Master Thesis

Design and Implementation of a Base Station for a Slope Movement Monitoring System with Cloud2Device Integration

Hannes Mock, BSc

Institute for Technical Informatics
Graz University of Technology

Assessor: Ass.Prof. Dipl.-Ing. Dr.techn. Christian Steger

Advisor: Ass.Prof. Dipl.-Ing. Dr.techn. Christian Steger
Dipl.-Ing. Leander Bernd Hörmann, BSc

Graz, March 2013

Abstract

The change of the global climate brings about an increased likelihood of slope movements. In the Alpine region this fact is enhanced by the progressive development of urban areas which leads to a deforestation of natural protection barriers. The project GeoWSN investigates, in which way the collected data should be pass to the population to initiate timely and efficient rescue and evacuation measures.

The goal of this thesis is the design and implementation of a database and visualization system that is characterized by reliable storage and intuitive display of data.

In the course of this thesis the data of the wireless sensor networks are transmitted over the Universal Serial Bus (USB) interface of one sensor node which serves as gateway node to a Linux server. Services running on the server receive, process and store the data into a MySQL database system. For the visualization two different approaches are chosen: on the one hand a web application with configuration options, and on the other hand a native Android application.

To achieve and guarantee a reliable alerting on the mobile device, an Extensible Messaging and Presence Protocol (XMPP), which interacts via the Metis Mobile Cloud (MMC), is integrated into the application.

Keywords

wireless sensor networks, environmental sensor networks, early warning and monitoring system, database and visualization system, Cloud2Device, Extensible Messaging and Presence Protocol

Kurzfassung

Durch die Veränderung des Weltklimas treten vor allem im alpinen Raum Massenbewegungen auf, welche Infrastrukturen gefährden. Das GeoWSN-Projekt befasst sich mit der kompletten Informationskette von der Datenerfassung mittels eines drahtlosen Sensor Netzwerks bis hin zur Visualisierung der bearbeiteten Daten für den Endbenutzer und soll die Frage klären, wie die gesammelten Daten an die Bevölkerung weitergegeben werden können, um etwaige Evakuierungsmaßnahmen einzuleiten. Das Ziel dieser Masterarbeit ist die Entwicklung eines Datenbank- und Visualisierungssystems, dass sich durch zuverlässige Speicherung und intuitive Darstellung der Daten auszeichnet. Im Zuge dieser Arbeit werden die gesammelten Daten des drahtlosen Sensornetzwerkes über einen als Basis dienenden Sensorknoten via ein USB (Universal Serial Bus) Interface an einen Linux Server weitergeleitet. Die auf diesem Server laufenden Dienste empfangen, verarbeiten und speichern die Daten in einer MySQL Datenbank ab. Für die Visualisierung wurde einerseits eine Web Applikation und andererseits eine native Android Applikation gewählt. Um eine ständige Alarmierung auf der mobilen Applikation zu bewerkstelligen und zu garantieren, wurde ein XMPP (Extensible Messaging and Presence Protocol) Push-Protokoll integriert, welches über die Metis Mobile Cloud (MMC) kommuniziert.

STATUTORY DECLARATION

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

....

date (signature)

Acknowledgment

I would take my master thesis as an opportunity to say thank you to those people, who have always supported me and always had a sympathetic ear. My thesis and my studies would not be possible without these people.

I want thank my mother, who supported me in any possible way during the whole studies and has made it all possible. I also express my special gratitude to my girlfriend Katharina for her support and the patience in the last years.

Concerning the time when I was working on my thesis I want to thank all the people of the GeoWSN consortium consisting of Geolith Consult Hermann & Loizenbauer OEG, the Department of Emergency Management and National Defence of the federal state of Styria, TeleConsult Austria GmbH, and Graz University of Technology (Institute of Navigation and Institute for Technical Informatics). Especially my two supervisors, Ass. Prof. Dipl.-Ing. Dr.techn. Christian Steger and Dipl.-Ing. Leander Berhander Hörmann, who supported me during this time.

Graz, March 2013 Hannes Mock

Contents

1 Introduction **1**
- 1.1 Motivation 2
- 1.2 Outline 3

2 Related Work **5**
- 2.1 Hazard Assessment and Early Warning Systems 5
- 2.2 Environmental Monitoring 5
- 2.3 Environmental Wireless Sensor Monitoring Systems 7
  - 2.3.1 Alpine Early Warning System 7
  - 2.3.2 Sensorbased Landslide Early Warning System 10
  - 2.3.3 X-Sense Project 12
  - 2.3.4 Further GPS-based Monitoring Systems of Landslide Movements . . 15
  - 2.3.5 Comparison between State-of-the-Art Projects and the GeoWSN Project 16
- 2.4 Instant Messaging and Presence Technologies 19
  - 2.4.1 Extensible Messaging and Presence Protocol 21
  - 2.4.2 Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions 23
- 2.5 Summary 25

3 Design **26**
- 3.1 System Architecture 26
  - 3.1.1 GeoWSN System Overview 26
  - 3.1.2 Gateway Node Architecture 28
  - 3.1.3 GeoWSN Base Station Architecture 29
- 3.2 GeoWSN Services 30
  - 3.2.1 Requirements 30
  - 3.2.2 Service Package Structure 31
- 3.3 Database 32

3.3.1 Requirements . . . 32
3.3.2 Database Structure . . . 33
3.4 Web Application . . . 34
3.4.1 Requirements . . . 34
3.4.2 User Interface . . . 35
3.5 Metis Mobile Cloud (MMC) . . . 35
3.5.1 MMC Architecture . . . 35
3.5.2 Distinction to the Android Cloud to Device Messaging Framework . 36
3.6 Mobile Client . . . 37
3.6.1 Requirements . . . 37
3.6.2 User Interface . . . 38
3.6.3 Metis Mobile Cloud (MMC) Interaction . . . 38
3.7 Summary . . . 39

4 Implementation 41
4.1 Development Environment . . . 41
4.1.1 Eclipse . . . 42
4.1.2 ZK Framework . . . 42
4.1.3 Android SDK . . . 43
4.1.4 Code Composer Sudio . . . 44
4.1.5 MySQL Database Management System . . . 45
4.2 Early Warning and Monitoring System . . . 46
4.3 Gateway Node . . . 47
4.3.1 Data Gathering Communication Flow . . . 48
4.4 Communication Protocols . . . 49
4.4.1 Base Frame . . . 49
4.4.2 Data Frame . . . 50
4.4.3 Ephemeris Frame . . . 52
4.5 GeoWSN Service Package . . . 52
4.5.1 GeoWSN Listener . . . 52
4.5.2 GeoWSN DataLogger . . . 57
4.5.3 GeoWSN AlertingService . . . 59
4.6 GeoWSN Web Service . . . 61
4.6.1 Object-Relational Mapping with Hibernate . . . 61
4.6.2 Operation . . . 62
4.7 Database . . . 64
4.7.1 Database Structure . . . 64
4.7.2 Entity Description . . . 66

4.8 Web Application . . . 78
    4.8.1 Class Description . . . 79
    4.8.2 User Interface and Interaction . . . 79
    4.8.3 API Communication . . . 80
4.9 Mobile Client . . . 81
    4.9.1 User Interface and Interaction . . . 82
    4.9.2 API Communication . . . 83
4.10 Experimental Results . . . 83
    4.10.1 Test Settings and Test Scenario . . . 84
    4.10.2 Test Results . . . 84
4.11 Evaluation . . . 86

5 Conclusion and Future Work **88**
5.1 Conclusion . . . 88
    5.1.1 COM Driver for the Gateway Node . . . 88
    5.1.2 Database System . . . 89
    5.1.3 Service Package . . . 89
    5.1.4 RESTful Web Service . . . 90
    5.1.5 Web Application . . . 90
    5.1.6 Mobile Client . . . 90
5.2 Future Work . . . 91

A Communication Protocol **92**

B Front-end Applications: User Interface **97**

Bibliography **101**

List of Figures

1.1 GeoWSN system. 2

2.1 Danger – precautionary measures – rest risk [1]. 6
2.2 Types of environmental sensor networks. 7
2.3 alpEWAS system overview [2]. 8
2.4 Structure of the SLEWS wireless sensor network [3]. 11
2.5 GPS-equipped wireless sensor node (reproduced from [4]). 13
2.6 GeoWSN wireless sensor node [5]. 17
2.7 Overview of presence service [6]. 20
2.8 Overview of instant message Service [6]. 20
2.9 Decentralized client-server architecture. 21
2.10 Security layers of the XMPP technology [7]. 23
2.11 SIMPLE presence architecture [8]. 24

3.1 GeoWSN system architecture. 27
3.2 Software architecture of the gateway node. 28
3.3 2-tier server architecture. 30
3.4 GeoWSN service/daemon package. 31
3.5 The flow of the database information. 33
3.6 Web application mock-up. 36
3.7 The architecture of the Metis Mobile Cloud Framework including the XMPP service (reproduced from [9]). 37
3.8 Smartphone operating systems and market share (*source: IDC Worldwide Mobile Phone Tracker, August 8, 2012*). 38
3.9 Detail-view mock-up. 39
3.10 Map-view mock-up. 39

4.1 ZK framework architecture (reproduced from [10]). 43
4.2 The structure of Android (reproduced from [11]). 44
4.3 MySQL architecture (reproduced from [12]). 45

4.4 GeoWSN system structure. 46
4.5 Communication flow between the data gathering components. 48
4.6 GeoWSN Listener class diagram. 53
4.7 Simplified program flow of the GeoWSN Listener RX task. 55
4.8 Simplified program flow of the GeoWSN Listener process task. 56
4.9 GeoWSN DataLogger class diagram. 57
4.10 Simplified program flow of the GeoWSN DataLogger process task. 59
4.11 GeoWSN Alerting Service class diagram. 59
4.12 Simplified program flow of the GeoWSN AlertingService alarm detection task. 60
4.13 Architectural overview of the web service components. 61
4.14 Hibernate architecture (reproduced from [13]. 62
4.15 Simplified program flow of the GeoWSN web service generate task. 63
4.16 Simplified program flow of the GeoWSN web service watchdog task. 64
4.17 Information flow for the location determination. 65
4.18 Input data RTK 65
4.19 Input data PPP 65
4.20 Web application class diagram 78
4.21 Web application map view. 80
4.22 Web application network view. 80
4.23 Web application detail and diagram view. 80
4.24 Web application communication flow. 81
4.25 Android app detail view. 82
4.26 Android app map view. 82
4.27 Communication flow of the mobile client. 83
4.28 Web application communication flow. 85
4.29 Web application communication flow. 85
4.30 Web application communication flow. 86

B.1 Android app detail view. 97
B.2 Android app node detail view. 97
B.3 Android app map view. 97
B.4 Web application map view. 98
B.5 Web application network view. 99
B.6 Web application detail view. 100

List of Tables

2.1 Slope failure detection with different sensor mechanisms [14]. 12
2.2 Tabular comparison of monitoring and early warning systems for critical slopes. 18
2.3 Client-server architectures [15]. 22
2.4 Advantages and disadvantages of the distributed architecture [6]. 23

4.1 Base frame format [16]. 49
4.2 Data frame format [16]. 50
4.3 GPS data frame format [16]. 51
4.4 Sensor state mask [16]. 51
4.5 Ephemeris frame format [16]. 52
4.6 Description of database entity **rdata_snode_0...n** 66
4.7 Description of database entity **sensor_nodes** 68
4.8 Description of database entity **ephem_base** 68
4.9 Description of database entity **ephem_broadcast** 69
4.10 Description of database entity **iono_klobuchar** 70
4.11 Description of database entity **ephem_precise** 71
4.12 Description of database entity **iono_ionex** 72
4.13 Description of database entity **iono_code** 72
4.14 Description of database entity **erp** 73
4.15 Description of database entity **dcb** 74
4.16 Description of database entity **pos_rtk_0...n / pos_ppp_0...n** 74
4.17 Description of database entity **data_logger_snode_0...n** 75
4.18 Description of database entity **threshold** 76
4.19 Description of database entity **alarm** 77

A.1 Number format [16]. 92
A.2 Subframe 1 [16]. 93
A.3 Subframe 2 [16]. 94
A.4 Subframe 3 [16]. 95

List of Abbreviations

3D	Three-dimensional
Ajax	Asynchronous JavaScript and XML
alpEWAS	Alpine Early Warning System
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AU	Asynchronous Update
BIOS	Basic Input/Output System
C2DM	Cloud to Device Messaging
CCA	Central Control Application
CCStudio	Code Composer Studio
CODE	Center of Orbit Determination in Europe
CPU	Central Processing Unit
CSMA-CA	Carrier Sense Multiple Access/Collision Avoidance
DCBs	Differential Code Biases
DDL	Data Definition Language
DGPS	Differential Global Positioning System
DML	Data Manipulation Language
DOM	Document Object Model
DSP	Digital Signal Processor
GeoWSN	Geological Wireless Sensor Network
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers

IGS	International GNSS Service
IM&P	Instant Messaging and Presence Technology
IM	Instant Messaging
IONEX	Ionosphere Map Exchange
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
JDBC	Java Database Connectivity
JID	Jabber Identifier
JTAG	Joint Test Action Group
L1 GPS	Single Frequency Global Positioning System
LC-GNSS	Low-cost Global Navigation Satellite System
LF	Line Feed
MEMS	Microelectromechanical Systems
MIME	Multipurpose Internet Mail Extension
MMC	Metis Mobile Cloud
MMC	Metis Mobile Cloud
MMS	Movement Measurement System
MSRP	Message Session Relay Protocol
NL	New Line
ORM	Object-Relational Mapping
OS	Operating System
POJOs	Plain Old Java Objects
POT	Power Optimized Tracking
PPP	Precise Point Positioning
REST	Representational State Transfer
RSSI	Received Signal Strength Indication
RTK	Real Time Kinematic
RXTX	Receive/Transmit
RX	Receive
SASL	Simple Authentication and Security Layer
SDK	Software Development Kit
SD	Secure Digital
SIMPLE	Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SLEWS	Sensorbased Landslide Early Warning System

SQL	Structured Query Language
SRAM	Static Random-Access Memory
SSL	Secure Sockets Layer
SV	Satellite Vehicle
TCP	Transmission Control Protocol
TDR	Time-Domain-Reflectometry
TEC	Total Electronic Content
TI	Texas Instruments
TLS	Terrestrial Laser Scan
TLS	Transport Layer Security
TPS	Terrestrial Positioning System
UART	Universal Asynchronous Receiver Transmitter
UI	User Interface
USB	Universal Serial Bus
UTC	Coordinated Universal Time
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
XDCtools	eXpress DSP Components tools
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
X-Sense	Extreme Sense
XUL	XML User Interface Language
ZUML	ZK User Interface Markup Language

Chapter 1

Introduction

Due to the changes in the global climate, the monitoring of slope movements is today more important than ever. Because of continuous measurement data of such endangered slopes, it is possible to generate exact geological models that can be used to make forecasts about the occurrence of an avalanche. Due to this fact, the opportunity to evacuate humans and protect public infrastructure, such as roads and utility networks, increases.

The national KIRAS project called *GeoWSN*[1] *("Frühwarnsystem zur Beurteilung der Gefährdung kritischer Infrastruktur durch Hangrutschungen")* is motivated by the lack of an adequate early warning and monitoring system for such slope movements. The proposed early warning and monitoring system uses a wireless sensor network. With such architecture it is possible to provide high-resolution measurement data that is highly accurate in time and space, for large-scale monitoring. The project investigates the complete information chain:

- raw data gathering over an energy-self-sufficient and maintenance-free wireless sensor network
- processing and parsing the data
- editing the data
- storing the data
- visualization
- hazard alert messages

Figure 1.1 shows the system overview of the GeoWSN project. The system behind the GeoWSN project determines the movement of critical slopes. Therefore, the back-end based on a wireless sensor network, which is equipped with a low-cost L1 Global Positioning

---

[1] Project grant GeoWSN (832344) by the Austrian Research Promotion Agency (FFG) under KIRAS PL3.3

System (GPS) receiver and several internal and external sensors. However, for the intended usage the accuracy of such low-cost GPS receivers is not sufficient. In the GeoWSN project, special positioning algorithms, like Real Time Kinematic (RTK), and Precise Point Positioning (PPP) [17] that determine a precise position of each sensor node are used.

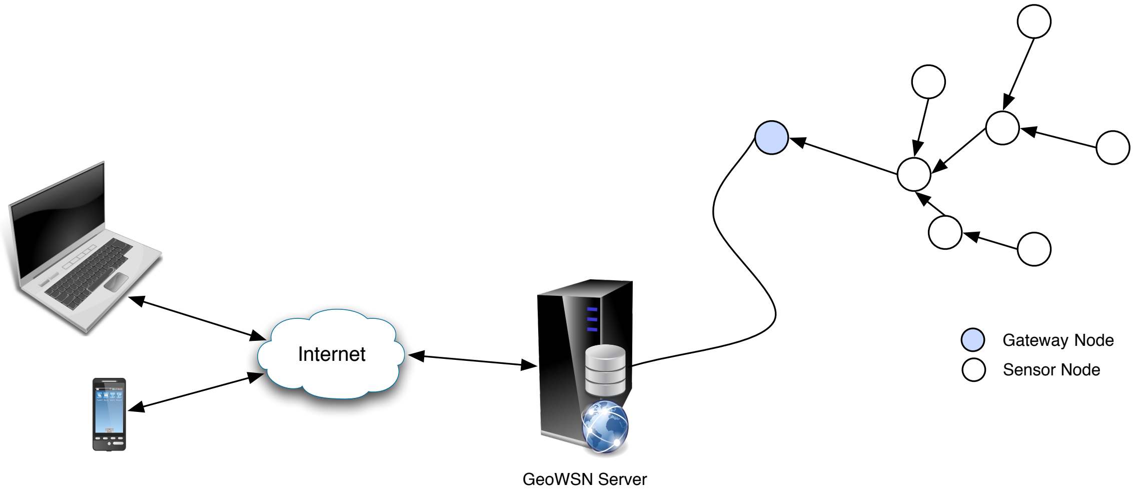


Figure 1.1: GeoWSN system.

1.1 Motivation

In the last few decades the change of the global climate and the progressive development of urban areas has lead to an increased risk level of descents of an avalanches.

Nevertheless, passive logging systems were applied to monitor slope movements. The problem of these systems is the disability to generate exact geological models out of the collected data. Such passive systems are enveloped into 'intelligent' environmental sensor networks [18]. An environmental sensor network comprises sensor nodes and a communication system, which allows an active forwarding of the measured sensor data to a server. With the help of spatially and temporally accurate data, it will be possible to generate exact geological models.

Why is it so important to generate exact geological models? Because highly accurate models and data are paramount for making forecasts in environments, where landslides jeopardize infrastructure and human life. Essential processes in such dangerous environments are often not logged by virtue of inaccessibility [18]. Thus, energy-self-sufficient and maintenance-free wireless sensor networks (environmental sensor networks) are a revolution in earth systems and environmental sciences. Another characteristic of environmental sensor networks is that spatially and temporally accurate data can be visualized and used for research and hazard alerts.

1.2 Outline

The aim of this master's thesis is the design and implementation of a base station for a slope movement monitoring and early warning system including a mobile hazard alert system with an Instant Messaging and Presence Technology (IM&P).

A MySQL database acts as the central component of the system. Some Linux services are functioning as fundamental link between the wireless sensor network, the database, and the front-end applications. One of those services is used to read the raw data of the sensor nodes via the communication interface and to parse the data for further processing. Another option, which is involved in this service, is the possibility of sending configuration statements to control the measurement interval of the sensor nodes. The second service, which is included in the GeoWSN service package, allows to compromise the raw data information with the help of an intelligent algorithm to reach a better performance for the visualization tools. The service package also integrates an alerting service. This service periodically checks if a velocity threshold exceedance occurs.

A greater usability is by the aforementioned reachability of the Linux services over a remote access service. This feature offers statistical information and opens possibilities for the operating surgeon via a graphical user interface.

To ensure the compatibility of the monitoring and early warning system for several front-ends a web application was chosen. With the web application it is able to monitor the whole wireless sensor network system, mange the access privilege and make settings for the monitoring and early warning system. To support these functionalities on the server a Representational State Transfer (REST) web service was implemented. Over the web service the user can query the visualization data and can also change configuration settings. The web service provides authentication and authorization mechanisms.

The design and implementation of the web application should meet the following requirements:

- intuitive usage
- alerting messages
- near-real time visualization
- configuration options
- live-view

Given the popularity of smartphones and mobile devices, a mobile application, using the Android platform, was developed. For a near-real time hazard alerting a cloud-based push technology was integrated. For the data transfer between the database and the mobile client, the functionality of the existing MMC (bases on the Extensible Messaging and Presence Protocol (XMPP) technology) was adapted and extended. Thereby, the user authentication takes place on the XMPP server.
The mobile application should meet the following requirements:

- energy-efficient
- near-real time alerting messages
- instant messaging and presence technology
- intuitive usage

The remainder of this thesis is structured as following. In **Chapter 2** the related work is discussed. Therefor several early warning and monitoring systems are analyzed. A further part of the chapter are push protocols for instant alerting. **Chapter 3** contains the concept of the database and visualization system. Also the requirements of such a system are mentioned here. In **Chapter 4** the detailed implementation of the system and the results are described. The conclusion and possible future work are provided in **Chapter 5** . Additional information can be found in the **Appendix**.

Chapter 2

Related Work

This chapter deals with environmental monitoring, and instant messaging and presence technologies. In order to develop a monitoring and early warning system for slope movements, the requirements of such a monitoring system and the risk of an avalanche occurring is from great importance. First, Section 2.1 discusses the topic hazard assessment and how to make forecasts as efficiently and effectively as possible. Then a general overview over environmental sensor networks is given in Section 2.2. In Section 2.3 different state-of-the-art environmental wireless sensor monitoring systems are mentioned. Finally, Section 2.4 shows different instant messaging and presence technologies.

2.1 Hazard Assessment and Early Warning Systems

Early warning systems for environmental processes are very complex and a part of a safety concept. Figure 2.1 shows that with the help of early warning system it is only possible to estimate the danger and not to eliminate the danger.

An early warning system consists of three main components: scientific component, technical component and the disaster management [19]. The scientific component supplies fundamental data and models as well as thresholds. The other two components are significant for this thesis. First, the technical part provides the tools and devices for gathering, relaying and storing data. And last, the disaster management cares on the one hand the monitoring and the assessment of the incoming data, and on the other hand the alerting and clearing up the population.

2.2 Environmental Monitoring

Environmental monitoring is one of the five major applications for wireless sensor networks [20]. By the use of sensor networks, to monitor the environment, the costs are significantly

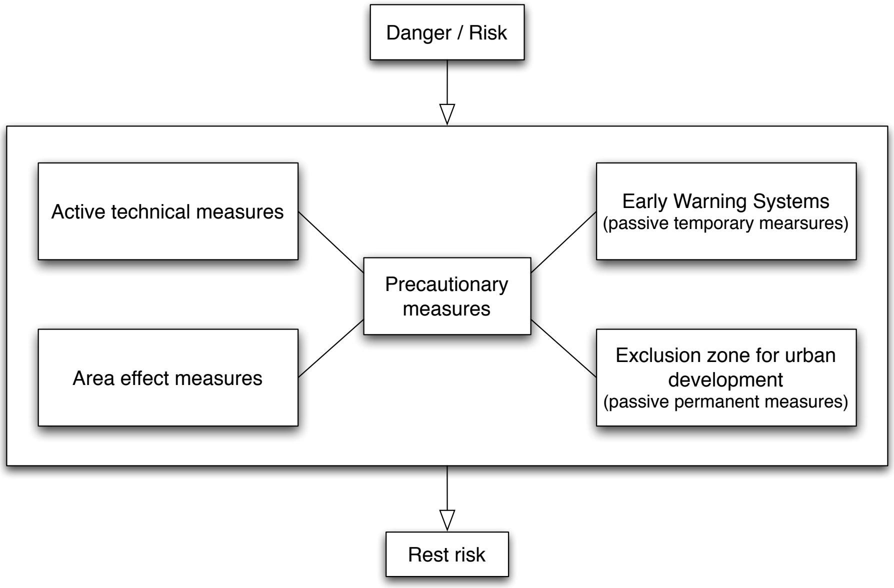


Figure 2.1: Danger – precautionary measures – rest risk [1].

reduced. The engaged system for monitoring environmental conditions vary in scale and function. Figure 2.2 shows the different types of environmental sensor networks.

Localized multifunction sensor networks are relevant for this thesis. In Section 2.3 such types of environmental monitoring systems are discussed. These ad-hoc networks contain small nodes with generic sensors to monitor the area in more detail. Related to sensor networks, there are three matters that we need to examine more closely: scale, variability and autonomy [21].

The term scale deals with the sampling interval, the system coverage and the number of sensor nodes. The smaller the factor of the sampling interval, the more detailed and innovative information can be collected. An important issue the sampling frequency are the used sensors. Temperature or humidity data may not need such a fine measurement resolution as a GPS sensor data for further processing. Another point is that higher data granularity allows for more exact geological models.

The system coverage is given by the size of the monitoring area. In fact, the spatial and temporal extents of systems varies widely.

Density deals with the number of nodes in the network. Basically, this is affected by extents of the system and of the network coverage. Also distributed sensing can be associated with the term density [22].

An important issue for localized multifunction sensor networks is variability. Variability describes how flexible and stable the network is when there are changes in the system structure. The variability increases if the system is tolerant against broken nodes or if

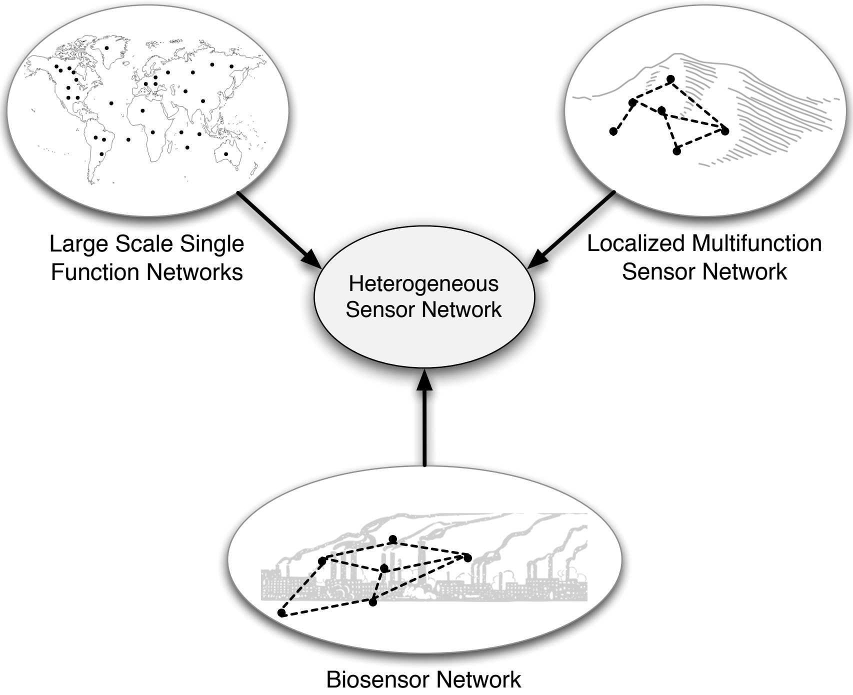


Figure 2.2: Types of environmental sensor networks.

the system can handle new nodes in the network. In more detail, the stronger the routing protocol the more changeable is the whole sensor network [23].

The most significant aspect for sensor networks is the degree of autonomy: maintenance-free, energy-self-sufficient and internal processing. Such an autonomous system increases the need for additional hardware, e.g. energy harvesting, further sensors and translation between external request and internal processing.

2.3 Environmental Wireless Sensor Monitoring Systems

This section deals with several state-of-the-art early warning systems. Section 2.3.1, Section 2.3.2 and Section 2.3.3 discuss those projects. In Section 2.3.4, GPS based monitoring systems are briefly mentioned. Further, in Section 2.3.5 a comparison of the described projects with the GeoWSN project is given.

2.3.1 Alpine Early Warning System

The Alpine Early Warning System (alpEWAS), shown in Figure 2.3, is a 3D monitoring and early warning system for landslides.

