university of groningen

FACULTY OF MATHEMATICS AND NATURAL SCIENCES
DEPARTMENT OF COMPUTING SCIENCE

SOFTWARE ARCHITECTURE - TEAM 2

Smart Flood Monitoring System

Architecture Description

Authors:
Thomas HOEKSEMA
Maarten KOLLENSTART
Toon ALBERS
Marijn VONK
Ynte TIJSMA

Coaches:
Antonis GKORTZIS
Sofia CHARALAMPIDOU

Lecturer:
prof. dr. ir. Paris AVGERIOU

Version 0.7
October 19, 2015

Revision History

The following students have contributed to the elaboration of this document:

ID	Author	Student no.	Email address
TH	Thomas Hoeksema	S2349639	T.M.Hoeksema@student.rug.nl
MK	Maarten Kollenstart	S2209053	M.Kollenstart@student.rug.nl
TA	Toon Albers	S2768372	T.Albers.2@student.rug.nl
MV	Marijn Vonk	S2066211	M.R.P.Vonk@student.rug.nl
YT	Ynte Tijsma	S2253216	Y.W.Tijsma@student.rug.nl

The table below lists all versions and specific revisions that have been performed on this document. All versions and revisions have been reviewed by at least one of the coaches.

Ver.	ID	Date	Revision
0.1	TH	31/08/15	Initial template of document.
		01/09/15	Addition of Revision History and A3 (Time Tracking Table).
		06/09/15	Refining draft versions of sections 1-2.
	MV	06/09/15	First draft for section 1-2.
0.2	TH	07/09/15	Modifying layout of A2 (Time Tracking Table).
		12/09/15	Combined sections (1: Introduction, 2: System Context) into section 1 (Introduction) for readability.
		13/09/15	Drafts for sections 2 and 3, added sections 1.3, added introductory texts, improved 3.2, 3.3, 3.4.
	MK	13/09/15	Draft versions of sections 3.2, 3.3, 3.4, 3.5.
	MV	10/09/15	Refining sections 1-2.
		13/09/15	Refining section 2.
	YT	13/09/15	Draft for section 4.
0.2.1	TH	15/09/15	Refining sections 1, 2, 3.2.
		16/09/15	Refining sections 3.2 to 3.4.
		17/09/15	Reworked and refined some sections in 2, 3.1-3.9.
	TA	15/09/15	Draft of 3.5 and 3.6.
		16/09/15	Draft of 3.5, 3.11 and 4.1.
	MK	15/09/15	Draft of use cases.
	MV	15/09/15	Working on section 2.4, 2.6 and 2.9.
		16/09/15	Working on section 2.3, 2.6, 2.7 and 2.9.
		17/09/15	Draft for section 3.1.
	YT	17/09/15	Working on section 4.1-4.3.
0.3	TH	19/09/15	Refining of introductory pages and sections 1-2, minor refining in section 3.
		20/09/15	Working on sections 2-3, refining document structure and layout (of table/figure templates), minor refining in sections 1, 4, and introductory pages.
	MK	19/09/15	Processing feedback.
		20/09/15	Working on sections 2.6, 2.7, 3.5-3.7, 3.10, 4.1 and 4.2.
	TA	19/09/15	Refining 3.11.

		20/09/15	Refining sections 3, 4 and 5.
	MV	19/09/15	Applying feedback of version 0.2.1.
		20/09/15	Editing diagram of 3.1, refining section 2 and 4.
	YT	20/09/15	Working on section 4.
0.4	TH	26/09/15	Modifications in sections 1-4 according to feedback.
		27/09/15	Initial version of section 5.1.
	MK	27/09/15	Updated section 2.8, started on section 5.1.
	TA	26/09/15	Rework section 4.3.
		27/09/15	Working on section 6.1 and A1, writing sections 6.2 and 6.3.
	MV	21/09/15	Updating 3.1 and UML diagrams for presentation.
		26/00/15	Start working on section 6.
		27/09/15	Proceed working on section 6.1.
	YT	26/09/15	Applying weekly feedback and feedback of group 1.
		27/09/15	Working on section 6.1.
0.5	TH	30/09/15	Refining sections 2.8, 3.3-3.6.
		02/10/15	Refining sections 3.5-3.9, 4.3, 5.1.
		03/10/15	Minor refining in sections 2-3, refining sections 5.1-5.2.
		04/10/15	Refining sections 5.1-5.3, templates for section 8.
	MK	01/10/15	Working on section 5.
		04/10/15	Working on section 5 and 7.5, refining section 2.8.
	TA	02/10/15	Start working on section 7.
		04/10/15	Working on sections 7.2-7.5.
	MV	01/10/15	Preparatory reading for and starting on section 7.
		02/10/15	Working on section 7.2.
		03/10/15	Working on section 7.6.
		04/10/15	Working on section 7.2 and 7.4.
	YT	02/10/15	Start working on section 7.
		03/10/15	Working on section 7.2.
		04/04/15	Working on section 7.2 and 7.4.
0.6	TH	09/10/15	Modifications to sections 3 and 5, layout of section 8.
		10/10/15	Working on section 8.1.
		11/10/15	Working on sections 8.1 and 8.2, layout of section 9.
	MK	09/10/15	Started working on section 8.
		10/10/15	Refining section 5.
		11/10/15	Working on section 8.
	TA	05/10/15	Working on diagrams of sections 7.2-7.5.
		09/10/15	Working on sections 6.2, 7-7.5.
		11/10/15	Add legend to Figure 10, working on sections 7.2-7.6.
	MV	05/10/15	Applying feedback of version 0.5, updating activity and process diagrams.
		09/10/15	Working on section 7.
		11/10/15	Working on section 7, reviewing sections 6-7, creating activity and sequence diagrams.
	YT	11/10/15	Working on section 6.4, refining section 7.
0.7	TH	12/10/15	Minor layout refinements throughout document.
		15/10/15	Modifying templates and adding cross referencing.

		16/10/15	Adding cross referencing for all traceable items in document, fixing spacing and formatting issues.
		17/10/15	Refining decisions, creating decision views, refining all requirements.
		18/10/15	Working on decision views, refining section 3, 8 and 9.
	MK	17/10/15	Working on section 8
		18/10/15	Working on chronological decision view and relational decision view
		18/10/15	Working on section 8
	TA	16/10/15	Working on section 7.2.1, diagrams in section 7.
		17/10/15	Working on section 7.2.
		18/10/15	Working on sections 7.2, 8.1 and relation decision view.
	MV	16/10/15	Incorporating feedback on decisions and diagrams.
		17/10/15	Updating section 3.1 and 7.6, refining section 7.
		18/10/15	Working on sections 7.2, 7.4 and 7.6, and refining section 7.
	YT	17/10/15	Applied feedback and refined section 6.
		18/10/15	Refined section 3.10, started section 7.7 and 9.
0.8	TH	00/00/15	
	MK	00/00/15	
	TA	00/00/15	
	MV	00/00/15	
	YT	00/00/15	
0.9	TH	00/00/15	
	MK	00/00/15	
	TA	00/00/15	
	MV	00/00/15	
	YT	00/00/15	
1.0	TH	00/00/15	
	MK	00/00/15	
	TA	00/00/15	
	MV	00/00/15	
	YT	00/00/15	

Contents

List of Figures

List of Tables

Glossary

ATAM - Acronym for Architecture Trade-off Analysis Method, a risk mitigation process used early in the software development life cycle, in particular during the inception and elaboration phases.

SDI-12 - Acronym for Serial/Digital Interface at 1200 baud, a wireless communication protocol.

SFM - Acronym for Smart Flood Monitoring, the product for which an architecture is described in this document, and also refers to the technique of using smart systems in order to monitor and generate warnings in areas where floods often occur.

Smart - The high-level property of a system to adapt its behaviour according to inputs of users or other systems in specific situations, and to learn and improve itself based on that behaviour and those situations, and to evaluate the consequences of its own behaviour.

1 Introduction

We are a software architecture team in a company that specializes in designing and developing smart systems for environmental monitoring. The company produces innovative products heavily based around ICT and provides services that aim to monitor natural environments. The current activities of the company revolve around a Smart Flood Monitoring system, which tries to limit the social and financial consequences of floods and, most importantly, to prevent the loss of human lives.

The goal of this project is to offer a system that will allow early and correct detection of floods. This initial section of the document describes the position of the envisaged system in its environment. First, it delivers the context in which the system-to-be (i.e. a system providing flood monitoring) is relevant. The limitations of the scope of the system will be described. The section then proceeds with an introduction to the company and it outlines the scope and core goals of the system, and a short statement on the company's core competences and approach to building the system. Finally, we outline the structure of this document.

1.1 System Context

Nowadays, floods are threatening many parts of Europe and the problem is aggravated every year through climate change and extreme weather phenomena. Floods can destroy farmlands and crops and cause massive problems in residential areas, even at the loss of human lives. There is an important question that needs to be answered: how can one gain insight into what areas are threatened by a flood? The answer to this question is the realization of an adequately designed system that can track the right data at the right moment, and that allows for the identification of an upcoming flood in a timely and automatic manner. With the emergence of new sensor technologies and novel aerial and terrestrial vehicles, the firm plans to produce a new product line of systems that provides flood monitoring in natural environments with a high occurrence of floods.

The scope of the system is limited to providing flood monitoring for events that occur in rural and urban areas (through rivers, canals, lakes, natural parks, etc) and events that happen near sea and other larger bodies of water (through damage to dikes or dams). For this system, we consider only rural and urban areas such as the ones that may be found in Europe, where there is sufficient to good access to the electricity network (or the possibility of extending the electricity infrastructure to make it so), and where it is possible to broadcast signals over various available networks.

1.2 Smart Flood Monitoring

The company has assigned the task of designing a new product, the Smart Flood Monitoring (SFM) system, that will provide flood monitoring in rural and urban areas close to bodies of water, such as rivers, canals, lakes, seaside areas, natural parks, structures such as dikes and dams, or special areas of interest that may suffer significant damage from flooding, such as farms. This product will serve as a pilot product for the firm for the next generation of environmental monitoring systems, paving the way for systems with more complicated sensors and vehicles and integration with emergency services.

The system-to-be monitors activities and properties of rivers, waterways and dikes, such as the water level and pressure or the consistency of the dike. In case of a prominent predicted emergency related to floods, the system warns relevant authorities and emergency services, but also directly alerts citizens who are in the relevant areas where the flood is occurring or who are subscribed for such messages. Subsequently, it provides guidance to its constituents and third parties.

With respect to product development, the company does not manufacture any hardware components itself. Rather, it reuses off-the-shelf hardware such as sensors and vehicle control units to build environmental monitoring systems. On the other hand, the firm focuses on the software system that controls the hardware in order to implement the flood monitoring functions.

1.3 Overview

The structure of the rest of this document will now be described.

- Section 2 (*Business Information*) describes the system in the context of its business environment, which includes the business vision and rationale, and a description of the product, its audience, and its key features.
- Section 3 (*Requirements*) identifies the stakeholders of the system, its key drivers, describes a set of requirements and use cases that the architecture should fulfill, and states the risk assessment for this architecture and the system.
- Section 4 (*Analysis*) contains the analysis of the system, consisting of the assumptions, technology roadmaps, and the design decisions and alternatives for the architecture.
- Section 5 (*System Architecture*) outlines the high level, reverse engineered description of the architecture of the hardware platform.
- Section 6 (*Hardware Architecture*) describes the architecture on the level of hardware.
- Section 7 (*Software Architecture*) describes the architecture of the software that is involved with the system.
- Section 8 (*Architecture Evaluation*) focuses on evaluating the architecture that has been designed and verifies if the requirements of the system have been met.
- Section 9 (*System Evolution*) will entail possible points of improvements to the system and defines how the system should evolve through its life cycle.
- Section 10 (*References*) and *Appendices* conclude the report with a list of all knowledge bases and resources that have been utilized during the elaboration of this assignment and acknowledges the efforts of external individuals and/or groups.

2 Business Information

This section elaborates on the business environment of the Smart Flood Monitoring (SFM) system. First, the business vision gives a description of the envisaged business opportunities and unique selling points. Advantages and disadvantages with respect to the current situation through a SWOT-analysis will be given. Subsequently, the main product functions and their evolution over time are discussed. These are followed by an overview of the target audience and business model, after which a roadmap about market development and product introduction is presented. A financial model is provided that covers among other things the unit price, turnover, required investments and operational costs, and finally, this section finishes with an identification of potential competitors.

2.1 Business vision

The Smart Flood Monitoring system will serve as a pioneering product that will revolutionize the market of environmental monitoring. In order to be successful, the SFM system is envisioned to accomplish the following three goals:

- **Monitoring activities** - Performing monitoring of activities and properties that are relevant for detecting premature floods through various types of sensors, such as water height level or water pressure, in and around areas with large bodies of water.
- **Warning activities** - Warn relevant constituents and parties in case of an emergency, such as governmental authorities, emergency services and simply the residents and companies in the affected areas.
- **Guidance activities:** Provide appropriate guidance to constituents or third parties on- and off-scene, for example guiding emergency services or residents through a flooded area, or showing information on the emergency events that are occurring and have occurred.

	Helpful (to achieve the objective)	Harmful (to achieve the objective)
Internal origin (product/company attributes)	S: High maintenance & services revenues. Experience with Smart Systems for Environmental Monitoring.	W: No prior experience with SFM systems. Hardware not manufactured in-house; high reliability on external suppliers.
External origin (environment/market attributes)	O: Increasing demand due to climate change & extreme weather phenomena.	T: Extremely competitive market. Interface incompatibilities.

Table 3: SWOT-analysis

The extent to which the Smart Flood Monitoring system is able to excel in these primary goals and will be successful is partly dependent on internal factors such as strengths and weaknesses. Moreover, external factors - i.e. opportunities and threats - play a significant role in determining the success of the system. For determining this business related information we utilize a so-called SWOT analysis (see [1] and [2]), which is a structural planning tool and methodology for the identification of the Strengths, Weaknesses, Opportunities and Threats of a product from a business perspective. The results of this table are depicted in Table 3, where the four categories of items are listed.

The unique selling points of the SFM can be described in the following way:

- **Highly scalable and future-proof:** The system's innovative and loose coupling of components allows the integration of new functions in the domain of environmental monitoring. Another explicit example of scalability includes the routing capabilities of each node (including sensor components). The data can reach its destination through multiple routing sequences of nodes, thus a node failure/removal or an expansion merely results into a decreased or increased variation of possible routing schemes, respectively.

- **Accuracy:** Collection of monitoring data from an abundant variation of sources to guarantee correctness, precision and reliability of the flood forecasts. It also safeguards availability of the system whenever a source (connection) fails.

- **High performance:** This guarantees early and correct flood warnings and guidance, despite of constituent systems of limited memory, battery life and processing power that exchange large amount of data across unstable environments.

- **Resilience:** The SFM system is highly available due to the high fault tolerance of the system. The system implements redundancy to make sure that every two nodes in the sensor network can reach each other in case some of the network nodes go offline. Aside from redundancy, the system can also detect automatically if a sensor or sub-network has become unreachable or has broken down, and marks these for maintenance.

Among these unique selling points, the system will also achieve:

- **Autonomous failure detection and management:** Automated detection of deviations in measurements by the sensors (a.o. through machine learning). This guarantees that divergences from the expected behavior are identified, and determines whether these exist either due to malfunctioning measurement units or unanticipated changes in the flood defence structures.

2.2 Business rationale

The mission statement of the company is to limit the social and financial consequences of floods and most importantly avoid the loss of human lives. To achieve this the company wants to have a pioneering role in the environmental monitoring market - and the market for flood monitoring in particular - by offering early and correct detection of floods.

2.3 Product description

Through collecting data on precipitation, groundwater and upstream water levels, and other types of properties that have to do with flood monitoring, governments and industries and utilities can get early warnings about upcoming floods and take appropriate action. Due to extreme competition in this particular market, many of such systems have already been proposed and partially implemented. Some of these systems have been described in Section 2.9 further on in this document. However, most of these systems fail to adapt to new software environments and changing user needs, which led to a limited public adoption of these systems. For example, some of these obsolete systems did not manage to provide real-time monitoring info. The risk of a flood could often not precisely be estimated, due to the lack of an accurate, correct, precise and timely supply of (sensor) data.

