas Groß, and Victor Shoup [2009]. *Anonymous credentials on a standard java card*. In *ACM CCS*, pages 600–610. ACM. (Cited on page 146.)
- Brands, Stefan [2000]. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press. (Cited on page 146.)
- Bray et al. [2006]. *Extensible Markup Language (XML) 1.1 (Second Edition)*. (Cited on pages 49 and 94.)
- Brzuska, Christina, Heike Busch, Oezguer Dagdelen, Marc Fischlin, Martin Franz, Stefan Katzenbeisser, Mark Manulis, Cristina Onete, Andreas Peter, Bertram Poettering, and Dominique Schröder [2010a]. *Redactable Signatures for Tree-Structured Data : Definitions and Constructions*. in *Applied Cryptography and Network Security*, 6123, pages 87–104. (Cited on page 83.)
- Brzuska, Christina, Marc Fischlin, Tobias Freudenreich, Anja Lehmann, Marcus Page, Jakob Schelbert, Dominique Schr, and Florian Volk [2009]. *Security of Sanitizable Signatures Revisited*. In *Proceedings of Irvine Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography - PKC*, pages 317–336. (Cited on pages 86, 87, 89 and 90.)

- Brzuska, Christina, Marc Fischlin, Anja Lehmann, and Dominique Schr [2010b]. *Unlinkability of Sanitizable Signatures*. in *Proceedings of Public Key Cryptography - PKC*, 6056, pages 444–461. (Cited on pages 86, 87 and 90.)
- Brzuska, Christina, Henrich C. Pöhls, and Kai Samelin [2012]. *Non-interactive Public Accountability for Sanitizable Signatures*. in *Proceedings of the 9th European PKI Workshop: Research and Applications (EuroPKI 2012)*, 7868, pages 178–193. ISSN 0302-9743. doi:10.1007/978-3-642-40012-4_12. (Cited on pages 87 and 88.)
- Buso et al. [2012]. *SPOCS Deliverable D2.4 Open Source Authentication Module (Version 1.4)*. (Cited on page 54.)
- Camenisch, Jan and Anna Lysyanskaya [2001]. *An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation*. In *Advances in Cryptology - EUROCRYPT 2001, LNCS*, volume 2045, pages 93–118. Springer. (Cited on page 146.)
- Canard, Sebastien and Amandine Jambert [2010]. *On Extended Sanitizable Signature Schemes*. in *Topics in Cryptology - CT-RSA*, 5985, pages 179–194. (Cited on pages 86, 89, 90, 91 and 94.)
- Canard, Sebastien, Amandine Jambert, and Roch Lescuyer [2012]. *Sanitizable Signatures with Several Signers and Sanitizers*. in *Proceedings of the 5th international conference on Cryptology in Africa - AFRICACRYPT*, 7374, pages 35–52. ISSN 0302-9743. doi:10.1007/978-3-642-31410-0_3. (Cited on pages 87, 90, 91 and 94.)
- Canard, Sebastien, Fabien Laguillaumie, and Michel Milhau [2008]. *Trapdoor Sanitizable Signatures and Their Application to Content Protection*. In *Proceedings of ACNS 2008*, 5037, pages 258–276. (Cited on page 83.)
- Chang, Ee-chien, Chee Liang Lim, and Jia Xu [2009]. *Short Redactable Signatures Using Random*. in *Topics in Cryptology - CT-RSA*, 5473, pages 133–147. (Not cited.)
- Clark, James and Steve DeRose [1999]. *XML Path Language (XPath) - Version 1.0*. <http://www.w3.org/TR/xpath/>. (Cited on page 113.)
- Cooper, D, S Santesson, S Farrell, S Boeyen, R Housley, and W Polk [2008]. *RFC 5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. (Cited on page 14.)
- Cooper et al. [2008]. *RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. (Cited on page 65.)
- Council, European [2000]. *Presidency Conclusions - Lison European Council*. http://www.consilium.europa.eu/uedocs/cms_data/docs/pressdata/en/ec/00100-r1.en0.htm. (Cited on pages 25 and 26.)
- Council of the European Union and European Commission [2000]. *eEurope - An Information Society for All - Action Plan*. (Cited on page 26.)
- Dahlström et al. [2011]. *W3C Recommendation, Scalable Vector Graphics (SVG) 1.1 (Second Edition)*. (Cited on page 134.)