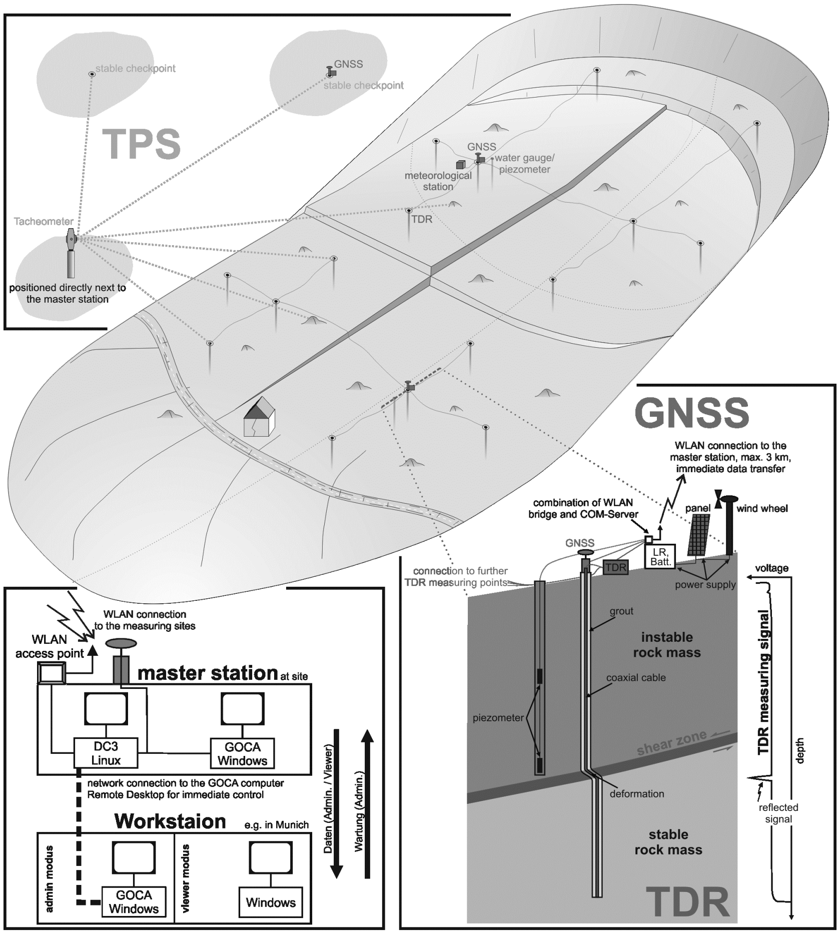


Figure 2.3: alpEWAS system overview [2].

An important issue of this project is the evaluation of innovative and economical measuring systems to detect deformations on surface and soil:

Time-Domain-Reflectometry (TDR). With a time-domain-reflectometry system it is possible to perform several measurement tasks. Such systems are used in earth and agricultural sciences to determine the soil moisture, in aviation wiring maintenance, in level measurement, in semiconductor device analysis or in geotechnical usage [24]. The latter one was examined in the alpEWAS project. For the detection of a soil deformation the TDR system consists of two components. One the one hand the TDR measurement device that is called cable tester and on the other hand a coaxial cable. The cable tester operates as transceiver (emits electric impulses) and receiver (receives electric impulses). At a deformation of the coaxial cable, the impulse is reflected back earlier in time to the receiver. With the help of this information it is able to analyze the depth of the deformation respectively to detect a slope movement.

Reflectorless Robot-Tacheometry. With the help of this system it is able to generate 3D models of the environment. For the detection of movements or deformations an exact 3D model of the initial epoch is needed. This model is generating from a Transport Layer Security (TLS) by scanning natural objects, e.g. rocks. After the initial period a Terrestrial Positioning System (TPS), the Reflectorless Robot-Tacheometer is used [25]. The tacheometer uses an integrated camera to detect contours of natural targets. Another use of the integrated camera is to realize an autonomous vision system by training a robot to search through suitable targets with the image information. By comparing the output model of the TPS with the 3D template model positional changes in space can be detected [26].

Low-cost Global Navigation Satellite System (LC-GNSS). The aim of this part was the integration of a LC-GNSS with an autonomous power supply. The system provides four stations, where three are installed in the surveillance area and the fourth one is located in a presumably stable range near the base station. For a high accuracy in space, post-processing is needed because such low-cost systems do not support RTK [27]. To achieve an accuracy of a few millimeters a continuous raw data stream in a 15 minutes interval is used [2].

Furthermore, a piezometer, a precipitation and a temperature sensor are integrated in the system. With the help of those, statements about the hydrostatic pore pressure and the climate conditions can be made [28].

2.3.1.1 Data Communication

For transmission of the sensor raw data to the network base station the standard Institute of Electrical and Electronics Engineers (IEEE) 802.11b/g is used. Because of the higher performance of the Wireless Local Area Network (WLAN) transmission ($11\,Mbps$) this standard is preferred over an ordinary radio data transmission ($9.6\,kbps$). Another argument for using the IEEE 802.11b/g standard is the data packet size of $274.5\,kByte$ of the GPS receiver by an epoch length of $15\,min$. So with the parameters mentioned before, the transmission by using an ordinary radio data transmission standard takes approximately $4\,min$, in comparison to $0.2\,sec$ when using the WLAN standard [29].

2.3.1.2 Power Management

An important property of the monitoring system is the secure power supply for all sensor nodes. A regular power supply over the grid is only given at the base station. The other sensor nodes in the network are stand-alone and need an autarkic power management. The power management system is based on rechargeable back-up batteries which can be

charged over a solar cell or as back-up solution by a fuel cell.
In the operation mode the GNSS node has a power consumption of about $2.9\,W$. With the adapted power management system a stand-alone operation without recharging is possible for an epoch length of about 22 days [29].

2.3.1.3 Data Processing

The raw data of the sensors mentioned above are transmitted directly over WLAN to a data management system on the base station. The data management system of the alpEWAS project consists of three components. First, with the Central Control Application (CCA) it is able to control and monitor the whole data management and data analyze system. The second component and central component of the data management system is the MySQL database. The last part is the sensor plugins which function as connector between the database and the sensor network. At this level a first interpretation of the received sensor raw data and the sensor state controlling takes place [30].

2.3.2 Sensorbased Landslide Early Warning System

The Sensorbased Landslide Early Warning System (SLEWS) is a landslide monitoring and early warning system based on a wireless sensor network shown in Figure 2.4. In the SLEWS project on the one hand the data quality and the detection of landslides should be improved, and on the other one the error rates of false alarms should be minimized. To reach this aim the wireless network must fulfill the following terms [14]:

- real-time ability
- self-organization and self-healing capacity
- energy efficiency
- bidirectional communication skills
- data interfaces

The network consists of several sensor nodes and one gateway (data collection point). Each component of the wireless network consists of an energy module, a solar panel and a sensor module. Furthermore, the gateway node is equipped with a Global System for Mobile Communications (GSM)/General Packet Radio Service (GPRS) module. With the attentional GSM/GPRS module a second option to transfer the collected sensor data to a processing unit is given, the other one is that the gateway node is connected directly with this unit.

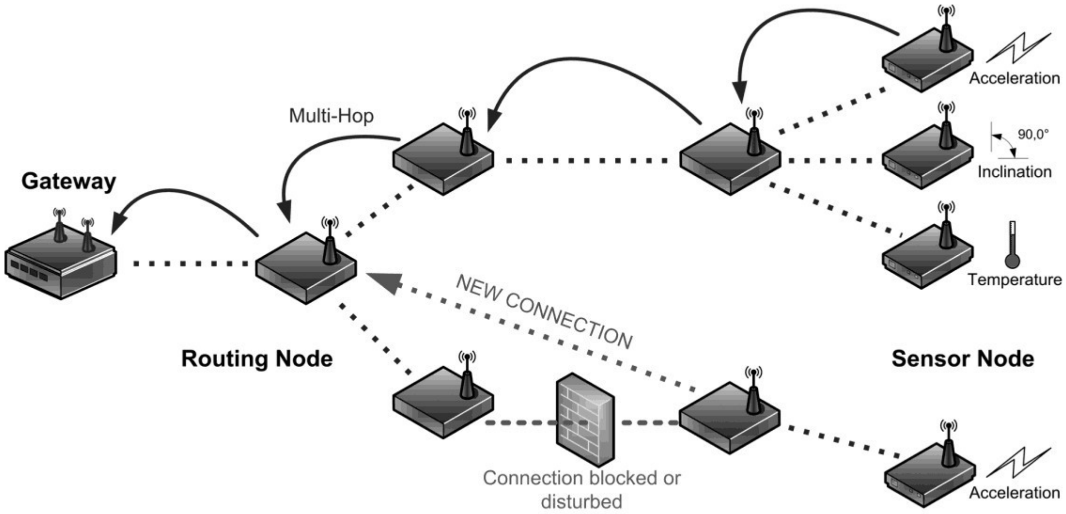


Figure 2.4: Structure of the SLEWS wireless sensor network [3].

2.3.2.1 Network and Nodes

The wireless sensor network is based on an ad-hoc network, which acts in a self-organizing manner. The distributed sensor nodes interact automatically with the network neighbor with the best link strength. This communication is needed to exchange the collected measurement data directly or over another node (multi-hop) to the gateway node. The multi-hop functionality allows long-range transmissions and reduces the transmission power [31].

The SLEWS wireless sensor network operates in the 868 MHz frequency band. Within this frequency band, 30 channels are used to avoid noise. By sending radio signals data rates from 4.8 $kbits/s$ up to 115 $kbits/s$ at a transmission range of 10 m to 1.2 km are reached [32].
Thereby, each wireless sensor node is organized in a strongly modular way with open sensor interfaces. With the help of this structure it is possible to integrate different sensors.

2.3.2.2 Sensors

To get information about surface and subsurface deformations, a permanent monitoring with precise measuring instruments is used. For this the sensor board contains a Microelectromechanical Systems (MEMS) [33]. Microelectromechanical Systems are small, precise and inexpensive which are positive aspects for the integration in a Wireless Sensor Network (WSN). The MEMS consists of sensing elements, transducers and mechanical and electronic units combined on a small microchip [3]. The MEMS, which are selected in the SLEWS Project are a 3D acceleration sensor, two different 2d tilting sensors and a barometric pressure sensor. These sensors are equipped with an integrated temperature sensor. Further, two position sensors for monitoring the opening and closing of fissures, a potentiometric displacement transducers and a linear magnetostrictive position transducers are integrated in the project [32].

Table 2.1 shows the main slope failure mechanisms and which sensor is able to monitor those deformations.

Table 2.1: Slope failure detection with different sensor mechanisms [14].

	Acceleration Transducer	Displacement Transducer	Angle Sensor
Falling	yes	no	maybe
Topple	yes	yes	yes
Rockslide	yes	yes	no
Rotational slide	yes	yes	no
Shallow translation	yes	yes	no

2.3.2.3 Sensor Fusion

To achieve a reliability of sensor information in terms of hazard monitoring and early warning, the SLEWS project uses sensor fusion [34]. In this method the sensor information of different devices is combined. That means by the use of sensor fusion the data of adequate sensors are compared, so a better interpretation of those data and a suppression of malfunctions are possible. The authors of [32] describe different approaches to implement and integrate sensor fusion which are listed in the following:

Redundant sensor fusion. The idea here is the integration of identical sensors on the same board.

Complementary sensor fusion. This approach uses the combination of different sensor types on a node. Therefor the system uses two or more sensors, which observe the same failure mechanism, like acceleration and inclination sensors. Applying this kind of sensor fusion no redundant sensor is necessary.

Spatial sensor fusion. Here the sensor information from distributed nodes is compared.

Network fusion or data Fusion. Fusion of different sensor networks to compare the gathered sensor data [35].

2.3.3 X-Sense Project

The aim of the X-Sense Project is to develop a wireless sensing technology for environmental monitoring. Within the project, a wireless network of low-cost L1 GPS equipped sensor nodes is built. The sensor nodes acquire the GPS raw data information for a further Differential Global Positioning System (DGPS) processing. The whole X-Sense system is based on an end-to-end system concept [4]. The concept includes the data acquisition, the data handling and processing, and the data application. The three sub-systems are briefly described in the following.

Data Acquisition. This system consists of a WSN with low-cost GPS equipped sensor nodes. The collected GPS raw data information of each node is sent over a multi-hop based radio communication to the base station. The base station acts as gateway between the wireless sensor network and the back-end infrastructure.

Data Handling and Processing. The data handling and processing system stores, processes and presents the acquired data to an end application. To reach a high spatial accuracy the process task performs a complex DGPS computation. Relational databases are used for storage. The processed information in the databases is published through a web interface.

Data Application. The data acquired from the wireless sensor network is published to the end user. With the position and velocity information combined with other sensor data, like temperature and high-resolution imagery, a geoscientific model of movements is possible [36].

2.3.3.1 Network and Nodes

The architecture of the wireless sensor node is shown in Figure 2.5. The main parts of the node are the ultra-low power wireless sensor module (Shockfish SA TinyNode184 [2]) and a low-cost L1 GPS receiver module (u-blox LEA-6T [3]).

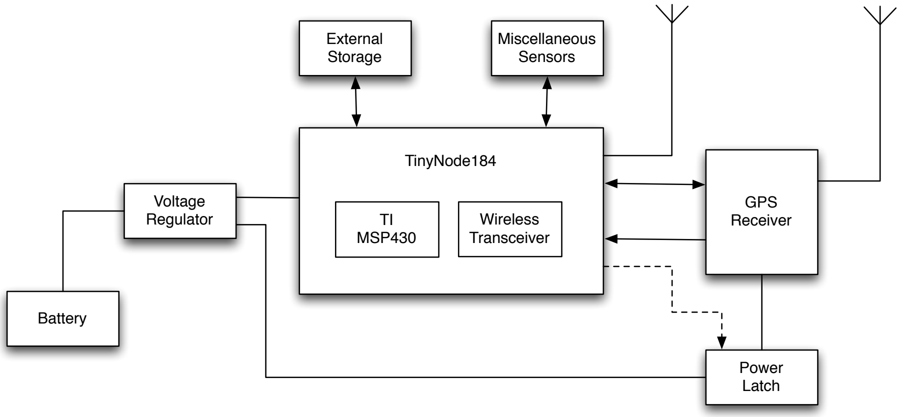


Figure 2.5: GPS-equipped wireless sensor node (reproduced from [4]).

The TinyNode184 comprises a MSP430 micro-controller and a 868 MHz wireless transceiver. The application software running on the micro-controller is implemented in TinyOS 2.1

---

[2] TinyNode184 Fact Sheet (effective [February 14(th), 2013])

[3] u-blox LEA-6T Product Summary (effective [February 14(th), 2013])

[37]. The communication in the network uses a low-power, low-bandwidth wireless network protocol stack called Dozer [38]. With the Dozer network protocol the communication from each node to the base station via multi-hop is possible.

The architecture of the sensor node integrates an Secure Digital (SD) card as an external storage device for redundant storage of measurement data. Also an interface for further specific sensors/actuators is defined.

2.3.3.2 GPS Module

The GPS module is a low-cost, off-the-shelf single frequency L1 GPS receiver and is qualified for precision timing applications. This module provides the raw data satellite information of the visible satellites in addition to the GPS position message. Over the Universal Asynchronous Receiver Transmitter (UART) interface the transfer of data and commands can be realized. An important issue in multi-hop networks is the time synchronization. This can be handled with the help of the GPS module, because the module provides an exact Coordinated Universal Time (UTC). With the help of this, all nodes in the network are synchronized [4].

The GPS module supports two different modes of measurement retrieval. In **polled mode** the micro-controller requests data by using a so called request packet. Whereas when the GPS module operates in **periodic mode** the data are supplied at a defined interval.

2.3.3.3 Power Management

The power analysis of stand-alone sensor nodes is necessary, because of a predication about the longevity of the device in the field. An important performance metric of the power analysis is the power dissipation of the node. The radio transceiver of the TinyNode184 and the GPS module are the main contributors to the power dissipation. To optimize the power consumption of the GPS module the Power Optimized Tracking (POT) mode is activated. The module has an average power dissipation of $137\,mW$ by a ratio of 1:5 between POT high and POT low. By enabling the low-power operation mode of the MSP430 micro-controller the main consumer of power of the TinyNode184 is the radio transceiver. The power analysis of the TinyNode184 shows that the main contributing components are the GPS data transmission, the beacon transmission to child nodes and the beacon reception from the parent and last the Dozer protocol, TinyOS, and data processing overhead. The power consumption of the TinyNode184 radio transceiver is when measurement is active $0.93\,mW$ and when measurement is inactive $0.37\,mW$. To summarize, the approximate power consumption of the node is $137.93\,mW$ when the GPS module is active, otherwise the power consumption is $0.37\,mW$ by a sampling interval of

30 sec [4]. This shows that the power dissipation of the GPS module is the major part.

The lifetime of the stand-alone sensor node is about 62 days using a 14000 mAh standard battery cell [39]. This lifetime can be achieved by optimizing the overall power dissipation due to POT, interrupt driven design, and synchronized measurement duty cycling.

2.3.4 Further GPS-based Monitoring Systems of Landslide Movements

In this Section further GPS-based monitoring systems for the detection of slope movements are discussed. First, a monitoring system with several mobile and one fixed GPS receiver unit is briefly described. The second system which is mentioned, deals with the approach where multiple antennas are connected to one receiver.

2.3.4.1 Permanent, autonomous monitoring of landslide movements with GPS

The system called Movement Measurement System (MMS) consists of several mobile GPS receiver units and at least one fixed GPS receiver unit, the reference receiver. To monitor the vulnerable area the mobile GPS receiver units are installed in this area. The GPS receiver is integrated in a measurement station.

The implementation of the communication protocols, the management of the measurement tasks and the data acquisition is done by the Central Processing Unit (CPU). The multi-task program, running on the CPU, takes care of reducing the power consumption of the remote sensor node by providing a doze state. The single-frequency GPS receiver, integrated in the sensor node, also provides a low power mode. An important issue in this project for getting precise position data is the stability of the local receiver reference clock. Such a high quality clock is needed because carrier phase measurement is used for positioning. The remote sensor node provides three additional analog input and two digital input ports. With those it is able to add additional sensors. Furthermore, for complex external sensors an RS232 serial port is provided. To achieve a total autonomous operation the remote sensor is fitted with a solar panel and an accumulator. This two components are connected with the power management unit and allows that the sensor node operates essentially maintenance-free [40]. The focus of the project is the monitoring of large structures. To fulfill this requirement a GSM module is used as communication interface [41]. The GSM module transmits the raw data to the control center. The control center stores the data, post-processes the GPS raw data and illustrates the data. In the post-processing state the relative position to the reference station is calculated, with which it is feasible to detect movements. The control center is reachable over a remote access and it is able to define system parameters, such as the measurement interval, sensor parameters or the alarm setup over the configuration screen.

2.3.4.2 Application of Multi-Antenna GPS Technology in Monitoring Stability of Slopes

In this project a new approach by integrating a multi-antenna GPS technology is used [42]. The multi-antenna GPS technology is bringing several benefits with it, like reducing the cost of GPS or lower power consumption. The system is structured in the following major components:
The **GPS multi-antenna control** unit switches between multiple inputs to one single output channel. This means that a number of GPS antennas are sequentially routed to one GPS receiver. Within the configured time interval the selected antenna can forward the data via the receiver to the communication interface. In the current design the maximum number of **GPS antennas** is limited to eight. To guarantee the data quality, the distance between the antenna and the **GPS receiver** should not exceed 120m. For longer distance transmission **signal amplifiers** are necessary. Via the **data link**, e.g., GPRS the GPS data are transmitted to the data procession and analysis center.
To achieve precise positions the DGPS [43] method becomes important. For DGPS one reference station with an exact known position is used to get information of the atmospheric influences. With the help of this information it is possible to get a precise position of the measurement stations in the network.
With the use of the multi-antenna GPS technology and the Differential Global Positioning System method a measurement accuracy in the three coordinate components of about $2\,mm$ can be achieved [42].

2.3.5 Comparison between State-of-the-Art Projects and the GeoWSN Project

First, in this section an overview of the wireless sensor node hardware[4] used in the GeoWSN project is given. Finally, a number of the previously described environmental wireless sensor monitoring projects are compared with the GeoWSN project.

2.3.5.1 Hardware of the GeoWSN Wireless Sensor Node

The GeoWSN wireless sensor node hardware [5], illustrated in Figure 2.6, is driven by a 16-bit Texas Instrument MSP430F5438A micro controller and equipped with several micro chips and interfaces.
The ultra low power TI MSP430F5438A micro controller, which has $16\,kB$ Static Random-Access Memory (SRAM) and $256\,kB$ program flash, is powered by a $16\,MHz$ crystal oscillator. The oscillator is needed due to the high data rates of the GPS receiver.

---

[4] developed by the Institute for Technical Informatics of the Graz University of Technology

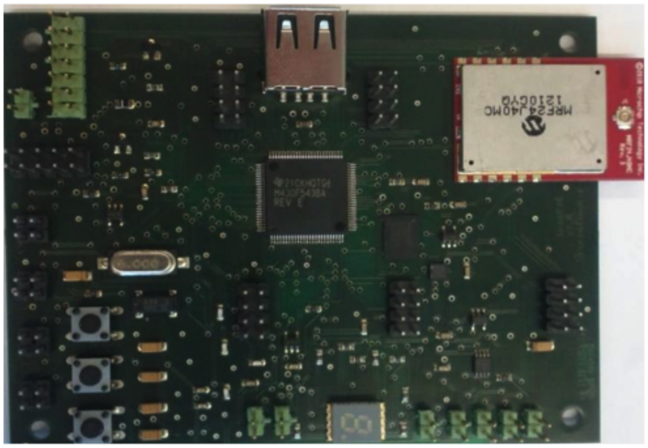


Figure 2.6: GeoWSN wireless sensor node [5].

The Microchip MRF24J40 wireless radio module operates in the $2.4\,GHz$ Industrial, Scientific and Medical (ISM) band and is compatible to the IEEE 802.15.4 communication standard. The module supports amongst others the ZigBee network protocol that is preferred in the GeoWSN project. This radio module provides several Carrier Sense Multiple Access/Collision Avoidance (CSMA-CA) features, like automatic retransmit, and automatic acknowledge response.

The over a Universal Serial Bus (USB) interface powered U-blox GPS receiver which gathers the satellite information is connected by a serial UART to the micro controller. With the serial UART interface it is also possible to configure the GPS receiver.

The sensor node board includes some sensors, e.g. temperature, and acceleration and interfaces for external sensors, e.g. humidity. The internal and external sensors are connected by an I2C bus to the MSP430.

The board of the gateway node of the wireless sensor network has an RS232 board additionally mounted, to support a communication to the server.

The wireless sensor network board also provides an interface for the usage of a GSM/GPRS module. This module supports a direct data transfer to the server.

2.3.5.2 Tabular Comparison

Table 2.2 gives a comparison of several state-of-the-art environmental monitoring system and the GeoWSN project.

Table 2.2: Tabular comparison of monitoring and early warning systems for critical slopes.

	GeoWSN	alpEWAS	SLEWS	X-Sense
Energy Source	solar cell, thermo-electric (intended), back-up batteries	solar cell, back-up batteries	solar cell, power grid	solar cell, back-up batteries
Communication module features	$2.4\,GHz$ $1\,km$ LOS CSMA-CA, auto ACK response, auto packet retransmit, AES-128	$2.4\,GHz$ $3\,km$ LOS max $54\,Mbit/s$	-	$868\,MHz$ $1\,km$ LOS data whitening, auto CRC, packet handling
Communication protocol (sensor network)	IEEE 802.15.4 based GeoWSN network protocol	IEEE 802.11b/g	IEEE 802.15.4	Dozer network protocol [38]
Sensors	3 temperature humidity luminance acceleration	TDR TPS piezometer precipitation	acceleration tilting way receiver altimeter temperature barometer	temperature humidity
GPS receiver	low-cost L1 GNSS	low-cost L1 GNSS	-	low-cost L1 GNSS
Positioning algorithms	PPP, RTK	NRTP (Near Real Time Processing)	-	RTK
Position accuracy	sub-centimeter	sub-centimeter	less than 10 centimeter	sub-centimeter
Monitoring system	web application (with configuration options), remote access (for administrator)	remote access (for administrator) web application (not published)	web application	web application
Warning system	XMPP push message (mobile client), near-real time online messages (desktop client)	-	XMPP push message (desktop client)	-
Mobile monitoring system	Android application	-	-	-

The environmental monitoring systems are compared by reference to several facts. All of the systems use a solar cell as energy source. All systems, except the SLEWS project, use rechargeable accumulators as back-up systems. The back-up solution of the SLEWS project is powered by the power grid.

The alpEWAS system uses the IEEE 802.11b/g standard, where the other systems are using the IEEE 802.15.4 standard as communication standard.

The environmental monitoring systems are equipped with commercial low-cost sensors. In addition to those the alpEWAS system also provides TDR and reflectorless robot-tacheometry as TLS.

To determine the position of the sensor nodes the GeoWSN, the alpEWAS, and the X-Sense project are using a low-cost L1 GPS receiver with post-processing algorithms. The SLEWS system uses the change of the network topology, extracted from the parameters of the radio communication and the commercial sensors, like way receivers to detect motions within the network. Based on the used technology the SLEWS can perceive motions in the range of less then ten centimeters, where the other systems have an accuracy in the sub-centimeter range.

Only the GeoWSN and the SLEWS project are integrating a public warning system in case of a slope failure. Both systems are based on a near-real time instant messaging and presence technology, the XMPP. The difference between the warning systems is that the alerting message of the GeoWSN warning system is displayed on a mobile device (smartphone) and the SLEWS warning system uses a desktop computer as front-end. In addition to the XMPP alerting service the GeoWSN system also supports near-real time alerting messages to a desktop computer, if the user is online.

Each project includes a monitoring system, which is available over a web application. The web application of GeoWSN system also provides configuration possibilities, like threshold settings, or user settings for the administrator. The administrator of the GeoWSN and the alpEWAS project has a remote access to the applications of the whole system.

The GeoWSN project supports additionally to the web application a mobile monitoring system. This system is implemented on the Android platform and gives an overview of the gathered parameters of the network.

2.4 Instant Messaging and Presence Technologies

The term Instant Messaging deals with the fact that a message exchange between online users can be realized in real time [44]. The feature real time communication separates instant messaging systems from an email system. An important part of an instant messaging system is that the user can determine if his contacts are online. This key feature is enabled by the Presence Technology.

The IM&P can be described by a generic model [45]. Within the model two services are defined:

The **presence service** accepts, stores and distributes presence information. Figure 2.7 displays the two distinct types of clients: *presentity* and *watcher*. Further information, which is displayed, is the information flow to the presence service. The presence information is provided by the presentity to the service, where the watcher receives presence information. The watchers can be classified in two kinds, the *fetchers* and the *subscribers*. The fetcher requests the presence information from the service. A special kind of a fetcher is the *poller* which is requesting those information on a regular basis. As opposed to this the subscriber subscribes to presence information of a presentity. If there is a change in the presence information of the special presentity the presence services transmits a notification to the subscriber [6].

The interaction between the service and the two clients is defined by the presence protocol [45].

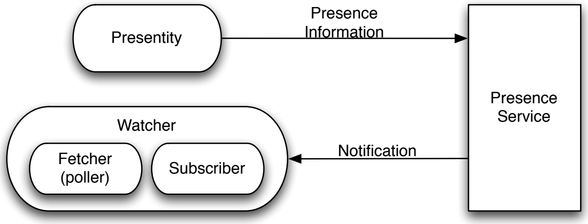


Figure 2.7: Overview of presence service [6].

The **instant message service** communicates, like the presence service, with two clients. On the one hand the instant message service, which is shown in Figure 2.8, communicates with the *sender*, which is the source of the instant message, and on the other hand with the consumer of the instant message, the *instant inbox*.

In essence, the instant message service tries to deliver the instant message, which was accepted from a sender to the addressed instant box. This communication is based on the instant messaging protocol [46].

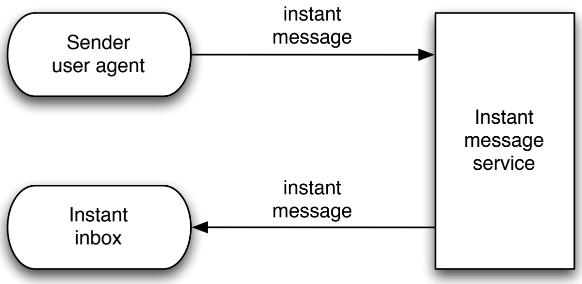


Figure 2.8: Overview of instant message Service [6].

Real-time messages and presence information are the characterizing technology of instant messaging, which distinguishes this communication service from other ones.

In Section 2.4.1 and Section 2.4.2 two standards of the instant messaging and presence technologies will be described.

2.4.1 Extensible Messaging and Presence Protocol

The XMPP is an Instant Messaging (IM) Protocol and is used for streaming Extensible Markup Language (XML) elements [47]. In essence, with XMPP it is possible to transmit XML elements from one subscriber to another in near-real time.

The Extensible Messaging and Presence Protocol provides several services and applications [15].

2.4.1.1 Architecture

XMPP uses similar the same architecture like the World Wide Web and the email architecture, a decentralized client-server architecture, which is shown in Figure 2.9. The XMPP architecture consists of three components: XMPP server, XMPP client and gateways. The server handles the connection management between authorized servers, clients and other entities in form of XML streams. The client is starting the XML Stream and also functions as end user. The gateway functions as a bridge between the XMPP network and a foreign network, like Short Message Service. Therefore, the gateway must translate between XMPP and the other foreign messaging protocol [7].