The Smart Flood Monitoring system proposed in this document makes use of state-of-the-art technologies that manages to overcome the inherent weaknesses of these traditional flood monitoring systems. The SFM system is first of all responsible for collecting relevant data, required for making accurate and reliable forecasts of potential floods. A substantial part of the data is obtained from numerous *Serial/Digital Interface at 1200 baud* (SDI-12) sensors, measuring for instance the (ground)water level, water flow and precipitation level. This data is supplemented with readily available data from external sources, such as weathers stations and meteorological experts. The Smart Flood Monitoring system subsequently consolidates and processes all this data in order to produce reliable estimations of future flooding scenarios. The consolidated data is stored in a database that can be accessed by the monitoring station or the web server. Numerous recipients of potential warnings and guidance from the monitoring station could be identified, the most important being governments, residents and firms. These recipients may also have access to the web server that provides a geographical overview with a real-time estimation of flood risk per area.

See Section 2.6 for a depiction of the business and domain model, which name the main entities that are part of, *or* relevant to the system.

2.4 Target audience

The product is targeted at authorities like governments and municipalities, emergency services and other organizations in the Netherlands and abroad that have either historically suffered floods or expect floods in the future (due to a rising sea level). The system should also directly issue warnings to citizens who are subscribed for such messages (e.g. through SMS or mobile apps). These target audiences require early warnings and guidance so that they may take appropriate actions.

2.5 System evolution

While the system becomes more broadly adopted it needs to keep pace with evolving technologies. Sensors are highly important elements of the SFM system, hence further studies on self-sufficient energy and wireless sensor technology need to be considered for replacing the current sensors. Thanks to an abundance of components and the scalability of the system alterations through sensor component replacements require little effort. Solar cells could be deployed as an energy source for the components to replace dependency on the electricity grid with energy self-sufficiency. Solar energy could also serve as a cheaper energy source, whenever system components are located in remote areas. Furthermore, alternative technologies such as drones could be adopted as complementary data sources for detecting floods.

2.6 Business and Domain Model

Figure 1 shows the business model, that presents the main entities which are most relevant to the system. An edge between two entities indicates the existence of a direct relation between these entities.

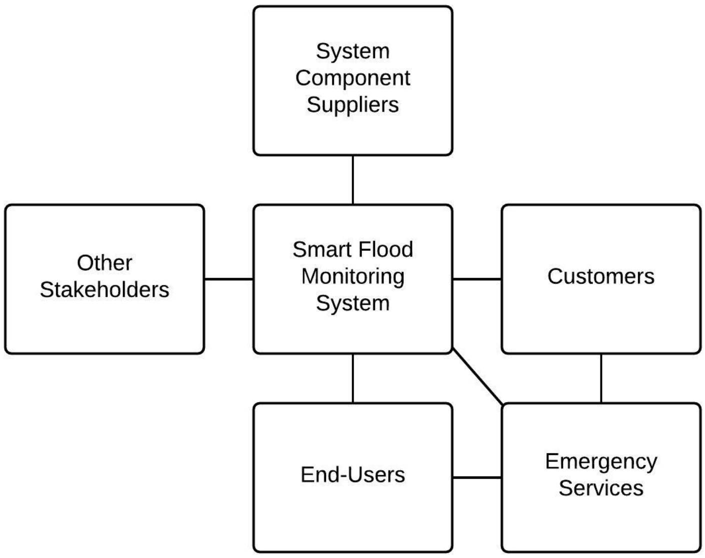


Figure 1: Business Model

Suppliers include all the parties which deliver the components that build up the system. The customers are the buyers of the system - most likely governmental bodies. The end-users are those stakeholders that do not necessarily pay for the system, but benefit strongly from its intended services (e.g. residents). Emergency services ensure public safety and health by addressing different emergencies. Some examples are fire departments, emergency medical services and the police. There are many other stakeholders which have an interest in the system. These are described in Section 3.2.

The domain model is depicted in Figure 2. It involves the most important components that are connected to the Smart Flood Monitoring system. Names of deployed interfaces, networks or protocols are provided on top of the connection lines.

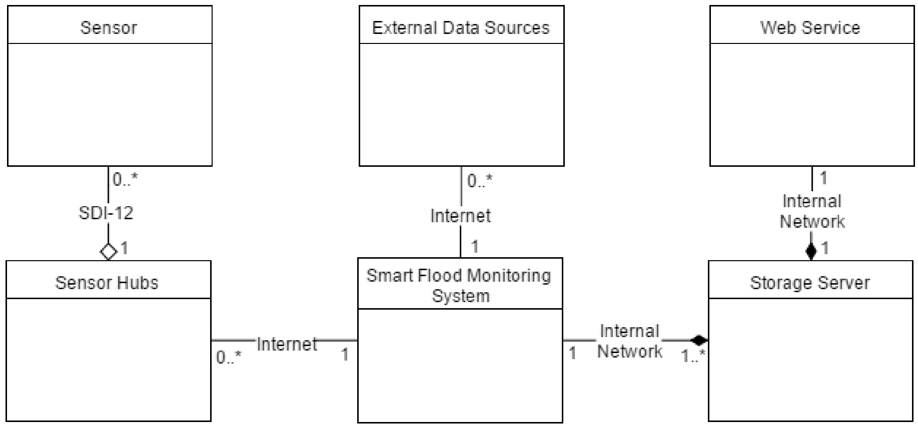


Figure 2: Domain Model

The central part of the domain model is the Smart Flood Monitoring System, which is part of the Storage Server. The SFM system is responsible for gathering all the data and applying the flood detection model, the resources the SFM system has at its disposal are a large number of Sensor Hubs and several External Data Sources. The Sensor Hubs communicate directly with the sensors via the SDI-12 protocol and sends the sensor data when the SFM system requests data of a specific sensor. External data sources, such as meteorological forecasts or other relevant data sources, are also gathered by the SFM system via a polling mechanism. All the communication done by the SFM system to gather data is done via the Internet. The Storage Server saves all the sensor data gathered by the SFM system, and also saves its flood predictions. Due to the unique selling point of the system, these Storage Servers are deployed redundantly to ensure the network resilience of the system. Each Storage Server is also equipped with a Web Service that allows interaction with the system via browser. It should be noted here that the Web Service refers to a collection of application instances that make up the service that users can access through the web portal of the system.

Sensor data includes measurement data on e.g. (ground)water level, water flow velocity and precipitation level. When the SFM system applies the flood detection model and the risk of a flood is too high the system will alert the users that there is a flood threat and will guide users to safer areas, these alerts will both be broadcast messages to groups of users and unicast messages to users individually.

2.7 Roadmap

Partnerships should be established as soon as possible such that the product can be discovered by a larger public. European countries will be targeted before moving to more distant areas. Central European countries including Austria, Czech Republic, Germany, Hungary, Poland, Slovakia and Switzerland will be targeted initially, since these countries encounter a relatively high amount of casualties and material damages [5]. In these countries we have to decide with the partners where and how many sensors should be placed. Subsequently, non-European target countries can include India, Bangladesh, China, Vietnam, Pakistan, Indonesia, Egypt, Brazil and Thailand, which all belong to the top 15 countries with most flood casualties [6]. Table 4 provides an overview of the anticipated periods for entering these (non-)European markets.

At the same time when partnerships are being established and locations to deploy the system are determined, the flood detection model could be developed and tests should be performed to check the desired infrastructure.

Period	Location	Milestone
2016 Q2	Europe	Establish deals with customers to identify high priority areas
2017 Q1	Europe	Launch SFM system
2018 Q3	Asia, Egypt, Brazil	Establish deals with customers
2019 Q1	Asia, Egypt, Brazil	Launch SFM system for customers

Table 4: Roadmap overview

2.8 Financial model

This section will discuss the financial model. All the different components of building and maintaining the system will be explained, including the software system, hardware components and labor costs.

2.8.1 Software architecture costs

The software architecture will be written by 5 team members working 12 hours a week for 10 weeks. This means that the total man hours of the Software Architecture is 600 hours. The result of this will be the Software Architecture Document. Each team member is valued at €100 per hour. This makes the total cost of the Software Architecture:

$$5 \cdot 12 \cdot 10 \cdot €100 = €60,000$$

2.8.2 Development costs

The easiest way to illustrate the costs of the development of the SFM system is with a table listing the main parts of the software system. Table 5 is such a table, the main part of the software is obviously the flood detection model that calculates the likelihood of a flood. An other large element of the software system is the server configuration, because the system must be available at all moments a good server configuration leads to a scalable and high available system.

The software developers should be experienced programmers for the harder parts of the software: the flood detection model and the server configuration. Therefore, the average hourly rate of the software developers is rated at €75. According to table 5 the total development hours will be 5920, so the total costs of the development is:

$$5920 \cdot €75 = \mathbf{€444,000}$$

Description	**Man-hours**
Sensor Registration	160
Sensor Data collector	320
External Data collector	160
Persistent Data storage	320
Flood Detection Model	1600
Alarm techniques	800
Web Service	800
Server Configuration	1200
Buffer	960
Total:	**5920**

Table 5: Overview of development costs in man-hours.

2.8.3 Hardware costs

The total hardware costs of the complete system are hard to determine in advance due to scalability factor of the system. We decomposed our hardware system into two main parts: a sensor hub with 10 sensors and a storage server.
The costs for installing a sensor hub with 10 sensors, that can cover an area with a radius of 6km, depends on several factors, therefore a rough estimate will be given. The hardware costs of a sensor hub and 10 sensors will be around €1500, €500 for the sensor hub and €100 for each sensor. Installing the sensor hubs and the sensors will take 2 men roughly 2 days, so 32 man-hours which will be around €1000. Additional costs, such as laying network and power cables, will be in difficult areas around the €1000 mark. So the total cost of installing sensor hub and 10 sensors will be around €3500.

Storage Servers must be deployed on relatively powerful servers. Dedicated servers managed by a hosting company are the most logical option for deploying the storage servers, such powerful dedicated servers will cost around €1000 per month per server.

We created a testcase for the country of Poland. For this model we used the following characteristics:

- 311.000 km^2 total area
- 38 million citizens
- 9000km of main waterways
- 440km of coast line

The length of the main waterways is determined by taking length of the 27 largest rivers in Poland and take the sum of these lengths, which results in 9900km.
Per 100km of main waterway we place on average 6 hubs and per 36km of coast line we also place 6 hubs. This leads to $9000/100 \cdot 6 = 540$ hubs for the main waterways and $440/36 \cdot 6 \approx 75$. So in the entire country of Poland we would have 615 hubs an. Given the estimate of €3500 per sensor hub the total costs are:

$$615 \cdot €3.500 = €2.152.500$$

We place a Storage Server roughly for each 100 hubs, so for the 615 hubs we would place 6 storage servers. Which would lead, if we take a 5 year lease for the servers, in the following costs:

$$6 \cdot 5 \cdot 12 \cdot €1.000 = €360.000$$

So the total costs for installing our system in Poland is:

$$€2.152.500 + €360.000 = \mathbf{€2.512.500}$$

2.8.4 Maintenance costs

The maintenance costs of the system will be dependent on the size of the system. The costs per component per year in man-hours is listed in table 6.

Description	**Man-hours per year**
Sensor maintenance	1
Sensor Hub maintenance	16
Storage server maintenance	120
Global Infrastructure Maintenance	360
Software Maintenance	1200

Table 6: Overview of development costs in man-hours.

Maintaining sensors covers routine checks of the individual sensors - every 12 months a check of the sensors is performed. The sensor hub maintenance, which includes maintaining the network connections and sensor communication connections, will take around 16 hours per sensor hub per year. Maintaining Storage Servers will take a lot more time, this includes updating software and monitoring server usage. Monitoring the complete system will take slightly more time than monitoring single storage servers as monitoring the complete system means identifying problems and passing the problems to more specific system monitors. The last point is the software maintenance, which is a large part of the maintenance due to the complexity of the flood detection model.

If we apply these maintenance costs to the previous example of Poland we get the following costs in man hours:

$$615 \text{ hubs} \cdot 16 \text{ man-hours} = 9840 \text{ man-hours}$$

$$6150 \text{ sensors} \cdot 1 \text{ man-hours} = 6150 \text{ man-hours}$$

$$6 \text{ storage servers} \cdot 120 \text{ man-hours} = 720 \text{ man-hours}$$

The maintenance costs of the Global Infrastructure and the Software are for the complete system so these aren't included in the overview of the maintenance costs for Poland. The total costs are roughly 16710 man-hours. With an estimated hourly rate of €30 the maintenance costs for Poland are

$$16710 \cdot €30 = \mathbf{€501.300}$$

2.8.5 Product pricing

The pricing of the product is hard, because the customers of our product will most likely be governmental bodies ordering our system for a large area. Therefore the product pricing will be on offer basis. So for each customer the total cost of the installation will be determined and on top of that a profit margin of 20% is counted.

2.9 Competitors

The Smart Flood Monitoring system proposed in this document is expected to be competitive. However, as mentioned before, the market is extremely competitive and a number of large scale projects with respect to flood signalling are currently in development. This section identifies the most prominent competitors in the field of flood monitoring and mentions and elaborates on the key characteristics of these flood monitoring providers.

2.9.1 Global Flood Awareness System

The Global Flood Awareness System (GloFAS) is a system of systems developed by the Joint Research Centre of the European Commission and the European Centre for Medium-Range Weather Forecasts (ECMWF). They assigned the NASA (and other large parties) to address and incorporate a number of its core functionalities. It is independent of administrative and political boundaries, couples state-of-the art weather forecasts with a hydrological model and with its continental scale set-up it provides downstream countries with information on upstream river conditions as well as continental and global overviews [7]. The system incorporates a number of subsystems: [7]

Real-time Integrated Global Flood Map The Critech Web Map Viewer offers an intuitive Web interface. Real-time Integrated Global Flood Map (Experimental).

Global Flood Monitoring System (GFMS) it is a NASA-funded experimental system using real-time TRMM Multi-satellite Precipitation Analysis (TMPAd) precipitation information.

Tropical Rainfall Measuring Mission (TRMM) The TRMM is a joint mission between NASA and the Japan Aerospace Exploration Agency (JAXA) designed to monitor and study tropical rainfall [7].

Global Flood Detection System The Global Flood Detection System monitors floods worldwide using near-real time satellite data. GFDS currently monitors around 10000 areas.

Dartmouth Flood Observatory Space-based Measurement and Modeling of Surface Water For Research, Humanitarian, and Water Management Applications.

Flood List Aims to bring information about floods and flooding from around the world, with the hope that it will inspire helpful discussion and collaboration in preparing for and dealing with the effects of flooding.

HEPEX To demonstrate the added value of hydrological ensemble predictions (HEPS) for emergency management and water resources sectors to make decisions that have important consequences for economy, public health and safety.

Global Flood News Monitoring of media is fairly common in many larger organisations. The use of social media in natural disasters has also demonstrated benefits. The same technologies can be used to monitor main stream and social media reports of floods.

Global Flood Alert System Is an attempt to make the best use of global satellite precipitation estimates in flood forecasting and warning.

GloFAS is in terms of functionality very comparable to the SFM system. However, it only provides information on flood risks to designated teams of different countries or states, and does not provide input directly to emergency services, the general public, or any other kind of stakeholder that may be interested in that information. The system is also fixed and it is not possible for any of the information-receiving parties to influence the construction of the system in any way. They can only request to receive information from the system. This means that the system is not directly adaptable to user needs and feedback which is harmful to the quality of the system. Furthermore, the fact that this system does not issue warnings to the general public and emergency services is a serious shortcoming that will not be present in the SFM system.

2.9.2 Global Flood Detection System

Another large scale initiative for flood monitoring for governments and organization is the Global Flood Detection System. It monitors floods worldwide using near-real time satellite data [8]. GFDS currently monitors around 10000 areas, defined in collaboration with partners [8]. GDACS is a cooperation framework between the United Nations, the European Commission and disaster managers worldwide to improve alerts, information exchange and coordination in the first phase after major sudden-onset disasters [8].