- DCMI [2012]. *DCMI Metadata Terms*. <http://dublincore.org/specifications/>. (Cited on page 106.)
- Derler, David [2013]. *On the Optimization of two Recent Proxy-Type Digital Signature Schemes and their Efficient Implementation in Java*. (Cited on pages 91, 94, 96, 97 and 98.)
- DLM Forum [2011]. *Core Services and Plug-in Modules - Version 1.1*. <http://www.moreq.info/specification/>. (Cited on page 106.)
- ECMA [2012]. *Standard ECMA-376 - Office Open XML File Formats*. <http://www.ecma-international.org/publications/standards/Ecma-376.htm>. (Cited on page 115.)
- Environmental Systems Research Institute [1998]. *Specification: ESRI Shapefile Technical Description*. (Cited on page 134.)
- ETSI [2009]. *Technical Specification 102 231, Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information*. (Cited on pages 60, 70, 71 and 72.)
- ETSI [2010a]. *Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 3: PAdES Enhanced - PAdES-BES and PAdES-EPES Profiles; TS 102 778-3*. (Cited on pages 21 and 52.)
- ETSI [2010b]. *Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES); TS 101 903*. (Cited on pages 19, 52, 101, 102 and 140.)
- ETSI [2012]. *Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC)*. (Cited on page 52.)
- ETSI [2013a]. *Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAAdES); TS 101 733*. (Cited on pages 17 and 18.)
- ETSI [2013b]. *Technical Specification 119 612, Electronic Signatures and Infrastructures (ESI); Trusted Lists*. (Cited on page 72.)
- European Commission [2000]. *eEurope - An Information Society for All - Communication on a Commission Initiative for the Special European Council of Lisbon, 23 and 24 March 2000*. (Cited on page 26.)
- European Commission [2002]. *eEurope 2005: An information society for all - An Action Plan to be presented in view of the Sevilla European Council*. (Cited on page 26.)
- European Commission [2003]. *Directive 2003/98/EC of the European Parliament and the Council of 17 November 2003 on the re-use of public sector information*. (Cited on pages 27, 32 and 126.)
- European Commission [2004]. *European Interoperability Framework for pan-European e-Government Services - Version 1.0*. <http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529>. (Cited on page 29.)
- European Commission [2005]. *i2010 - A European Information Society for growth and employment*. (Cited on page 26.)

- European Commission [2006a]. *Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market*. (Cited on pages 31, 48, 50 and 160.)
- European Commission [2006b]. *i2010 eGovernment Action Plan: Accelerating eGovernment in Europe for the Benefit of All*. <http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52006DC0173&from=EN>. (Cited on page 27.)
- European Commission [2009a]. *Commission Decision of 16 October 2009 setting out measures facilitating the use of procedures by electronic means through the 'points of single contact' under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market*. (Cited on pages 40 and 72.)
- European Commission [2009b]. *Corrigendum to Commission Decision 2009/767/EC of 16 October 2009 setting out measures facilitating the use of procedures by electronic means through the 'points of single contact' under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market*. (Cited on page 72.)
- European Commission [2009c]. *IDABC. 2009. eID Interoperability for PEGS: Update of Country Profiles*. (Cited on page 144.)
- European Commission [2010a]. *A Digital Agenda for Europe*. <http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52010DC0245&rid=4>. (Cited on pages 24, 27, 41 and 42.)
- European Commission [2010b]. *A Digital Agenda for Europe*. (Cited on pages 48 and 126.)
- European Commission [2010c]. *Commission Decision of 28 July 2010 amending Decision 2009/767/EC as regards the establishment, maintenance and publication of trusted lists of certification service providers supervised/accredited by Member States*. (Cited on page 72.)
- European Commission [2010d]. *The European eGovernment Action Plan 2011-2015 - Harnessing ICT to promote smart, sustainable & innovative Government*. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0743:FIN:EN:PDF>. (Cited on pages 24, 28, 42 and 43.)
- European Commission [2011a]. *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*. (Cited on page 17.)
- European Commission [2011b]. *European Interoperability Framework for pan-European e-Government Services - Version 2.0*. http://ec.europa.eu/isa/documents/eif_brochure_2011.pdf. (Cited on pages 29, 30 and 41.)