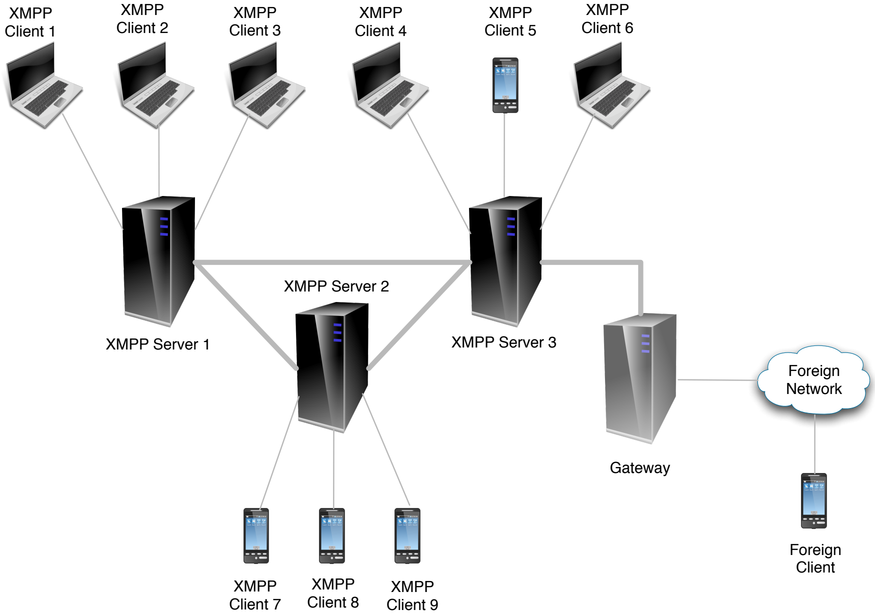


Figure 2.9: Decentralized client-server architecture.

Due to this architecture the system is not influenced by a failure of a single point. Another benefit behind the distributed client-server architecture is that the servers are able to enforce security policies. Table 2.3 shows the architectural differences between the World Wide Web, the email and the XMPP architecture:

Table 2.3: Client-server architectures [15].

Architecture	Interdomain Connections	Multiple Hops
World Wide Web	no	n/a
Email	yes	yes
XMPP	yes	no

2.4.1.2 Streams and Stanzas

The term XML stream is defined as *"a container for the exchange of XML elements between any two entities over a network"* [48]. Before a XML stream can be established, the underlying Transmission Control Protocol (TCP) connection must be opened. One TCP connection can handle multiple sessions (context in which instant messages are exchanged). Due to this a unique identifier (address), the so-called Jabber Identifier (JID) is needed. The JID consists of three parts, the local part (username) which must be unique within the domain, the domain and the resource [49]. The resource is necessary to separate between different connections of one user [50]. The JID resembles the following pattern: user@domain/resource.

As mentioned before the communication of two entities takes place within a XML stream. This connection can be used for exchanging XML data elements or else XML stanzas [51]. The XMPP technology comprises three core stanza types [52]:

1. <message/>
2. <presence/>
3. <iq/>

2.4.1.3 Encryption and Authentication

The XMPP technology integrates security methods for XML streaming to avoid tampering and eavesdropping. Figure 2.10 shows the two used encryption and authentication protocols for a secure connection and communication to achieve data integrity and mutual authentication [7], the Transport Layer Security (TLS) protocol [53] and the Simple Authentication and Security Layer (SASL) protocol [54]. For the channel encryption the TLS protocol is used. The SASL protocol does the authentication.

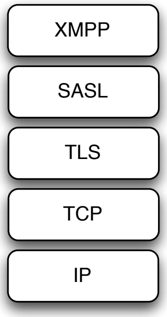


Figure 2.10: Security layers of the XMPP technology [7].

2.4.2 Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions

The Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) based on the Session Initiation Protocol (SIP) [55]. The aim of SIMPLE was to define an extension to integrate instant messaging into the SIP standard. Therefor a presence protocol which instantiates the usage of SIP for notification and subscription and also acts as a signaling protocol is defined [56].

2.4.2.1 Architecture

The architecture of the SIMPLE technology is a distributed model, like the XMPP architecture. In contrast to a centralized where the whole traffic, like registration and lookup tasks is handled by one server, in a distributed architecture the traffic is managed over a network of servers. Table 2.4 compares both models.

Table 2.4: Advantages and disadvantages of the distributed architecture [6].

	Centralized architecture	Distributed architecture
Performance	limited	good
Attack vulnerability	high	moderate
Administration / Maintenance	easy	moderate
Scalability	limited	good

2.4.2.2 Instant Messaging

SIMPLE provides two instant messaging modes, the page mode and the session mode. In page mode the message can be exchanged outside a session (*context in which instant messages are exchanged*). This means that there are no association between each instant message and the other messages between the communication partners. If a conversion takes place in the context of a session the session mode is used. By the usage of this mode the explicit conversation has a clear beginning and end in contrast to the page mode. The use of the Message Session Relay Protocol (MSRP) [57] brings several benefits with it, such as direct peer-to-peer operations, high security, high privacy and explicit rendezvous.

2.4.2.3 Presence

The availability of an user to communicate is indicated by the presence information. This information (presence status) is gathered by a presence service. Therefor each entity transmits the presence information to the presence server. The architecture of the presence technology for SIMPLE is shown in Figure 2.11.

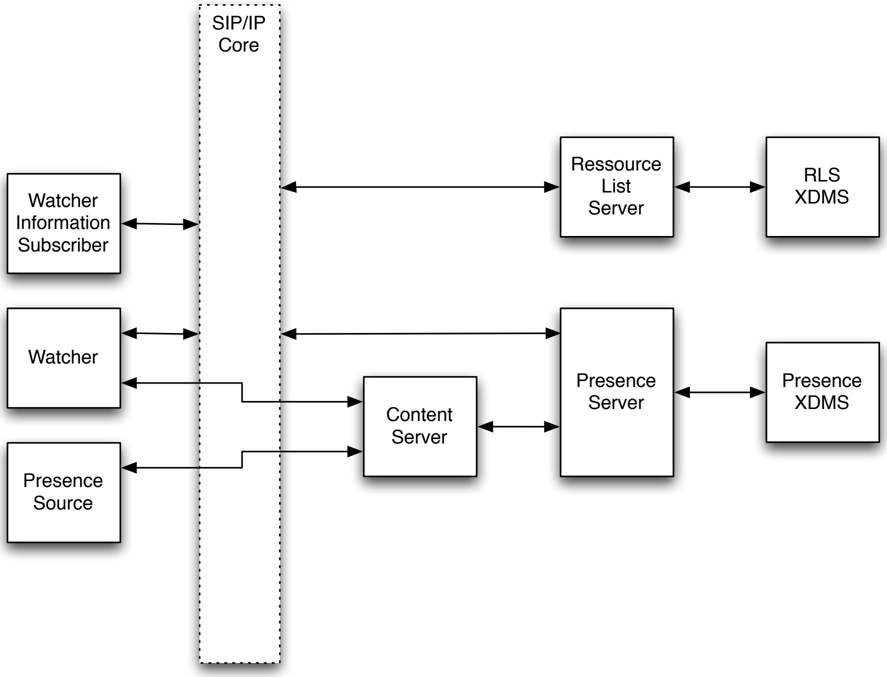


Figure 2.11: SIMPLE presence architecture [8].

On the one hand the Presence Server accepts and stores presence information published by authorized Presence Sources and on the other one the server provides the information for the watchers. As mentioned before, the Presence Source (an user or any network entity) makes its presence information available. The entity, which requests information about the presentity from the Presence Server, is called Watcher. The Resource List Server manages the presence list and thus enables the watcher information exchange of multiple entities with one transaction. With the help of the Content Server it is able to manage Multipurpose Internet Mail Extension (MIME) [58] for presence [59].

2.5 Summary

In this section related work has been analyzed. First, general information about hazard assessment and early warning systems has been given. In this part the main components of an early warning system were described. Then, environmental monitoring concepts that based on wireless sensor networks have been discussed generally. Here the basic structure and the main aspects of the sensor network design have been mentioned. The next section has been dealt with several state-of-the-art environmental wireless sensor monitoring systems. Therefore, these systems, like alpEWAS, SLEWS, the X-Sense project, and some GPS-based monitoring systems, have been described briefly. In the following a comparison between some state-of-the-art project and the GeoWSN Project has been given. Finally, two instant messaging and presence technologies, XMPP and SIMPLE has been discussed.

The aspects, which have been mentioned before, are influencing the design of an early warning and monitoring system that is based on a database and visualization system for critical slopes.

Chapter 3

Design

This chapter focuses on the different development aspects of a database and visualization system for landslide monitoring. Section 3.1 describes the system behind the GeoWSN project in general. In Section 3.2 different data processing daemons and services are mentioned. Section 3.3 discusses the implementation details of the database. The visualization is realized by two different approaches. The first approach, a web application with different visualization and configuration possibilities is explained in Section 3.4. The enhanced MMC is described in Section 3.5. This communication system is used as interface between the server and the mobile client. Finally, Section 3.6 gives an overview of the native application running on the mobile client.

3.1 System Architecture

This section focuses on two main parts, first the gateway node architecture and second the architecture behind the GeoWSN server. The first section describes the GeoWSN system. The system components and the interaction between the components are also discussed in this section. Further, the gateway node architecture and the requirements of this are mentioned. Finally, the architecture of the database and visualization system is described. The requirements that this system must fulfill are also given in the section.

3.1.1 GeoWSN System Overview

The components of the GeoWSN system and the interaction between them are shown in Figure 3.1. The system uses a four components design which consists of a GPS-equipped wireless sensor network, the (GeoWSN) server, the MMC and the front-end applications.

GPS-equipped wireless sensor network. The back-end of the GeoWSN system is the wireless sensor network. The network consists of several nodes with different sensor,

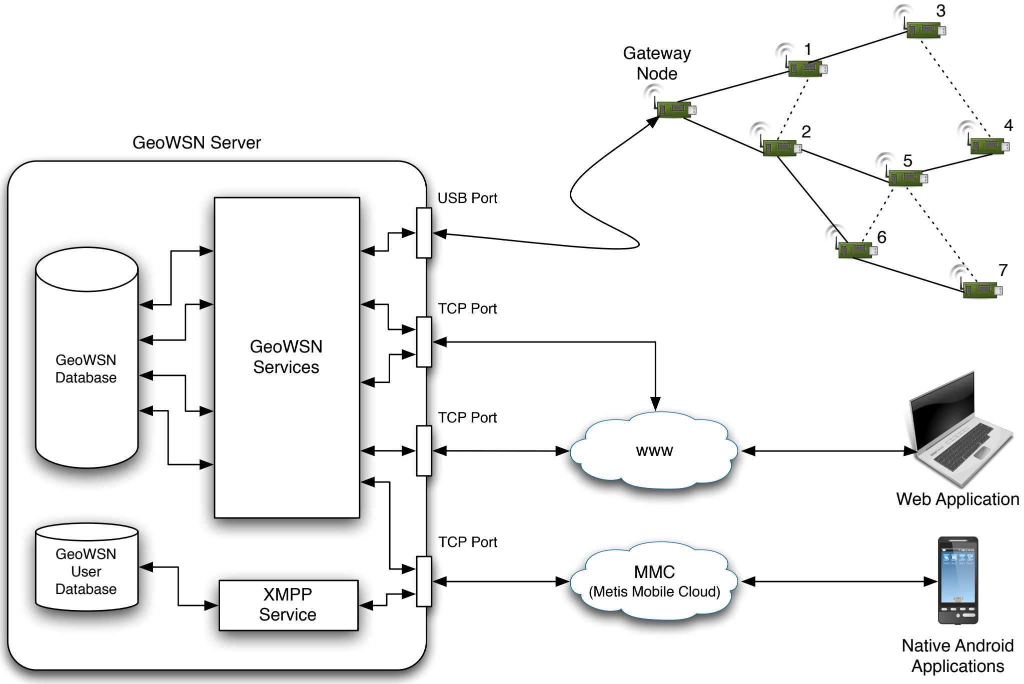


Figure 3.1: GeoWSN system architecture.

e.g. temperature, and acceleration. One of the most important component of the sensor node is the low-cost L1 GPS receiver. The GPS-equipped sensor nodes gather satellite and sensor raw data information. Over multi-hop routing the data are transmitted to the gateway node (additionally includes an USB interface) by radio. For the exchange of information the gateway node is directly connected via USB to the GeoWSN server.

GeoWSN server. The GeoWSN server is the main component of the GeoWSN database and visualization system. The GeoWSN server contains a number of services and databases. The services are processing and parsing the raw data, determining the position of the sensor nodes, and making the solutions available to user applications. The parsed data from the sensor network as well as the data for the positioning service and the calculated positions are stored in a database. Further, the user information for the instant messaging and presence service are held in a database.

Metis Mobile Cloud. The Metis Mobile Cloud is used for reliable near-real time communication between GeoWSN server and the mobile client. The MMC contains two main components a XMPP Server and an Android client, including a XMPP application and a client application. The components interacting like a peer-to-peer connection over the World Wide Web.

Front-end applications. As front-end application two approaches are provided, the web application and the native android application. The native android application gives an actual overview of the system parameters, like status, environment parameters, and positions. Due to the communication over the MMC, a near-real time alerting is possible. The other option is the web application, which brings along visualization and configuration possibilities. By using the web application useful parameters in the space of the critical slope can be observed in several intervals and different configuring settings can be made.

3.1.2 Gateway Node Architecture

The software of the gateway node is an extended sensor node software version of the other nodes in the WSN. Additional to the main components – the radio unit, the processing unit and the measurement unit – the gateway node integrates a serial communication unit. Figure 3.2 illustrates the software parts of the gateway node.

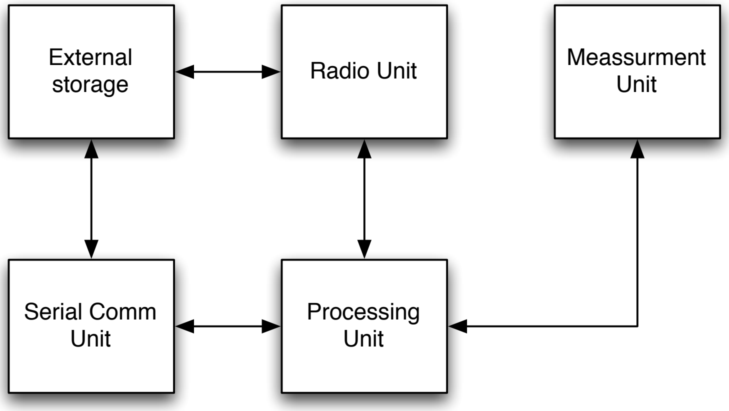


Figure 3.2: Software architecture of the gateway node.

A number of requirements on the gateway node are presented to ensure a safe operation of the whole system.

Fault tolerant. A monitoring and alerting system uses fault tolerant components. Therefore, the gateway node must integrating fault tolerant mechanisms. Since the gateway node only distinguishes itself by serial communication unit from the other nodes in the network such mechanisms should be included for this unit.

Near-real time. To achieve an actual and precise position of the sensor nodes a near-real time communication between the gateway node and the server is necessary.

In the following the components of the software architecture, shown in Figure 3.2, are briefly described.

Measurement Unit. The *measurement unit* gathers periodically the information from the environment with the I2C connected sensors of the node. In this part also the satellite raw data are received from the GPS sensor, which is connected over a serial interface.[5]

Radio Unit. The radio unit contains the routing and network functionality, and provides the methods for sending and receiving messages. In case of a communication problem via the radio interface, the messages are stored in an external flash memory.[5]

External Storage Unit. The hardware of the external storage are enabled via the external storage unit, the flash driver.[5]

Processing Unit. The *processing unit* manages the incoming messages and reacts to them.[5]

Serial Comm Unit. The *serial comm unit* is a COM driver, which is necessary for the communication to the GeoWSN server. By the use of this unit, the processing unit can directly transmit the data or if there was any problem with the existing communication the data can be transmitted from the external storage to the server.

3.1.3 GeoWSN Base Station Architecture

The GeoWSN base station can be described on a 2-tier architecture shown in Figure 3.3. The back-end of this architecture is the database tier and the service tier interacts with this one.
The database tier consists of two databases that are strong the whole information for further operations.

GeoWSN database. This database is the backend of the whole monitoring, alerting, and visualization system. The data gathered from the wireless sensor network or from other external sources, e.g. WWW, are stored in the GeoWSN database. The database also contains the determined positions of the sensor nodes. A detailed description of that database design is mentioned in Section 3.3.

GeoWSN user database. The user database contains the information, which is necessary for the XMPP service to achieve a near-real time communication with the end-users.

---

[5] part of the existing sensor node software

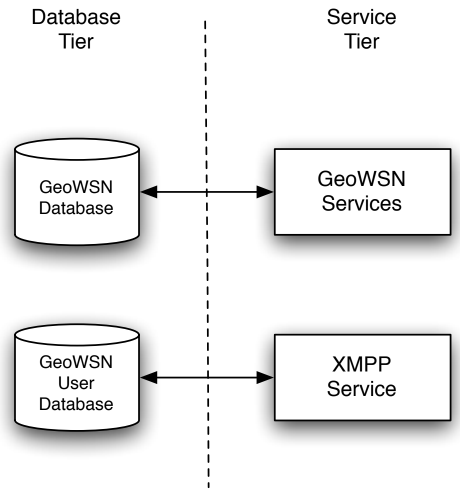


Figure 3.3: 2-tier server architecture.

There are several services, which are building up the service tier. These services process, observe and manipulate the gathered data.

GeoWSN Services. The services are interacting between the database and the environment. The term GeoWSN Services represents a software package of several Linux daemons/services. This daemons and services processe incoming data and transfer them to the database. The design of the different daemons/services are discussed in Section 3.2 in more detail.

XMPP Service. The XMPP service defines the interface between the mobile client and the databases. The authentication of the mobile monitoring and alerting systems is included in this service.

3.2 GeoWSN Services

GeoWSN Services is a software package that includes different kinds of daemons/services. This daemons/services are the fundamental link between the wireless sensor network, the database, and the front-end applications and offer special functionalities.

3.2.1 Requirements

This section deals with the essential software requirements that should be fulfilled by the GeoWSN service package.

Stability. The most important issue of software is the stability. To achieve a high stability different mechanisms must be included into the software, like a restart in case of a non-predictable error. A further term, which deals with the stability of software is bug-tracking or error management.

Flexibility. Flexibility describes the requirement that the core functionality should be transparent. That means it should be easy to install another interface to the environment without changing the main software components.

Usability. The daemons/services should provide a user interface, over which the user could check the state of the daemons, can tweak essential settings, or can get statistical information of the sensor network.

3.2.2 Service Package Structure

Figure 3.4 illustrates the components of the service package. The package can be roughly divided into two parts, the daemons and services processing the incoming data and storing them into a database – further called process daemons/services – and those which gather the data from a database and presents them to the user – called visualization daemons/services.

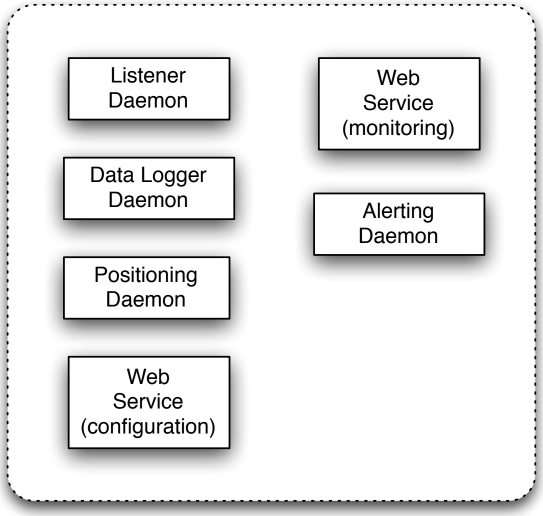


Figure 3.4: GeoWSN service/daemon package.

In the following the components, which are building up the process daemons/services tier, are mentioned:

Listener Daemon. The Listener Daemon is the interface between the wireless sensor network and the database. The daemon parses and processes the incoming data stream from the sensor network and stores the data in the database.

Data Logger Daemon. This daemon compresses the information that are represented over charts to the user. With other words, the gathered information are temporal combined to enhance the performance of the visualization applications.

Positioning Daemon. The Positioning Daemon determines the precise position of the wireless sensor network nodes. In this daemon the two types of positioning algorithms are integrated, the RTK and the PPP. [6]

Web Service (configuration). The web service integrates configuration options, which can be used to make threshold or user settings.

The second tier, the visualization daemons/services of the GeoWSN service package contains two components:

Web Service (monitoring). This service provides the information, which are illustrated through the web application by using a polling server push technology. The client queries the web service periodically, and the data to the client will be changed automatically depending on change of the information that the server gathers from database.

Alerting Daemon. The Alerting Daemon checks periodically the position and the velocity of the wireless sensor nodes and triggers an alarm if a predefined threshold is exceeded.

3.3 Database

The gathered measurement data of the wireless sensor network are stored for further processing in a database. The database is the back-end of the GeoWSN project and it is the general basis for further applications, like visualization, data modeling, and positioning.

3.3.1 Requirements

The database structure must fulfill several defined requirements to ensure the correct functionality of the GeoWSN system.

Handle large data volumes. Due to the high data rates of the wireless sensor network, large data volumes must be stored in the database. Another challenge the large data volume brings with it is the performance issue.

---

[6] implemented by Institute of Navigation, Graz University of Technology and TeleConsult Austria GmbH

Long lifetime. To achieve a reliable alerting system over a long period, the lifetime aspect is an important one. Therefore, intelligent algorithms to compress data and to delete decaying data must be developed.

Consistent and reliable data storage. The positioning techniques to determine the exact position of the sensor node need consistent and periodical data. Furthermore, only relevant data should be stored in the database.

3.3.2 Database Structure

For the calculation of the precise sensor node position, the gathered raw data information is transmitted to the base station where the information is stored in the database. Depending on the positioning technique different information must be stored in the database. Figure 3.5 illustrates the information flow for the two different techniques used in this project: the Real Time Kinematic (RTK) technique, and the Precise Point Positioning (PPP) technique [17].

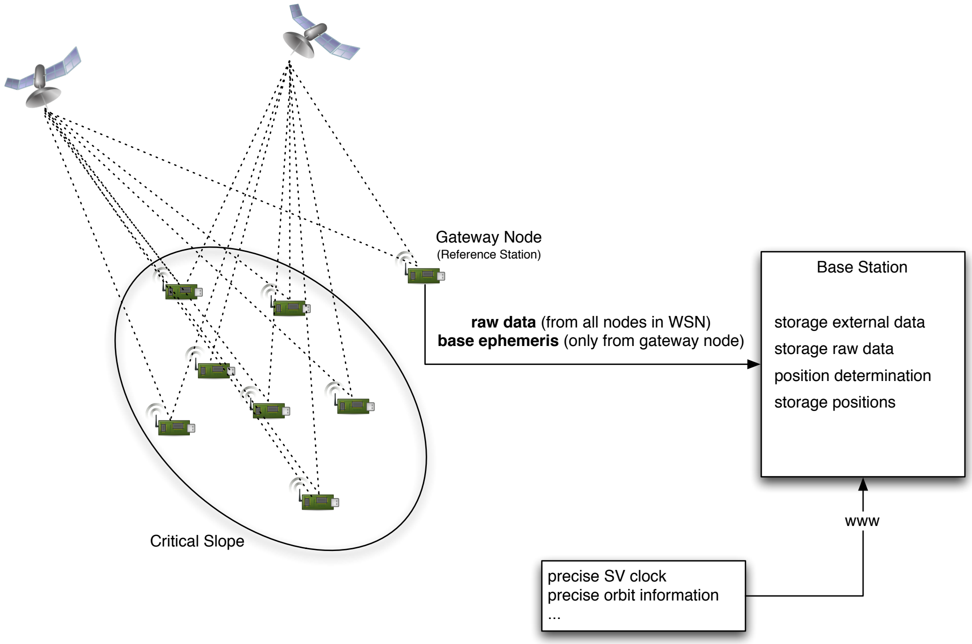


Figure 3.5: The flow of the database information.

The structure of the database can be split in following components. These components are storing the gathered and processed data.

Satellite and sensor raw data. The sensor data and raw data of the satellites, which are gathered from each wireless sensor network node, are stored in this database part.

Base Station ephemeris. This part stores the ephemeris information that are received by the gateway node of the wireless sensor network.

Broadcast ephemeris. This part holds the ephemeris and the satellite clock information that are received by the WWW.

Ionospheric model. The parameters of the different ionospheric models that influence the determination of the precise node positions are stored in this part.

Precise ephemeris, precise satellite clock. For the PPP positioning technique further precise ephemeris and satellite clock information, which are gathered from the WWW, are stored in this part of the database.

Tropospheric model. The positioning algorithms must regard tropospheric parameters, like the earth's rotation parameter, that are stored in this section of the database.

Sensor node positions. This part stores the positions and velocities that are determined by the positioning algorithms.

Temporal combined sensor and position information. This section is needed for the monitoring and visualization system.

Alarm and threshold information. The alarm and threshold information is stored in this database part.

3.4 Web Application

The architecture of a web application can be separated into two different parts, the server side part and the client side part. Thereby the server side part provides the actual information over a web service, whereas the client side part displays the information by the user's browser. The design of a web application can be influenced by a number of factors. In this case the web application should support a mobile client (smartphone) as client side part, so different platforms and frameworks, e.g. flash should not be used in this part.

3.4.1 Requirements

By the design of a web application several requirements must receive attention. During the design and development phase also the human factors should be appropriately taken into consideration.

Usability. The usability of the web application needs to be easy to handle and self-explaining for the end-user.

Security and Reliability. The web application illustrates sensitive data and also several modifications of the alerting system can be made via this, so this sections has to be securely treated. The alerting functionality must be reliable regarding correctness.

Unambiguousness. Within the application there are a number of different possibilities to visualize the information, so there should be no ambiguousness between the particular options.

Near-real time. As mentioned before an alerting option is integrated in the web app, thereby the near-real time requirement is a critical one. This term deals also with the correct visualization of the system state information.

3.4.2 User Interface

The web application is developed for two kinds of users, the normal user and the privileged user. The normal user has the possibility to choose three different views. In the map view the positions of the wireless sensor network nodes are displayed. The mock-up in Figure 3.6 shows the detail view of the web application. This view provides a number of options for illustrating the sensor and velocity information of each sensor node. The last view, which can be accessed by the normal user, is the network view. In this view traffic and energy state information are given. The privilege user has additionally the option to access the configuration view. The configuration view enables modification functionalities for the user management and the threshold settings. Both users have to authenticate themselves to the system before joining.

3.5 Metis Mobile Cloud (MMC)

A XMPP communication network consists of a XMPP server and a number of clients, which connects itself to the server. Therefore, the network can contain any number of XMPP servers (clustering) and XMPP clients.

Section 3.5.1 shows the architecture of the Metis Mobile Cloud Framework. In Section 3.5.2 the differences between the MMC Framework and the Android Cloud to Device Messaging (C2DM) are given.

3.5.1 MMC Architecture

The architecture of the Metis Mobile Cloud Framework is illustrated in Figure 3.7. The framework consists of two components, the Android service and the Android library (XAFLib). The XAFLib is a small Android library that is integrated in the Android service, which is installed as an application. The Android service, called Xmppd, is a

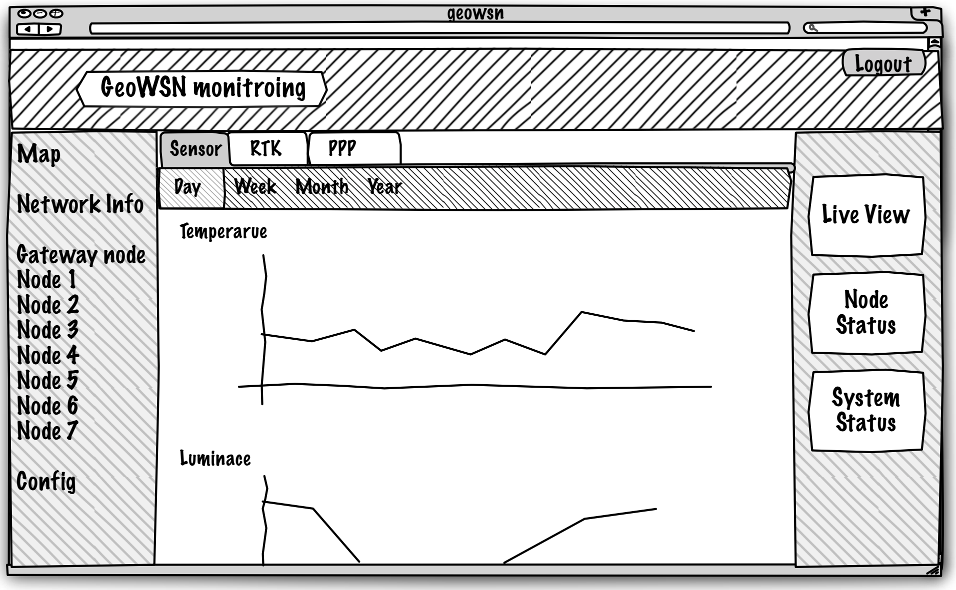


Figure 3.6: Web application mock-up.

daemon that is providing the functionality for the XMPP communication. By the use of an interface any client application can be connected to the Xmppd daemon [9].