2.9.3 Small-Scale Competitors

Apart from the global competitors mentioned above, there exists a large number of smaller-scale organizations which offer solutions for (smart) flood monitoring. These firms usually target governments, municipalities or companies. These systems usually deal with a fixed amount of sensors and are only accessible by a specific set of parties, so the quality of this system will quickly degrade as it does not adapt to user needs or new technologies.

3 Requirements

In this section, the requirements of the Smart Flood Monitoring system will be described. First of all, the architectural vision of the system will be outlined. Secondly, the stakeholders and their concerns are identified, and the key drivers for the architecture will be determined. Finally, the use-cases, requirements and risk assessment of the system are given.

3.1 Architectural vision

The following diagram shows the environment of the SFM system, its components, and interactions between those components.

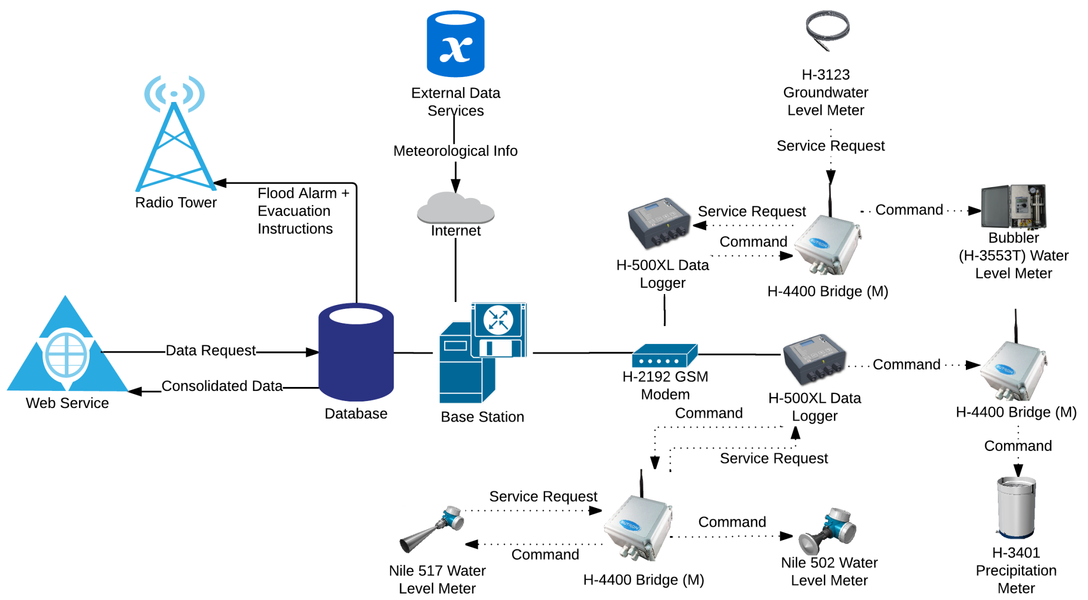


Figure 3: High-Level Vision of the SFM system

The system consists of a web portal and a sensor network that are both connected to the Internet. They exchange information with a central database that keeps track of measurements that happen in the sensor network. The sensor network itself, on the right of the diagram, is made as a tree structure, where the nodes of the three are propagation hubs (SDI-12 data loggers) and the leaves are the different sensors. We distinguish five types of relevant sensors for flood monitoring that will each measure one of the following properties: ground water level, water current speed, local water level, local precipitation and pressure sensors.

Converters are put into place to convert the data obtained by the SDI-12 sensors to a format that can be used on the base station. Furthermore, we utilize external data sources to improve the prediction of floods. For every sensor in the sensor network, the base station will request that the sensor submits data every so often. This request is propagated down through the network of hubs until it reaches one (or multiple due to redundancy for fault tolerance) measuring hubs, who will poll the sensor over a wireless network for data. The data is then propagated back towards the central server and stored in the database.

The web portal allows users, customers, emergency services and maintenance staff to view information on determined flood warnings and to access the sensor network directly. Furthermore, these stakeholders can subscribe via a web portal to receive flood warning notifications through email or other types of digital addressing.

3.2 Stakeholders

The stakeholders of the system will now be identified. We will describe the concerns of the stakeholders using quality attributes as defined in the ISO/IEC 9126 [3] and ISO 25010 [4] standards. The quality metrics that will be used in this section are:

Standard	**Quality attributes**
ISO/IEC 9126	Fault tolerance, Interoperability, Authenticity, Modularity, Adaptability, Efficiency, Performance
ISO 25010	Usability, Maintainability

Table 7: Used definitions of quality attributes.

Furthermore, we define the following quality attributes:

Profitability The ability for the manufacturer, distributor, or licensing authority of a product to create a profit through the sales of a product and the various maintenance activities for that product thereafter.

Affordability The ability for stakeholders to gain a net profit or other kind of significant advantage that weighs up against the installation and maintenance costs of the system.

3.2.1 Primary stakeholders

The following stakeholder groups are distinguished and their concerns are described with the introduced definitions:

Product owner - The product owner finances and oversees the development of the product. The main concerns of the product owner are the affordability of the product for potential customers, the ability for the system to interact with existing and future infrastructures and components, and how well the system performs in general. This correlates loosely to the unique selling points of the system.

Customers - The governmental authorities or private companies that will purchase an instance of the SFM system for a certain area, and will compensate our company for the continued maintenance of its components. The damage that this kind of stakeholder attempts to limit by utilizing the system is both material and personal. The concerns of this group of stakeholders is that the system should be affordable, that the system should be highly available and fault tolerant, as well as a high performance and the ability for the system to change over time due to novelty sensor technology or changing user needs. Theses type of stakeholders would also like to receive warnings issued by the system and see information about emergency situations on the web interface of the system.

End users - The individuals or groups of individuals that will use the SFM system to receive flood warnings when they are in the vicinity of the area that has been flooded, or who want to receive information on recent floods on request through a web interface. Opposed to Customers, the most important damage to prevent for this type of stakeholder is personal, in other words human lives. The concerns are high performance, the ability for the system to communicate with other systems (to warn these stakeholders and emergency services) and that the system should adapt to changing user needs.

Emergency services - This includes the first responders of organisations such as the police, the fire department and hospitals. Their primary concerns are that the system should be highly available so that, in the case of a flood warning, the system will be up and running with high performance. Furthermore, the authenticity of issued warnings is important to prevent false alarms.

Maintenance team - The group of technicians that is responsible for the maintenance of the software and the variety of hardware components that are concerned with the system, as well as the back-end, front-end and customer support of all peripheral systems such as web interfaces. Their concerns are that the system should be modular, easy to adapt in terms of sensor components or other information sources, and that the hardware and source code should have high maintainability. Furthermore, the system should be fault tolerant with high performance, so that as little as possible excessive maintenance will have to be performed.

Development team - The developers and installers of the software and hardware that is needed for the Smart Flood Monitoring system to function correctly. They aim to make the system fault tolerant, adaptive in terms of information streams and user needs, and high performing. Secondary concerns consist of authenticity of requests being sent, the system being modular and efficient.

3.2.2 External stakeholders

Besides the primary stakeholders that are actively part of the system, there are external stakeholders that indirectly influence and are influenced by the SFM system:

Researchers - Researchers and scientists that are interested in accessing and utilizing the data that is captured by the sensor network of the Smart Flood Monitoring system, with the goal of research that may improve the system's capabilities of forewarning floods. Researchers do not actively take part in the SFM system, but they can benefit hugely from the system.

System component suppliers - The providers of the sensor equipment, information hubs, and other types of hardware included in the system. All the system component suppliers are essential to the SFM system, but the products used from these suppliers are off the shelf products.

Network and infrastructure providers - The providers of infrastructure such as mobile and wired networks, power grids, telephone networks, which may be used by the SFM system. Just like the products used from the system component suppliers the products of the network and infrastructure providers are used but do not impact the system hugely.

External data providers - The external information experts or information streams that are, in conjunction with the data obtained from the sensor network, to determine whether to generate flood warnings in specific circumstances. The information from the external data providers is used to assist the decision taking in the SFM system, but the information doesn't change the system.

3.2.3 Stakeholder concerns

The demands of the different types of primary stakeholders do not weigh uniformly. The following table displays the amount of points that can be distributed in the concerns matrix per stakeholder type. We do not consider the external stakeholders (researchers, system components suppliers, network and infrastructure providers and external data providers) in the exploration of the concerns, due the fact that these stakeholders will not have any direct influence in the requirements of the system.

The development team is the least significant type of stakeholder, since they are only concerned with the implementation of the system. The maintenance team will maintain the system on a daily basis, so it is more involved into the system than the development team. The emergency authorities are more significant due to the fact that they will have to respond to warnings given by the system to prevent personal and material damage. The product owner is the most significant stakeholder, since the product owner oversees the architectural vision and design decisions of the system. While customers will accommodate financially for the placement and maintenance of the system, we determined end users to be just as important, due to the fact that we consider personal damage more important than material damage, so these stakeholders are equally relevant in the concerns exploration.

Stakeholders	**Points**
Product owner	300
Customers	300
End users	300
Emergency authorities	200
Maintenance team	150
Development team	100

Table 8: Priorities of stakeholders indicated by the amount of points to spend.

	Concerns										
Stakeholders	Profitability	Affordability	Fault tolerance	Interoperability	Authenticity	Modularity	Adaptability	Efficiency	Usability	Maintainability	Performance
Product owner	50	25	45	45			45		10	30	50
Customers		50	80	40			50				80
End users				60			60		80		100
Emergency authorities			80	20	40						60
Maintenance team			25			25	25	20	10	25	20
Development team			30			20	20	10			20
Total:	50	75	260	165	40	45	200	30	100	55	330

Table 9: Matrix of stakeholders concerns.

The stakeholders and concerns, as described above, are summarized in the concern matrix above. The stakeholders are listed in decreasing order of importance and each type of stakeholders is given the amount of points to spend on the various concerns that may drive the system architecture.

3.3 Key Drivers

From the matrix of stakeholders and their concerns, we can deduce that the following concerns may be listed as key drivers for the architecture, as they reflect the concerns in the matrix that have received the highest combined scores from the different stakeholder groups. Due to the fact that the architecture description needs to be completed within a limited time frame, we only assume the three most significant concerns as key drivers.

Performance - The main purpose of the system is to warn end users and companies of increased chances of floods in their vicinity. These stakeholders should receive these warnings as early as possible when they are identified by the system through its information sources, and the system should be engineered and tuned in such a way that the occurrence of false warnings is as low as possible. The validity, accuracy and timeliness of these warnings are crucial factors in the success of the system.

Fault tolerance - Since the described system is essentially a distributed sensor network which consists of many different components which should be highly available, attention should be closely paid to ensure that a failure somewhere in the system has minimal to no impact to the service provided by other parts of the system.

Adaptability - It should be possible to attach new components and detach old components from the system with relative ease, and to adapt components of the system to work in new environments. The system should be designed in such a way that it is capable of incorporating new (future) information sources into its warning system to improve accuracy, performance or usability for users. Being able to adapt to new environments and to keep up with the rapid evolution of sensor technology will increase the life span of the system and the accuracy of the system.

There is a fourth concern that is deemed as important according to the concerns of the stakeholders as listed in the matrix, which is Interoperability. We consider this concern as a secondary key driver, due to the fact that it is significantly important to the system compared to the three primary key drivers, and will help maintain adaptability and performance (through the addition of extra communication and information networks, channels or emergency services).

Interoperability - This describes the ability of the system to cooperate with other systems, applications and interfaces. In particular, the system should be capable of issuing warnings to residents of a certain area through the use of the telephone network, mobile networks or the Internet. Furthermore, the system is required to interact with the relevant emergency authorities in case of an established flood.

3.4 High-Level Requirements

The requirements, that are listed below, represent the high level functionality of the SFM system. Each of the high level requirements will be refined by various requirements.

HL-1	**Must**	**Collecting sensor data**
The Smart Flood Monitoring system must collect data from a wide variety of sensors. This data must be saved to be able to analyse this data to see patterns in this data. Besides the collection of sensor data, the system should request data from different kind of external sources that will support the analysis of the sensor data. These external sources consist of data on the weather and external experts.		

HL-2	**Must**	**Analyzing sensor data**
The Smart Flood Monitoring must be able to make a risk analysis given the stored data. The decision should consist of some kind of rating that indicates the likelihood of a flood to occur within a specific area. This way, part of the end users can be informed when the likelihood is uncommonly high in a particular area that is covered by the sensor network.		

HL-3	**Must**	**Detecting failing components**
The Smart Flood Monitoring should be able to identify faulty sensors or hubs in the system automatically, and should assist the maintenance team by marking or pointing out these faulty components to the maintenance team.		

HL-4	**Must**	**Sending flood warnings to users**
The Smart Flood Monitoring system must alert users in case of an imminent thread of a flood in the vicinity of the users. By sending out uni-cast messages, via i.e. cell broadcasting, to all the people that are in the area of the possible flood.		

HL-5	**Must**	**Extending measuring network**
After the installation of the sensor network, the maintenance team should be able to add new components of the system which utilize new measuring technologies, or to incorporate new external information streams in the analyzing process. New sub-networks of sensors can be connected to the main system with relative ease to increase the diversity and/or the area of the entire sensor network.		

HL-6	**Must**	**Integrating third parties and services**
The system should communicate with third parties such as the emergency services and the organisations and authorities that own a specific sub-network of the SFM system. Furthermore, the system should be capable of utilizing mobile networks in order to broadcast notifications to mobile devices. The system should be designed in such a way that it is possible to integrate new and existing services in the future.		

HL-7	**Must**	**Web portal**
Apart from warning the user directly through various communication methods, the SFM system should also allow its users to browse the statistics of flood threats and real-time information about incidents happening within the sensor network through a web portal. The system could also make low-level data available for researchers, such as raw sensor data, and the respective organisations or authorities and maintenance teams can view detailed status information on the sensors within (a part of) the sensor network.		

HL-8	**Must**	**Machine learning**
The system applies supervised machine learning techniques and algorithms, along with a model that has been constructed from available training data, to predict the occurrences of floods in certain areas with high performance.		

HL-9	**Optional**	**Self and manual adaptation**
The system is able to process automatically whether or not a warning that has been issued by the system was valid by examining the sensor data that is being gathered after the warning has been issued. A warning is not only valid if a flood is actually happening after the warning is issued, but also if there are obvious (or even extreme and dangerous) deviations in the measurements made by sensors in that specific area, such as the water level being abnormally high. The maintenance team can also modify the model of the system manually through the web portal.		

HL-10	**Optional**	**Enhanced guidance for users and emergency services**
When a flood warning has been issued for a specific area, the system should provide guidance for users and emergency services in and around that area, so that they may safely evacuate the area, or so that emergency services can reach parts of the affected area where their service may be needed. Not only does the system accomplish this through giving information on the location and severity of flood warnings through the web portal (see HL-7), but it may also provides information about the safety of roads in the area, depending on their flood risk level. This may help users and emergency services plan routes in, through, and out of the affected area.		

3.5 Stories and Use Cases

The diagram below describes the system in terms of use cases. We will elaborate upon the non-trivial use cases in this section and provide detailed descriptions of the most architecturally relevant use cases.