- European Commission [2012]. *The functioning and usability of the Points of Single Contact under the Services Directive - State of Play and Way Forward*. http://ec.europa.eu/internal_market/services/docs/services-dir/study_on_points/final_report_en.pdf. (Cited on pages 43, 106 and 161.)

- European Commission [2013a]. *Commission Implementing Decision of 14 October 2013 amending Decision 2009/767/EC as regards the establishment, maintenance and publication of trusted lists of certification service providers supervised/accredited by Member States*. (Cited on pages 55, 59, 60 and 72.)
- European Commission [2013b]. *Directive 2013/37/EU of the European Parliament and of the Council of 26 June 2013 amending Directive 2003/98/EC on the re-use of public sector information*. (Cited on pages 32, 126, 127, 128 and 129.)
- European Commission [2014a]. *European Commission Decision: Amending Decision 2011/130/EU establishing minimum requirements for the crossborder 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(2014) 1640*. (Cited on pages 17, 39, 55, 85, 94, 95 and 141.)
- European Commission [2014b]. *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*. (Cited on pages 33, 173 and 181.)
- European Commission [2014c]. *Study on eGovernment and the Reduction of Administrative Burden*. http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=5155. (Cited on pages 43, 106, 120, 161 and 172.)
- Eurostat [2013a]. *Enterprises using the Internet for interacting with public authorities, Code: isoc_bde15ee*. (Cited on page 160.)
- Eurostat [2013b]. *Individuals using the Internet for interacting with public authorities, Code: isoc_bde15ei*. (Cited on page 160.)
- Fotiou et al. [2012]. *Pilot evaluation. Annex to D5.9 Pilot and evaluation of the e-Services related to the selected Professions, Version 1.0*. (Cited on pages 57 and 58.)
- Hanser, Christian and Daniel Slamanig [2013]. *Blank digital signatures. Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13*, page 95. doi:10.1145/2484313.2484324. <http://dl.acm.org/citation.cfm?doid=2484313.2484324>. (Cited on pages 76, 80, 83, 87, 90, 91, 94 and 151.)
- Höhne, Focke, Henrich C. Pöhls, and Kai Samelin [2012]. *Rechtsfolgen editierbarer Signaturen. Datenschutz und Datensicherheit - DuD*, 36(7), pages 485–491. ISSN 1614-0702. doi:10.1007/s11623-012-0165-8. <http://link.springer.com/10.1007/s11623-012-0165-8>. (Cited on pages 87 and 88.)
- Housley [2009]. *RFC 5652, Cryptographic Message Syntax (CMS)*. (Cited on page 17.)
- Huang, Zewu, Hai Jin, Pingpeng Yuan, and Zongfen Han [2006]. *Header Metadata Extraction from Semi-structured Documents Using Template Matching*. In Meersman, Robert, Zahir Tari, and Pilar Herrero (Editors), *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Lecture Notes in Computer Science*, volume 4278, pages 1776–1785. Springer Berlin Heidelberg. ISBN 978-3-540-48273-4. doi:10.1007/11915072_84. http://dx.doi.org/10.1007/11915072_84. (Cited on pages 109 and 117.)

- ISO/IEC [2008]. *ISO/IEC 32000-1, Document management - Portable document format - Part 1: PDF 1.7, First Edition*. (Cited on pages 52 and 94.)
- ISO/IEC [2008]. *PDF (Portable Document Format), version 1.7, Base level (ISO 32000-1:2008)*. (Cited on page 49.)
- J. R. Gil-Garcia [2007]. *Exploring E-Government Benefits and Success Factors*. In *Encyclopedia of Digital Government*, pages 803–811. (Cited on pages 43, 160 and 161.)
- Johnson, Robert, David Molnar, Dawn Song, and David Wagner [2002]. *Homomorphic Signature Schemes*. *RSA Security Conference Cryptographers Track, 2271* (Lecture Notes in Computer Science), pages 244–262. (Cited on pages 75, 76, 83 and 85.)
- Kaliski [1998]. *RFC 2315, PKCS #7: Cryptographic Message Syntax*. (Cited on pages 17 and 18.)
- Klonowski, Marek and Anna Lauks [2006]. *LNCS 4296 - Extended Sanitizable Signatures*. In *Proceedings of Information Security and Cryptology - ICISC, 4296*, pages 343–355. (Cited on pages 86, 89, 90 and 91.)
- Krawczyk, Hugo and Tal Rabin [2000]. *Chameleon Signatures*. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS 2000)*, pages 143–154. (Cited on page 79.)