To achieve a reliable data communication in case of terminated connection a database is added. If there is no connection to the XMPP server the stanzas are stored in the database and are resent when the connection is stable.

3.5.2 Distinction to the Android Cloud to Device Messaging Framework

With Android 2.2, Google Inc. presents a service, which allows an easy and reliable possibility for the data exchange between servers and clients. The Android Cloud to Device Messaging (C2DM) Framework usually informs the client that data are available on the server. In the following the client fetches the data on the server. In contrast to this approach the XMPP service in the MMC framework can send the data immediately. The issue also underpins the fact that with the C2DM Framework the packet size and the number of data packets, are limited.

Another aspect, which favours the MMC Framework over the C2DM is that by the last one, a registration by Google is necessary. On the contrary, the XMPP service is running on any free XMPP server.

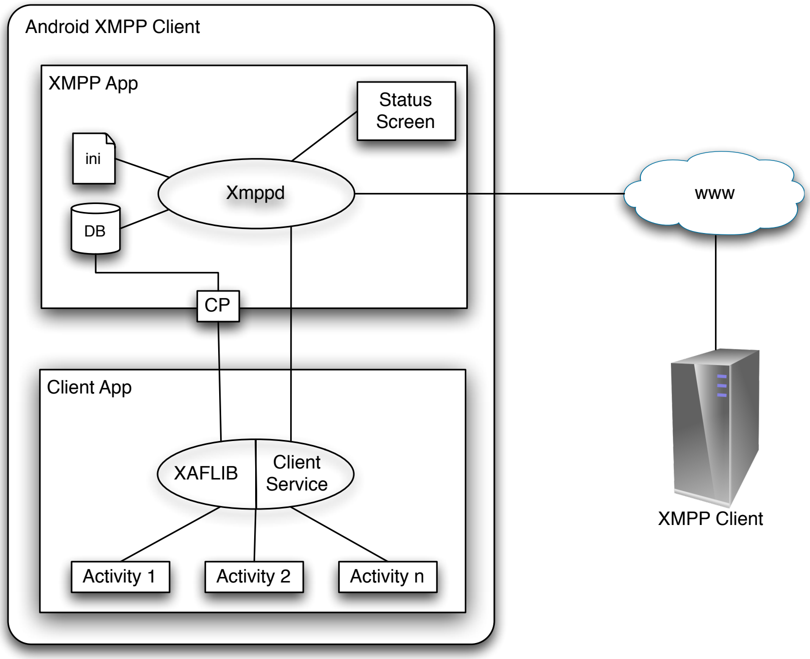


Figure 3.7: The architecture of the Metis Mobile Cloud Framework including the XMPP service (reproduced from [9]).

3.6 Mobile Client

Due to the fact that smartphones have gained much popularity during the past years these devices are suitable for the GeoWSN project as portable end-device for the alerting system.

In order to demonstrate the functionality of the monitoring and alerting system a client using the Android platform is used. The Android platform by Google was chosen, because it is an open-source framework that packages key applications, middleware, and an operation system [60]. Another fact for selecting the Android platform was the increasing market share shown in Figure 3.8.

3.6.1 Requirements

The mobile client is used as an end-device for the monitoring and alerting system in the GeoWSN project. Thereby several requirements must be fulfilled to ensure a correct functionally.

Usability. Due to the requirements on a mobile device the usability is a big issue. The application should give a quick overview of the important parameters of the system to the end-user.

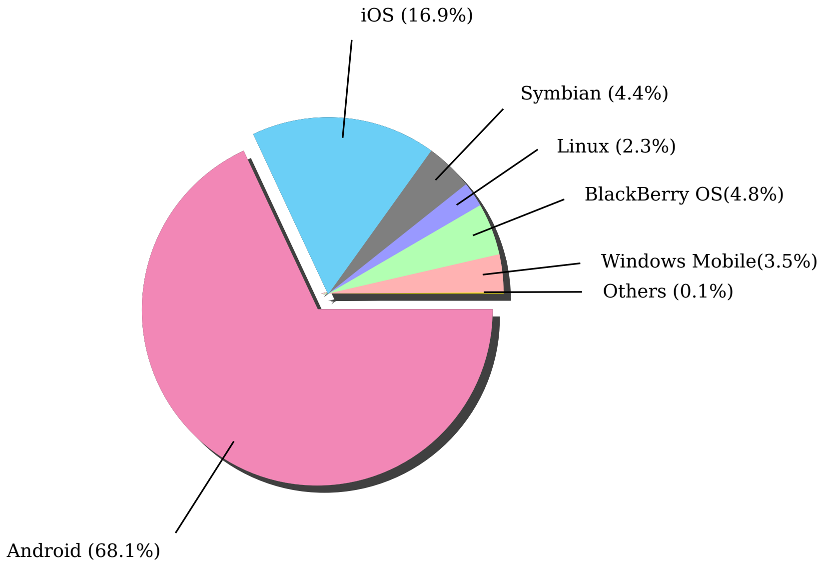


Figure 3.8: Smartphone operating systems and market share (*source: IDC Worldwide Mobile Phone Tracker, August 8, 2012*).

Energy efficient. Due to the fact that smartphones have gained the limited energy supply of the mobile client to application needs to implement power saving modes, like push by alerting or refresh on commit.

Unambiguousness. There should be no ambiguousness between the different data sets.

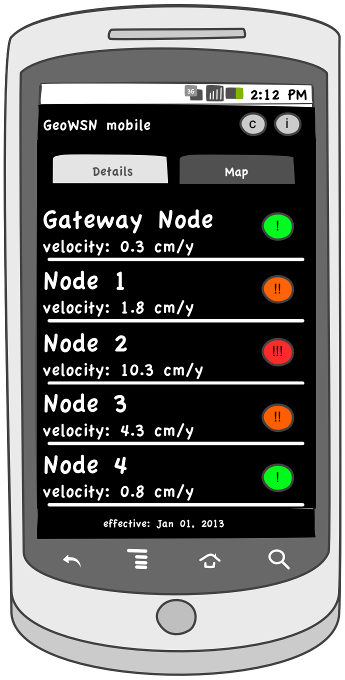
Near-real time. The mobile client is also used as end-device of the alerting system the near-real time requirement is a critical one.

3.6.2 User Interface

The Android application consists of two main views. Figure 3.9 shows the design of the detail view. With the help of the detail view a quick overview of the most important information is given. Thereby an exceedance of the threshold levels should be illustrated visually. Another view option is shown in Figure 3.10, the map view. In this view the current positions of all sensor nodes in the wireless sensor network should be displayed on a map.

3.6.3 Metis Mobile Cloud (MMC) Interaction

As Application Programming Interface (API) for the mobile client the existing Metis Mobile Cloud communication is used. The XML stream, which contains the information



Gateway Node
velocity: 0.3 cm/y
Node 1
velocity: 1.8 cm/y
Node 2
velocity: 10.3 cm/y
Node 3
velocity: 4.3 cm/y
Node 4
velocity: 0.8 cm/y
effective: Jan 01, 2013

Figure 3.9: Detail-view mock-up.

Figure 3.10: Map-view mock-up.

that should be displayed on the mobile client, are encapsulated in an XMPP stanza.

For the information transfer between the mobile client and the server the MMC features are adapted. To ensure a near-real time and consistent XML data exchange to the XMPP application running on the mobile client and to the XMPP service on the server special functionalities must be added.

3.7 Summary

This section has shown the concept of the early warning and monitoring system. First, the architecture of the whole GeoWSN system and the integration of the early warning and monitoring system has been briefly described. Thereafter, the architecture of the gateway node of the wireless sensor network has been described. The architectural extensions to serve as bridge between the server and the wireless sensor network have been defined. A serial interface has been chosen for the data transfer between the communication partners. To support this concept on the gateway node a COM driver, which fulfills the mentioned requirements, was designed.

The architecture of the GeoWSN server has been shown as a two tier architecture, where one tier describes the database structure and the other one the service structure. First, the concept and the requirements of the GeoWSN services has been mentioned. The structure of the service package has been separated into two parts, the process services/-

daemons and the visualization services/daemons.

The requirements, which have to be fulfilled by the database structure, have been described. Then, an overview of the designed structure of the database and the communication flow for storing the required information has been given.

Finally, the concept front-ends of the early warning and the API of the mobile client have been described. First, the requirements and the user interface of the web application were mentioned. Then, a general description of the existing Metis Mobile Cloud and the designed extension that the MMC functions as API for the mobile client were presented. Last, the user interface and the requirements of the mobile client have been shown. For the integration of the mobile client the open-source Android platform has been chosen.

The defined requirements and the mentioned aspects for the components of the early warning and monitoring system should be appropriately taken into consideration in the development phase.

Chapter 4

Implementation

This chapter deals with the implementation of a database, a visualization and an early warning system for slope movements. The implementation must fulfill several requirements, like near-real time, high data quality, and minimizing the error rate.

This part shows the implementation of the whole chain of an early warning and monitoring system. First, in Section 4.1 the Development Environment is briefly described. Section 4.3 deals with the software extension, which must be made for the gateway node of the wireless sensor network. In Section 4.4 the protocols for the data exchange between the gateway node and the GeoWSN server are mentioned. The GeoWSN service package includes a number of services for processing, editing and storing the gathered data. This package is described in Section 4.5. The database structure, which holds the parsed information, is mentioned in Section 4.7. Section 4.8 focuses on the implementation of the web application. The web application is one of the two front-ends from the early warning and monitoring system, and provides several visualization and configuration features. Finally, the second front-end from this system, the mobile client including a native Android application is shown in Section 4.9. In the mobile client an instant messaging and presence technology for near-real time communication is integrated. In Section 4.10 experimental results are mentioned. Section 4.10 deals with the evaluation of the database and visualization system.

4.1 Development Environment

In this section the used technologies and environments for the development of an early warning and monitoring system are mentioned. First, the focus lies on the basis for the software development, the platform Eclipse. This is followed by the plug-ins for the development of the web application, the ZK Framework, and for the native Android application, the Android Software Development Kit (SDK). The mentioned plug-ins interact with the core of the Eclipse platform. The next part deals with the development environment for enveloping applications for Texas Instruments (TI) embedded processors, the Code Composer Studio (CCStudio). Finally, the database management system is explained.

4.1.1 Eclipse

Eclipse [61] is a platform for multi-language software development, comprising a workspace and an extensible plug-in system. The software development environment is structured around the plug-in concept. The core of the open Eclipse architecture handles the dynamic discovery, loading, and running of plug-ins. A plug-in is a bundle of code with a manifest. The plug-in manifest contains information about the plug-in, of dependencies and its utilization. There are different kinds of functionalities that a plug-in provides, e.g. code libraries (Java classes with public API), or platform extensions. The provided common User Interface (UI) model includes a standard user navigation system [62].

4.1.2 ZK Framework

ZK [63] is an open-source, event-driven Asynchronous JavaScript and XML (Ajax) web application framework from the Potix Coorparation. The component-based ZK Framework enables the creation of rich user interfaces. An overview of the ZK Framework architecture is shown in Figure 4.1

The Ajax engine of ZK consists of a server and a client component that communicate with each other. The user interface can be described by two different sets of components: the XML User Interface Language (XUL) component set and Extensible HyperText Markup Language (XHTML). A markup language for rich user interface definition of the ZK framework supports ZK User Interface Markup Language (ZUML). ZUML, like HyperText Markup Language (HTML), describes the components that are added to the page.

A big advantage of the ZK framework over other ones is that the Ajax codes are completely transparent to the developer. This means that ZK uses Ajax as a behind-the-scene-technology [10]. Therefor, the framework provides a ZK Asynchronous Update (AU) Engine and a ZK Client Engine. The ZK AU Engine is running on the server and the ZK Client Engine is running on the browser, working together to update the Document Object Model (DOM) tree at the browser. The events, which occur in the browser, are delivered to the server application. This functionality is known as event-driven technology. Thereby the event handling is developed in Java code, which is executed on the server [64]. To summarize, the ZK framework simplifies the rich Ajax web application developments by the following approaches[10]:

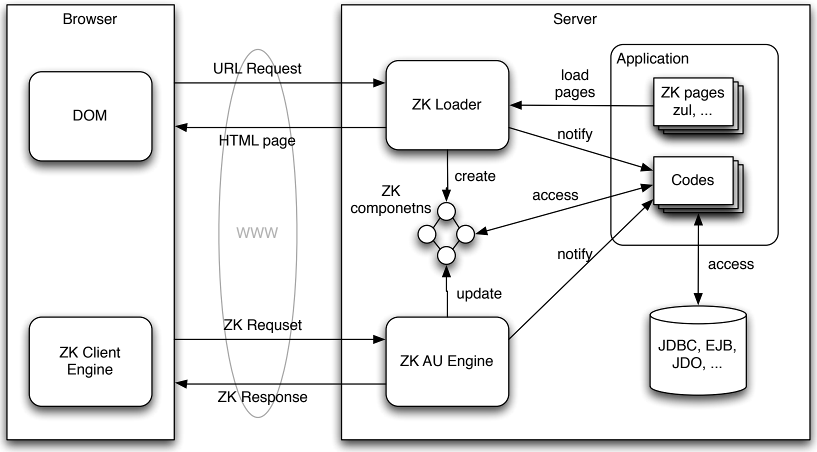


Figure 4.1: ZK framework architecture (reproduced from [10]).

- The design of rich user interfaces is done by the ZUML markup language which makes the design as simple as authoring HTML pages
- The core of the framework is an Ajax-based event-driven engine which comes along with intuitive desktop programming
- Supported XUL and XHTML enrich the web application

4.1.3 Android SDK

Android is an open-source operating system and software stack for mobile devices. The architecture of the software stack is shown in Figure 4.2 and based on a Linux 2.6 Kernel. The Kernel handles the storage and process management, builds up the hardware abstraction layer for the further software and provides several device drivers for the system. The library layer consists of several native C/C++ libraries, like Secure Sockets Layer (SSL) or SQLite and the android runtime. The android runtime based on a Dalvik Virtual Machine and their Java core libraries. The next layer is the application framework. The application framework includes all kinds of APIs, which are needed for the development of an Android application. By the use of the functionality of the application framework it is able to exchange data between applications, display information on the screen and so on. The lifecycle of the application can be also managed here. The application layer contains the applications developed with Java [11].

The Android SDK [65] enables the development of applications for the Android platform. Via the Android SDK the developer can use the functionality from the application framework layer. The package of the Android SDK includes in addition to the development tools also an emulator for debugging and prototyping [66].

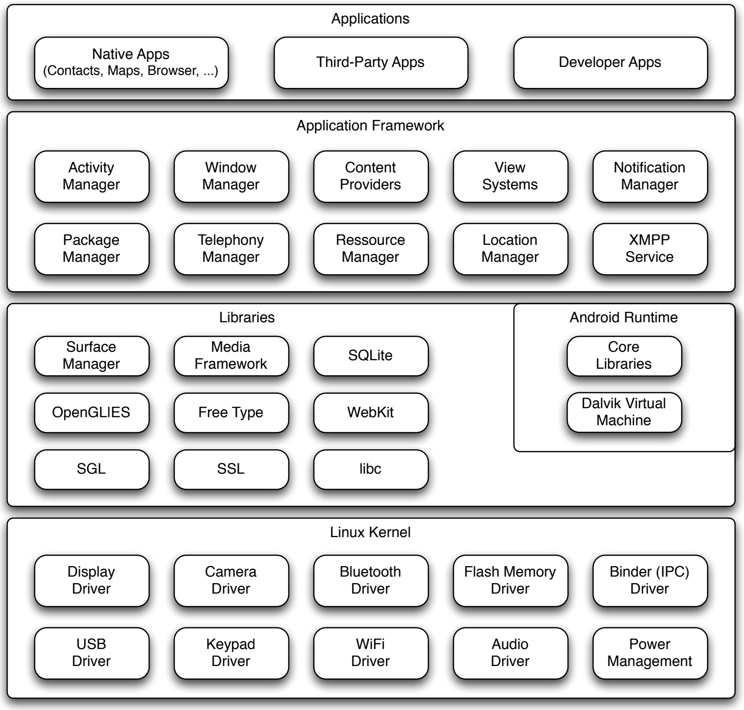


Figure 4.2: The structure of Android (reproduced from [11]).

4.1.4 Code Composer Sudio

The Code Composer Studio [67] is an Eclipse based Integrated Development Environment (IDE) developed by Texas Instruments. The CCStudio contains several tools for developing and debugging applications for TI embedded processors such as the MSP430. In the Code Composer Studio real time operating systems, like DSP/BIOS or SYS/BIOS, are included. The Texas Instruments IDE supports low-level Joint Test Action Group (JTAG) based development as well as application debugging in the operating system level.

The SYS/BIOS operating system, which is included in the CCStudio, is used for real-time scheduling and synchronization in embedded applications. Therefor SYS/BIOS provides memory management, preemptive scheduling, and hardware abstraction [68].

For the use of the SYS/BIOS components the eXpress DSP Components tools (XDCtools) are necessary. The XDCtools also integrate configuration tools of SYS/BIOS for the application. Furthermore, the XDCtools APIs include memory management services, logging services, and system control services.

4.1.5 MySQL Database Management System

The open-source Structured Query Language (SQL) database management system MySQL [69] is developed by Oracle Corporation. Figure 4.3 shows the modular architecture of the MySQL database management system.

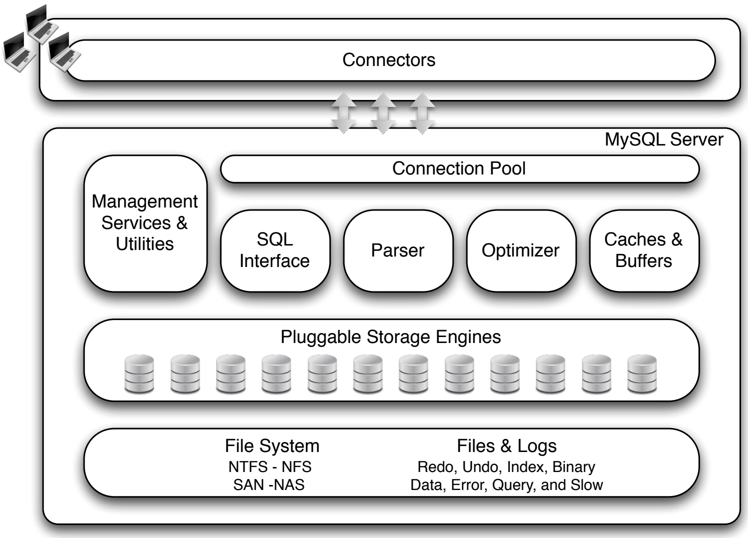


Figure 4.3: MySQL architecture (reproduced from [12]).

MySQL provides standard-based drivers, called connectors, like Java Database Connectivity (JDBC) that enables the client to interact with the MySQL server. Each client, who connects itself with the server, has its own thread. The connection pool manages those connections on the MySQL server. Also part of this component is the authentication of the client by username and password. The SQL interface for example includes a Data Manipulation Language (DML) pre-compiler for receiving a request from a client in the application layer, and a Data Definition Language (DDL) compiler that processes the request from an administrator in the application layer. The parser extracts the internal structure of the MySQL query. Further, the optimizer increases the performance of the database management system by optimizing the query. The caches and buffers component indicates global and engine specific caches and buffers. One of the main differences between the MySQL database management system and other ones are that the storage engines in the MySQL systems are pluggable. Pluggable means that the developer of the application can choose a storage engine that fits best in his application. Therefor, MySQL provides several storage engines, like InnoDB, BDB, or Black hole. The storage engines are used for the data storage and index management in the database management system. Thereby, the MySQL server communicates via defined API with the storage engines [12].

4.2 Early Warning and Monitoring System

The structure of the GeoWSN Early Warning and Monitoring System is illustrated in Figure 4.4. The structure shows the different components and the interaction between those.

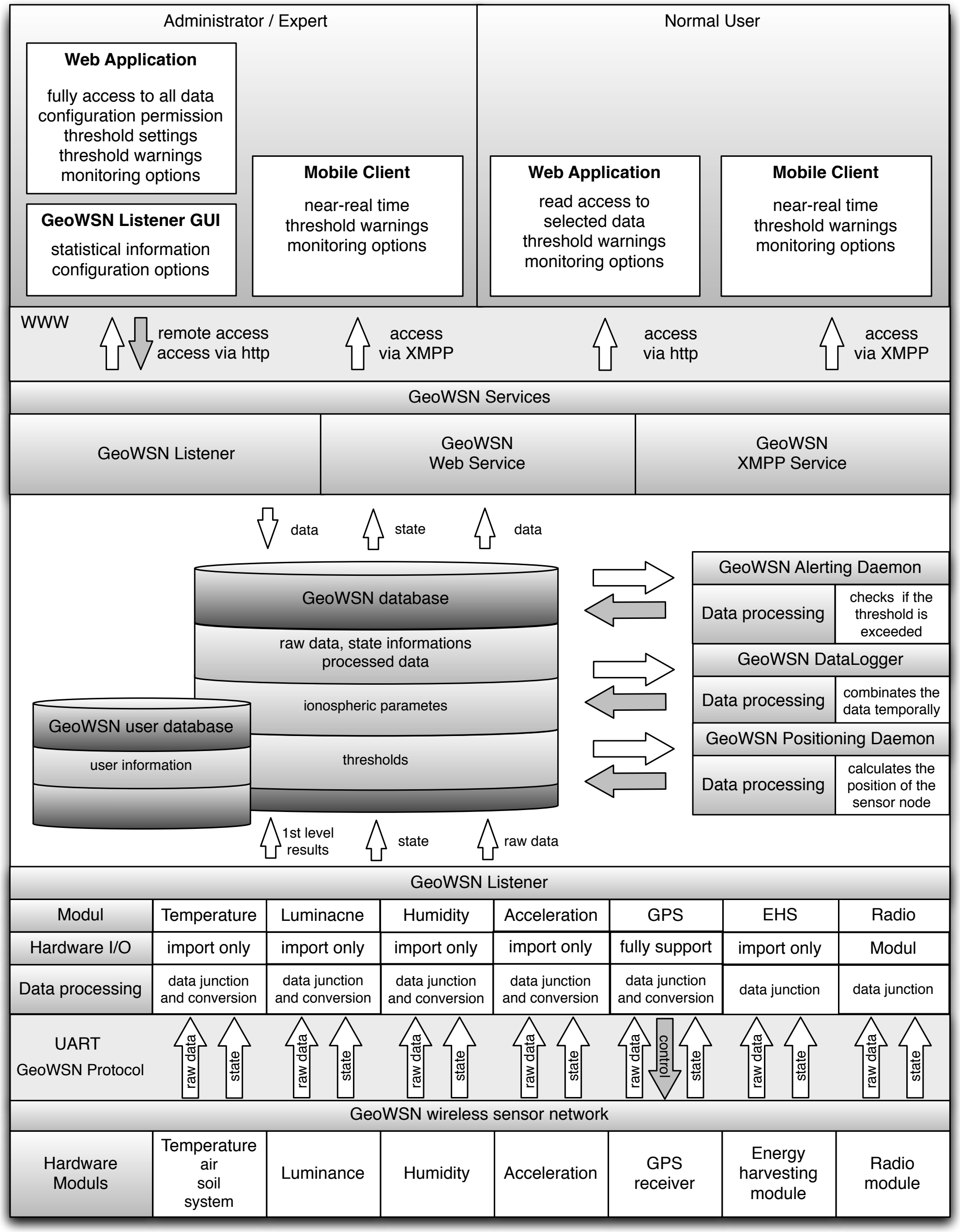


Figure 4.4: GeoWSN system structure.

The early warning and monitoring system uses a bottom-up architecture. The basis of this architecture is the wireless sensor network, from which the sensor data are gathered respectively to which configuration streams are transferred. The communication is handled over an UART interface by using an internal protocol. The communication partner of the sensor network is the GeoWSN Listener. This application receives the incoming sensor data stream and performs a first-level processing. With the help of the Listener it is also possible to transfer configuration streams via the UART connection to the gateway of the wireless sensor network. The received and processed data are stored in a database. The database system builds the back-end of the early warning and monitoring system. Parallel to the database several applications are running. These applications on the one hand perform a further data processing task, and on the other hand a monitoring task of different parameters that are stored in the database takes place. The next level of the early warning and monitoring architecture represents the middleware. The middleware defines the interface to the front-end application, which can be used by two kinds of users. The GeoWSN system offers two bi-directional communication options for the administrator/expert. The first option is the access via Hypertext Transfer Protocol (HTTP), the web application, and the second one is the graphical user interface of the GeoWSN Listener that uses a remote access. The mobile client uses an instant messaging and presence technology for the interaction. This application is displaying current information of the whole system and can be accessed of both kinds of users. The normal user can also use the visualization options of the web application, as can the administrator.

4.3 Gateway Node

The gateway node is part of the wireless sensor network and represents the interface between the GeoWSN server and the sensor network. For the data transfer an extension to the node software, the COM driver is implemented. Therefor, a modular structure was chosen that supports the current node architecture and also further extensions to those. Listing 4.1 shows the methods that are implemented in the COM driver. These methods provide a number of sending and receiving functionalities.

In case of a broken communication between the gateway node and the server, a safety mechanism to guarantee reliable data is integrated. The mechanism does not accept incoming data packages from the other network participants. Each node stores the data package, which includes the sensor information, on its own flash memory. The mentioned mechanism avoids that the integrated flash memory of the gateway node has an over-run, by storing the data packages of the whole sensor network.

Listing 4.1: COM driver methods for sending and receiving.

```
// send methods
void sendByte(uint8 byte);
void sendNextByte();
void sendHex(uint8 byte);
void sendMSG(Message * msg);
void sendString(char* byte);
// receive methods
void receiveByte();
uint8 getReceivedByte();
```

4.3.1 Data Gathering Communication Flow

Figure 4.5 illustrates the communication flow of the sensor and satellite data to the database. The gathered data of the gateway node are transmitted over a RS232 communication interface to the server. The GeoWSN Listener is a service that is running on the server. This service receives and parses the data. The parsed data are transmitted by the use of a JDBC driver to the GeoWSN database. The data are stored in the assigned database table. In case of a connection problem between the GeoWSN Listener and the database the transfer data are stored in an external file.

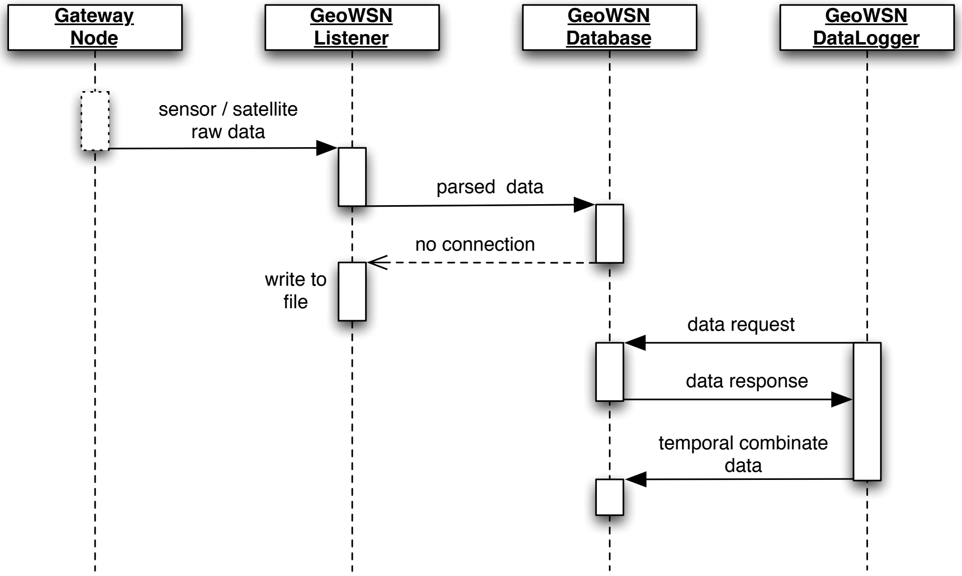


Figure 4.5: Communication flow between the data gathering components.

Also another service is involved in this communication flow, the GeoWSN DataLogger. This service requests data from the database, combines the responded data temporally, and transmits the combined data to the database. The GeoWSN DataLogger reduces the amount of data without the loss of any information to enhance the performance of the visualization system.