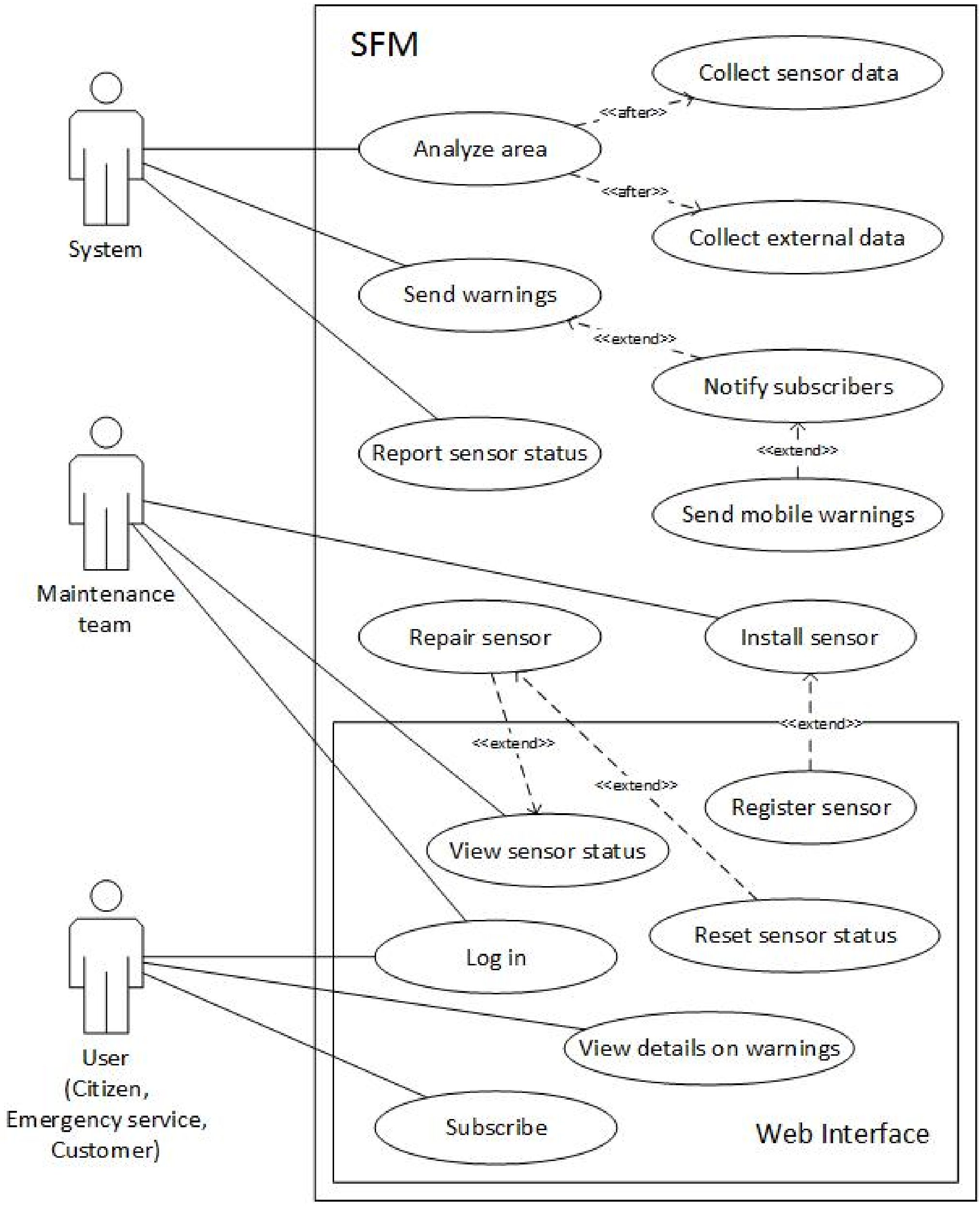


Figure 4: Use case diagram featuring the most significant use cases of the SFM system.

UC-1.1 - Collect sensor data

Primary Actor	System
Goal	Collect and save a measurement of a specific sensor in the sensor network of the system.

Preconditions

The central server of the system is connected to the sensor network, and the sensor was once registered to the sensor network and identified by a unique identifier. A certain time threshold has been passed since the last time that the specific sensor has been polled for data, or has not yet been polled at all.

Main success scenario

1. The central server of the system broadcasts a message requesting the data of a specific sensor asynchronously, and continues to perform other tasks.
2. The message is routed through the hubs of the sensor network until it reaches a hub at which that specific sensor is registered.
3. The hub makes a request to the sensor to submit its data asynchronously, and continues to perform other tasks.
4. The sensor makes a measurement or looks up a recent cached measurement and responds to the hub with this data.
5. The hub propagates the data obtained from the sensor back towards the central server of the network.
6. The data of the specific sensor is logged by time, identifier and location, and saved on the database of the central server.

Extensions

2a. The sensor is not registered at any hub in the sensor network.

- 2a-1. See UC-3.1.

4a. The sensor is broken or is running but does not respond to requests.

- 4a-1. See UC-3.1.

Post conditions

The system has polled recent data for the sensor which has been saved to the database, or the status of the sensor has been set accordingly in any other case.

Related requirements	HL-1, FR-1.1, FR-1.2, FR-3.2, TR-1.1, TR-1.2, TR-1.4, TR-3.2

UC-1.2 - Collect external data

Primary Actor	System
Goal	Poll data from an external sensor or information stream.

Preconditions

The system is connected to the Internet and it is possible to label the obtained data on time, location and type.

Main success scenario

1. The system makes contact with the external data provider.
2. The data provider responds with a data set that indicate a certain environmental property in specific locations on specific times, which are embedded in the data.
3. The central server of the system stores this data in the database of the system to be used for the analysis.

Extensions

- 2a. The data is not properly labeled on time and location.
  - 2a-1. The system estimates the time at which the data samples were observed and uses contextual information about the data provider to determine how the data samples should be labeled on location.
- 2b. The data is in a data format that is not compatible with the database.
  - 2b-1. If no algorithm is available for the data format conversion, the maintenance team is notified and the system aborts the operation.
  - 2b-2. If an algorithm is available that can convert the data from the current format to the target format, then the system applies the algorithm to the data and continues from step 3.

Post conditions

The data has been polled and is labeled right to be used in the analysis of flood risks.

Related requirements	HL-1, FR-1.3, FR-1.4, FR-1.5, FR-1.6, TR-1.3, TR-1.4

UC-2.1 - Analyze area

Primary Actor	System
Goal	Analyze a certain area that falls within the extents of the sensor network to determine the likelihood of floods in the upcoming hours.

Preconditions

Data has been gathered on the area that is to be analyzed according to UC-1.1 and UC-1.2.

Main success scenario

1. The system retrieves the data samples from the database that have been measured in the particular area and within the past few hours.
2. The system applies machine learning models and thresholds to determine if there is an abnormally elevated risk factor for a flood to occur in that specific area.
3. The system detects that there is no elevated risk in the area.
4. The system saves the risk factor to the database and the analysis aborts.

Extensions

2a. There is too little available data to make a meaningful analysis.

- 2a-1. The system sends a debug report to the maintenance team and marks the analysis as invalid in the database.

3a. The system detects that there is an elevated risk in the area.

- 3a-1. The system proceeds sending warnings to the relevant stakeholders through the information channels that have been configured at the central server of the system. See UC-4.1 and UC-6.1.

Post conditions

The analysis on the area has been performed and the appropriate actions have been taken if necessary.

Related requirements	HL-2, FR-2.1, FR-2.2, FR-2.3, FR-2.4, TR-2.1, TR-2.2, TR-2.3

UC-3.1 - Report sensor status

Primary Actor	System
Goal	Indicating the status of a specific sensor in the sensor network.

Preconditions

The sensor was once registered to the network and the central server of the system is connected to the sensor network. The specific sensor is identified by a unique identifier.

Main success scenario

1. The system notes that a sensor has recently been registered or has recently been polled.
2. The system performs an analysis on the values obtained by the sensor compared to nearby sensors.
3. The system decides that there are no determinable abnormalities in the data from the sensor.
4. The sensor is marked as "online", meaning it will be polled for data.

Extensions

1a. The system notes that a sensor that was about to be polled is not registered at any hub in the sensor network.

  1a-1. The sensor is marked as "not present" and will not be polled for new data until handled by the maintenance team.

1b. The system notes that a sensor that was about to be polled has not replied within a determined time threshold.

  1b-1. The sensor is marked as "offline" and will not be polled for new data until handled by the maintenance team.

3a. The system detects that there are significant abnormalities in the data obtained from the sensor.

  3a-1. The sensor is marked as "needs calibration" and will not be polled for new data until handled by the maintenance team.

Post conditions

The status of the sensor has been set accordingly.

Related requirements	HL-3, FR-3.1, FR-3.4, FR-3.5, TR-3.1, TR-3.2, TR-3.4

UC-4.1 - Notify subscribers

Primary Actor	System
Goal	Notify users and services who have subscribed to the system about flood warnings.

Preconditions

It is known how to contact the subscribed users, either via email or some other defined interface address.

Main success scenario

1. The system has analyzed that there is an elevated risk of flood in a certain area.
2. The system iterates over all services that subscribed to this area (higher priority), and sends a flood warning in the appropriate protocol to each of the contacts for the listed subscribers.
3. The system notifies users near the area for which there is a flood warning. See UC-6.1.
4. The system now iterates over all users that subscribed to this area (lower priority), and sends the same flood warning as done in step 2.

Extensions

2a. The system cannot contact one of the subscribed services with high priority.

  2a-1. The maintenance and support teams are notified that the service cannot be reached and the system continues with the next subscriber in step 2.

4a. The system cannot contact one of the subscribed users with low priority.

  4a-1. A warning is logged to the database, but is not directly propagated to maintenance and support teams.

Post conditions

All subscribed users have been informed about the flood warning, or appropriate error handling has been put in place to mitigate the inability to contact a subscriber.

Related requirements	HL-4, HL-6, FR-4.1, FR-4.2, FR-4.3, FR-4.4, FR-4.5, FR-6.1, TR-4.1, TR-6.2

UC-5.1 - Register sensor

Primary Actor	Maintenance team
Goal	Registering a new sensor to the system, so that it can be polled for information.

Preconditions

A sensor has been installed in the sensor network with a unique identifier. The sensor has been registered at one or more lower-level hubs. The location of the sensor is determined by the maintenance team. The maintenance team is logged into an authorized account on the web portal and has a connection to the Internet.

Main success scenario

1. The maintenance team navigates to the "Sensor information" page and initiates the registration procedure on the web portal.
2. The maintenance team enters the identifier and location of the sensor and confirms.
3. The system performs a check to see if the sensor is reachable and that the sensor responds with valid sensor data for that specific type of sensor.
4. The system concludes the check successfully and registers the sensor under the given identifier.

Extensions

4a. The sensor is determined to be not reachable after a certain timeout has passed without successful communication with the sensor.

  4a-1. The system informs the maintenance team that the sensor is not reachable.

4b. The sensor provides data that is not within the expected range of that sensor type and/or is not comparable to surrounding sensors.

  4b-1. The system informs the maintenance team that the location is incorrect or that the sensor is not calibrated correctly.

Post conditions

The sensor has been added to the system's register, or the maintenance team has been informed about the failure that happened during the registration

Related requirements	HL-5, FR-5.1, FR-5.2, FR-5.3, FR-5.4, FR-5.5, TR-5.1

UC-6.1 - Broadcast mobile networks

Primary Actor	System
Goal	Broadcasting a flood warning to cellphones and smart phones in a certain area experiencing an elevated flood risk.

Preconditions

The system knows the positions of telephone towers in and around the affected area and has the ability (via a governmental authority) to command these telephone towers to broadcast warnings to mobile users connected via those telephone towers. The system has a connection to the telephone network.

Main success scenario

1. The system has analyzed that there is an elevated risk of flood in a certain area.
2. The system determines all of the telephone towers that overlap the affected area.
3. The system commands these telephone towers to broadcast a flood warning to mobile users.
4. The system determined that the warning broadcasts have been propagated.

Extensions

2a. The system determines that there are no telephone towers in the affected area, or that the covered area by the telephone towers is insufficient.

  2a-1. The system notifies the support team so that they can decide the course of action.

Post conditions

A high fraction of the mobile users in the affected area have been warned via telephone networks.

Related requirements	HL-4, HL-6, FR-4.1, FR-6.1, FR-6.2, FR-6.3, TR-4.1, TR-6.1

UC-7.1 - Log in

Primary Actor	User (End user, Customer, Maintenance team)
Goal	Logging a user into an account (and registering for an account if they do not yet have an account).

Preconditions

The user is not logged in, has a valid email address, and is connected to the Internet.

Main success scenario

1. The user starts a log-in attempt on the web portal.
2. The web portal asks the user to either log-in or register for a new account using forms.
3. The user has an account and fills in their credentials in the form.
4. The web portal checks the credentials with success and creates a log-in session for the user.

Extensions

1-3a. The user cancels the action at any time.

- 1-3a-1. The user will not be logged in.

3b. The user does not have an account.

- 3b-1. The user provides the required information (email address and password) in the form for the registration of an account.
- 3b-2. The web portal checks the information and sends a confirmation email to the user's email address when the check succeeds, or shows an error to the user if the check fails and asks the user to retry their registration.

4a. The web portal checks the credentials and concludes that they are incorrect.

- 4a-1. The web portal shows an error to the user and continues from step 2.

Post conditions

The user is logged into their account.

Related requirements	HL-7, FR-7.4, FR-7.5, FR-7.6, FR-7.7, TR-7.1, TR-7.2, TR-7.4, TR-7.5

UC-7.2 - View sensor status

Primary Actor	Maintenance team
Goal	Discovering the status of a certain sensor in the sensor network.

Preconditions

The maintenance team is logged into the web portal using an authorized account, see UC-7.1. The sensor is identified by a unique identifier. The sensor was once registered to the system and has a status label in the system. There is a connection to the Internet.

Main success scenario

1. The maintenance team navigates to the "Sensor information" page.
2. The web portal shows a list of all sensors of the sensor network.
3. The maintenance team uses a "Search" option to manually find a sensor with their identifier or finds the specific sensor in the displayed list.
4. The web portal shows detailed information on the sensor, including last recorded data and status.
5. The maintenance team inspects the data and the status of the sensor, observes that the status of the sensor is "online", and does not need to take any action.

Extensions

5a. The maintenance team discovers that the sensor is marked "offline", "not present" or "needs calibration".

5a-1. The maintenance team will inspect the sensor for defects and failures and reset the status of the sensor afterwards through the web portal.

Post conditions

The maintenance team is aware of the status of the sensor.

Related requirements	HL-7, FR-3.3, FR-7.4

UC-7.3 - Subscribe

Primary Actor	User (End user, Emergency service, Customer)
Goal	Subscribing to flood warnings of the entire system via email.

Preconditions

The user is logged into an account, see UC-7.1. The user has an email address, telephone number, or some address for a different messaging protocol that they would like to use for a subscription to the SFM system for receiving warnings, and there is a connection to the Internet.

Main success scenario

1. The user navigates to the "Manage subscriptions" page and performs the "Add subscription" action on the web portal.
2. The web portal asks which areas the user would like to receive warnings for and how severe the warning should be before receiving a digital warning, and asks for a way to contact the user (email, telephone, RSS feed).
3. The user selects his personal preferences and fills in the required addressing information, and confirms.
4. The web portal shows a confirmation message to the user and the user will be notified via subscription of relevant future warnings.

Extensions

1-3a. The user cancels the process anywhere before confirming.

1-3a-1. The user will not be subscribed to flood warnings.

3b. The user provides invalid input.

3b-1. Notify the user of the mistakes and continue from step 2.

Post conditions

If the user has completed the subscription process, they are now subscribed to relevant flood warnings.

Related requirements	HL-7, FR-7.5, TR-7.3

3.6 Functional Requirements

The following requirements are indexed in such a way that the first number refers to the high-level requirement that they support.

FR-1.1	**Must**	The system shall request and receive data from sensors, by polling sensors for their data asynchronously.
FR-1.2	**Must**	The sensor network will store the data collected from sensors and external data providers for prospective risk analysis.
FR-1.3	**Must**	The system shall contact external data providers and services to collect data about precipitation.
FR-1.4	**Optional**	The system shall contact external data providers and services to collect data about water currents.
FR-1.5	**Optional**	The system shall contact external data providers and services to collect data about sea levels.
FR-1.6	**Optional**	The system shall contact external data providers and services to collect data about wind speeds.
FR-2.1	**Must**	If an area needs to be analyzed for flood risks, the system shall load data that has been obtained beforehand from sensors in that area for the analysis.
FR-2.2	**Must**	The system shall load external data that has been gathered from external sources, from which the measuring locations are contained in the area that is being analyzed.
FR-2.3	**Must**	The system shall apply the flood detection model to the set of relevant (as defined in FR-2.1 and FR-2.2) data.
FR-2.4	**Must**	The system shall store the calculated flood likelihood for the analyzed area.
FR-3.1	**Must**	The system shall evaluate status labels for the sensors which can be one of the following: "online", "offline", "not present", "needs calibration".
FR-3.2	**Must**	The system shall not poll sensors for data that are marked with a status that is anything other than "online".
FR-3.3	**Must**	The maintenance team is able to view the status labels for all of the (once) connected sensors.
FR-3.4	**Must**	The system shall detect when a component (hub or sensor) is not able to communicate using a defined timeout on the connection between the server and that component.
FR-3.5	**Must**	The system shall detect when a sensor is not yet registered to any of its children hubs.
FR-4.1	**Must**	The system shall broadcast warnings about floods in the nearby area to users using available data networks that are integrated.