- Krnjic, Vesna, Klaus Stranacher, Tobias Kellner, and Andreas Fitzek [2013]. *Modular Architecture for Adaptable Signature-Creation Tools*. In Wimmer, MariaA., Marijn Janssen, and HansJ. Scholl (Editors), *Electronic Government, Lecture Notes in Computer Science*, volume 8074, pages 274–285. Springer Berlin Heidelberg. ISBN 978-3-642-40357-6. doi:10.1007/978-3-642-40358-3_23. http://dx.doi.org/10.1007/978-3-642-40358-3_23. (Cited on page 209.)
- Lei, Hao and Dengguo Feng [2011]. *Selective Disclosure on Encrypted Documents*. In Li, Yingjiu (Editor), *Data and Applications Security and Privacy XXV, Lecture Notes in Computer Science*, volume 6818, pages 255–262. Springer Berlin Heidelberg. ISBN 978-3-642-22347-1. doi:10.1007/978-3-642-22348-8_21. http://dx.doi.org/10.1007/978-3-642-22348-8_21. (Cited on page 145.)
- Leitold, Herbert and Karl-Christian Posch [2004]. *The Austrian Citizen Card: A Bottom-Up View* book title: *Security and Privacy in Advanced Networking Technologies*, volume 193, pages 231 – 236. IOS Press. (Cited on pages 74 and 144.)
- Leitold, Herbert, Reinhard Posch, and Thomas Rössler [2010]. *Reconstruction of electronic signatures from eDocument printouts*. *Computers & security*, 29, pages 523 – 532. (Cited on page 55.)
- Leitold H., Posch R., Hollosi A. [2002]. *Security Architecture of the Austrian Citizen Card Concept*. In *Proceedings of 18th Annual Computer Security Applications Conference (ACSAC'2002), Las Vegas, 9-13 December 2002*. pp. 391-400, IEEE Computer Society, ISBN 0-7695-1828-1, ISSN 1063-9527., page n/a. (Cited on pages 74 and 149.)
- Lenz, Thomas, Klaus Stranacher, and Thomas Zefferer [2013a]. *Enhancing the Modularity and Applicability of Web-Based Signature-Verification Tools* book title: *Webist 2013 Selected and revised papers*. In *Lecture Notes in Business Information Processing*. Springer. In press. (Cited on pages 139 and 209.)

- Lenz, Thomas, Klaus Stranacher, and Thomas Zefferer [2013b]. *Towards a Modular Architecture for Adaptable Signature-Verification Tools*. In *Proceedings of the 9th International Conference on Web Information Systems and Technologies*, pages 325 – 334. (Cited on pages 69, 139 and 209.)
- Lenz, Thomas, Bernd Zwattendorfer, Klaus Stranacher, and Arne Tauber [2014]. *Identitätsmanagement in Österreich mit MOA-ID 2.0*. *eGovernment review*, 13, pages 18 – 19. (Cited on page 209.)
- Lithuanian Archives Department [2009]. *Specification ADOC-V1.0 of the electronic document signed by the electronic signature*. (Cited on page 55.)
- Margraf, Marian [2011]. *The New German ID Card*. In *ISSE 2010 Securing Electronic Business Processes*, pages 367–373. Vieweg+Teubner. ISBN 978-3-8348-1438-8. doi:10.1007/978-3-8348-9788-6_35. (Cited on pages 42, 144 and 145.)
- Mathes, Judith [2008]. *Diener des Königs - Diener des Staates*. <http://www.judithmathes.de/history/beamte.html>. (Cited on page 8.)
- Menezes, Alfred J., Paul C. van Oorschot, and Scott A. Vanstone [1996]. *Handbook of Applied Cryptography*. CRC Press. ISBN 0-8493-8523-7. <http://www.cacr.math.uwaterloo.ca/hac/>. (Cited on page 13.)
- Miyazaki, Kunihiro, Seiichi Susaki, Mitsuru Iwamura, Tsutomu Matsumoto, Ryoichi Sasaki, and Hiroshi Yoshiura [2003]. *Digital Document Sanitizing Problem*. In *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, pages 61– 67. (Cited on pages 41, 74 and 94.)
- Modinis [2006]. *The Status of Identity Management in European eGovernment initiatives*. Deliverable D3.5. (Cited on page 144.)