4.4 Communication Protocols

This section deals with the communication protocols which are used by the GeoWSN system. The communication protocols describes the skeleton of the data transfer between the gateway node of the wireless sensor network and the server where the data are processed. Therefore different protocols are supported. The communication protocol that is used for the exchange of the sensor and satellite raw data is described in Section 4.4.2. In Section 4.4.3 the protocol which is used for the ephemeris information is mentioned.

4.4.1 Base Frame

The base frame for the serial communication between the sensor gateway node and the GeoWSN server is shown in Table 4.1. For the data exchange each byte is transmitted as a hexadecimal American Standard Code for Information Interchange (ASCII) code, which reserves two bytes. The negative effect of needing more bandwidth comes along with a better readability and consequently an easier debugging possibility for developers.

Table 4.1: Base frame format [16].

ID	SN	Src. addr.	MSG Type	Data size	Data

Short form	Bytes	Description
ID	1	Start identifier
SN	2	Sequence number
Src. addr.	4	Source address in the multihop WSN
MSG Type	2	Defines the message type of the frame
Payload size	2	Data size of this packet
Payload	2*n	n data Bytes

By the start identifier field it is possible to distinguish between a satellite/sensor data message and an ephemeris data message. Therefor, unique start sequences are used: a 'P' indicates a satellite/sensor data message where a 'Q' indicates an ephemeris data message. Further, because of the unique sequence the process service running on the server can be synchronized by this.

4.4.2 Data Frame

One possibility of the frame format that is identified by the payload field in Table 4.1 is described in Table 4.2. The data frame contains the information which are gathered form the sensor node, the sensors, and the GPS receiver.

Table 4.2: Data frame format [16].

	Byte Offset	Number Format [7]	Name	Unit	Description
Net info	0	U1 [6]	hopPath	-	Address of every hop
	6	U1 [7]	hopRSSI	-	RSSI of every hop transmission
	13	U2	netload	b/s	Node network load in Byte per second
	15	X1	netNeighbor	-	Bit mask of which nodes are in range
Sensor data	16	U2	tempSys	-	System temperature
	18	U2	tempAir	-	Air temperature
	20	U2	tempGnd	-	Soil temperature
	22	U3	light	-	Meassured illumination value
	25	I1	accX	mG	Acceleration x-axis in milli G
	26	I1	accY	mG	Acceleration y-axis in milli G
	27	I1	accZ	mG	Acceleration z-axis in milli G
	28	U2	humidity	-	Meassured humidity
States	30	X1	senseState	-	Sensor status mask described in Table 4.4
	31	X1	GPSState	-	GPS sensor status
	32	U2	eInput	-	Energy input level
	34	X1	eState	-	Energy source state
	35	U2	eLevel	-	Energy store level
	37	X1	eLevelState	-	Energy store state
GPS	38	I4	iTOW	ms	GPS time in millisecond of GPS week
	42	I2	week	weeks	GPS week at measurement
	44	U1	numSV	-	Number of satellites (N)
	45	U1	reserved	-	Reserved
	Repeated block from Table 4.3				

The data frame can be separated into four different tiers, the net information, the sensor data, the state information, and the GPS data. In the net information tier information about the network topology and statistical network performance parameter are covered. The hopRSSI (Received Signal Strength Indication) and hopPath fields are representing the route and the transmission quality. Therefore, each node on the route sequentially adds its address to the hopPath and the received signal strength of the incoming package to the hopRSSI field.

The sensor data tier contains the gathered and processed data from the internal and external sensors.

---

[7] The number formats are described in Table A.1.

The state information tier consists of several state and energy source information fields. The state fields show the operating state of the internal and external sensors or the GPS receiver. By the energy source information fields the selected energy source and the energy state of the accumulator are mentioned.

The GPS data tier contains information about the number of satellites and the precise GPS measurement time. If these fields indicate that GPS data are available the GPS raw data are added to the frame for each satellite. The frame format of the repeated block is shown in Table 4.3. This frame format is an adapted format from the RXM-RAW frame which is supported by the GPS receiver.

Table 4.3: GPS data frame format [16].

Byte Offset	Number Format	Name	Unit	Description
Start of repeated block (numSV times)				
0+24*N	R8	cpMes	cycles	Carrier phase measurement
8+24*N	R8	prMes	m	Pseudorange measurement
16+24*N	R4	doMes	Hz	Doppler measurement
24+24*N	U1	sv	-	Space Vehicle Number
25+24*N	I1	mesQI	-	Nav measurement quality indicator
26+24*N	I1	cno	dbHz	Singnal strength C/No.
27+24*N	U1	lli	-	Loss of lock indicator
End of repeated block				

As mentioned above the data frame format includes a state information tier. The state fields describes in a binary way the operating state of the respective component. In Table 4.4 the structure of the used binary mask for the sensor state is shown. Each bit in the mask indicates another sensor. If the bit is set the sensor is in the correct operating state and vice versa.

Table 4.4: Sensor state mask [16].

Sense state	Description
b_0	Humidity sensor
b_1	Luminance senor
b_2	Air temperatur sensor
b_3	Soil temperatur sensor
b_4	System temperatur sensor
b_5	Acceleration sensor

4.4.3 Ephemeris Frame

The second option, which can be included in the payload field in Table 4.1 is the ephemeris frame, described in Table 4.5. This frame format is adapted from the RXM-EPH frame. The parameters, included in the ephemeris frame, are support the determination of an ionospheric model. If the frame contains ephemeris, they are transmitted in a block of three subframes. The structure of the subframe are mentioned in Table A.2, Table A.3, and Table A.4.

Table 4.5: Ephemeris frame format [16].

Byte Offset	Number Format	Name	Unit	Description
0	U4	svid	-	SV id
4	U4	how	-	Hand-Over Word of first subframe 0 indicates no ephemeris
Start of optional block				
8	U4[8]	sf1d	-	Subframe 1 Words 3..10 (SF1D0..SF1D7)
40	U4[8]	sf2d	-	Subframe 2 Words 3..10 (SF2D0..SF2D7)
72	U4[8]	sf3d	-	Subframe 3 Words 3..10 (SF3D0..SF3D7)
End of optional block				

Each subframe is adapted from the GPS navigation satellite message format and divided into 8 words of 30 bits. High accuracy ephemeris and clock offset data are contained in the subframes.

Subframe 1 (Table A.2) consists of parameters for the determination of the clock offset. In subframe 2 (Table A.3) and subframe 3 (Table A.4) orbital parameters are transmitted.

4.5 GeoWSN Service Package

This section deals with the services, which are the fundamental link between the database and the wireless sensor network on the one hand, and between the database and the front-end applications on the other hand. Thereby, the services can be distinguished into two parts: those which offer information and those which gather information.

4.5.1 GeoWSN Listener

The GeoWSN Listener is the link between the gateway node of the wireless sensor network and the database system. This service has a modular structure, which allows extensions and an easy usage. Each module is encapsulated in a single class. Figure 4.6 gives an overview of the classes and their dependencies. Here every class represents the functionality of the respective GeoWSN Listener component.

The GeoWSN Listener provides two functionalities, receiving the gather information from the gateway node, and sending configuration statements to the gateway node over a communication interface.

By the receiving functionality the collected data are parsed according to the underlying frame structure. After the parsing process a first level data processing task takes place. The processed data are stored in a database and as a back-up option in a file. To inform other services that a new data set is available in the database a communication interface over a TCP socket is supported. In the GeoWSN Listener system a graphical user interface is included. The data transfer between the service tier and the visualization tier of the system is also done via a TCP socket. The graphical user interface of the GeoWSN Listener offers the possibility of reporting statistical information, like error rate, and retransmit rate to the system administrator.

A second option, which is supported by the graphical user interface and thus also from the GeoWSN Listener, is the possibility of sending configuration statements, e.g. the measuring interval to the wireless sensor network using the existing communication interface.

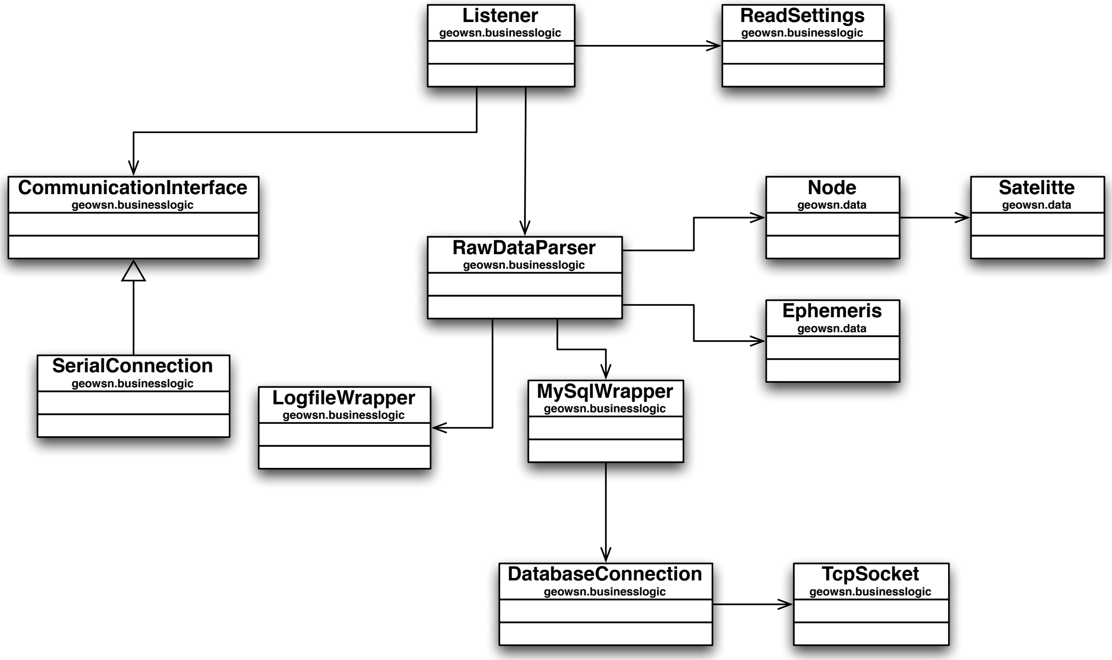


Figure 4.6: GeoWSN Listener class diagram.

4.5.1.1 Class Description

A brief description of the classes of the GeoWSN Listener, shown in Figure 4.6, is listed in the following.

Listener. This class is the main class and instances the required objects.

ReadSettings. This class reads the configuration settings used by the GeoWSN Listener, especially by the DatabaseConnection component from a file.

CommunicationInterface. The CommunicationInterface provides the sending and receiving methods. By the use of this interface any connection module is supported.

SerialConnection. This class gathers the data stream from the serial interface of the server. Another option, which is provided by this class, is the possibility of sending configuration statements.

RawDataParser. The RawDataParser parses the incoming raw data stream and stores the parsed information. Also a first level processing of the information is done in this state.

Node. This class holds the information of a wireless sensor node.

Satellite. The Satellite class stores the GPS information of each satellite.

Ephemeris. This class contains the gathered ephemeris data as attributes.

MySqlWrapper. This class generates the query string of the node, satellite and ephemeris object attributes.

DatabaseConnection. The DatabaseConnection class manages the database connection by the use of a JDBC driver.

TCP Socket. This class provides a server socket (listen for a client communication) and a client socket (establishes a connection to a server on defined host and port).

LogFileWrapper. This wrapper class is used as a back-up option in case of a broken database connection. Hence the data are stored in a readable format in a file.

4.5.1.2 Operation

The GeoWSN Listener can be split into two separated tasks. On the one hand the RX task and on the other hand the process task. The basic operation of those tasks is described in the following.

The assignment of the **Receive (RX) task** is to read the data from the serial interface of the GeoWSN server. Figure 4.7 shows the simplified program flow of the RX task.

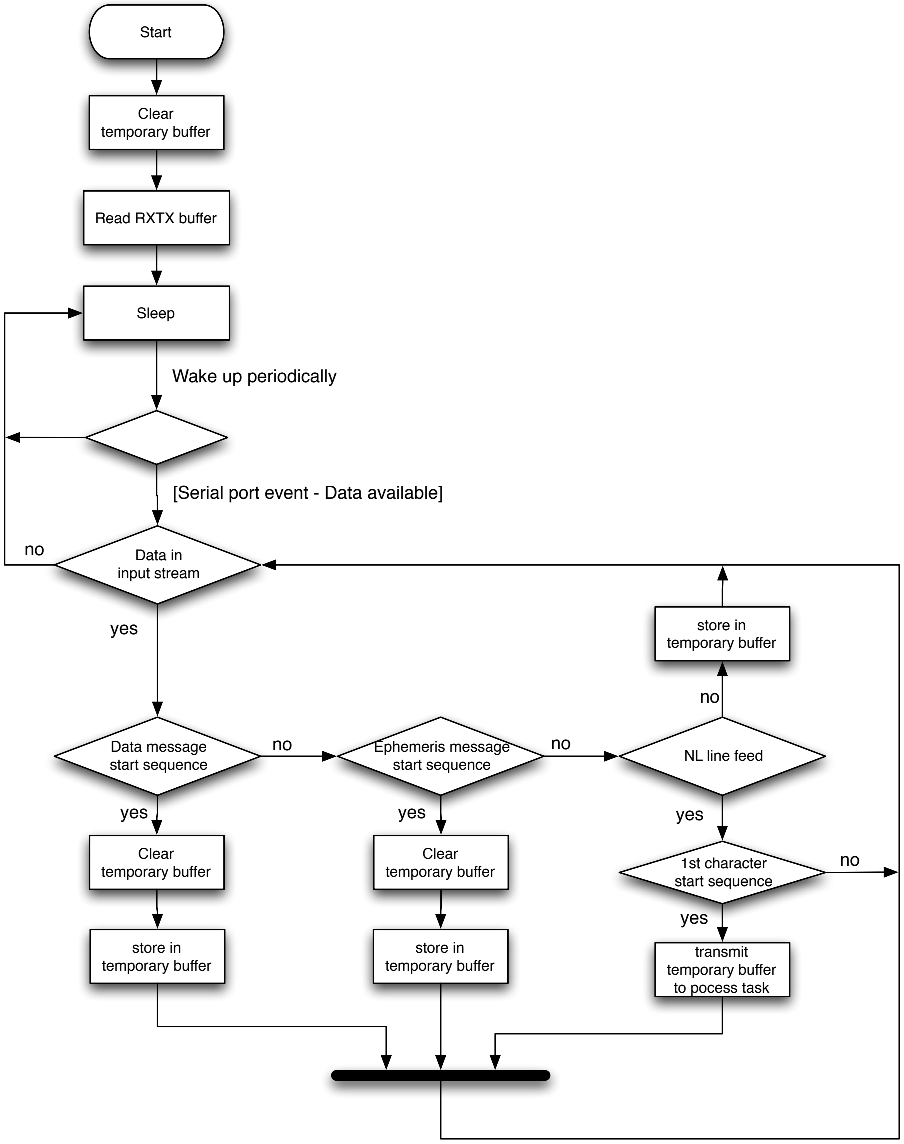


Figure 4.7: Simplified program flow of the GeoWSN Listener RX task.

The task activates the event listener of the Receive/Transmit (RXTX) buffer of the serial connection and switches in the sleep state. Periodically the task wakes up and checks if a serial port event has been occurred. This indicates that data available at the RX buffer. When this event occurs the task reads character by character from the input stream of the RX buffer. This can be done, because the content of the communication frames is represented by an ASCII format. The input character can indicate a data message start sequence, an ephemeris message start sequence, another ASCII character, or an end of line indicator called New Line (NL) Line Feed (LF). By the first two options the temporary buffer is cleared and the starting sequence is added to the buffer. The third option stores the read character in the buffer. The last option is active if the read character indicates a new line, and the first character of the stream was a defined starting sequence. When these requirements are fulfilled the content of the temporary buffer are transmitted to the process task. If there are no further characters in the input stream available the task switches to the sleep state.

The **process task** parses, makes a 1st level processing, and stores the data that are read by the RX task. In Figure 4.8 the program flow of the process task is shown.

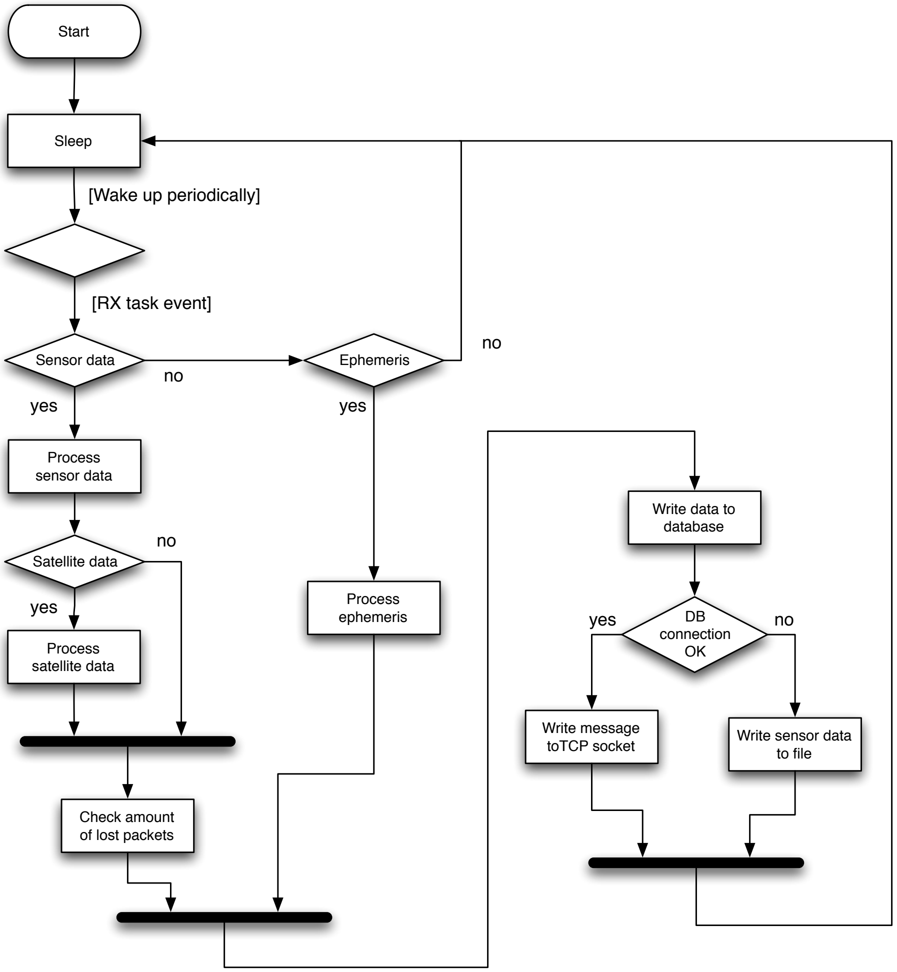


Figure 4.8: Simplified program flow of the GeoWSN Listener process task.

The task starts and switch immediately to the sleep state. In this state the task wakes up periodically (default 1 *sec*). If the RX task event triggers the following decision has to be made: indicates the stream, transmitted from the RX task, sensor data, ephemeris, or none of both. If the incoming stream indicates sensor data, this stream is parsed based on the underlying frame format. If the sensor data stream includes the satellite raw data, they are also parsed in this section. In both parsing states also a 1st level processing takes place. If the incoming stream indicates ephemeris the stream are parsed and also a 1st level processing takes place. After the data are present in the desired format, the amount of lost packets (in the wireless sensor network) for the sensor data of each node is calculated. In the next state the parsed and processed data are stored in the database. If there was a successful connection to the database, storage and statistical information are transmitted via a TCP socket to external services. Otherwise, if a database connection error occurs the processed sensor data are stored in a back-up file.

4.5.2 GeoWSN DataLogger

The GeoWSN DataLogger fundamentally offers the data for the chart visualization of the web application. This service increases the performance of the web application by combining the considerable quantity of data temporally. Therefore, two algorithms are used to combine the data. The first one builds the mean value of the data, like temperature, and luminance or the maximum value of the data, like velocity for the defined interval. The second algorithm is used for combining the acceleration data. The algorithm determines the maximum change to the previous interval and uses this acceleration point, which leads to this.

An overview of the classes and their dependencies is given in Figure 4.9, where each class represents a component of the GeoWSN DataLogger

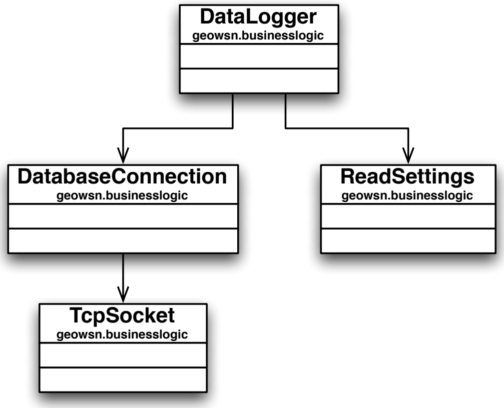


Figure 4.9: GeoWSN DataLogger class diagram.

4.5.2.1 Class Descritpion

In this section the classes of the GeoWSN DataLogger, shown in Figure 4.9 are described briefly.

DataLogger. This class is the main class and instances the required objects. The class also includes the algorithms for the data processing.

ReadSettings. The configuration settings for the database connection are read from this class from an external file.

DatabaseConnection. This class configures the database connection and provides the methods for interacting with the database by using a JDBC driver.

TCP Socket. This class provides a socket for the data transfer to the graphical user interface of the GeoWSN DataLogger.

4.5.2.2 Operation

In Figure 4.10 the simplified program flow of the process task of the GeoWSN DataLogger is illustrated. The process task starts and runs the operation for combining the data for the day interval. First, the desired data are read from the database. Secondly, the gathered data are processed by the use of two different algorithms.

Algorithm I calculates the mean value of the temperature, luminance, and humidity data and the maximum value of the velocity data over the selected interval.

Algorithm II determines the acceleration, which occurs in the selected interval. Therefor the maximum and the minimum acceleration of the current interval and the average acceleration of the previous interval are necessary. In the following the mathematical description of the algorithm is given.

$$\begin{aligned} acc_1 &= previousAccAvg - currentAccMin \\ acc_2 &= previousAccAvg - currentAccMax \\ |acc_1| \geq |acc_2| \quad \Rightarrow \quad acceleration &= currentAccMin \\ |acc_1| < |acc_2| \quad \Rightarrow \quad acceleration &= currentAccMax \end{aligned}$$

After the processing states the data is stored in the database. In the next step the decision, if the week counter, the month counter, or the year counter reaches the defined state has to be made. If one or more counter reaches the state, the program flow mentioned above is processed sequentially for the respective interval. In the case of no further processing has to be made, the task switches into the sleep state. From this state the task wakes up periodically (default 30 sec – interval for the day visualization).

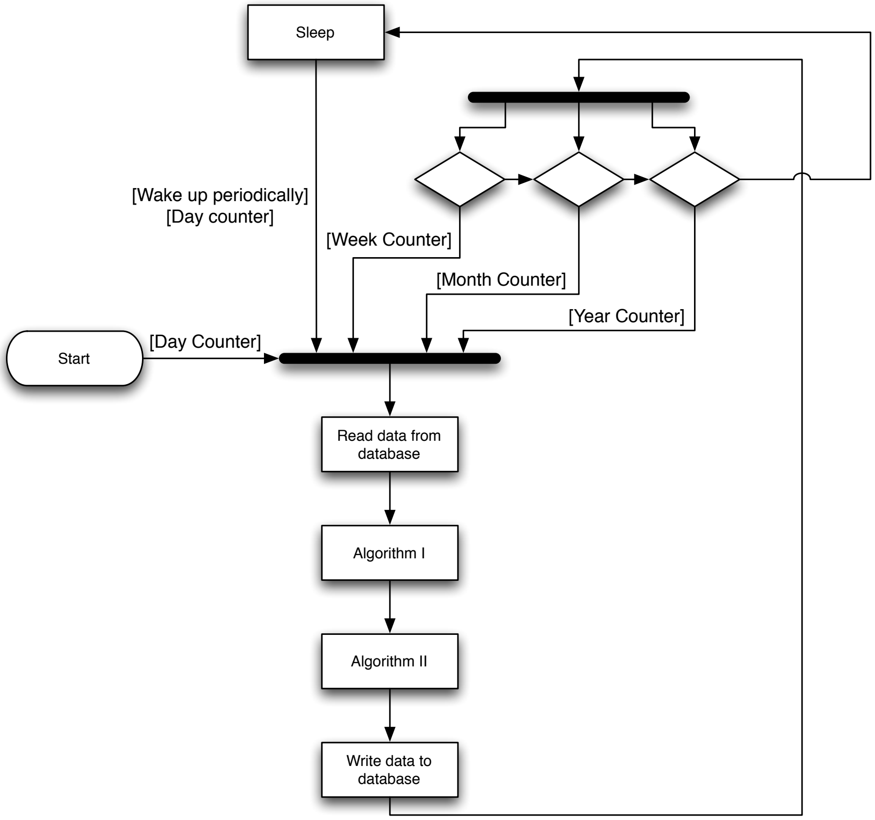


Figure 4.10: Simplified program flow of the GeoWSN DataLogger process task.

4.5.3 GeoWSN AlertingService

The GeoWSN AlertingService checks periodically if the predefined velocity, acceleration, or energy threshold is exceeded and performs the required database transaction. Figure 4.11 shows the class structure of this service. The figure also illustrates the dependencies between each component.

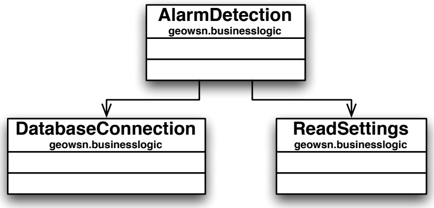


Figure 4.11: GeoWSN Alerting Service class diagram.

4.5.3.1 Class Description

A brief description of the classes of the GeoWSN AlertingService, shown in Figure 4.11 is listed in the following.

AlarmDetection. This class is the main class, instances the required objects and holds the functionality of the alarm detection. Alarm detection means verifying the velocity value of each node compared to the threshold value.

ReadSettings. This class reads the configuration settings for the database connection from a file.

DatabaseConnection. This class manages the database connection and provides a number of methods for the interaction with the database.

4.5.3.2 Operation

The program flow of the alarm detection task of the GeoWSN AlertingService is illustrated in Figure 4.12. The alarm detection task starts and switches in the sleep state. From this state the task wakes up periodically (default 1 *sec*) and checks if a triggered alarm is confirmed. If not, the threshold information and the velocity data for each node are gathered from the database. In the next state the alarm detection takes place. Therefore, the algorithm checks if a velocity value of a node exceeded the defined threshold. After this state, the algorithm updates the alarm and the alarm confirm flag in the database.

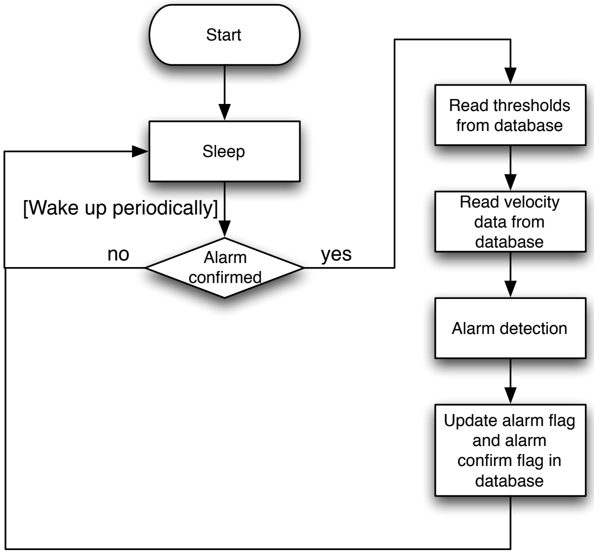


Figure 4.12: Simplified program flow of the GeoWSN AlertingService alarm detection task.

4.6 GeoWSN Web Service

The GeoWSN Web Service defines the interface between the web application and the database system. This API is implemented as a REST Web service in Java. Figure 4.13 shows the components and the embedding into the GeoWSN monitoring and early warning system. As a web server an Apache Tomcat[8], which contains a HTTP server and servlet containers, was chosen [70].

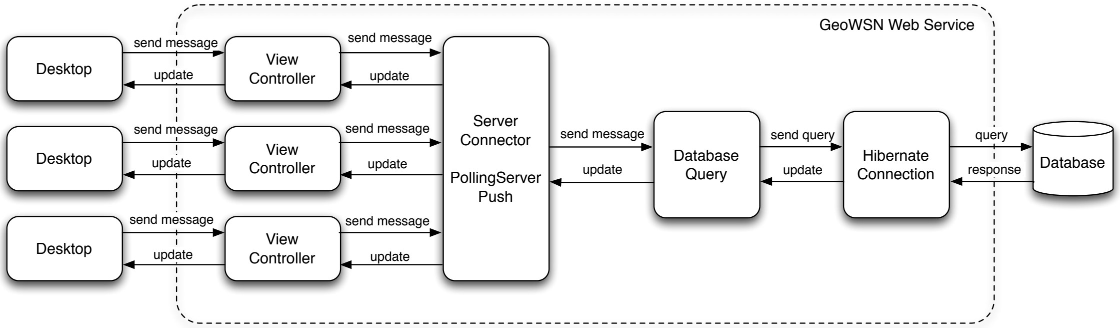


Figure 4.13: Architectural overview of the web service components.