FR-4.2	**Must**	The system shall determine, from a specified area for which an elevated flood risk has been determined, what the relevant network nodes are to be used for the warning procedure through the data networks.
FR-4.3	**Must**	The system shall calculate the areas to which warnings are to be sent, given a data map of flood risk values.
FR-4.4	**Must**	The system can send warnings to subscribers via email, if they have subscribed with their email address.
FR-4.5	**Must**	The system can issue warnings to subscribers via SMS messages, if they have subscribed with their telephone number.
FR-5.1	**Must**	The system shall supply the maintenance crew with a form to register new sensors, where a unique identifier, physical location of the sensor and its type have to be entered.
FR-5.2	**Must**	The maintenance team can insert new sensors in the sensor list by completing the form as defined in FR-5.1.
FR-5.3	**Optional**	The system shall supply the maintenance crew with an interface to calibrate sensors.
FR-5.4	**Optional**	The system shall apply the calibration to the sensor values that arrive at the system, given the type of sensor.
FR-5.5	**Must**	After registration, the system shall check if the newly registered sensor can be reached through the sensor network, utilizing a defined timeout.
FR-6.1	**Must**	The system shall be able to communicate with telephone stations to broadcast flood warnings to mobile users in the affected area by those flood warnings.
FR-6.2	**Must**	The system shall determine the telephone stations to use for broadcasting warnings, given a certain area in the sensor network with elevated flood risk.
FR-6.3	**Future**	The system shall be able to, through radio networks, distribute messages to FM radios of vehicles in the areas affected by flood warnings.
FR-7.1	**Must**	Users shall be able to view detailed information (including the area, time, and severity) about distributed flood warnings through a web portal.
FR-7.2	**Must**	Users shall be able to view information on the web portal about flood warnings without needing to create an account or provide personal information.
FR-7.3	**Must**	The system shall store the aforementioned detailed information about flood warnings, including their cause, location and communication information regarding the warning.

FR-7.4	**Must**	The maintenance team and customers can log into the web portal to view detailed information about the sensor network, the web portal, the status of the sensors, and the data that has been obtained.
FR-7.5	**Must**	Logged-in users, customers and emergency services can subscribe to flood warnings using a custom addressing method, including email and SMS, by filling in their personal information and addressing details, e.g. an email address or telephone number, and by choosing the areas for which they want to receive flood warnings.
FR-7.6	**Must**	Logged-in users, customers and emergency services can view their subscriptions to flood warnings and can unsubscribe from flood warnings.
FR-7.7	**Must**	Users can create an account by providing their email address and a password, and can subsequently use these credentials to log into their account.
FR-7.8	**Future**	Logged-in researchers can access low-level sensor data that has been gathered by the sensor network of the system for research purposes.
FR-8.1	**Must**	The system uses a machine learning model, which implements supervised learning techniques so that it may be modified to enhance performance, to analyze the risks of a flood occurring in a specific area within the sensor network.
FR-9.1	**Optional**	The system shall analyze the output of relevant (as defined in FR-2.1 and FR-2.2) sources after a flood warning has been given, to determine if the warning was valid or invalid automatically.
FR-9.2	**Optional**	The maintenance team can modify the machine learning model that calculates the risk of occurrence of a flood in a specific area, according to system (see FR-9.1) and user feedback.
FR-10.1	**Optional**	When a flood warning is issued for a specific area, the system will generate a heatmap of flood risk values and overlay these with a map of the roads in the area, so that stakeholders in that area are informed about which roads are safer or riskier to use through the web portal.
FR-10.2	**Future**	Stakeholders can use the web portal to plan the safest route (in/out/through the area affected by an elevated flood risk) between two arbitrary points defined by the stakeholder, using the generated heatmap and a map of the roads as explained in FR-10.1.

3.7 Technical Requirements

The following requirements are indexed in such a way that the first number refers to the group of functional requirements that they support.

TR-1.1	**Must**	The system can communicate over wired networks with the hubs of the sensor network.
TR-1.2	**Must**	The hubs of the sensor network can communicate both wired and wireless with the individual sensors.
TR-1.3	**Must**	The system shall have a connection to the Internet and thus to the external data sources.
TR-1.4	**Must**	There shall be a database in the system that stores the data obtained from the sensors and external data providers.
TR-1.5	**Must**	The system has non-blocking polling mechanisms for obtaining data from its sensors and external data sources.
TR-1.6	**Must**	The system is able to make distinctions between sensors by identifying them using their uniquely assigned identifier.
TR-2.1	**Must**	The database should support both reading and writing of information in real-time.
TR-2.2	**Must**	The locations of the hubs and sensors in the sensor network are accurately known to the system.
TR-2.3	**Must**	The system can apply the pipeline of a prediction algorithm to a large set of data in real-time.
TR-3.1	**Must**	The communication between components has a defined timeout of 30 seconds, so that the system may conclude that a polling request for data has succeeded or failed.
TR-3.2	**Must**	The measuring hubs keep track of the sensors that it can reach through wired or wireless networks.
TR-3.3	**Optional**	The central server can request registration data of sensors from lower level hubs.
TR-3.4	**Must**	The system implements an algorithm that determines whether a range of values from one sensor is significantly different from the range of values from another nearby sensor or external data provider.
TR-3.5	**Must**	The system is capable of storing polling requests it has made and revisit these after the connection timeout of 30 seconds has passed, for the purpose that has been stated in TR-3.1.
TR-3.6	**Must**	The central server can request lower-level hubs to confirm or deny that a particular sensor, identified by its identifier, is registered at that hub.

TR-4.1	**Must**	The system can communicate with the various integrated networks, such as wireless networks, wired networks, telephone and radio networks, to send digital messages and warnings such as SMS and email.
TR-4.2	**Must**	The system has knowledge about the position and range of broadcasting nodes such as radio towers or telephone stations, so that it can calculate which broadcasting nodes should be used for issuing warnings to a specific area.
TR-5.1	**Must**	The system must be reachable for the maintenance team at all locations where sensors are placed through a remote interface.
TR-5.2	**Optional**	The system stores calibration offsets for each sensor in the system which can be set remotely through the interface mentioned in TR-5.1.
TR-6.1	**Must**	The system is licensed to broadcast messages over radio and telephone towers in real-time at any moment.
TR-6.2	**Must**	The system must be able to send direct messages to users via SMS
TR-6.3	**Must**	The system must be able to send direct messages to users via E-mail
TR-6.4	**Optional**	The components of the sensor network are capable of recognizing existing networks in a particular area and attempt to communicate over these networks with other components.
TR-6.5	**Optional**	The system is capable of producing Radio Data System code 31 signals [9] that will interrupt the the radio audio in cars in range of the transmitter.
TR-7.1	**Must**	There should be a clear distinction between authorized (customers and maintenance team) and non-authorized (all other stakeholders) users, which is determined through the user logging in on the web portal.
TR-7.2	**Future**	The sensor data and status in the database is available for authorized users over a secure encrypted connection.
TR-7.3	**Must**	The personal and address data for subscriptions of users should be saved securely on a database and linked to the relative accounts.
TR-7.4	**Must**	Authorized users can change status data in the database directly through the web portal, e.g. the maintenance team can mark a sensor as "online" when they have repaired/inspected/calibrated the sensor.
TR-7.5	**Must**	The log-in name or email address, a salted hash of the user's password and other relevant account information must be stored securely on the database.

TR-7.6	**Must**	It is possible to distinguish areas within the sensor network so that users may refer to particular subsets of the total sensor network area during their subscription.
TR-8.1	**Must**	The system stores all obtained data values over time, from every sensor in the sensor network, which may be retrieved at any time.
TR-9.1	**Optional**	The monitoring units store the warnings issued by itself, so that they may be reviewed and verified later.
TR-9.2	**Optional**	Authorized users can view and modify the parameters of the machine learning module.
TR-9.3	**Optional**	Authorized users can assist the supervised learning process by providing direct feedback (confirm/deny) on the accuracy and performance of the system for a given flood warning.
TR-10.1	**Optional**	The system can generate a heatmap based on the risk values that aggregates the risk values of several analyses that have been performed in the preceding minutes.
TR-10.2	**Optional**	The system has access to data which identifies the road network in the area in which the heatmap is contained.
TR-10.3	**Optional**	The system is capable of interpreting the road network and the heatmap to designate a risk level to each road in the road network.
TR-10.4	**Optional**	The web portal is capable of serving a complex road network with interpreted risk values to users in real time during peak web traffic.
TR-10.5	**Future**	The system is capable of interpreting connections between roads and apply a weighted path finding algorithm (modified version of the A* algorithm) to calculate the safest route between two user-defined points within an area with elevated flood risk.

3.8 Commercial Non-Functional Requirements

This section describes the commercial non-functional requirements.

CNR-1	**Must**	The price of running the entire system for 5 years should not exceed €100.000 for a system that contains 20 hubs, 100 sensors and a central server. The maintenance of such a network should not exceed €36000 per year for software and €27000 per year for hardware and infrastructure.
CNR-2	**Must**	The sensors themselves should have a lifespan of at least 5 years. The hubs should have a lifespan of at least 10 years.

CNR-3	**Must**	The sensors and hubs should not hinder users or non-users of the system, and should be placed in locations where they are not a hindrance, as well as still being able to perform their measurements correctly.
CNR-4	**Must**	The components of the system should be connected to the power grid or be self sufficient through solar cells in combination with batteries. In either case, the availability of the sensor is of paramount importance, and in the case of the power grid care should be taken that the component does not use too much power to be ecologically responsible.
CNR-5	**Must**	The communication methods of the components should not interfere, obstruct or scramble communication between devices that are not a part of the system, unless in necessary situations where a flood warning occurs.

3.9 Technical Non-Functional Requirements

This section describes the technical non-functional requirements of the system, based on the following categories: fault tolerance, performance, adaptability, interoperability, integration.

3.9.1 Fault tolerance

TNR-1.1	**Must**	Every sensor in the network should be reachable by at least two hubs.
TNR-1.2	**Must**	A sensor that is marked as broken does not hinder the evaluation of risks and the sensor should not be taken into account until it is fixed.
TNR-1.3	**Must**	One web portal should handle up to 50.000.000 page views per day, should be able to handle up to 1.000.000 concurrent users (measured within an interval of five minutes), and should be highly scalable, as the amount of page views will vary greatly depending on the presence of flood warnings and the fact that the measuring system can span across entire countries with high population.
TNR-1.4	**Must**	One web portal should have at peak traffic, as defined in TNR-1.3, at least 500 instances for redundancy in case part of the instances fail.

3.9.2 Performance

TNR-2.1	**Must**	The sensors should be polled frequently, at least once per 15 minutes.
TNR-2.2	**Must**	The variety of sensor data (including external data providers) in the network should cover at the least the following measured properties: water level, ground water level, precipitation, precipitation forecast.

TNR-2.3	**Must**	When a risk is assessed, at least two different types of sensor data should be taken into account.
TNR-2.4	**Must**	Sensors should be calibrated when indicated by the system, or otherwise at least once every 12 months.
TNR-2.5	**Must**	Warnings that are issued by the system should be accurate in 90% of the cases. The warning is considered accurate when (1) a flood actually occurs, at any scale, in the designated area for the flood warning, or (2) a set of customers (or meteorologists) confirm that the predicted increase of flood risk was appropriate.
TNR-2.6	**Must**	A flood warning should occur at least an hour before the predicted situation occurs in the designated area for that flood warning.
TNR-2.7	**Must**	The system should be running 99.99% of the time on at least 90% of the sensor network, and 90% of the instances of the web portal.
TNR-2.8	**Must**	When the system detects an increase in flood risk for a particular area, warnings to all stakeholders which are either subscribed to or located in that area will have been sent within ten minutes.

3.9.3 Adaptability

TNR-3.1	**Must**	The installation and testing time of adding a new sensor as a node in the network should altogether take less than 20 minutes.
TNR-3.2	**Must**	The communication between components should happen with a mainstream, highly applicable and sustainable communication standard to guarantee to a high extent the support of new and existing measuring components.
TNR-3.3	**Must**	The time it takes to de-register and remove a sensor from the network is less than 10 minutes.
TNR-3.4	**Must**	The system should be designed in such a way that sensors which measure other environmental properties, such as wind speed, water current, dike pressure and consistency, water leakage, water pollution or extreme weather phenomena, can be added to the system.

3.9.4 Interoperability

TNR-4.1	**Optional**	The system should be able to exchange data related to flood risks with external parties that perform guidance activities for stakeholders.

TNR-4.2	**Must**	The system should be able to directly, within an interval of five minutes, inform emergency services of flood warnings with the highest priority, before any other stakeholder type is warned.

3.9.5 Integration

TNR-5.1	**Must**	The various sub-networks of the sensor network, controlled by lower-level hubs, are seamlessly integrated through wireless and wired networks.

3.9.6 Security

TNR-6.1	**Optional**	The communication between the components of the system should be authorized by means of a public key authentication system to ensure the authentication and integrity of sensor data.
TNR-6.2	**Optional**	The components themselves should be hidden from sight or protected with casing to avoid tampering with the components and thus the functionality or resulting data of that component.

3.10 Evolution Requirements

This section describes requirements that apply to the evolution of the system after its release in terms of extending its functionality or maintenance.

3.10.1 Dike sensors

Currently a system is being developed in order to measure water leakage in dikes. The research is showing good results so far in order to detect these leakages. This system can be used for the SFM system. It would be an autonomic part of the SFM system that sends its results to the central server, so it would not behave according to the polling mechanism that is present in other sensors, but it would be regarded as an external data provider instead.

3.10.2 Connected cars

The two largest mobile operating system providers both have plans to create solutions for so called 'Connected cars', Apple CarPlay and Google Android Auto. These solutions create cars that are connected to the Internet, which will bring a large amount of new functionality to these cars. For the SFM system these solutions can be used to give users in cars more information and possibly can add navigation points to the cars navigation, this way users can be guided more precisely to safer grounds in areas where flood warnings are in effect.

3.10.3 Mobile applications

The rising amount of smart phones has led to the point that a large fraction of the population now carries a smart phone with them at all times. Applications on these smart phones can give users the same insight as discussed in Section 3.10.2. However, because both systems work on mainly cellular and Wi-Fi networks, the chance of network failures in cases of bad weather will increase. Nevertheless, these techniques can support the information and guidance methods already discussed in this document.

3.11 Rerouting of vehicles

Many citizens use route planners to find their way to their destination. Working together with those companies, we could provider better guidance for citizens and emergency services, by guiding them through flooded areas through the inclusion of flood risk in the calculation of the fastest (and safest) route between two locations.

3.12 Risk Assessment

This section describes the risks affecting the system. They are categorised by impact and probability, ranging from low to high. The impact and probability are then used to calculate the severity of a risk, using the schema of table 10. The risks are categorised in Appendix A1.

Impact	**high**	medium	high	critical
	medium	low	medium	high
	low	very low	low	medium
		low	**medium**	**high**
	Probability			

Table 10: Severity calculation

4 Analysis

This section will cover the analysis of the architecture. The analysis will consist of three parts: assumptions, technology roadmap, and the high level design decisions.

4.1 Assumptions

Some assumptions need to be made about the environment in which the Smart Flood Monitoring system operates.