- Nuñez, David, Isaac Agudo, and Javier Lopez [2012]. *Integrating OpenID with Proxy Re-Encryption to enhance privacy in cloud-based identity services*. In *IEEE CloudCom 2012*, pages 241 – 248. ISBN 978-1-4673-4511-8. ISSN 978-1-4673-4509-5. doi:10.1109/CloudCom.2012.6427551. (Cited on page 145.)
- OASIS Security Services TC [2005]. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. (Cited on pages 144 and 151.)
- Open Geospatial Consortium Inc. [2008]. *KML, Version: 2.2.0*. (Cited on page 134.)
- Open Geospatial Consortium Inc. [2012]. *Geography Markup Language (GML) - Extended schemas and encoding rules, Version: 3.3.0*. (Cited on page 134.)
- Open Government Working Group [2007]. *8 Principles of Open Government Data*. (Cited on pages 126, 127 and 128.)
- OpenId Cons. [2007]. *OpenID Authentication 2.0 - Final*. (Cited on page 144.)
- Pöhls, Henrich C., Kai Samelin, and Joachim Posegga [2011]. *Sanitizable Signatures in XML Signature - Performance , Mixing Properties , and Revisiting the Property of Transparency*. in *Applied Cryptography and Network Security*, 6715, pages 166–182. (Cited on page 88.)

- Posch, Karl-Christian, Reinhard Posch, Arne Tauber, Thomas Zefferer, and Bernd Zwattendorfer [2011]. *Secure and Privacy-preserving eGovernment - Best Practice Austria book title: Rainbow of Computer Science*, pages 259 – 269. Lecture Notes in Computer Science, Christian S. Calude, Grzegorz Rozenberg, Arto Salomaa. (Cited on page 10.)
- Posch, Reinhard, Clemens Orthacker, Klaus Stranacher, Arne Tauber, Thomas Zefferer, and Bernd Zwattendorfer [2012]. *Open Source Bausteine als Kooperationsgrundlage book title: E-Government - Zwischen Partizipation und Kooperation*, pages 185 – 209. Wolfgang Eixelsberger, Jürgen Stember. (Cited on page 209.)
- Republic of Austria [2004]. *Austrian Federal Act on Provisions facilitating electronic communications with public Bodies; part I, Nr. 10/2004*. Federal law Gazette. (Cited on page 150.)
- Republik Österreich [2010]. *Bundesgesetz über elektronische Signaturen (Signaturgesetz - SigG)*. (Cited on page 31.)
- Rivest, Ronald L., Adi Shamir, and Leonard M. Adleman [1978]. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. *Communications of the ACM*, 21(2), pages 120–126. (Cited on pages 12 and 13.)
- Rössler et al. [2011]. *SPOCS Deliverable D2.1 Inventory of standard documents and relations to open specifications (Version 1.1)*. (Cited on pages 36 and 49.)
- Shafranovich [2005]. *RFC 4180, Common Format and MIME Type for Comma-Separated Values (CSV) Files*. <http://tools.ietf.org/html/rfc4180>. (Cited on page 133.)
- Siddhartha, Arora [2008]. *National e-ID card schemes: A European overview*. *Inf. Secur. Tech. Rep.*, 13(2), pages 46–53. (Cited on page 144.)
- Slamanig, Daniel and Stefan Rass [2010]. *Generalizations and Extensions of Redactable Healthcare. in Communications and Multimedia Security CMS*, 6109, pages 201–213. (Cited on page 83.)
- Slamanig, Daniel, Klaus Stranacher, and Bernd Zwattendorfer [2014]. *User-Centric Identity as a Service-Architecture for eIDs with Selective Attribute Disclosure*. In *19th ACM Symposium on Access Control Models and Technologies (SACMAT 2014), London (Ontario), Canada, 25-27 June*. ACM. In press. (Cited on page 209.)
- SPOCS Consortium [2011]. *PROPOSAL PART B: Annex I - "Description of Work"*. (Cited on page 48.)
- Stasis et al. [2012]. *SPOCS Deliverable D5.9 Final report Work Package 5, Pilot and evaluation of the e-Services related to the selected Professions*. (Cited on page 173.)
- Steinfeld, Ron, Laurence Bull, and Yuliang Zheng [2001]. *Content Extraction Signatures*. In *Proceedings of Information Security and Cryptology - ICISC*, 2288, pages 285–304. doi:10.1007/3-540-45861-1_22. (Cited on pages 75, 76, 83 and 85.)