A monitoring and early warning system must update each user's content periodically. The View Controller maintains the desktop content of each user. The View Controller indicates a thread running on the web service. These thread sending request messages to a single Server Connector thread. By the use of the Server Connector component a polling server push technology is provided. Here, the client thread (view controller) periodically queries the server connector thread. The client thread changing the data automatically depending on the data, which the server connectors thread gathers from the back-end system. That means the user content is only changed, if the gathered data set has been changed. The Server Connector manages the requests of the view controllers and forwards these requests to the Database Query component. Over the Hibernate Connection component the required data are gathered from the database.

4.6.1 Object-Relational Mapping with Hibernate

Object-Relational Mapping (ORM) means storing Plain Old Java Objects (POJOs) in a relational database and creating POJOs from a corresponding dataset of the relational database [71]. The architecture of Hibernate is shown in Figure 4.14 and provides transparent persistence for POJOs [13].

---

[8] Apache Tomcat 7: http://tomcat.apache.org/download-70.cgi (effective [February 14(th), 2013])

Therefore, persistent classes that are mapped to a database table have to be defined and must fulfill the following requirements:

- no-argument constructor of the class
- setter/getter methods for the attributes

The information how the class is made persistent is provided in a mapping file. The mapping file is available in an XML format. For the data manipulation of a database and the database query different interfaces supported by the Hibernate. In this project the criteria interface is used. With the help of this interface it is possible to create and execute object-oriented queries.

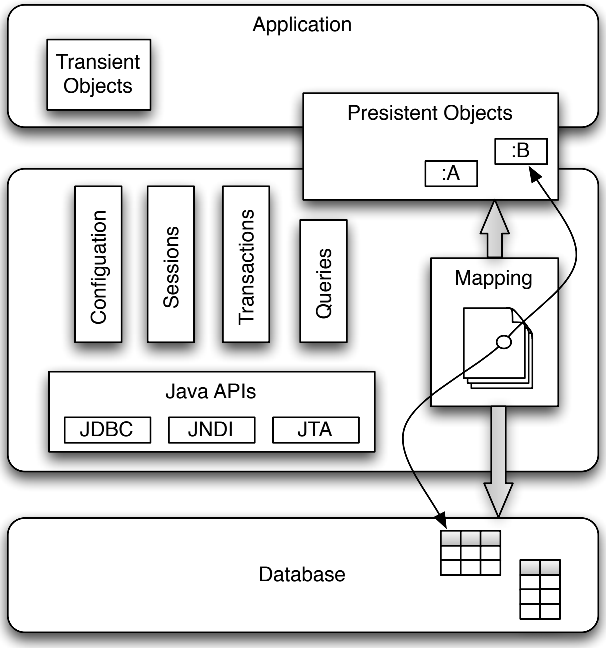


Figure 4.14: Hibernate architecture (reproduced from [13].

4.6.2 Operation

The web service can be separated into three tasks. First, the web application task, which represents the client threads. Then, the generate task that gathers the required data from the database. Finally, the watchdog task, which controls the generate task. Figure 4.15 shows the program flow of the generate task of the web service. As mentioned before, this task gathers the required information for the visualization in the browser.

If the web application task requests data the generate task is initialized and starts the operation. After the start process the Hibernate connection is initialized. If the generate task is set active, the state, network and position information is generated. Before the generate chart information state is reached the following requirement must be fulfilled: Is any client thread gathering chart information or should a new chart visualized. Is this true the generate chart information state gathers the required information for the selected chart. In the next state the minute timer for the selected chart is started. The next step checks, if an event occurs, which signals that a chart information must be updated. If this is the case, the chart information for the selected charts are generated. During the next operation the task switches in the sleep state, where the task wakes up periodically (default 1 *sec*).

In case of a killing event, this means no data request over a defined time interval, the task closes the opened connections and stops the operation.

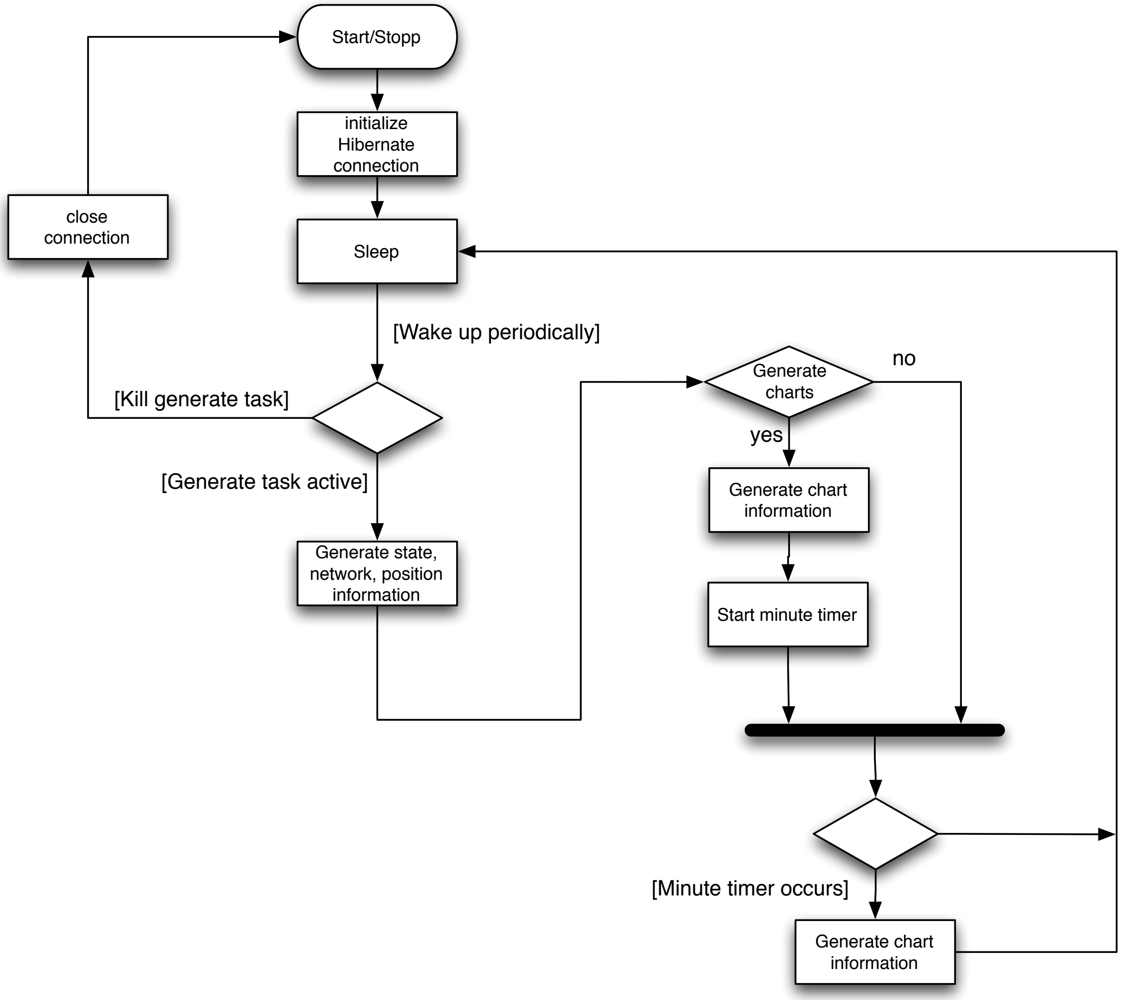


Figure 4.15: Simplified program flow of the GeoWSN web service generate task.

The program flow of the watchdog task is shown in Figure 4.16. This task checks periodically (default 20 *sec*) if data are requested from the client threads. Depending on the outcome of this query the watchdog task controls the generate task of the web service. In case that no further information has to be gathered, the watchdog task triggers the kill event for the generate task and stops its own operation.

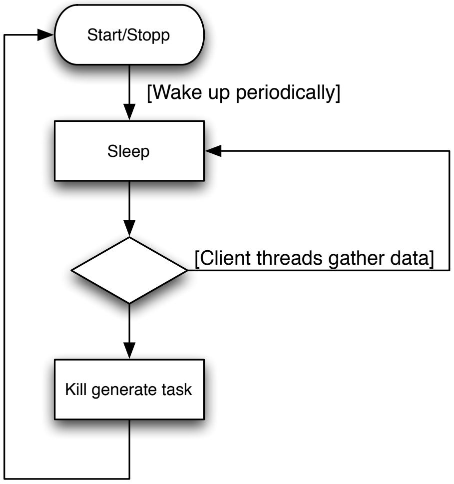


Figure 4.16: Simplified program flow of the GeoWSN web service watchdog task.

4.7 Database

The GeoWSN Database is the back-end of several applications, like positioning, data modeling, and visualization, of the monitoring and early warning system.

As database management system MySQL was chosen. This database management system brings a number of benefits with it. Two of the greatest benefits are, on the one hand that MySQL is an open-source database, which is constantly evolving, and on the other hand it is possible to use application specific storage engines, like InnoDB and in the foreground an optimizer increases the performance of the whole database management system.

4.7.1 Database Structure

The information, which are stored in the MySQL database, are gathered from the wireless sensor network and the WWW. Figure 4.17 shows the information flow.

For the RTK algorithm a reference station that gathers additional satellite data (satellite clock, ionospheric parameters) to the satellite raw data information of the wireless

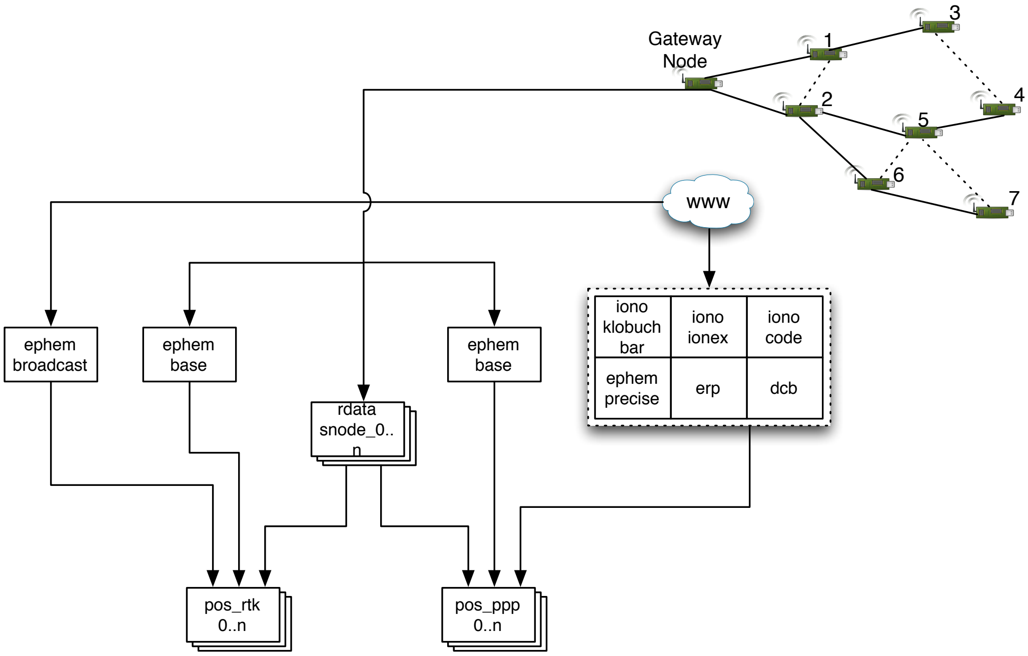


Figure 4.17: Information flow for the location determination.

sensor network is needed. In the GeoWSN Project the gateway node of the sensor network serves as reference station.

The PPP positioning technique needs additional precise information of the Global Positioning System from the World Wide Web, like the orbit parameter, or clock error parameter. The database consists of several tables that are described in more detail in Section 4.7.2. Figure 4.18 and Figure 4.19 show the flow of the input data of the respective near-real time positioning technique.

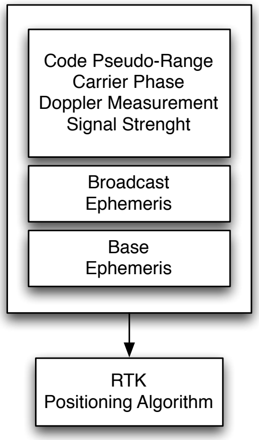


Figure 4.18: Input data RTK

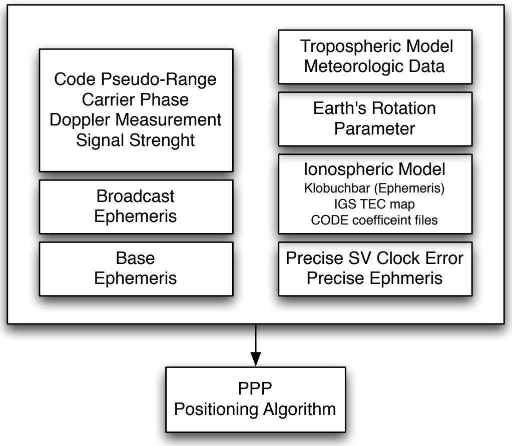


Figure 4.19: Input data PPP

4.7.2 Entity Description

The following section describes each entity of the database in more detail and information about the further usage is given. Also the source to the data for the described entity is mentioned.

4.7.2.1 Satellite and Sensor Raw Data

The raw data information of each sensor node is stored in an own database table. The structure of these database tables is shown in Table 4.6. This table contains several satellite data and sensor raw data of the sensor node. Further information about the network topology, like net load or transmission routes and about the energy extraction of the energy harvesting module are gathered in the table. The entity rdata_snode_0...n is dimensioned for 32 satellites, the maximum amount of satellites within the GPS.

The data in this table are the basis for the positioning algorithms and the visualization and are gathered from the sensor nodes in the sensor network.

Table 4.6: Description of database entity **rdata_snode_0...n**

Entitiy: rdata_snode_0...n				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	observation time, GPS week
3	gps_msecond	unsigned int	msec	observation time, GPS millisecond
4	epoch_data	timestamp	YMDhms	storage time (UTC); index
5	time_of_measurement	timestamp	YMDhms	time of the measurement (UTC); index
6	pr_prn1	unsigned double	m	code pseudo-range satellite PRN1
...	...	...	...	...
37	pr_prn32	unsigned double	m	code pseudo-range satellite PRN32
38	cp_prn1	double	cyles	carrier phase satellite PRN1
...	...	...	...	...
69	cp_prn32	double	cyles	carrier phase satellite PRN32

70	ds_prn1	double	Hz	Doppler measurement satellite PRN1
...	...	...	...	...
101	ds_prn32	double	Hz	Doppler measurement satellite PRN32
102	sn_prn1	unsigned double	dBHz	signal strength satellite PRN1
...	...	...	...	...
133	sn_prn32	unsigned double	dBHz	signal strength satellite PRN32
134	gps_state	tinyint	-	state of GPS receiver
135	sens_temp_sys	double	°C	sensor node temperature
136	sens_temp_air	double	°C	air temperature
137	sens_temp_gnd	double	°C	soil temperature
138	sens_luminance	double	lux	luminance
139	sens_humidity	double	%rf	relative humidity
140	sens_acc_x	smallint	mg	acceleration X direction
141	sens_acc_y	smallint	mg	acceleration Y direction
142	sens_acc_z	smallint	mg	acceleration Z direction
143	sens_state	tinyint	-	state of the sensors
144	net_neighbors	tinyint	-	connection with net neighbors
145	net_load	unsigned smallint	Byte/s	net load of sensor node
146	net_retransm_count	unsigned smallint	-	count of retransmission
147	net_transm_route_0	tinyint	-	1st participant on the transmission route
...	...	...	...	...
152	net_transm_route_5	tinyint	-	6th participant on the transmission route
153	net_transm_rssi_0	tinyint	-	transmission quality 1st hop
...	...	...	...	...
159	net_transm_rssi_6	tinyint	-	transmission quality 7th hop
160	engy_input	integer	mJ	energy extraction in the last epoch
161	engy_state	tinyint	-	kind of energy source
162	engy_stroe_level	integer	mV	charge level of the accumulator
163	engy_store_state	tinyint	-	state of the accumulator

4.7.2.2 Sensor Nodes Information

The Table 4.7 holds information of all sensor nodes in the wireless sensor network. With the table it is possible to address and manage other tables, like the raw data table in the section before.

Table 4.7: Description of database entity **sensor_nodes**

Entitiy: sensor_nodes				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	id_node	unsigned int	-	id of the sensor node in the WSN
3	x_coordinate	double	m	position of the sensor node in X direction
4	y_coordinate	double	m	position of the sensor node in Y direction
5	z_coordinate	double	m	position of the sensor node in Z direction

4.7.2.3 Base Station Ephemeris

The Table 4.8 holds the ephemeris of the satellites that are gathered by the GPS receiver of the gateway node . The gateway node is a part of the base station, so no extra communication amongst the sensor nodes is necessary.

Table 4.8: Description of database entity **ephem_base**

Entitiy: ephem_baset				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	prn	unsigned int	-	satellite id
3	epoch_data	timestamp	YMDhms	storage time (UTC)
4	iode	unsigned int	-	issue of data (ephemeris)
5	crs	double	m	sine harmonic correction term in the orbit radius
6	delta_n	double	rad/s	mean motion difference from computed value
7	m_0	double	rad	mean anomaly at reference time
8	cuc	double	rad	cosine harmonic correction term to the argument of latitude
9	ecce	double	-	eccentricity
10	cus	double	rad	sine harmonic correction term to the argument of latitude

11	sqrt_a	double	$\sqrt{m}$	square root of the semi-major axis
12	toe	unsigned int	s	reference time ephemeris; GPS second
13	cic	double	rad	cosine harmonic correction term to the angle of inclination
14	omage_0	double	rad	longitude of ascending node of orbit plane at weekly epoch
15	cis	double	rad	sine harmonic correction term to the angle of inclination
16	i_0	double	rad	inclination angle at reference time
17	crc	double	m	cosine harmonic correction term in the orbit radius
18	kl_omega	double	rad	argument of perigee
19	omega_dot	double	rad/s	rate of right ascension
20	i_dot	double	rad/s	rate of inclination angle

4.7.2.4 Broadcast Ephemeris and Satellite Clock Information

The Table 4.9 holds the broadcast ephemeris of the satellites. The broadcast ephemeris are gathered from the WWW. The ionospheric parameters included in the broadcast ephemeris are stored in the Table 4.10. These parameters are required for the Klobuchbar Model [72].

Table 4.9: Description of database entity **ephem_broadcast**

Entitiy: ephem_broadcast				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	prn	unsigned int	-	satellite id
3	epoch_data	timestamp	YMDhms	storage time (UTC)
4	iode	unsigned int	-	issue of data (ephemeris)
5	crs	double	m	sine harmonic correction term in the orbit radius
6	delta_n	double	rad/s	mean motion difference from computed value
7	m_0	double	rad	mean anomaly at reference time
8	cuc	double	rad	cosine harmonic correction term to the argument of latitude
9	ecce	double	-	eccentricity
10	cus	double	rad	sine harmonic correction term to the argument of latitude

11	sqrt_a	double	$\sqrt{m}$	square root of the semi-major axis
12	toe	unsigned int	s	reference time ephemeris; GPS second
13	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
14	cic	double	rad	cosine harmonic correction term to the angle of inclination
15	omage_0	double	rad	longitude of ascending node of orbit plane at weekly epoch
16	cis	double	rad	sine harmonic correction term to the angle of inclination
17	i_0	double	rad	inclination angle at reference time
18	crc	double	m	cosine harmonic correction term in the orbit radius
19	kl_omega	double	rad	argument of perigee
20	omega_dot	double	rad/s	rate of right ascension
21	i_dot	double	rad/s	rate of inclination angle
22	clk_bias	double	s	SV clock bias correction coefficient
23	clk_drift	double	s/s	SV clock drift correction coefficient
24	clk_driftrate	double	s/s^2	SV clock drift rate correction coefficient
25	leap	tinyint	s	clock data reference time of week
26	sv_accur	double	-	SV clock accuracy index
27	sv_health	unsigned int	-	SV clock accuracy change index
28	tgd	double	s	estimated group delay differential
29	iodc	unsigned int	-	issue of data (clock)

Table 4.10: Description of database entity **iono_klobuchar**

Entitiy: iono_klobuchar				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	ion_alpha1	double	s	ionospheric parameter α_1
6	ion_alpha2	double	s/rad	ionospheric parameter α_2
7	ion_alpha3	double	s/rad^2	ionospheric parameter α_3
8	ion_alpha4	double	s/rad^3	ionospheric parameter α_4
9	ion_beta1	int	s	ionospheric parameter β_1

10	ion_beta2	int	s/rad	ionospheric parameter β_2
11	ion_beta3	int	s/rad^2	ionospheric parameter β_3
12	ion_beta4	int	s/rad^3	ionospheric parameter β_4

4.7.2.5 Precise Ephemeris and SV clock error

The PPP positioning technique needs further information for determining the sensor node position. One of this additional information is the precise ephemeris and the Satellite Vehicle (SV) clock error. This information is acquired from an external source like the International GNSS Service (IGS) and represented in a SP3 file. The contained data – the actual satellite position and the satellite clock error – are an input source of the PPP algorithm and stored in the Table 4.11.

Thereby, different data types can be distinguished, e.g. ultra-rapid, or rapid. To make the determination of the satellite position in real-time available ultra-rapid data, which contains predicted values, are used.

Table 4.11: Description of database entity **ephem_precise**

Entitiy: ephem_precise				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	prn	unsigned tinyint	-	satellite id
6	x_coordinate	double	m	X coordinate, precise ephemeris
7	y_coordinate	double	m	Y coordinate, precise ephemeris
8	z_coordinate	double	m	Z coordinate, precise ephemeris
9	clock	double	s	SV clock correction coefficient
10	x_sdev	unsigned double	m	standard deviation x
11	y_sdev	unsigned double	m	standard deviation y
12	z_sdev	unsigned double	m	standard deviation z
13	c_sdev	unsigned double	s	standard deviation clock correction coefficient

14	clock_event	unsigned tinyint	-	clock event flag
15	clock_prediction	unsigned tinyint	-	clock prediction flag
16	maneuver_flag	unsigned tinyint	-	maneuver flag
17	orbit_prediction	unsigned tinyint	-	orbit prediction flag
18	type	char	-	precise orbit type (ultra-rapid, rapid, final)

4.7.2.6 Ionospheric Models

To achieve accurate positions the ionospheric influence must be considered. To determine the parameters, various ionospheric models can be used. One of the modest model is the Klobuchbar Model (Table 4.10). The ephemeris contains the information for building up this type of model. The Klobuchbar Model has been used previously by the RTK algorithm. For preciser or attentional ionospheric parameters two other models exist.

First, the Total Electronic Content (TEC) Model which is provided by the IGS as Ionosphere Map Exchange (IONEX) TEC maps. The in the IONEX file included parameters are stored in the Table 4.12. The second model is the Center of Orbit Determination in Europe (CODE) Coefficient Model. This type of model is a regional one and the model values are stored in Table 4.13:

Table 4.12: Description of database entity **iono_ionex**

Entitiy: iono_ionex				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	lat	double	deg	latitude
6	lon	double deg	longitude	
7	value_tecu	unsigned double	tecu	VTEC value

Table 4.13: Description of database entity **iono_code**

Entitiy: iono_code				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week

3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	degree_iono	int	deg	degree
6	order_iono	int	deg	order
7	value_tecu	double	TECU	VTEC value
8	rms_tecu	double	TECU	root mean square of the VTEC value
9	height_sl	unsigned int	m	height of single layer
10	lat_gmp	double	deg	latitude of the geomagnetic pole
11	lon_gmp	double	deg	longitude of the geomagnetic pole

4.7.2.7 Earth's Rotation Parameters

Table 4.14 shows the database structure how the earth's rotation parameters are stored. The parameters are obtained form the IGS.

Table 4.14: Description of database entity **erp**

Entitiy: erp				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	x_pole	double	arcs	pole coordinate x
6	y_pole	double	arcs	pole coordinate y
7	ut1utc	double	s	difference between UT1 and UTC
8	lod	double	s/d	day length
9	x_rt	double	arcs/d	change of the pole coordinate x
10	y_rt	double	arcs/d	change of the pole coordinate y

4.7.2.8 Differential Code Biases

Another input source for the PPP positioning technique are the Differential Code Biases (DCBs) that are acquired from an external source, e.g. IGS. Table 4.15 describes the structure of the DCB information.

Table 4.15: Description of database entity **dcb**

Entitiy: dcb				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	prn	unsigned tinyint	-	satellite id
6	dcb	double	s	differential code bias
7	dcb_rms	double	s	differential code bias RMS
8	type	char	-	DCB data type

4.7.2.9 Sensor Node Postion

For each sensor node and positioning algorithm (PPP and RTK) the determined position is stored in a database (Table 4.16). The content of the sensor node position tables are used as an input parameter in the GeoWSN alerting system.

Table 4.16: Description of database entity **pos_rtk_0...n / pos_ppp_0...n**

Entitiy: pos_rtk_0...n / pos_ppp_0...n				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	x_coordinate	double	m	X coordinate
6	y_coordinate	double	m	Y coordinate
7	z_coordinate	double	m	Z coordinate
8	std_x	double	m	standard variance X

9	std_y	double	m	standard variance Y
10	std_z	double	m	standard variance Z
11	velocity _x	double	cm/y	velocity of the sensor node in X
12	velocity _y	double	cm/y	velocity of the sensor node in Y
13	velocity _z	double	cm/y	velocity of the sensor node in Z
14	n_sat	unsigned int	-	number of satellites
15	gdop	unsigned double	-	geometric dilution of precision
16	pdop	unsigned double	-	position DOP
17	hdop	unsigned double	-	horizontal DOP
18	vdop	unsigned double	-	vertical DOP
19	integrity	unsigned int	TBD	integrity parameter
20	event	text	-	description of the event

4.7.2.10 Temporal Combined Sensor and Postion Information

The stored position and sensor data solutions are temporally combined by an algorithm. With the help of this a better visualization performance of the different solutions like velocity or temperatures can be realized. The structure of the data is shown in Table 4.17.

This table contains the data, which are generate by the GeoWSN DataLogger. Furthermore, this table is the main interface of the visualization option in the web application.

Table 4.17: Description of database entity **data_logger_snode_0...n**

Entitiy: data_logger_snode_0...n				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	gps_week	unsigned smallint	w	reference time ephemeris; GPS week
3	gps_second	unsigned int	s	reference time ephemeris; GPS second
4	epoch_data	timestamp	YMDhms	storage time (UTC)
5	time_of_measurement	timestamp	YMDhms	time of the measurement (UTC); index
6	epoch_category	char	-	epoch category; d(ay), w(eek), m(onth), y(ear)
7	sens_temp_sys	double	°C	mean value of the sensor node temperature
8	sens_temp_air	double	°C	mean value or the air temperature

9	sens_temp_gnd	double	°C	mean value of the soil temperature
10	sens_luminace	double	lux	mean value of the luminance
11	sens_humidity	double	%rF	mean value of the relative humidity
12	sens_acceleration_x	double	mg	peak value of the sensor node acceleration in X
13	sens_acceleration_y	double	mg	peak value of the sensor node acceleration in Y
14	sens_acceleration_z	double	mg	peak value of the sensor node acceleration in Z
15	ppp_velocity_x	double	cm/y	peak value of the sensor node velocity in X with PPP algorithm
16	ppp_velocity_y	double	cm/y	peak value of the sensor node velocity in Y with PPP algorithm
17	ppp_velocity_z	double	cm/y	peak value of the sensor node velocity in Z with PPP algorithm
18	rtk_velocity_x	double	cm/y	peak value of the sensor node velocity in X with RTK algorithm
19	rtk_velocity_y	double	cm/y	peak value of the sensor node velocity in Y with RTK algorithm
20	rtk_velocity_z	double	cm/y	peak value of the sensor node velocity in Z with RTK algorithm

4.7.2.11 Threshold Information

The threshold values and ranges for landslide events and low battery notifications are stored in Table 4.18.

Table 4.18: Description of database entity **threshold**

Entitiy: threshold				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	threshold_changed	tinyint	-	threshold settings changed (1 - yes, 0 - no)

3	velocity_threshold_red	double	cm/y	velocity warning level red
4	velocity_threshold_orange	double	cm/y	velocity warning level orange
5	velocity_threshold_min	double	cm/y	defined minimum value of velocity warning levels
6	velocity_threshold_max	double	cm/y	defined maximum value of velocity warning levels
7	acc_threshold_red	double	mg	acceleration warning level red
8	acc_threshold_orange	double	mg	acceleration warning level orange
9	acc_threshold_min	double	mg	defined minimum value of acceleration warning levels
10	acc_threshold_max	double	mg	defined maximum value of acceleration warning levels
11	engy_threshold_red	double	%	energy warning level red
12	engy_threshold_orange	double	%	energy warning level orange
13	engy_threshold_min	double	%	defined minimum value of energy warning levels
14	engy_threshold_max	double	%	defined maximum value of energy warning levels

4.7.2.12 Alarm Information

The Table 4.19 holds the alarm information, like type of the alarm. This table contains the data, which are generated by the GeoWSN AlertingService and the XMPP service.