- Power and communication infrastructure is available for the hubs of the system.
- There is sufficient solar activity to power sensors with solar panels.
- It is possible to assign a SDI-12 sensor to multiple data loggers.
- The system can send messages via Cell Broadcasting via some kind of API to all Cellular providers in a given area.
- Security of communication is provided by the SDI-12 sensors and SDI-12 data loggers.
- There are hosting providers that can deliver a scalable network infrastructure and network hardware.

4.2 Technology roadmap

The first iteration of the system will be a minimum viable product. The most recent sensors will be used to measure various parameters. When processing the data machine learning will not be applied. The web-interface will be minimal. The first iteration should be finalised in Q1 2017.

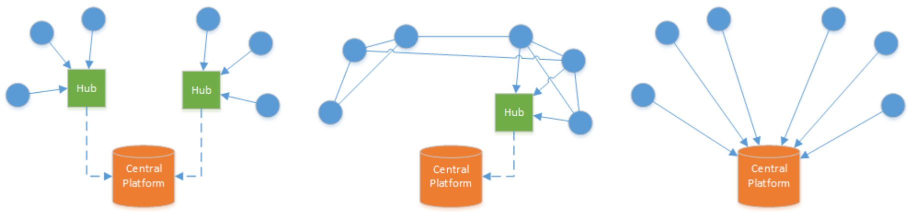
In the second iteration of the system the existing features will be further refined and more features will be added. Different sensors might be used depending on their costs, functionality, and efficiency. The processing of data will be optimised in this iteration as well as using machine learning to improve the analysis of the data. Furthermore the web-interface will be improved using feedback sent in by users up to that point and third parties and services will be integrated into the system. This version should be in use at the end of 2017.

During the third iteration the system will be further refined where necessary and might bring cost reductions by using better sensors and/or infrastructure. At this point the system could evolve to include the features discussed in section 3.10 (Evolution Requirements). Further iterations will continue until such point that the product becomes obsolete.

4.3 High level design decisions

Decision HD1	**Data storage on central platform**
Status	*Approved*
Problem/issue	Gathered data needs to be stored.
Decision	Data will be stored in a central database.
Alternatives	Local storage - Data could be stored locally on each sensor. Since we want to perform calculations with the data each sensor should support this functionality as well. This will raise the costs of the system. Mixed storage - Data could be stored partially on a central server and it could be stored locally. This would require that each sensor has storage and calculation functionality.
Arguments	The system needs to do a lot of calculations with data. This will require hardware that is able to do this. Performing broad analysis on the data would also require the data to be sent to a central processing server for each calculation, negatively impacting performance. Furthermore having a processing unit at each data logger would increase the costs.

<table>
<tr><th>Decision HD2</th><th>Hubbed sensor network</th></tr>
<tr><td>Status</td><td>Approved</td></tr>
<tr><td>Problem/issue</td><td>Communicating sensor data to central platform.</td></tr>
<tr><td>Decision</td><td>The system will use hubs to transfer data from multiple sensors to the central platform, where multiple sensors are connected to a single hub. See figure 5a.</td></tr>
<tr><td>Alternatives</td><td>Mesh network - The system could transfer sensor data through a network of sensors to an upload hub. See figure 5b.
Individual network - The system could allow each sensor to upload its data to the central platform. See figure 5c.



(a) Hubbed network (b) Mesh network (c) Individual network

Figure 5: Network types</td></tr>
<tr><td>Arguments</td><td>Equipping each sensor with their own networking equipment would be too expensive - either each sensor would need to be equipped with a networking cable, or use a cellular or satellite connection. A mesh network could be less expensive, but making it fault tolerant would require the sensors to be placed close together to allow each sensor to connect to multiple other sensors. Large networks could also overload the communication capabilities of sensors because of high traffic volume. This can affect both the fault tolerance and performance of the system. Using a hubbed network, the hubs would not be overloaded since only a certain number of sensors are attached to a single hub. Fault tolerance can still be achieved through the use of multiple hubs around a measurement location. Losing a small group of sensors then does not impact the system.</td></tr>
</table>

Decision HD3	Wireless and wired communication
Status	*Approved*
Problem/issue	Connecting sensors to hubs.
Decision	The system will support both wireless and wired communication to communicate data from sensors to hubs.
Alternatives	Wired only - The system could transfer sensor data through network cables from the sensor to the hub. Large distances would require very long cables. Wireless only - The system could transfer sensor data through wireless communication. This could be affected by interference and cases where sensors are located very close to the hubs a wireless connection is still required.
Arguments	Supporting both wireless and wired communication allows us to cover large distances through wireless communication, while cases where a sensor is placed next to a hub would only require a simple network cable.

Decision HD4	SDI-12 protocol
Status	*Approved*
Problem/issue	Communicating sensor data to hubs.
Decision	The system will use SDI-12 for communication, since it is supported by virtually all industrial sensors.
Alternatives	RS-232 - The sensors could use RS-232 serial communication to send data to hubs. Current - The sensors could use current to send data to hubs, which are then interpreted by the system. REST - The sensors could use a REST-based protocol to send data to hubs. This would in turn require interpretation of RS-232 or current data which would also require a processing unit at each sensor.
Arguments	RS-232 for communication is not always supported, where virtually all industrial sensors support the SDI-12 protocol. Sending the data through modulation of current across a wire would require custom interpretation and calibration, where SDI-12 type sensors do not require this [10]. SDI-12 type sensors therefore enhance the adaptability of the system.

Decision HD5	**Private cloud hosting**
Status	*Approved*
Problem/issue	A server is needed that can store and process data.
Decision	Services of a private cloud provider will be used to store and process the data.
Alternatives	Dedicated servers - The system could make use of dedicated servers Cloud servers - The system could use a cloud provider to buy instances on colocated servers
Arguments	Dedicated servers do not scale well and would require expensive high-performance servers per client. It is also more difficult to guarantee high availability. Cloud servers scale extremely well, however they are co-located with other customers of a cloud provider which can affect the performance of the system. A private cloud combines these two, by hosting the cloud network on isolated machines on an isolated network

Decision HD6	**Diverse power supplies for sensors**
Status	*Approved*
Problem/issue	The sensors need to be powered.
Decision	Sensors can be powered using either solar cells with a battery or cables that connect it to the electricity grid.
Alternatives	Using mains power - The sensors could be powered using long cables that attach it to mains electricity. Using a battery and solar cells -The sensors could be powered using a battery and solar cells which make the sensors completely wireless.
Arguments	Using mains power would require using long power cables to connect each sensors, which could be done in urban environments but is not always feasible in remote locations. At the same time using mains power can increase reliability and maintainability so using it in urban environments is more preferable. A hybrid approach is therefore the best option.

Decision HD7	**Manual monitoring in high-risk areas**
Status	*Approved*
Problem/issue	In situations where large groups of hubs are unavailable, areas with elevated flood risk need to be inspected to guarantee safety.
Decision	Workers will manually monitor areas with an elevated flood risk
Alternatives	Drones - The system could use drones to fly over areas with an elevated flood risk to provide visual inspection Remotely controlled boats - The system could use remotely controlled boats (floating drones) for visual inspection from the side of the water.
Arguments	Drones are extremely expensive and require a lot of certification. Remotely controlled boats are less expensive but require a large number in order to cover the whole sensor network. There is not always enough water for the boats to move on, and they can only see the area from the side of the water (e.g. not over dikes or other obstructions. Using workers requires some travel but they can perform up-close inspection.

5 System Architecture

In this section, a high-level and reverse engineered description of the SFM system will be given. This is accomplished by first describing the initial model for the system architecture and providing the alternatives that have been discussed for the initial model. The chosen initial model will be developed into an elaborated model with a detailed description and verification of the feasibility of this model.

5.1 Initial model

The initial model for the architecture is distilled from the high-level design decisions that are listed in the previous sections. These decisions are summarized in figure 6. In this figure, the actors and the global system components are shown and their logical connections and relationships.

5.1.1 Actors and roles

The following list shows the entities that communicate with our system and the kind of communication they have with the system:

Customers The customers of the system, of course, communicate with the system. This communication is mainly to create an overview of the system for the customer, which enables the customers to see what is actually happening with the system. This also includes that all the warnings that are sent via the system will be forwarded to the customers.

Emergency Services The communication between emergency services and our system is one-way communication, the emergency services only receive warnings from the Alert media component but do not have to provide any information to the system.

Citizens Since citizens are able to subscribe to certain area's to receive warnings even if they aren't present in that area, the citizens do have to send their information to the system so that the system can reach them. Furthermore, citizens can get access to less detailed information about the system to see statistics or flood risks. When the system detects a flood, messages are sent to all people in that area even when they aren't subscribed to messages.

Maintenance Team The maintenance team is supposed to solve problems with the sensor hubs and the sensors in combination with the central server. So the maintenance team is only interested in the status of sensors and if they are working correctly, this also includes the calibration of sensors.

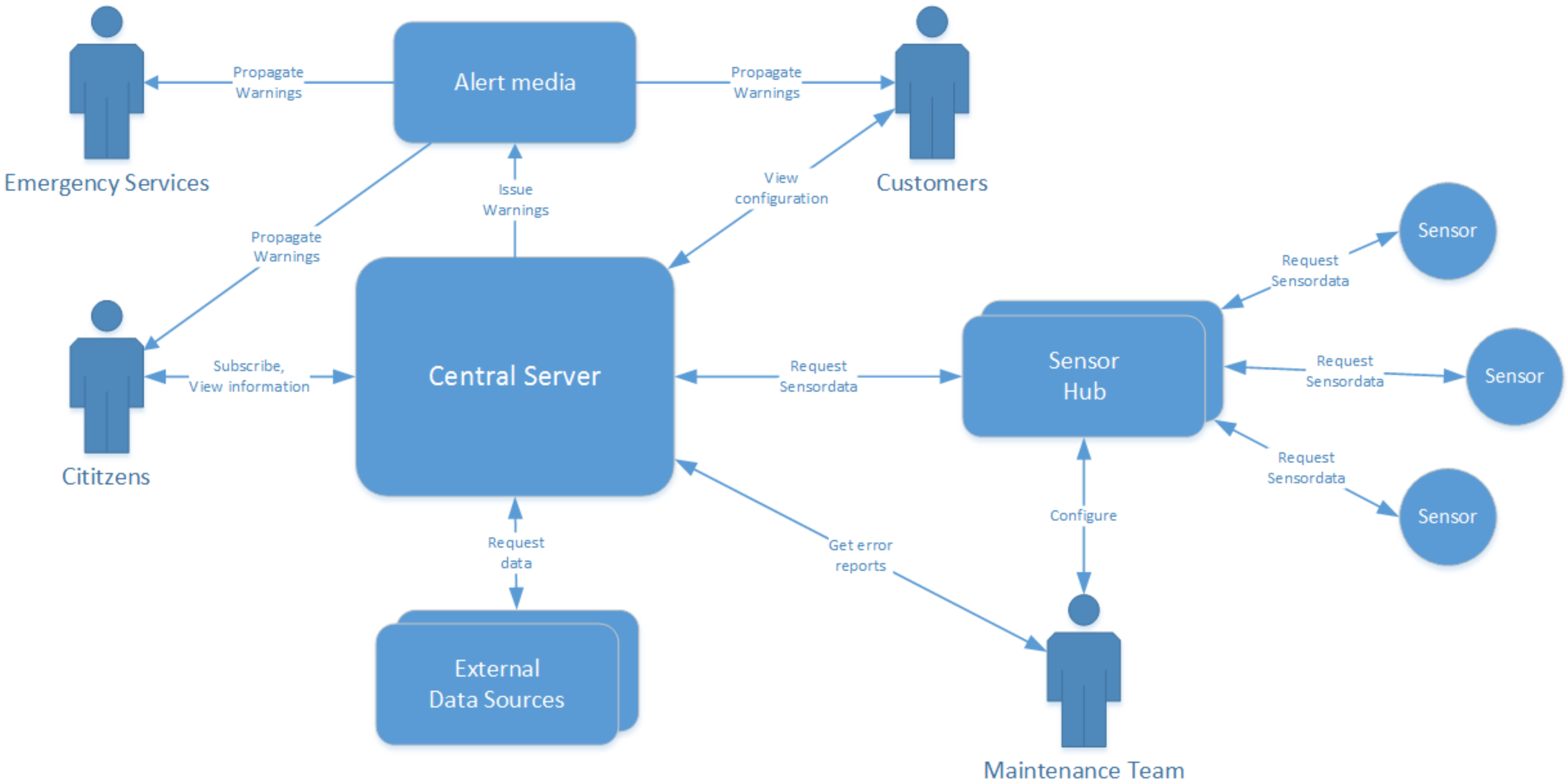


Figure 6: Initial Model

5.1.2 Components and external systems

The following list contains all of the components of the system, including systems and subsystems of external parties, that together fulfill the intended functionality of the system as defined in the requirements. The components are:

Central Server This component stores all of the sensor data, account and subscription information in a database. Furthermore, it provides a web service for actors to access the status of sensor; the sensor data itself; to access, modify or remove their subscriptions; and to log-in and log-out of their account. Once every so often, the central server will also perform the flood risk analysis for areas inside of the sensor network.

Alert Media The alert media component is the part of the system that propagates warnings to actors via diverse media, for instance cell broadcasting, email and SMS. The Central Server issues warnings to the Alert media component, this way the method of sending warnings to users can be adapted easily by changing only the Alert media component. This component is a system of external parties.

External Data Sources The external data sources provide the system with additional information on top of the sensor data. The Central Server will send a request for new data regularly to the external data sources to have up to date information.

Sensor Hub The Sensor Hubs will interact as intermediaries to propagate sensor data requests from the Central Server to the actual sensors, and also to create the communication channel between the sensors and the Central Server.

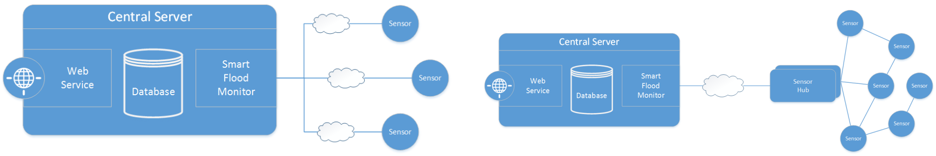
Sensor The sensor component speaks for itself, these are the actual sensors that gather the data. The sensors will be as simple as possible to be able to add more types of sensors easily and to reduce the maintenance of the sensors.

5.2 Alternatives

In the initial model some of the high level decisions are already visible, alternatives for these decisions and the other decisions will be discussed in this section.

5.2.1 Sensor network

The type of sensor network indicates the way sensors communicate with the system. In the initial model, we chose the hubbed sensor network type is chosen. There are some alternatives to that: an individual network and an mesh network. An individual network is a network where all the nodes communicate directly with one central server, which means that all of the nodes need communication hardware to communicate with this central server as seen in Figure 7a. For instance, a system where all the sensors have a cellular module that sends it's sensor data directly over the Internet to the central server will have an individual network type. An mesh network is quite similar to a hubbed sensor network, however in the mesh network type the nodes are able to communicate directly with each other instead of only communicating with a hub as seen in Figure 7b.

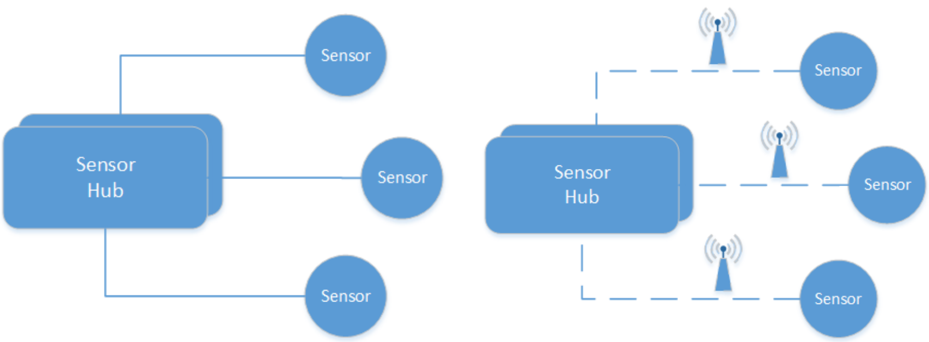


(a) Individual network (b) Mesh network

Figure 7: Sensor network alternatives

5.2.2 Sensor communication

The communication between sensors and hubs can be either wired or wireless. In the system a distinction can be made between a complete wireless system, a complete wired system and a hybrid system with both wired and wireless sensor communication. Both alternatives are shown in Figure 8. Wired connections have a great advantage in terms of reliability of the network, but such a wired network will cost a lot more than a wireless solution. Especially when sensors are placed on difficult terrains, laying a wired connection is sometimes nearly impossible. However, wireless connections hand in a great portion of the reliability, because wireless connections are much more susceptible to interference than wired connections. Wired and wireless connections can also be combined to have the advantages of both techniques, in cases the terrain is suitable for a cable network the communication can be wired. When the terrain isn't suitable for a cable network the sensors could be equipped with wireless technology to communicate with a nearby sensor hub.