- Stranacher [2013]. *SPOCS Final Report Work Package 2, eDocuments (Version 1.1)*. (Cited on pages 58, 71 and 181.)

- Stranacher, Klaus [2010]. *Foreign Identities in the Austrian E-Government - An interoperable eID Solution*. In Center, The Norwegian Computing (Editor), *IDMAN 2010 - 2nd IFIP WG-11.6 International Conference on Identity Management*, pages 31 – 40. (Cited on page 209.)
- Stranacher, Klaus and Tomazs Kawecki [2012]. *Interoperable Electronic Documents*. In *Electronic Government and Electronic Participation - Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart 2012, Informatik*, volume 39, pages 81 – 88. Trauner. (Cited on page 208.)
- Stranacher, Klaus, Vesna Krnjic, and Thomas Zefferer [2012]. *Vertrauenswürdige Open Government Data*. In Brigitte Lutz, Günther Tschabuschnig (Editor), *1.OGD D-A-CH-LI Konferenz*, pages 27 – 39. (Cited on page 208.)
- Stranacher, Klaus, Vesna Krnjic, and Thomas Zefferer [2013a]. *Authentische und integritätsgesicherte Verwaltungsdaten*. *eGovernment review*, 11, pages 30 – 31. (Cited on page 207.)
- Stranacher, Klaus, Vesna Krnjic, and Thomas Zefferer [2013b]. *Trust and Reliability for Public Sector Data*. In *Proceedings of International Conference on e-Business and e-Government*, volume 73, pages 124 – 132. (Cited on page 208.)
- Stranacher, Klaus, Vesna Krnjic, Bernd Zwattendorfer, and Thomas Zefferer [2013c]. *Assessment of Redactable Signature Schemes for Trusted and Reliable Public Sector Data*. In Ferrari, Elena and Walter Castelnovo (Editors), *Proceedings of the 13th European Conference on e-Government*, pages 508 – 516. ACPI. (Cited on page 208.)
- Stranacher, Klaus, Vesna Krnjic, Bernd Zwattendorfer, and Thomas Zefferer [2013d]. *Evaluation and Assessment of Editable Signatures for Trusted and Reliable Public Sector Data*. *Electronic journal of e-Government [Elektronische Ressource]*, 11, pages 360 – 372. (Cited on page 207.)
- Stranacher, Klaus, Thomas Lenz, and Konrad Lanz [2013e]. *Trust-Service Status List Based Signature Verification*. In Ko, Andrea, Christine Leitner, Herbert Leitold, and Alexander Prosser (Editors), *Technology-Enabled Innovation for Democracy, Government and Governance, Lecture Notes in Computer Science*, volume 8061, pages 29–42. Springer Berlin Heidelberg. ISBN 978-3-642-40159-6. doi:10.1007/978-3-642-40160-2_4. http://dx.doi.org/10.1007/978-3-642-40160-2_4. (Cited on page 208.)
- Stranacher, Klaus, Arne Tauber, and Thomas Rössler [2009]. *Ausländische Identitäten im österreichischen E-Government*. In Patrick Horster, Peter Schartner (Editor), *D-A-CH Security 2009*, pages 263 – 272. (Cited on page 209.)
- Stranacher, Klaus, Arne Tauber, Thomas Zefferer, and Bernd Zwattendorfer [2013f]. *The Austrian Identity Ecosystem - An e-Government Experience book title: Architectures and Protocols for Secure Information Technology*, pages 288 – 309. *Advances in Information Security, Privacy, and Ethics (AISPE)*, IGI Global. (Cited on page 209.)
- Stranacher, Klaus and Bernd Zwattendorfer [2009]. *Web-Service based Transformation of Digital Signature Formats*. In *Taking the eGovernment Agenda Forward: Meeting the Challenges of Digital Governance, Justice and Public Sector Information*, pages 523 – 532. Austrian Computer Society (OCG). (Cited on page 209.)

- Stranacher, Klaus and Bernd Zwattendorfer [2012]. *Ein interoperabler Container für elektronische Dokumente*. In Peter Schartner, Jürgen Taeger (Editor), *D-A-CH Security 2012*, pages 421 – 431. (Cited on page 208.)