Table 4.19: Description of database entity **alarm**

Entitiy: alarm				
No.	Field	Type	Unit	Description
1	id	unsigned int	-	primary key, auto-increment
2	alarm_flag	tinyint	-	indicates an alarm (1 - yes, 0 - no)
3	alarm_acknowledged	tinyint	-	alarm acknowledged (1 - yes, 0 - no)
4	alarm_status	tinyint	-	status flag for the XMPP service
5	alarm_type	tinyint	-	type of the alarm (velocity, acceleration, energy)

4.8 Web Application

The web application, called GeoWSN monitoring, provides a visualization and a configuration environment. Today, the list of web application frameworks seems to be endless. This issue makes it hard to decide the correct framework for the web application, which fulfills several requirements. For the GeoWSN monitoring web application the ZK open-source Ajax web application framework was chosen. Through the integration of various open-source JavaScript libraries, e.g. dygrahs[9], and canvas[10], additional functionality was achieved.

As previously mentioned the web application can be separated into two environments, the visualization, and the configuration environment. The visualization environment offers the possibility to illustrate different wireless sensor network specific parameters and also gathered information of the sensor network to the end-user. By the use of the configuration environment that can be only entered by authorized users, it is possible to change configuration settings, like threshold levels. The web application also provides near-real time alerting messages in case of slope failure in both environments.

The whole web application was designed to require as little user interactions as possible.

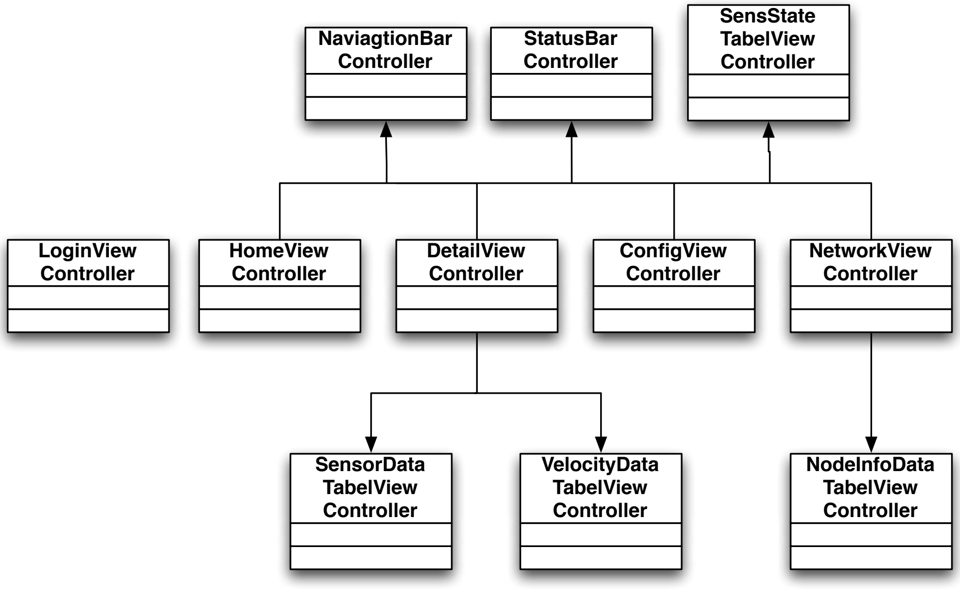


Figure 4.20: Web application class diagram

---

[9] dygraphs JavaScript Visualization Library (effective [February 14(th), 2013])

[10] HTML5 Canvas (effective [February 14(th), 2013])

4.8.1 Class Description

A brief description of the classes is listed in the following.

LoginViewController. The LoginViewController builds up the login screen.

HomeViewController. This class represents the map view. The position of the nodes is visualized on a map. Therefore, the class communicates with the Google Maps API.

DetailViewController. The detail / diagram view is represented by the DetailViewController class.

NetworkViewController. This class handles the visualization of the network topology and energy state information of each node.

ConfigViewController. With the help of this class the authorized user are able to change threshold settings for the early warning system.

NavigationBarController. This class contains the navigation controller.

StatusBarController. This class displays the warning level of the early warning system to the user.

SensStateTabelViewController. This table view controller class provides the state of the different external and internal sensors and the GPS receiver of each node.

NodeInfoDataTabelViewController. The energy information of each sensor nodes are illustrated tabular, which is handled by this class.

SensorDataTabelViewController. This class provides the sensor data to the end-user.

VelocityDataTabelViewController. This class gives an overview of the velocity and acceleration data.

4.8.2 User Interface and Interaction

The web application consists of five main views. The first view that the user enters is the login view. In this view the user authenticates himself to the system. After the authentication the map view, shown in Figure 4.21, is reached. In this view the position of the nodes are illustrated, and also some statistical information are mentioned. Figure 4.22 shows the network view of the web application. The network view provides two options, the network topology including network specific parameters, and the visualization of sensor node energy information. The fourth view that is supported is the detail/diagram view. The detail/diagram view, shown in Figure 4.23, contains several diagrams and data tables of the measurement data. The last view of the web application is the configuration view, which can be entered by authorized users only. In this view threshold and user configuration settings can be changed.

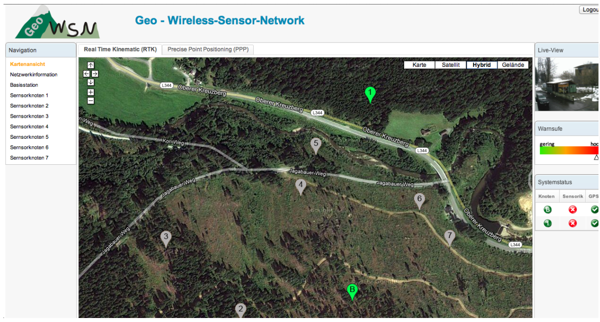


Figure 4.21: Web application map view.

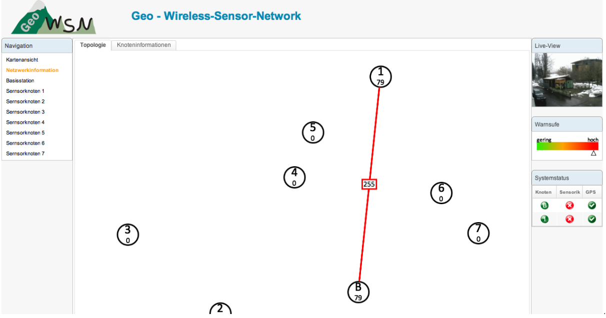


Figure 4.22: Web application network view.

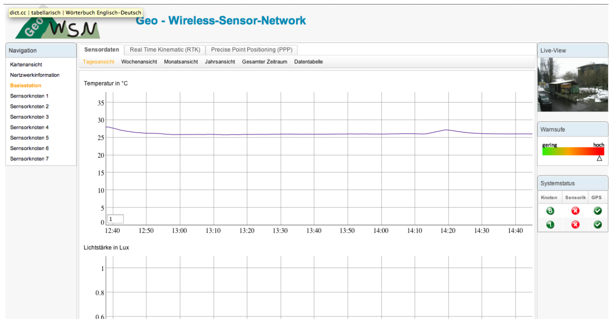


Figure 4.23: Web application detail and diagram view.

The user interface of the GeoWSN monitoring web application is illustrated in more detail in Figure B.4, Figure B.5, and Figure B.6.

4.8.3 API Communication

The communication flow in Figure 4.24 shows the interaction between the GeoWSN web application and the back-end. As interface for this communication the GeoWSN web service is used.

First, a login statement with username and password information is sent to the web service. This device queries the GeoWSN user database if a tuple with the transmitted username and password exists. In case of non-matching elements a 'no authentication' statement is sent back to the web application. By a matching entry the web service requests data from the GeoWSN database. The GeoWSN web service generates a HTML page with the requested data. The generated page is displayed by the web application in the browser of the user.

The GeoWSN web application can be used for monitoring as well as for configuration. As mentioned before, for both options the user has to authenticate himself to the system. To use the configuration option a special permission is required. By an authorized access the selected database table is updated with the new information. After this transaction the GeoWSN web service generates a HTML page with the changed configuration information, which is finally displayed to the user.

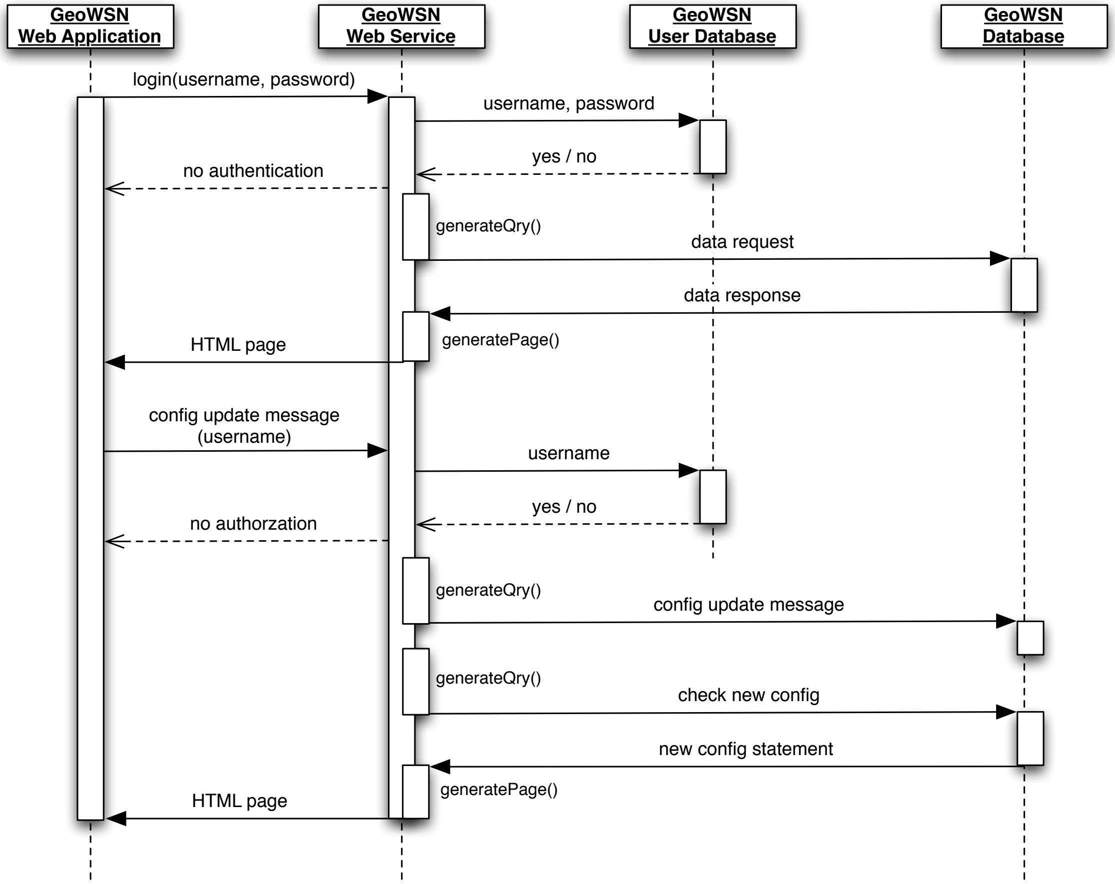


Figure 4.24: Web application communication flow.

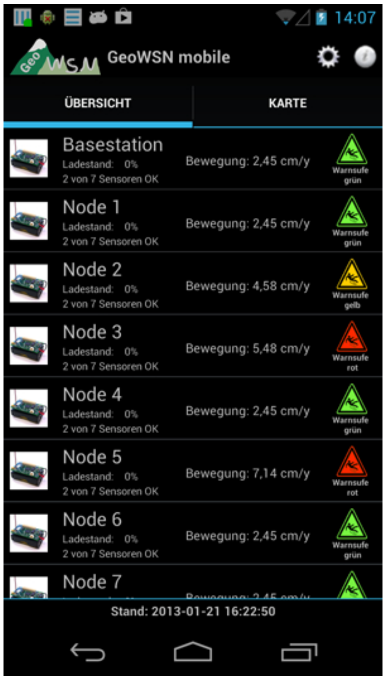
4.9 Mobile Client

In order to achieve a portable early warning system, a client using the Android platform has been implemented. The mobile client, called GeoWSN mobile, communicates via the MMC with the back-end of the early warning and monitoring system and runs on any Android smartphone.

4.9.1 User Interface and Interaction

The Android application consists of three main views. The first view, which is entered after start up, is the detail view. The detail view, shown in Figure 4.25, gives a quick overview of each node including the velocity information, warning level indicator, and node specific parameters, like sensor state. In this view also the option to display a number of sensor parameters is integrated. Selecting the desired node cell enters this functionality. The view in Figure 4.26 shows the map view of the Android client application. To display the position of each node a Google Map API is used. By selecting a node indicator detailed position information is displayed. In both views the user has the possibility to select one of the two positioning algorithms by pressing the configuration button.

Since the application was designed for energy-efficient operation no polling algorithm is implemented. To update the view the user must pull down the table of the detail view for instant updating. However, in case of a threshold violation the client application is automatically reloaded. This is realized over the MMC, which integrates the XMPP instant message and presence protocol.



Basestation
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 1
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 2
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 4,58 cm/y
Warnsufe gelb

Node 3
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 5,48 cm/y
Warnsufe rot

Node 4
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 5
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 7,14 cm/y
Warnsufe rot

Node 6
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 7

Stand: 2013-01-21 16:22:50

Figure 4.25: Android app detail view.

Figure 4.26: Android app map view.

The user interface of the native Android application (GeoWSN mobile) is illustrated in more detail in Figure B.1, Figure B.2, and Figure B.3.

4.9.2 API Communication

Figure 4.27 illustrates the communication flow between the GeoWSN Android App and the back-end of the system.

The Android application sends a data and authentication request over an internal broadcast where the request is packed into a XMPP stanza. This stanza is forwarded to the Openfire XMPP Server. The XMPP server verifies the user with the help of the GeoWSN user database. If the authentication is correct the data request is committed to the GeoWSN XMPP Service. The service unpacks the XMPP stanza, which includes the data request, and queries the GeoWSN Database. The XMPP service converts the gathered data from to database to a XML stream. This stream is packed to a XMPP stanza and takes the way back to the XMPP application running on the mobile device. There, the XMPP stanza containing the requested data is unpacked and the data are forwarded to the Geological Wireless Sensor Network (GeoWSN) mobile application. This application displays the system information to the end-user.

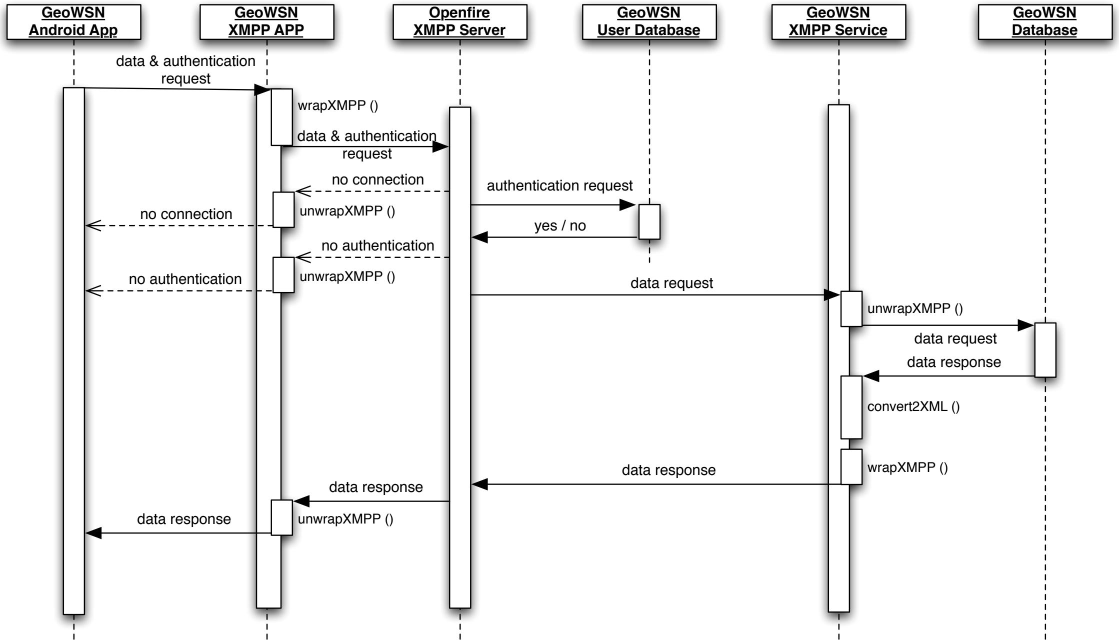


Figure 4.27: Communication flow of the mobile client.

4.10 Experimental Results

This section shows the evaluation of the GeoWSN database system. The evaluation should ensure that the implemented database system fulfills the requirements regarding handling large data volumes, long lifetime, and consistent and reliable data storage. Due to the database system is the back-end of the GeoWSN database and visualization system (GeoWSN early warning and monitoring system) the evaluation of the database system is an important issue.

4.10.1 Test Settings and Test Scenario

The tests have been performed on a 4-cores Intel Xeon E3 3.1GHz server with 8GB DDR3 RAM. As operating system the Linux openSUSE 12.1 (x86_64) is running on the server.

A Java program simulates a periodical MySQL database access and insert data sets over a JDBC driver into the database tables (mentioned in Table 4.6, and Table 4.16). In a defined interval (default: 30 *sec*) a join operation between the raw data table and the position table is performed and the the last ten database entries are selected. Per day 43200 entries are stored in each table. Listing 4.2 shows an example of the SQL query that is used for the evaluation.

Listing 4.2: Evalution SQL query.

```
SELECT r.id FROM rdata_snode_0 AS r LEFT JOIN pos_ppp_0 AS p
ON(r.gps_week = p.gps_week AND r.gps_msecond = p.gps_msecond)
WHERE r.epoch_data >= '2013-03-03 12:00:00'
```

4.10.2 Test Results

For the evaluation of the database system three test cases have been performed: indexed database, compressed database, and database memory consumption.

4.10.2.1 Indexed Database

The first test comprises the performance between a database that include database indexes and a database that include no database indexes on table columns for different intervals, like a day, or a week. In Figure 4.28 the results of the test are illustrated. The first case uses a database, with an index on *gps_week* and *gps_msecond*. The second one a non indexed database is used.

The results of the test scenario show that the performance of the database can be increased by the usage of database indexes. Looking at the "day" and "2 days" intervals a uprating with the factor 1.65 can be reached. By the "week" interval the factor is 16.5. This could be explained by the fact that the MySQL database indexes are stored in a B-tree. Such B-trees are balanced trees, which have a worst case access time of $O(log(n))$, where n are the number of elements in the table. By large databases the B-tree is split into smaller ones. Thereby the performance of databases that store a high number of elements is increased.

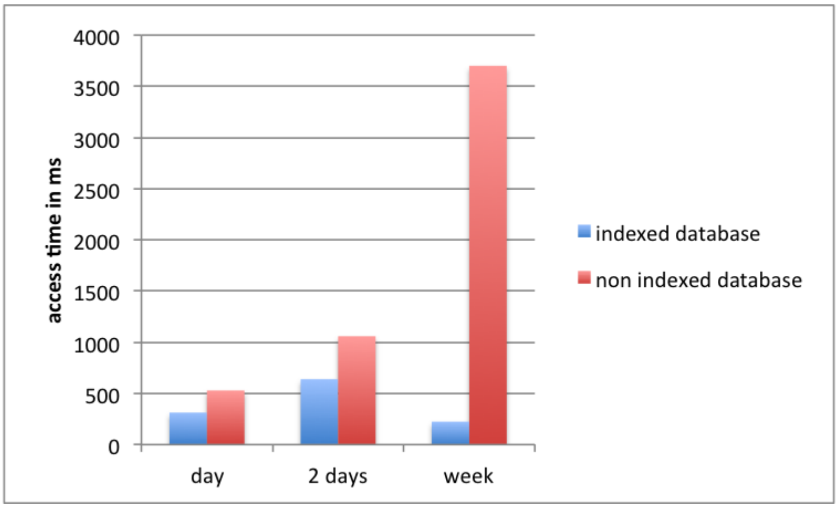


Figure 4.28: Web application communication flow.

4.10.2.2 Compressed Database

The second test shows how the performance of the database system and the visualization system is influenced by the usage of the GeoWSN DataLogger. Due to the DataLogger service the data of the raw data table and the position tables are temporally combined with algorithms. Hence, the number of elements in the database tables is reduced by this service.

The test results, represented in Figure 4.29, show that the performance of the database and visualization system is increased.

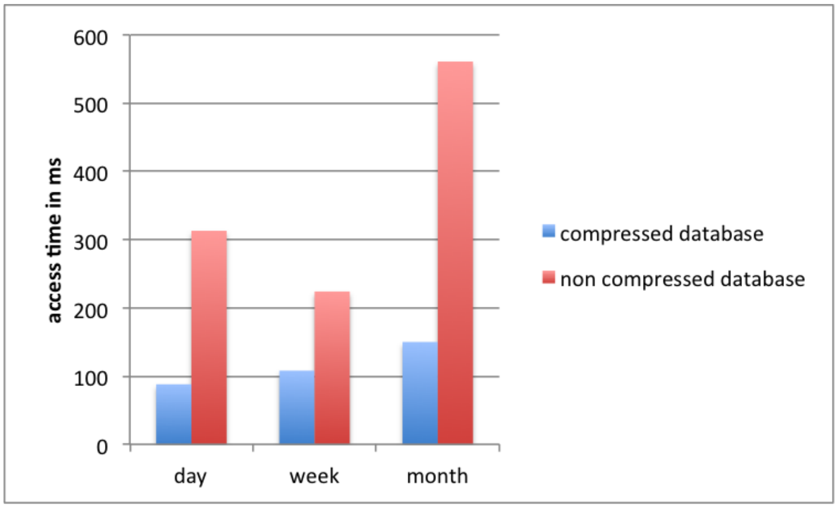


Figure 4.29: Web application communication flow.

4.10.2.3 Memory Consumption

The third test treats with the memory consumption of the database system. Figure 4.30 illustrates the monthly memory consumption of the database system. The used memory of the raw data table and the position tables are included in the computation.

The result highlights that with the use of the GeoWSN DataLogger a reduction of memory consumption by a factor of 47.67 can be achieved. To avoid a memory over-run, the GeoWSN database and visualization system implement Linux cronjobs for deleting decaying data (older than two days) of several database tables, like raw data table. The compressed data are not influenced by the deleting algorithm of the cronjob.

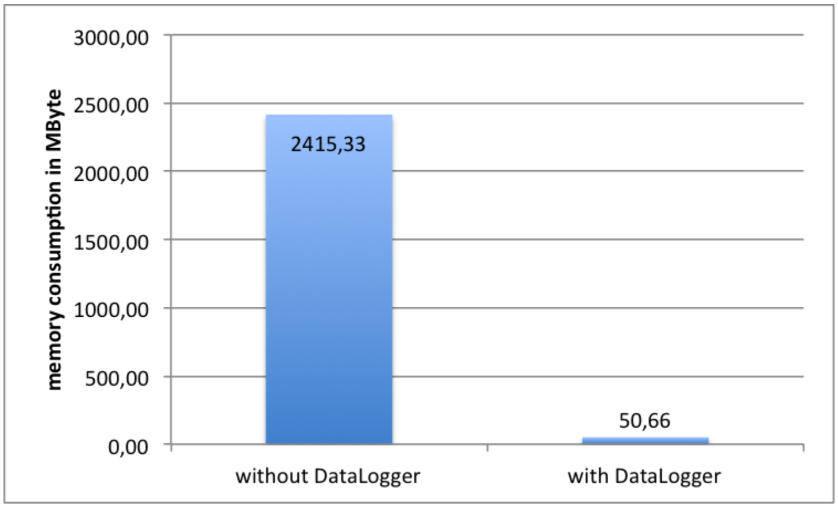


Figure 4.30: Web application communication flow.

4.11 Evaluation

Environmental monitoring in the context of early warning and monitoring systems is a wide research area and defines concepts that influences the development of such systems. The development of an early warning and monitoring system for slope movements depends on several requirements, which must be fulfilled.

The evaluation of the implemented database, visualization, and altering system shows that:

- the database system can handle large data volumes with high performance, due to the usage of database indexes.
- by using intelligent algorithms, which compress and delete decaying data, the life time of the database system is increased.

- through the MySQL database management system a consistent and reliable data storage is possible. This fact is given by the architecture of the database management system, where the best fitting storage engine, here InnoDB, can be selected.
- the integration of new components into the GeoWSN Linux services is easy to handle, due to the modular structure of this services.
- the GeoWSN Linux services are easy-to-use by providing a graphical user interface over a remote access.
- the execution of the GeoWSN tools, like Listener, DataLogger, and AlertingService, as a Linux service implicates a good run-time stability.
- the front-end applications provide secure mechanisms, such as authentication and authorization, to use the system.
- the front-end applications require as little user interactions as possible.
- a reliable and near-real time alerting system is integrated in the system.
- the web application provides an easy way for modifying the threshold and user settings.
- the web application provides chart and tabular data visualization, which is easy to handle.
- the mobile client is easy to handle and provides a self-explaining interface.

Chapter 5

Conclusion and Future Work

This chapter summarizes and concludes the results gathered in this thesis. An outlook gives an overview of the further implementation opportunities.

5.1 Conclusion

This thesis dealt with an early warning and monitoring system for slope movements and is part of the GeoWSN project. Thus, in this thesis relevant research in the field of environmental monitoring, and instant message and presence technology is presented. Environmental monitoring is one of the major research areas of wireless sensor networks, which builds the back-end of the GeoWSN project. For the near-real time early warning concept an Instant Messaging and Presence Technology (IM&P) is integrated.

The following sections summarize the single units of the GeoWSN early warning and monitoring system.

5.1.1 COM Driver for the Gateway Node

The gateway node software are extended by the COM driver. This extension is necessary for the data transfer between the gateway node of the wireless sensor network and the back-end server. The driver uses a modular structure and provides several transfer options for existing units and future extensions. Additionally to the COM driver a safety mechanism is integrated. In case of a broken communication between the gateway node and the server, no further data packets from the other network participants are accepted. With the help of this a fast memory over-run of the gateway node flash memory is prevented.

5.1.2 Database System

The database system represents the back-end of the early warning and monitoring system. The database system consist of two databases. On the one hand the user database, which contains the user information, and on the other hand the parameter database that stores the information for the monitoring, alerting, and visualization units. To fulfill several requirements, like handling large data volumes, long life time, and reliable data storage, the MySQL database management system was chosen.

The user database holds the information that is necessary for the authentication and authorization of the user against the system.

The parameter database consist of a number of tables, which stores the required data. Thereby, the database structure can be divided roughly into seven parts: the satellite and sensor raw data, the broadcast ephemeris, the ionospheric model, the precise ephemeris and satellite clock, the tropospheric model, the temporal combined sensor and position information, and the alarm and threshold information.

The information that is stored in the databases are gathered by the wireless sensor network, the World Wide Web, or SQL configuration scripts.

5.1.3 Service Package

The service package can be distinguished into two different kinds of services, those that offer information and those that gather information. Thus, the services are the fundamental link between the wireless sensor network and the database, and also the fundamental link between the database and the front-end monitoring and alerting applications.

The single components of the service package structure are mentioned in the following.

GeoWSN Listener. The Listener service can be described based on two tasks. First, the RX task that receives the incoming data stream of the sensor network via the USB interface of the server. Therefor, the gateway node and the GeoWSN Listener build up the interface between the sensor network, where the data are gathered, and the server, where the data are processed. The second task is the process task. This parses the received data stream based on the defined communication protocols and performs a first-level processing. Finally the processed data are stored in the database. In case of a broken database communication a back-up solution is integrated. Thereby, the not submitted SQL query is stored in a file, which can be later loaded into the database.

GeoWSN DataLogger. This service enhances the performance of the monitoring system. Thereby, two different algorithms combine, according to the data set, the sensor and position information temporally. In order not to influence the accuracy of the chart visualization the data are combined at varying intervals, for the day, week, month, and year representation, depending on the visualization interval.

GeoWSN AlertingService. This service triggers an alarm (setting the alarm flag in the database) if a predefined velocity threshold is exceeded.

The described components are implemented in general-purpose, concurrent, class-based, object-oriented computer programming language Java and are running as a Linux service.