(a) Wired network (b) Wireless network

Figure 8: Sensor communication alternatives

5.2.3 Sensor power supply

Most of the off the shelf sensors that are applicable to this project have relative low power demands, going up to a couple of watts. With such low energy needs the power supply isn't restricted to just the main power grid. There are solutions available with solar cells in combination with a battery to power sensors. Of course, using only batteries is also a good alternative, which will need regular maintenance to replace batteries. Both alternative solutions will need a lot more maintenance than using the standard power grid, which is much more reliable.

5.2.4 Data storage

There are several options on where to store the data gathered from the sensor: on a central server, on local storage of the sensor and a mixed storage. Storing data on a central server has the advantage that all of the data is available when algorithms need to be performed on a data set. Scale advantages reduce the cost of storing data at a central server, but storing data at a central server will increase the risk of losing all the data of all the sensors in case of a system failure. Local storage requires all sensors to have local data storage and requires more processing power of the sensors and will therefore raise the price of single sensors considerably. Communicating larger sets of data with a central processing unit will decrease the performance of the flood detection, because all the data needs to be sent each time. An combination could also be created, by storing data at the sensors themselves and at a central server, which tackles the point of information loss in case of a system failure at the central server. But this approach will cost a lot more than the other two alternatives.

5.2.5 Central Server implementation

A Central Server, as shown in the initial model, can be deployed as a pure central server were there is one large server in one location. However the Central Server can also be implemented in a cloud based way, in which there are a multitude of separate nodes all working together having a single point of entry. Such a cloud implementation can also be divided into two subgroups: a public cloud environment and a private cloud environment. In a public cloud environment the servers used in the cloud are shared over all the users of the cloud and most of the times customers do not have the ability to choose in which data-center their application runs, the upside of such cloud environment is that the application can scale up and down very easily. A private cloud is a software solution to enable all of the features of a public cloud but maintaining the privacy and security of having own physical servers. With a private cloud you hand in some of the scalability aspects but the system can distribute the work over multiple physical servers in different data-centers.

5.3 Elaborated model

The chosen initial model with the given components and communication channels will be developed into an elaborated model. This elaborated model is shown in Figure 9. The details of the elaborated model will now be described.

5.3.1 Central server

The central server consists of a web service (that provides the web portal as a service, as discussed in the requirements), a database, and the Smart Flood Monitor. The latter communicates with the sensor network, obtains the data of the sensors, and performs flood risk analyses on particular areas in the sensor network. Interaction with the web service is performed through HTTP and HTTPS using REST APIs. For issueing flood warnings, the system utilizes a message broker, Apache ActiveMQ, as an intermediate for delivering warnings to the various alert media.

5.3.2 Communication in sensor network

It has been established that the sensor network will be hubbed, where a sensor is connected to the central server via one or multiple network hubs. The communication between these hubs and the central server is done via Telnet interfaces. The communication between the hubs and the sensors themselves is performed both wired and wireless via the SDI-12 protocol.

5.3.3 Alerting users

The central server uses a message broker to distribute warnings to alert media. The system can send warnings to cell towers and email servers, which will respectively broadcast warnings via Cell-Broadcast, and broadcast warnings via emails to emergency services and other subscribed actors.

5.3.4 External data sources

The central server will connect to external data providers over HTTP (or HTTPS in case the data provider does not provide data to the general public). We assume that all of our external data providers are accessible through the Internet and through a HTTP/HTTPS interface that implements the REST methodology.

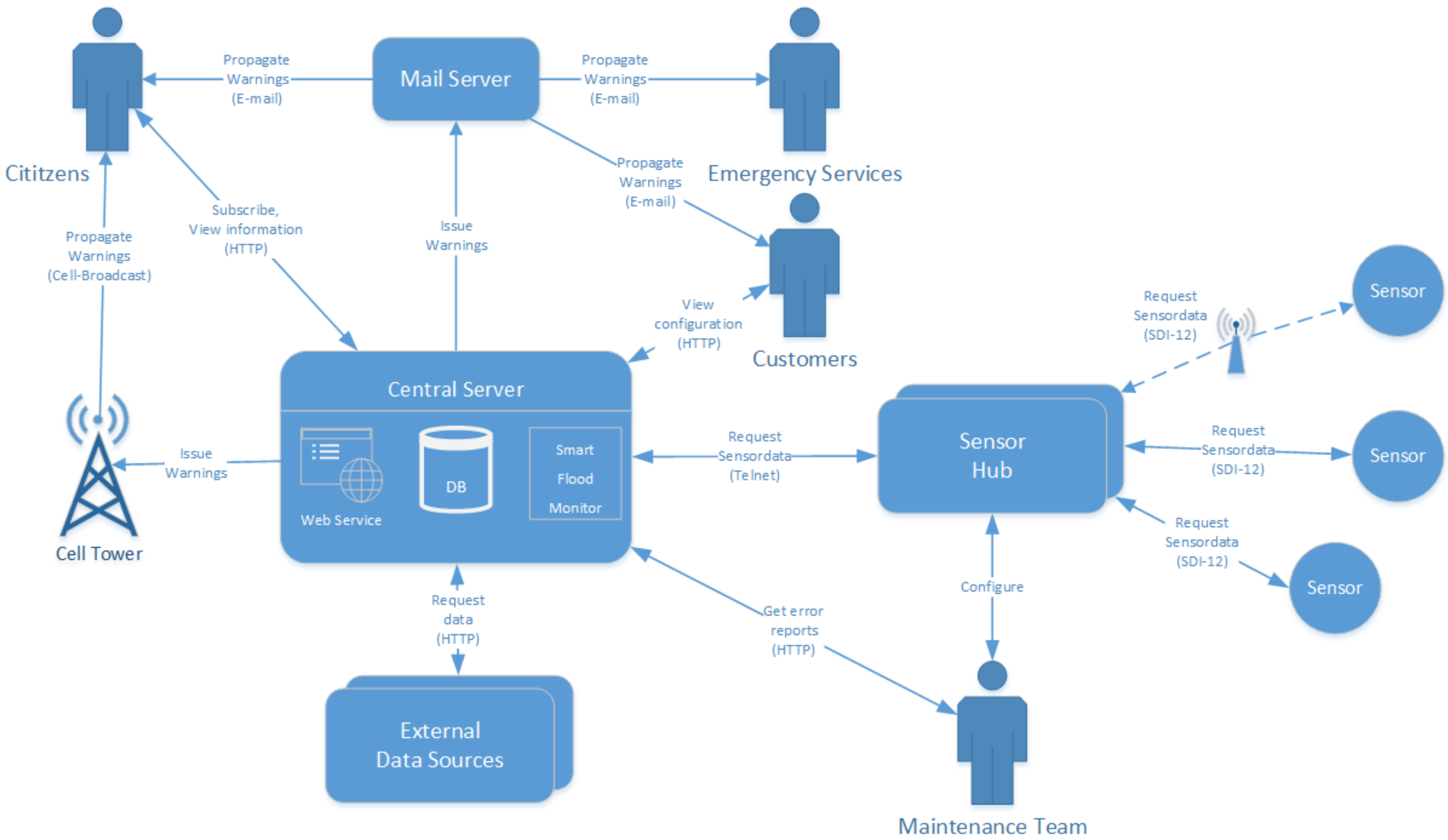


Figure 9: Elaborated Model

5.4 Verification

The feasibility verification of the system architecture is discussed in this section.

5.4.1 Availability

The availability of the system can be described by estimating the mean time to failure (MTTF) of components in the system, including communication errors, and the mean time to repair (MTTR) such component. The availability can then be described as the percentage of time the component has worked divided by that time including the repair time, so translated into a formula this gives: $\frac{\text{MTTF}}{\text{MTTF+MTTR}}$
The components with their estimated MTTF and MTTR values and given these values the availability is shown in table 11.

Component	**MTTF**	**MTTR**	**Availability**
Web Service	1 year	30 minutes	99.98859%
Database	1 year	30 minutes	99.98859%
Smart Flood Monitor	1 year	30 minutes	99.98859%
Sensor Hub	3 year	1 day	99.90876%
Sensor	3 year	12 hours	99.95436%
Mail server	1 year	30 minutes	99.98859%
Internet Connection	3 year	1 day	99.90876%
External Data Sources	3 year	1 day	99.90876%
Cellular Tower	1 year	6 hours	99.93155%

Table 11: System Architecture Availability

All the components in that are located in the private cloud are images that can be easily restarted and can be deployed redundantly, therefore the MTTR is estimated quite low with only 30 minutes repair time. Combining all the availability scores in table 11 we get a global availability of 99.56733%.

5.4.2 Time to Market

The SDI-12 protocol is a widely available protocol for outdoor sensors, which results in enough off the shelf sensors. This has a good impact on the time to market of the complete system. The central server will be implemented in a private cloud environment, this results in simpler and smaller images that are responsible for only one task. The advantage of this approach is that less time has to be spent on creating the individual components of the central server. The biggest part of the Central Server will be the Flood Detection Model, which is the component that has the effect on the time to market. However, the launch of the system is scheduled for Q1 2017, with the 1600 man-hours that stand for the Flood Detection Model that are roughly 150 man-hours per month which will not pose a threat to the scheduled launch.

5.4.3 Cost and Return on Investment

The cost of the product cannot be specified clearly, because the system is of a different scale for each customer. If we take the example of Poland, as described in Section 2.8.4, we see that the hardware costs are roughly €2,5 million. The software development costs are estimated to be €444 thousand, which isn't that much if we take into account that the software is created for more than one customer. The maintenance costs of the example system will be roughly €500 thousand per year, so for the expected life span of the system of 5 years this will result in €2,5 million. So the total costs of the system are: €5,5 million. With the profit margin of 20% the total price for the Polish government will be around €6,6 million. This is only 17 cent per citizen.
Since the product will not be distributed on a large scale, the return on investment is hard to determine. But with the profit of roughly €1 million for a customer on the scale of Poland, the investment on the software architecture and the maintenance of the global system will be covered by this profit.

6 Hardware Architecture

This section briefly outlines the hardware architecture. The section starts with the hardware design decisions. Subsequently, a description is provided of the hardware platform. The section ends with an overview of the application interfaces.

6.1 High-level architecture

There are several decisions to be made with regards to the hardware components. This section will outline the hardware design decisions.

Decision HW1	**H-500XL Data Logger**
Status	*Approved*
Problem/issue	Provide temporary storage for sensor data at hubs
Decision	The system will make use of the H-500XL Data Logger
Alternatives	H-522 - a data logger with embedded satellite communication. H-5000 - a data logger equipped with a touchscreen. Storm 3 - a data logger tailored for use with Waterlog's proprietary data hosting.
Arguments	The H-500XL data logger is the simplest of the alternatives. The H-222 provides embedded communication to satellites that are not available globally and are also not required for the hub network. The H-5000 is equipped with a touchscreen. However, since the data will be processed remotely such a feature is unnecessary. The Storm 3 is meant for use with Waterlog's Storm Central, which is again a feature not required in our system. Furthermore, the H-500XL offers near real-time data retrieval, which adds to performance and allows for early flood detection. Since the H-500XL is sufficient and does not include unnecessary features, this is the best choice for temporary data storage.

Decision HW2	Line-of-Sight Radio series (H-4400)
Status	*Approved*
Problem/issue	Support completely wireless communication between sensors and hubs in sensors areas
Decision	The system will make use of the the Line-of-Sight Radio Series (H-4400)
Alternatives	Sutron SDI-Link Wireless Bridge - a bridge that offers the same core functionality as the H-4400 series, but only has a range up to 1.6km line-of-sight (http://www.sutron.com/product/sdi-link-wireless-bridge.htm/).
Arguments	The Line-of-Sight Radio series (H-4400) bridges are capable of communicating via radio link or repeating radio receiver. The wireless bridge is useful where one or more SDI-12 sensors must be physically located up to 8km from the data logger. The radio link is compatible with any data logger that uses transparent SDI-12 compliant communication. It can seamlessly replace parts of an existing system that is based on wired communication. Furthermore, each remote site can have multiple SDI-12 sensors. Two H-4400 series components are required to incorporate the bridging functionality; a *master* component is connected to a RS-232 serial port on the data logger (instead of the SDI-12 port). One or more *slave* units are located at the remote sensor sites. 7-bit even parity is used as a simple form of error detecting code.

Decision HW3	SDI-12 Rain Gauge (H-3401)
Status	*Approved*
Problem/issue	The amount of rainfall needs to be measured, while maintaining accuracy at increased rates of rainfall
Decision	The system will make use of the SDI-12 Tipping Bucket Rain Gauge (H-3401)
Alternatives	T-200B Precipitation Gauge - A precipitation meter for all weather conditions. Rain gauge WTB100 - Rain Gauge with bounce-free reed contact.
Arguments	The Tipping Bucket Rain Gauge (H-3401) is the only sensor that is equipped with an SDI-12 output on default. Besides, it has a magnetic reed bucket tip sensor and an internal leveling mechanism that allows it to provide the highest accuracy and validity at increased rates of rainfall ranging from 0-25 inches (635 mm) per hour. The rain gauge also offers two removable stainless steel funnel screens and a rust-proof enclosure, which help to prevent faults/failures (e.g. due to heavy weather conditions) from occurring. Furthermore, it has a built-in microprocessor that automatically corrects errors and it operates on low power.

Decision HW4	Bubbler / Pressure Sensor (H-3553T)
Status	*To be reviewed*
Problem/issue	Accurately monitor river, lake and well water levels
Decision	The system will make use of the contact water level Bubbler + Pressure Sensor (H-3553T)
Alternatives	H-3551T Bubbler - Constant Flow Bubbler for water level monitoring. Sutron Compact Constant Flow (CF) Bubbler - A bubbler suitable for water level monitoring.
Arguments	The H-3553T operates in depths up to 35m, almost double that of the Sutron CF Bubbler. With a max purge pressure of 90 pounds per square inch (PSI), the model has the highest pure pressure in the industry (e.g. the H-3551T and Sutron variants only allow for 50 PSI). Unlike the H-3551T Bubbler, it has a built-in calibrated pressure sensor that guarantees high accuracy, such that no external pressure sensor is needed. Moreover, with respect to adaptability of the system, it is compatible with other manufactures' pressure sensors and data loggers. This summation shows that the H-3553T is capable of providing the highest performance. Accordingly, we decide to choose for the H-3553T for water level monitoring.)

Decision HW5	**Nile 502/504/517**
Status	*To be reviewed*
Problem/issue	Accurately monitor river, lake and well water levels in extreme environmental conditions
Decision	The system will make use of the non-contact water level sensors of the Nile radar series
Alternatives	Bubbler / Pressure Sensor (H-3553T) - A bubbler requires direct contact with the water to measure the level, by measuring the pressure required to pas air through a tube in the water. Radar Series (H-3611/12/13): Predecessor of the Nile radar series.
Arguments	The design was built for extreme environmental conditions, making this series suitable for rough sites or high bridges. It outperforms its predecessor (H-3611/12/13) with a measurement range up to 70m (Nile 517) with high accuracy (2mm). It uses continuous, non-contact transmission, through which it is able to prevent fault/failures due to heavy environmental conditions. The interface and SDI-12 and RS-232 communication ensure seamless integration with other system components. As a result, the Nile series will be deployed for water level monitoring in extreme environmental conditions and tough-to-reach areas.