- Stranacher, Klaus and Bernd Zwattendorfer [2013]. *Redigierbare Signaturen in e-Business Anwendungen*. In *Proceedings of D-A-CH Security Konferenz*, pages 19 – 30. (Cited on page 208.)
- Stranacher, Klaus and Bernd Zwattendorfer [2014]. *Efficient Public Administration Procedures Across Borders*. In *Proceedings of 10th Central and Eastern European eGov Days*, pages 317 – 327. (Cited on page 208.)
- Stranacher, Klaus, Bernd Zwattendorfer, Sandra Fruhmann, and Patrick Koch [2014]. *Umsetzung eines vertrauenswürdigen Open Government Data*. In *D-A-CH Security 2014*. In press. (Cited on page 208.)
- Stranacher, Klaus, Bernd Zwattendorfer, and Vesna Krnjic [2013g]. *Secure and Efficient Processing of Electronic Documents in the Cloud*. In *Proceedings of IAIDIS International Conference e-Society*, pages 217 – 224. (Cited on page 208.)
- Stranacher, Klaus, Bernd Zwattendorfer, and Clemens Orthacker [2008]. *EU-Projekt "eGov-Bus". eGovernment review*, 1, pages 8 – 9. (Cited on page 209.)
- Stranacher et al. [2011]. *SPOCS Deliverable D2.2 Standard Document and Validation Common Specifications (Version 1.4.0)*. (Cited on pages 50 and 51.)
- Stranacher et al. [2012]. *SPOCS Deliverable D2.3 Open Source Standard Document Processing Module (Version 1.2)*. (Cited on pages 54 and 55.)
- Sultana, Nik, Moritz Y. Becker, and Markulf Kohlweiss [2013]. *Selective Disclosure in Datalog-Based Trust Management*. In Accorsi, Rafael and Silvio Ranise (Editors), *Security and Trust Management, Lecture Notes in Computer Science*, volume 8203, pages 160–175. Springer Berlin Heidelberg. ISBN 978-3-642-41097-0. doi:10.1007/978-3-642-41098-7_11. http://dx.doi.org/10.1007/978-3-642-41098-7_11. (Cited on page 145.)
- Tauber, Arne [2012]. *Cross-border Certified Electronic Mailing: A Scalable Interoperability Framework for Certified Mail Systems*. PhD Thesis, University of Technology Graz. (Cited on page 171.)
- Tauber, Arne, Klaus Stranacher, and Daniel Medimorec [2012]. *SPOCS: Interoperable eGovernment Services in the Context of the Services Directive*. *European journal of ePractice [Elektronische Ressource]*, 14, pages 90 – 106. (Cited on page 207.)
- Tauber, Arne, Bernd Zwattendorfer, and Klaus Stranacher [2013]. *Elektronische Identität und Stellvertretung in Österreich*. In Peter Schartner, Peter Trommler (Editor), *D-A-CH Security 2013*, pages 1 – 9. (Cited on page 209.)
- Tauber, Arne, Bernd Zwattendorfer, Thomas Zefferer, and Klaus Stranacher [2011]. *Grenzüberschreitendes E-Government in Europa*. *eGovernment review*, 8, pages 8 – 9. (Cited on page 209.)

- Tews, Hendrik and Bart Jacobs [2009]. *Performance Issues of Selective Disclosure and Blinded Issuing Protocols on Java Card*. In Markowitch, Olivier, Angelos Bilas, Jaap-Henk Hoepman, ChrisJ. Mitchell, and Jean-Jacques Quisquater (Editors), *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks, Lecture Notes in Computer Science*, volume 5746, pages 95–111. Springer Berlin Heidelberg. ISBN 978-3-642-03943-0. doi:10.1007/978-3-642-03944-7_8. http://dx.doi.org/10.1007/978-3-642-03944-7_8. (Cited on page 145.)
- The Council of the European Union [2000]. *Directive 1999/93/EC the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures*. (Cited on pages 15, 27, 31, 59, 83 and 147.)
- Vullers, Pim and Gergely Alpar [2013]. *Efficient Selective Disclosure on Smart Cards Using Idemix*. In Fischer-Hübner, Simone, Elisabeth Leeuw, and Chris Mitchell (Editors), *Policies and Research in Identity Management, IFIP Advances in Information and Communication Technology*, volume 396, pages 53–67. Springer Berlin Heidelberg. ISBN 978-3-642-37281-0. doi:10.1007/978-3-642-37282-7_5. http://dx.doi.org/10.1007/978-3-642-37282-7_5. (Cited on page 145.)