5.1.4 RESTful Web Service

The web service is the interface between the database system and the web application client and it is designed according to the RESTful architectural style. The web service uses a modular structure that is implemented in Java. The view controller indicates a thread, which represents the desktop content of the user. For a multi-user communication a server connector layer has been integrated as communication interface between the client threads and the database. By this layer a polling server push technology is implemented in the system. The technology minimize the interactions between the web application client and the web service to a minimum. An objected-relational mapping technology, Hibernate, represents the interface to the database system.

5.1.5 Web Application

The web application – GeoWSN monitoring – has been developed using the ZK web application framework and provides a number of visualization and configuration environments. The configuration environment allows authorized users to change several parameters of the early warning and monitoring system, like threshold settings, or user configurations. Different visualization possibilities of the gathered data from the sensor network are offered by the visualization environment. This is done by chart, topology and tabular representations.

The mentioned environments provide an instant illustration of the important information for the early warning system, like sensor state, or warning level. Furthermore, in case of a threshold exceedance an alerting message is displayed.

The web application also integrates a web cam stream to monitor the hazardous area.

5.1.6 Mobile Client

As mobile client an application using the Android platform has been developed. The interface between the database system and the mobile clients is an adapted version of the Metis Mobile Cloud (MMC). With the MMC a near-real time alerting and monitoring over the XMPP instant message and presence technology is possible. The mobile client application – GeoWSN mobile – functions as the portable front-end of the early warning and monitoring system. The application provides two views. The first view gives an overview of the most important system parameters, and the second one displays the sensor node positions on a map.

5.2 Future Work

The implemented database and visualization system offers an open and flexible interface. A USB interface is used for communication between the gateway node of the wireless sensor network and the base station Linux server. In a follow project, a GPRS module will be added to the existing gateway node hardware. To support this module for data exchange an additional software component (a TCP SSL socket) has to be added to the GeoWSN Listener service. This TCP SSL socket will be used to receive incoming data and control the data exchange.

The visualization system of the GeoWSN early warning and monitoring system provides several mechanism, like authentication, to ensure a secure connection and communication. The implemented mechanisms between the web application (GeoWSN monitoring) and the GeoWSN web service should be analyzed to guarantee a secure connection.

Appendix A

Communication Protocol

Table A.1: Number format [16].

Short	Type	Size (Bytes)
U1	Unsigned Char	1
I1	Signed Char	1
X1	Bitfiled	1
U2	Unsigned Short	2
I2	Signed Short	2
X2	Bitfiled	2
U4	Unsigned Long	4
I4	Signed Long	4
X4	Bitfiled	4
R4	Float	4
R8	Double	8
Bn	Bitfield with length n	n bit
B'n	Signed Bitfield (LSBs & MSBs)	n bit

Table A.2: Subframe 1 [16].

	Bit Offset	Number Format	Scale Factor (LSB)	Name	Unit	Description
Word 3	0	X10	1	WN	weeks	Week number
	10	B2	1	C/A or P on L2	-	Codes on L2 channel
	12	B4	-	URA index	-	SV accuracy
	16	B6	1	SV health	-	SV health
	22	B2	-	IODC (MSBs)		Issue of Data, Clock
	24	B6	-	P	-	Parity bits
Word 4	30	X1	1	L2 P data flag	-	Data flag for L2 P-code
	31	B23	-	R	-	Reserved
	54	B6	-	P	-	Parity bits
W 5	60	X24	-	R	-	Reserved
	84	B6	-	P	-	Parity bits
W 6	90	X24	-	R	-	Reserved
	114	B6	-	P	-	Parity bits
Word 7	120	X16	-	R	-	Reserved
	136	I1	2^{-31}	T_{GD}	s	Estimated group delay differential
	144	B6	-	P	-	Parity bits
Word 8	150	U1	-	IODC (LSBs)		Issue of Data, Clock
	158	U2	2^{4}	t_{oc}	s	SV clock correction parameter
	174	B6	-	P	-	Parity bits
Word 9	180	I1	2^{-55}	a_{f2}	s/s^2	SV clock correction parameter
	188	I2	2^{-43}	a_{f2}	s/s	SV clock correction parameter
	204	B6	-	P	-	Parity bits
Word 10	210	X'22	2^{-31}	a_{f0}	s	SV clock correction parameter
	232	B2	-	t	-	Noninformation bearing bits
	234	B6	-	P	-	Parity bits

Table A.3: Subframe 2 [16].

	Bit Offset	Number Format	Scale Factor (LSB)	Name	Unit	Description
Word 3	0	U1	-	IODE	-	Issue of Data (Ephemeris)
	8	I2	2^{-5}	C_{rs}	m	Amplitude of the Sine Harmonic Correction Term to the Orbit Radius
	24	B6	-	P	-	Parity bits
Word 4	30	I2	2^{-43}	Δn	semi-circles/s	Mean motion difference from computed value
	46	I1	2^{-31}	M_0 (MSBs)	semi-circles	Mean anomaly at reference time
	54	B6	-	P	-	Parity bits
W 5	60	I3	2^{-31}	M_0 (LSBs)	semi-circles	Mean anomaly at reference time
	84	B6	-	P	-	Parity bits
Word 6	90	I2	2^{-29}	C_{UC}	radians	Amplitude of the Cosine Harmonic Correction Term to the Argument of Latitude
	106	U1	2^{-33}	e	-	Eccentricity
	114	B6	-	P	-	Parity bits
W 7	120	U3	2^{-33}	e	-	Eccentricity
	144	B6	-	P	-	Parity bits
Word 8	150	I2	2^{-29}	C_{US}	radians	Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude
	166	U1	2^{-19}	$\sqrt{A}$ (MSBs)	$\sqrt{m}$	Square root of the semi-major axis
	174	B6	-	P	-	Parity bits
W 9	180	U1	2^{-19}	$\sqrt{A}$ (LSBs)	$\sqrt{m}$	Square root of the semi-major axis

	204	B6	-	P	-	Parity bits
Word 10	210	U2	2^4	t_{oe}	s	Reference time ephemeris
	226	B1	-	Fit interval flag	-	
	227	B5	-	AODO	-	Age of Data Offset
	232	B2	-	t	-	Noninformation bearing bits
	234	B6	-	P	-	Parity bits

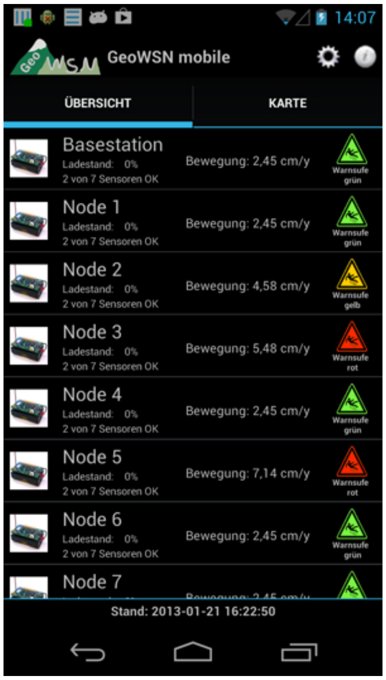
Table A.4: Subframe 3 [16].

	Bit Offset	Number Format	Scale Factor (LSB)	Name	Unit	Description
Word 3	0	I2	2^{-29}	C_{ic}	radians	Amplitude of the cosine harmonic correction term to the angle of inclination
	16	I1	2^{-31}	Ω_0 (MSBs)	semi-circles	Longitude of ascending node of orbit plane at weekly epoch
	24	B6	-	P	-	Parity bits
W 4	30	I3	2^{-31}	Ω_0 (LSBs)	semi-circles	Longitude of ascending node of orbit plane at weekly epoch
	54	B6	-	P	-	Parity bits
Word 5	60	I2	2^{-29}	C_{is}	radians	Amplitude of the sine harmonic correction term to the angle of inclination
	76	I1	2^{-31}	i_0	semi-circles	Inclination angle at reference time
	84	B6	-	P	-	Parity bits
W 6	90	I3	2^{-31}	i_0	semi-circles	Inclination angle at reference time
	114	B6	-	P	-	Parity bits
Word 7	120	I2	2^{-5}	C_{rc}	m	Amplitude of the cosine harmonic correction term to the orbit radius

	136	I1	2^{-31}	ω (MSBs)	semi-circles	Argument of perigee
	144	B6	-	P	-	Parity bits
W 8	150	I3	2^{-31}	ω (LSBs)	semi-circles	Argument of perigee
	174	B6	-	P	-	Parity bits
W 9	180	I3	2^{-43}	$\dot{\Omega}$ (LSBs)	$\sqrt{m}$	Rate of right ascension
	204	B6	-	P	-	Parity bits
Word 10	210	U1	-	IODE	-	Issue of data (ephemeris)
	218	B'14	2^{-43}	IDOT	-	Rate of inclination angle
	232	B2	-	t	-	Noninformation bearing bits
	234	B6	-	P	-	Parity bits

Appendix B

Front-end Applications: User Interface



Basestation
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 1
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 2
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 4,58 cm/y
Warnsufe gelb

Node 3
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 5,48 cm/y
Warnsufe rot

Node 4
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 5
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 7,14 cm/y
Warnsufe rot

Node 6
Ladestand: 0%
2 von 7 Sensoren OK
Bewegung: 2,45 cm/y
Warnsufe grün

Node 7

Stand: 2013-01-21 16:22:50

Figure B.1: Android app detail view.

Node 2

Ladestand: 0%
2 von 7 Sensoren OK

Bewegung: 4,58 cm/y
Beschleunigung X: 1.5 g
Beschleunigung Y: 2.3 g
Beschleunigung Z: 1.8 g

Lufttemperatur: 25.69 °C
Lichtstärke: 99 lux
Luftfeuchtigkeit: nA %rF

Stand: 2013-01-21 16:22:50

Figure B.2: Android app node detail view.

Figure B.3: Android app map view.

Figure B.4: Web application map view.

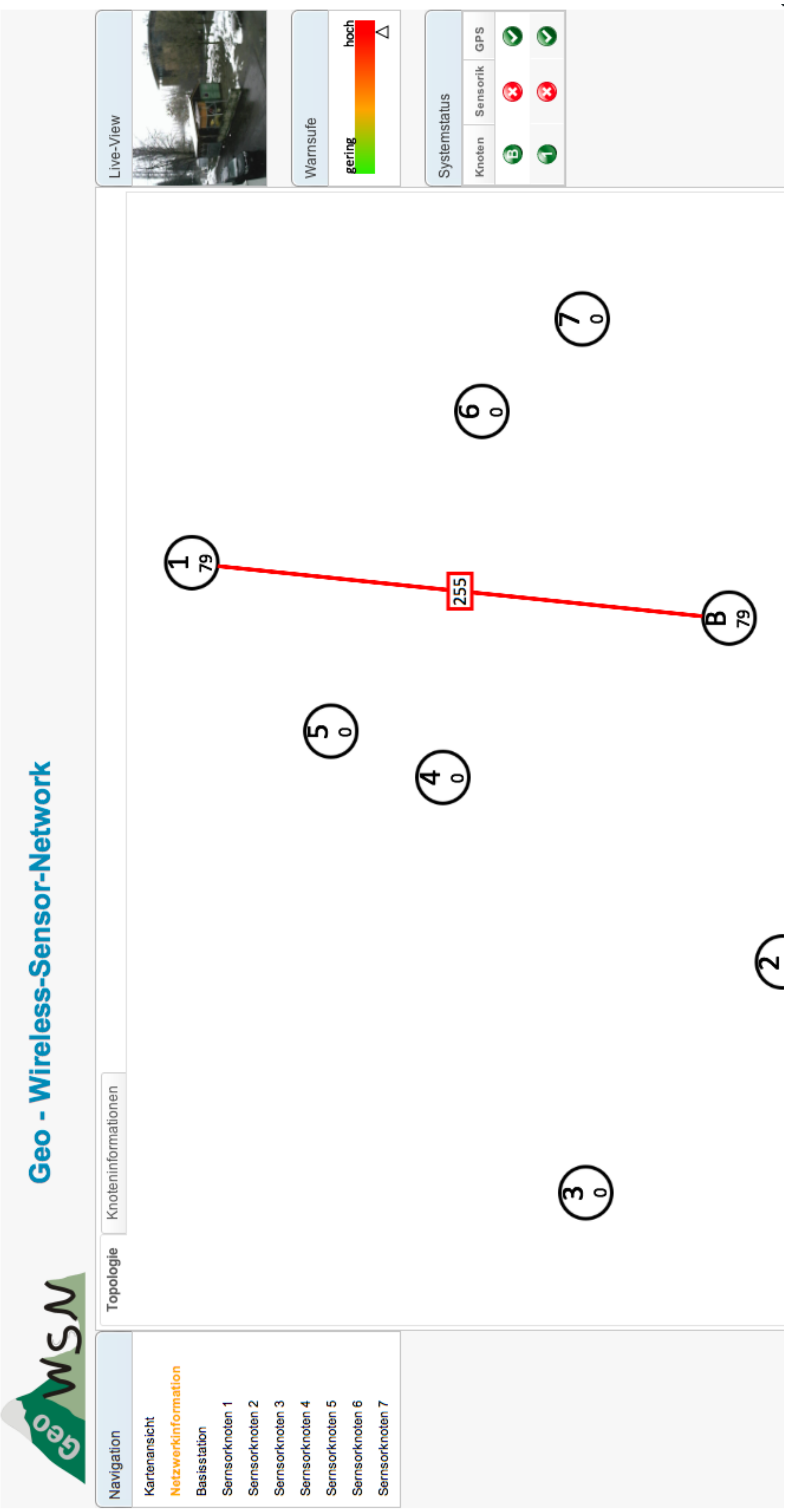


Figure B.5: Web application network view.

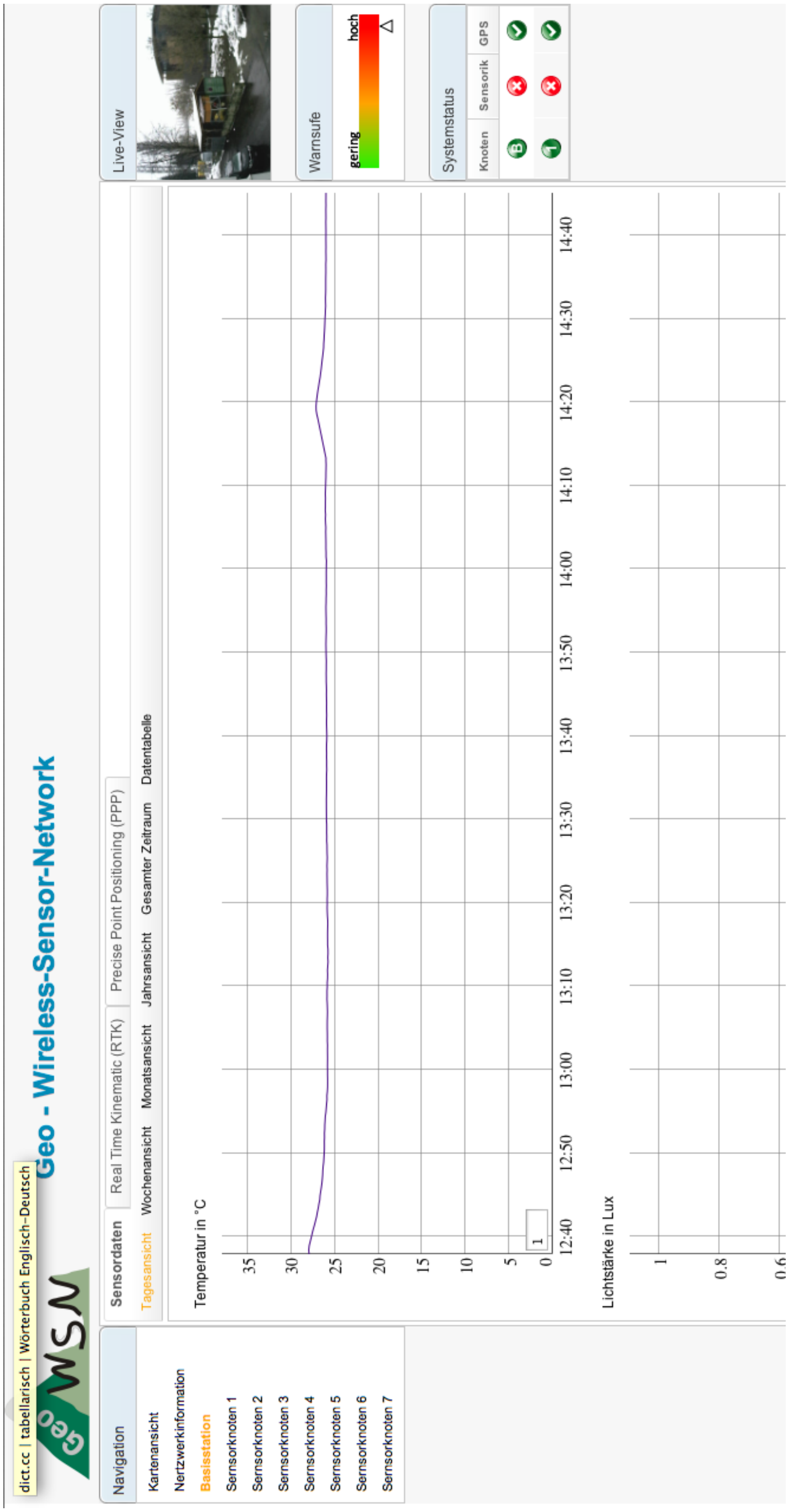


Figure B.6: Web application detail view.

Bibliography

[1] W. Klaus. Literaturstudie Murenwarnsysteme. Report 17, Institut für Alpine Naturgefahren, Universität für Bodenkultur Wien, 1999.

[2] K. Thuro, T. Wunderlich, and O. Heunecke. Development and testing of an integrative 3D early warning system for alpine instable slopes (alpEWAS). *Geotechnologien Science Report, S*, pages 101–112, 2007.

[3] R. Azzam, T.M. Fernandez-Steeger, C. Arnhardt, H. Klapperich, and KH Shou. Monitoring of Landslides and Infrastructures with Wireless Sensor Networks in an Earthquake Environment. In *5th International Conference on Earthquake Geotechnical Engineering*, pages 10–13, 2001.

[4] B. Buchli, F. Sutton, and J. Beutel. GPS-equipped wireless sensor network node for high-accuracy positioning applications. *Wireless Sensor Networks*, pages 179–195, 2012.

[5] M. Dietl. Entwurf und Implementierung der Hardware-Architektur eines drahtlosen Sensorknotens zur Überwachung gefährdeter Hänge mittels GPS. to be published, Institute of Technical Informatics, Graz University of Technology, 2013.

[6] P. Salin. *Mobile instant messaging systems-a comparative study and implementation.* PhD thesis, Helsinki University of Technology, 2004.

[7] P. Nei. An open standard for instant messaging: eXtensible Messaging and Presence Protocol (XMPP). *Found in: www. tml. tkk. fi/Publications/C/21/nie_ready. pdf*, 2006.

[8] Open Mobile Alliance. Presence SIMPLE architecture. *Approved Version*, 1(1):21, 2008.

[9] F. Schramm. Metis Mobile Cloud (MMC). Internal Progress Report TU Graz, 2013.

[10] Henri Chen and Robbie Cheng. *ZK: Ajax without the Javascript Framework.* Apress, 2007.

[11] Reto Meier. *Professional Android 4 application development.* Wrox, 2012.

[12] Oracle Corporation. MySQL 5.1 reference manual. *Sun Microsystems*, 2007.

[13] C. Bauer and G. King. Hibernate in action. 2005.

[14] C. Arnhardt, K. Asch, R. Azzam, R. Bill, TM Fernandez-Steeger, SD Homfeld, A. Kallash, F. Niemeyer, H. Ritter, M. Toloczyki, et al. Sensor based Landslide Early Warning System-SLEWS. Development of a geoservice infrastructure as basis for early warning systems for landslides by integration of real-time sensors. *Geotechnologien science report*, 10:75–88, 2007.

[15] P. Saint-Andre, K. Smith, and R. Tronçon. *XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies.* O'Reilly Media, Incorporated, 2009.

[16] L.B. Hörmann, S. Gether, Mock H., M. Dietl, and Ch. Steger. GeoWSN Implementierung und Anwendungsrichtlinien. Technical Report 1, Institute for Technical Informatics, Graz University of Technology, 2013.

[17] M. Lytvyn, Ch. Pöllabauer, M. Troger, K. Landfahrer, L. Hörmann, and Ch. Steger. Real-Time landslide monitoring using single-frequency PPP: Proof of concept. In *Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing,(NAVITEC), 2012 6th ESA Workshop on*, pages 1–6. IEEE, 2012.

[18] J.K. Hart and K. Martinez. Environmental Sensor Networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3):177–191, 2006.

[19] Rudolf Schmidt. *Frühwarnung in Wildbacheinzugsgebieten.* Rudolf Schmidt und Forsttechnischers Dienst für Wildbach- und Lawinenverbauung (WVL) Voralberg, 2002.

[20] Th. Arampatzis, J. Lygeros, and S. Manesis. A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 719 –724, june 2005.

[21] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *Pervasive Computing, IEEE*, 1(1):59 –69, jan.-march 2002.

[22] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 4, pages 2033 –2036 vol.4, 2001.

[23] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6–28, 2004.

[24] DA Robinson, Scott B Jones, JM Wraith, D Or, and SP Friedman. A review of advances in dielectric and electrical conductivity measurement in soils using time domain reflectometry. *Vadose Zone Journal*, 2(4):444–475, 2003.

[25] W. Stempfhuber and T. Wunderlich. Auf dem Weg zur Sensorsynchronisation von GPS und TPS für kinematische Messaufgaben. *Allgemeine Vermessungs-Nachrichten (AVN)*, 5:2004, 2004.

[26] T. Wunderlich, T. Schäfer, K. Zilch, E. Penka, and V. Cetl. Schadenserkennung an einer Spannbetonbrücke durch reflektorlose Deformationsmessungen. *Festschrift zum*, 60, 2005.

[27] G.R. Hu, H.S. Khoo, P.C. Goh, and C.L. Law. Development and assessment of GPS virtual reference stations for RTK positioning. *Journal of Geodesy*, 77:292–302, 2003.

[28] K. Thuro, J. Singer, J. Festl, T. Wunderlich, P. Wasmeier, C. Reith, O. Heunecke, J. Glabsch, and S. Schuhbäck. New landslide monitoring techniques–developments and experiences of the alpEWAS project. *Journal of Applied Geodesy*, 4(2):69–90, 2010.

[29] J. Günther, O. Heunecke, S. Pink, and S. Schuhbäck. Developments towards a low-cost GNSS based sensor network for the monitoring of landslides. In *13th FIG Symposium on Deformation Measurement and Analysis, LNEC, Lisbon, Portugal*, pages 12–15, 2008.

[30] K. Thuro, J. Singer, and J. Festl. Ein geosensornetz-basiertes Monitoring-und Frühwarnsystem für Rutschungen. 2011.

[31] K. Sohraby, D. Minoli, and T. Znati. *Wireless sensor networks: technology, protocols, and applications*. Wiley-Interscience, 2007.

[32] TM Fernandez-Steeger, C. Arnhardt, K. Walter, S. Haß, F. Niemeyer, B. Nakaten, SD Homfeld, K. Asch, R. Azzam, R. Bill, et al. SLEWS-A Prototype System for Flexible Real Time Monitoring of Landslides Using an Open Spatial Data Infrastructure and Wireless Sensor Networks. *Geotechnologien science report*, 13:3–15, 2009.

[33] MEMS (Micro-Electro-Mechanical Systems). In Bharat Bhushan, editor, *Encyclopedia of Nanotechnology*, pages 1305–1305. Springer Netherlands, 2012.

[34] F. Kobayashi, Y. Tanabe, T. Fukuda, and F. Kojima. Acquisition of sensor fusion rule based on environmental condition in sensor fusion system. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, volume 4, pages 2096 –2101 vol.4, july 2001.

[35] Li Li and Fan Bai. Analysis of data fusion in wireless sensor networks. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 2547 –2549, sept. 2011.

[36] V. Wirz, P. Limpach, J. Beutel, B. Buchli, and S. Gruber. Temporal characteristics of different cryosphere-related slope movements in high mountains. In *Proc. 2nd World Landslide Forum. Springer, Heidelberg*, 2011.

[37] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al. TinyOS: An operating system for sensor networks. *Ambient intelligence*, 35, 2005.

[38] N. Burri, P. Von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 450–459. IEEE, 2007.

[39] J. Beutel, S. Gruber, A. Hasler, R. Lim, A. Meier, C. Plessl, I. Talzi, L. Thiele, C. Tschudin, M. Woehrle, et al. PermaDAQ: A scientific instrument for precision sensing and data recovery in environmental extremes. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 265–276. IEEE, 2009.

[40] L. Manetti, M. Frapolli, and A. Knecht. Permanent, autonomous monitoring of landslide movements with GPS. In *1st European Conference on Landslides. Prague, Czech Republic*, 2002.

[41] L. Manetti and A. Knecht. GPS-based system for autonomous and permanent monitoring of large structures. In *First International Conference on Bridge Maintenance, Safety and Management. Barcelona, Spain*, 2002.

[42] XL Ding, WJ Dai, WT Yang, XW Zhou, J. Lam, Q. Zhang, and L. Wang. Application of multi-antenna GPS technology in monitoring stability of slopes. *Proceedings of Strategic Integration of Surveying Services, FIG Working Week*, 2007.

[43] B. Churcher. Differential global positioning system. 2011.

[44] S.J. Vaughan-Nichols. Presence technology: more than just instant messaging. *Computer*, 36(10):11–13, 2003.

[45] J. Rosenberg and M. Day. A model for Presence and Instant Messaging. 2000.

[46] M. Day, S. Aggarwal, G. Mohr, and J. Vincent. Instant messaging/presence protocol requirements. *Request for Comments*, 2779, 2000.

[47] Peter Saint-Andre. Extensible messaging and presence protocol (XMPP): Core. 2011.

[48] Peter Saint-Andre. Extensible messaging and presence protocol (XMPP): Core. page 19, 2011.

[49] C. Kaes. Definition of Jabber Identifiers (JIDs). 2003.

[50] J. Moffitt. *Professional XMPP programming with JavaScript and jQuery.* Wrox, 2010.

[51] S. Chatterjee, T. Abhichandani, H. Li, B. TuIu, and J. Byun. Instant messaging and presence technologies for college campuses. *Network, IEEE*, 19(3):4–13, 2005.

[52] P. Saint-Andre. Streaming XML with Jabber/XMPP. *Internet Computing, IEEE*, 9(5):82–89, 2005.

[53] D. Meyer and P. Saint-Andre. XMPP Transport Layer Security. 2009.

[54] A. Melnikov and K. Zeilenga. Simple Authentication and Security Layer (SASL). Technical report, RFC 4422, June, 2006.

[55] G. Camarillo, M. Handley, J. Peterson, J. Rosenberg, A. Johnston, H. Schulzrinne, and R. Sparks. SIP: session initiation protocol. 2002.

[56] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. Session initiation protocol (SIP) extension for instant messaging. 2002.

[57] B. Campbell, R. Mahy, and C. Jennings. The message session relay protocol (MSRP). *Internet Engineering Task Force (IETF), Setembro*, 2007.

[58] N. Freed and N. Borenstein. Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies, 1996.

[59] E. Burger. A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages. 2006.

[60] Google Inc. Android http://www.android.com, February 14(th) 2013.

[61] Eclipse Foundation. Eclipse. http://www.eclipse.org, February 14(th), 2013.

[62] David Gallardo, Ed Burnette, and Robert McGovern. *Eclipse in action: a guide for java developers.* Manning Publications, 2003.

[63] Potix Corporation. Zk. http://www.zkoss.org, February 14(th), 2013.

[64] Markus Stäuble and Hans-Jürgen Schumacher. *ZK Developer's Guide.* Packt Pub Limited, 2007.

[65] Google Inc. Android SDK. http://developer.android.com/sdk/index.html, February 14(th) 2013.

[66] James Steele and Nelson To. *The Android Developer's Cookbook: Building Applications with the Android SDK.* Addison-Wesley Professional, 2010.

[67] Texas Instruments Incorporated. Code Composer Studio. http://www.ti.com/tool/ccstudio, February 14(th) 2013.

[68] Texas Instruments Incorporated. *TI SYS/BIOS v6.34 Real-time Operating System.* Texas Instruments Incorporated, September 2012.

[69] Oracle Corporation. MySQL. http://www.mysql.de, February 14(th), 2013.

[70] A. Vukotic and J. Goodwill. *Apache Tomcat 7.* Apress, 2011.

[71] M. Keith and M. Schnicariol. Object-relational mapping. *Pro JPA 2*, pages 69–106, 2010.

[72] R. Filjar, T. Kos, and I. Markezic. GPS ionospheric error correction models. In *Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006 focused on*, pages 215–217. IEEE, 2006.