- W3C [2008]. *W3C Recommendation, Extensible Markup Language (XML) 1.0 (Fifth Edition)*. (Cited on page 134.)
- W3C [2012]. *XML Schema 1.1*. <http://www.w3.org/XML/Schema.html>. (Cited on pages 113, 115 and 119.)
- Wikipedia.org [2014a]. *Ancient Egypt*. http://en.wikipedia.org/wiki/Ancient_Egypt. (Cited on page 8.)
- Wikipedia.org [2014b]. *Bureaucracy*. <http://en.wikipedia.org/wiki/Bureaucracy>. (Cited on page 8.)
- Wikipedia.org [2014c]. *Public administration*. http://en.wikipedia.org/wiki/Public_administration. (Cited on page 8.)
- Wikipedia.org [2014d]. *Seal (emblem)*. [http://en.wikipedia.org/wiki/Seal_\(emblem\)](http://en.wikipedia.org/wiki/Seal_(emblem)). (Cited on page 8.)
- Zefferer, Thomas, Vesna Krnjic, Klaus Stranacher, and Bernd Zwattendorfer [2014]. *Measuring Usability to Improve the Efficiency of Electronic Signature-based E-Government Solutions book title: Measuring E-government efficiency. The opinions of Public Administrators and other Stakeholders*, pages 45 – 74. Springer. (Cited on pages 41, 43, 160, 161, 162 and 209.)
- Zefferer, Thomas, Arne Tauber, Bernd Zwattendorfer, and Klaus Stranacher [2012]. *Qualified PDF signatures on mobile phones*. In *Electronic Government and Electronic Participation - Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart 2012, Informatik*, volume 39, pages 115 – 123. (Cited on page 209.)
- Zefferer et al. [2011]. *Secure and Reliable Online-Verification of Electronic Signatures in the Digital Age*. In Bebo White, Pedro Isaias and Flavia Maria Santoro (Editors), *Proceedings of the IADIS International Conference WWW/INTERNET 2011*, pages 269 – 276. (Cited on page 138.)

- Zwattendorfer, Bernd [2014]. *Towards a Privacy-Preserving Federated Identity as a Service-Model*. (Cited on page 156.)
- Zwattendorfer, Bernd, Daniel Slamanig, Klaus Stranacher, and Felix Hörandner [2014a]. *A Federated Cloud Identity Broker-Model for Enhanced Privacy via Proxy Re-Encryption*. In *15th IFIP TC6/TC11 International Conference on Communications and Multimedia Security, CMS'2014, September 25th - 26th, 2014, Aveiro, Portugal*. In press. (Cited on page 209.)
- Zwattendorfer, Bernd, Klaus Stranacher, and Arne Tauber [2012a]. *Bürgerkarten-Authentifizierung zur Public Cloud*. In Peter Schartner, Jürgen Taeger (Editor), *D-A-CH Security 2012*, pages 136 – 147. (Cited on page 209.)
- Zwattendorfer, Bernd, Klaus Stranacher, and Arne Tauber [2013a]. *Towards a Federated Identity as a Service Model*. In *Technology-Enabled Innovation for Democracy, Government and Governance*, pages 43 – 57. Lecture notes in computer science, Springer. (Cited on page 209.)
- Zwattendorfer, Bernd, Klaus Stranacher, Arne Tauber, and Peter Reichstädter [2013b]. *Cloud Computing in E-Government across Europe*. In *Technology-Enabled Innovation for Democracy, Government and Governance*, pages 181 – 195. Lecture notes in computer science, Springer. (Cited on pages 168 and 209.)
- Zwattendorfer, Bernd, Arne Tauber, Klaus Stranacher, and Peter Reichstädter [2012b]. *Cross-Border Legal Identity Management*. In *Electronic Government 11th IFIP WG 8.5 International Conference, EGOV 2012*, pages 149 – 161. Springer. (Cited on page 209.)
- Zwattendorfer, Bernd, Thomas Zefferer, and Klaus Stranacher [2014b]. *An Overview of Cloud Identity Management-Models*. In INSTICC (Editor), *Proceedings of the 10th International Conference on Web Information Systems and Technologies*, pages 82 – 92. (Cited on page 209.)