Decision HW6	**H-3123 Submersible Pressure Transducer to measure groundwater level**
Status	*Approved*
Problem/issue	Accurately measure ground water levels
Decision	The system will make use of the Waterlog H-3123 Submersible Pressure Transducer placed inside a tube in the ground. It has a statistical accuracy of $\pm 0.02\%$ of Full Scale Output (FSO).
Alternatives	MEAS KPSI 353 submersible hydrostatic level transducer - A system similar to that of the H-3123, with a statistical accuracy of $\pm 0.10\%$ FSO. AquiStar SDI-12 submersible pressure transmitter - A similar system with a statistical accuracy of $\pm 0.06\%$ FSO.
Arguments	The Waterlog pressure transducer is more accurate than the MEAS and AquiStar products, which makes it the best choice for our system.

Decision HW7	H-211 VDC Rechargeable Battery series
Status	*Approved*
Problem/issue	The data acquisition components (e.g. sensors) need a reliable power supply to function
Decision	The system will make use of the H-211 VDC Rechargeable Battery series
Alternatives	UB1270 12 Volt X Ah Rechargeable Battery - Rechargeable sealed lead acid battery. ExpertPower EXP12180 12 Volt X Ah - Most trusted and highest reviewed sealed lead acid batteries on Amazon. Power Patrol X Ah 12 Volt - Rechargeable sealed lead acid battery. A large number of other suitable rechargeable batteries exist.
Arguments	The H-211 series of sealed rechargeable battery can be recharged with a solar panel or VDC charger. Their voltage and electrical charge (Ampere-hour) most closely match with the required charge of all Waterlog products. Since the data collection components require low power, it enables the use of low cost 12 VDC power supplies as the primary energy source.

Decision HW8	H-209 VDC Battery Charger series
Status	*Approved*
Problem/issue	Provide sufficient charging of batteries of components that are close to AC electrical sources
Decision	The system will make use of the H-209 VDC Battery Charger series
Alternatives	NOCO Genius 12v 4 Amp Marine On-Board Battery Charger GENMini1 - A 4 Amp compact and waterproof charger for 12V lead acid batteries. Battery Tender 12v 800 mA Waterproof Marine Smart Charger - A 800 mA compact and waterproof charger for 12V lead acid batteries. A large number of other suitable battery chargers exist.
Arguments	The H-209 series AC plug-in provides a continuous charge to a VDC rechargeable battery. As these series are specifically designed to match with the H-211 battery series they exist in both 12 Volt 800 mA and 12 Volt 4 Amp editions. Accordingly, these are determined to be the best solution for the system.

Decision HW9 - Optional	H-227 Regulated Solar Panel series
Status	*Approved*
Problem/issue	Provide sufficient charging of batteries of components in remote locations, far from AC electrical sources
Decision	The system will make use of the H-227-X Regulated Solar Panel series
Alternatives	Campbell Scientific SP series Solar Panels - Solar panels in the range of 5-100 watt.
Arguments	The H-227 Regulated Solar Panel series on-board voltage regulator provides direct connection to an external battery. Charging power of the H-227-10 is sufficient for charging batteries in tropical to temperate latitudes, while that of the H-227-20 is typically sufficient for charging batteries in high elevation and latitudes. Hence, these solar panels seem highly adequate for the system.

Decision HW10	H-2191 Telephone Modem
Status	*To be reviewed*
Problem/issue	Provide communication between hubs and the central server
Decision	The H-2191 Telephone Modem will be used to connect hub data to the central server using landlines
Alternatives	H-2192 GSM Modem - The H-2192 could be used to connect hubs to the server using GPRS
Arguments	GPRS connections might not be very reliable during bad weather conditions. One should also consider a scenario of panic, where cellular networks are overloaded. The H-2191 landline modem is then the best choice to satisfy the fault tolerance key driver.

Decision HW11	**Minimize the loss of data gathered by the sensors**
Status	*To be reviewed*
Problem/issue	The loss of data gathered by the sensors should be minimized in order to increase the fault tolerance of the SFM system.
Decision	A master node will share a subset of its sensors(slaves) with another master node.
Alternatives	Backup sensors - for each sensor there will be a backup sensor that is on standby. This sensor takes over when the primary sensor stops working.
Arguments	Using the backup sensors will double the amount of sensors used in the SFM system. Furthermore it would force us to use a watchdog timer [27]. This will increase the costs greatly. Not only the sensor costs are doubled and watchdog timers need to be purchased, there need to be double the power sources as well. The other option is cheaper, but is less fault tolerant. However it is unlikely that multiple sensors will fail simultaneously.

6.2 Description of hardware platform

Measurement sites are covered in hubs which have a communication radius of 8km add spec ref here. To increase fault tolerance we will only place sensors in a radius of 6km. The hubs are connected wireless to multiple sensor sites, which in turn contain one or more sensors connected to a single communication antenna. The communication between the hub and sensors can also be done via a wire, in which case the sensors are powered by the power supply of the hub. In the case of wireless communication, the sensor sites are powered by a battery which can be charged by either solar cells or mains power, depending on the availability of mains power at the sensor site. Hubs are powered directly by mains power, since they can be placed in more convenient locations than the sensors. The central server will be hosted in a private cloud which is to be selected based on the country that the system operates in. Each country thus has their own central server. An overview of the hardware deployment can be seen in figure 10. The blue nodes in figure 10 are part of the sensor sites, green nodes are the hubs and the orange node is the central server.

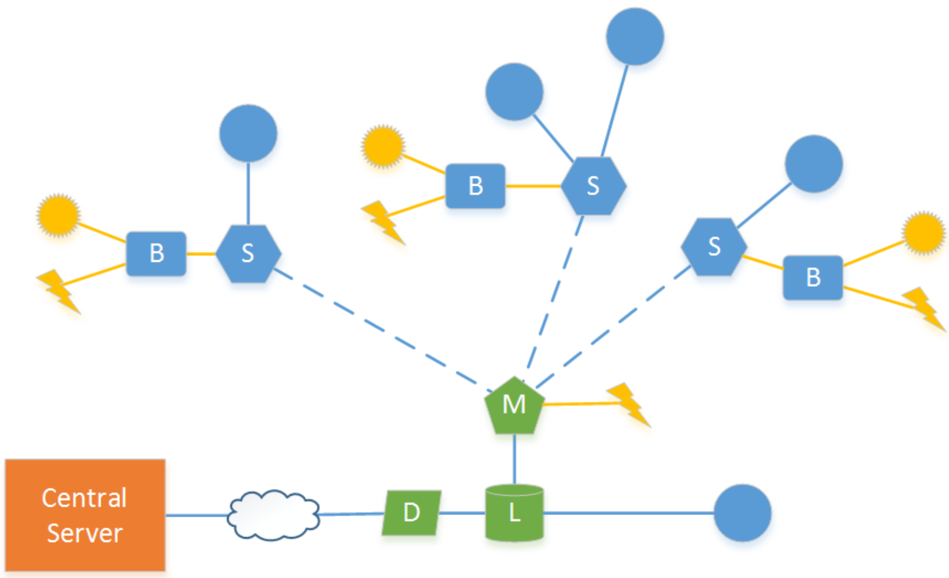
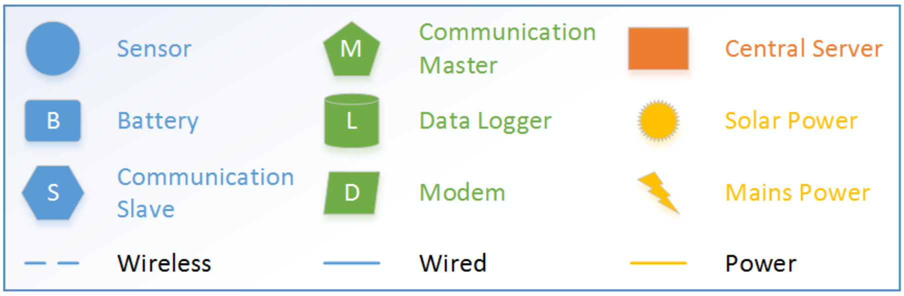


Figure 10: Overview of the hardware architecture.



6.3 System interfaces

Communication between sensor and hub can be done in two ways. In the wired version, sensors connect to, and are powered by, the H-4400 communication slaves, using the SDI-12 protocol. These are connected using a proprietary wireless protocol. The H-4400 communication masters then use the SDI-12 protocol to store data in the data logger. In the wired version, the sensors communicate with the data loggers directly, and are also powered by them. In both cases, the H-2191 modem uses an RS-232 connection to read data from the data logger and sends it to the central server over the Internet. See table 12 for an overview of these connection interfaces.

Version	Connection	From	To
Wired	SDI-12	Sensor	H-500XL data logger
Wireless	SDI-12	Sensor	H-4400 slave
Wireless	SDI-12	H-4400 master	H-500XL data logger
Wireless	Proprietary RF	H-4400 slave	H-4400 master
Both	RS-232	H-500XL data logger	H-2191 modem
Both	Telnet	H-2191 modem	Central server

Table 12: System connection interfaces

6.4 Hardware overview

This section will outline the specifications of the hardware components that are used. See Section 6.1 for the decisions regarding these components.

H-500XL Data Logger

The H-500XL Data Logger collects and stores data from sensors that use SDI-12 interface for communication. It is depicted in figure 11. Source of the specification details and image of the H-500XL Data Logger can be found at [13].

Figure 11: H-500XL Data Logger

H-500XL Data Logger	**Specifications**
Accuracy	±0.5% FSO (Full scale output)
Dimensions	180 x 220 x 110 mm
Weight	2kg
Operating temperature	-40 ° to +60 °C
Storage temperature	-40 ° to +80 °C
Humidity	100% non-condensing
Voltage input	10.0 to 16.0 V
Current	Standby: 5 mA Max Active:
Storage	256MB Non-volatile FLASH

Line-of-Sight Radio series (H-4400)

The Line-of-Sight Radio series supports wireless communication between sensors and hubs in sensor areas via a radio link or repeating radio receiver. It is depicted in figure 12. Source of the specification details and image can be found at [14].

Figure 12: Line-of-Sight Radio series (H-4400)

Line-of-Sight Radio series (H-4400)	**Specifications**
Input range	902.5 to 927.5 MHz
Line of sight	5+miles (depending on site and antenna conditions)
Dimensions	8 x 6 x 4 inch
Operating temperature	-40 ° to +60 °C
Storage temperature	-50 ° to +70 °C
Humidity	0 – 100% non-condensing
Voltage input	9.6 to 18 V DC
Current	less than 2.0 mA on average

SDI-12 Rain Gauge (H-3401)

The SDI-12 Rain Gauge measures rainfall accurately. It uses the SDI-12 protocol to communicate with external components. The SDI-12 Rain Gauge is depicted in figure 13. Source of the specification details and image can be found at [15].

Figure 13: SDI-12 Rain Gauge (H-3401)

SDI-12 Rain Gauge (H-3401)	**Specifications**
Accuracy	$\pm 3\%, 0 - 25$ in. per hr (0-635 mm/hr) MHz
Funnel aperture	200 mm
Weight	3.7kg
Operating temperature	-40 ° to +60 °C
Storage temperature	-50 ° to +70 °C
Mechanical operating temperature	0 ° to +50 °C
Voltage input	10.0 to 16 V DC
Current	raining: 2.8 mA for 60 seconds communication: 6 mA

Bubbler / Pressure Sensor (H-3553T)

The Bubbler/Pressure Sensor (H-3553T) monitors river, lake and well water levels accurately. This sensor is shown in figure 14. Source of the specifications and image of the H-3553T sensor can be found at [16].

Figure 14: Bubbler / Pressure Sensor (H-3553T)

Bubbler / Pressure Sensor (H-3553T)	**Specifications**
Pressure hysteresis	less than 0.01% of FSO
Dimensions	266.7 x 228.6 x 160 mm
Weight	4.3kg
Operating temperature	-40 ° to +60 °C
Storage temperature	-40 ° to +80 °C
Compensated range	-40 ° to +60 °C
Voltage input	10.0 to 16.5 V DC
Current	based on a 60 bubbles/min flow rate with a 15 min measurement cycle and a 1 purge/day frequency. Average active current 20 mA

Nile 502/504/517

The Nile 502/504/517 accurately monitors river, lake and well water levels in extreme environmental conditions. The Nile 517 is depicted in figure 15. Source of the specification details and image can be found at [17].

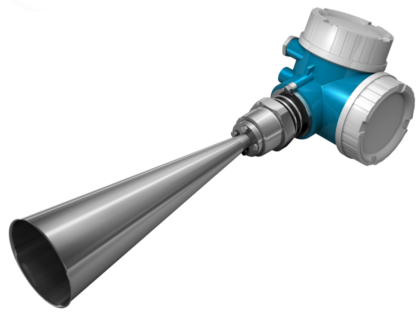


Figure 15: Nile 517

Niles 502/504/517	**Specifications**
Accuracy	±4 mm
Range	20 m
Dimensions (housing)	168 x 144 mm
Dimensions (horn)	Nile 502: 80 mm, 115 x 137.9 mm Nile 504: 100 mm, 135 x 150.5 mm Nile 517: 100 mm, 95 x 287 mm
Weight	Nile 502, 504: 2.7kg Nile 517: 4.2kg
Operating temperature	−40 ° to +80 °C
Storage temperature	−40 ° to +80 °C
Voltage input	10 to 16 V DC
Current	active: 13.5 mA

H-3123 Submersible Pressure Transducer

The H-3123 is designed to measure ground and surface water pressure, temperature and levels. The H-3123 is shown in figure 16. Source of the specifications and image of the H-3123 can be found at [18].

Figure 16: H-3123 Submersible Pressure Transducer

H-3123 Submersible Pressure Transducer	**Specifications**
Pressure hysteresis	less than or equals to 0.02% of FSO
Dimensions	22.2 x 165 mm
Cable length	max 304.8m
Operating temperature	0 ° to +40 °C
Storage temperature	−10 ° to +75 °C
Voltage input	9.6 to 16 V DC
Current	standby: 1 mA max active: 15 mA max

H-2191 Telephone modem

The H-2191 Telephone modem will provide the communication between hubs and the central server. Figure 17 shows such a telephone modem. Source of the specifications and image of the H-2191 can be found at [19].

Figure 17: H-2191 Telephone modem

H-2191 Telephone modem	Specifications
Type	GPRS, quad band, 14,400 bps
Frequency	850/900/1800/1900 MHz
Dimensions	4.0 x 6.5 x 1.5 inch
Operating temperature	-40 ° to +60 °C
Storage temperature	-40 ° to +60 °C
Humidity	0 - 95% non-condensing
Voltage input	9 to 16 V DC
Current	sleep, all subsystems off: 900 uA standby, telephone: 900 uA standby, GPRS active: 25 mA transmitting, GPRS: 120 mA max standby, radio: 95 mA

6.5 Server specifications

The SFM system will make use of a private cloud provider. Since the private cloud provider will be decided per country, we need to define some minimal specifications for the server in order to let the SFM system run smoothly.

Processing instance	**Specifications**
Operating system	Debian, CentOS, OpenSUSE, CoreOS, FreeBSD, SELinux, or other provided OS
Instance type	vCPU's: 16, RAM: 104 GB
Local SSD	375 GB
Data server location	Variable per country
Average hours up-time per day	24
Average days up-time per week	7

Web instance	**Specifications**
Operating system	Debian, CentOS, OpenSUSE, CoreOS, FreeBSD, SELinux, or other provided OS
Instance type	vCPU's: 4, RAM: 15 GB
Data server location	Variable per country
Load balancing support	Yes
Average hours up-time per day	24
Average days up-time per week	7

6.6 Database specifications

The SFM will make use of Cassandra and MongoDB, as described in Section 7. In order to maximize the performance there needs to be at least 3 instances of each database type. The minimum specifications are listed below.

Cassandra	Specifications
Operating system	Debian, CentOS, OpenSUSE, CoreOS, FreeBSD, SELinux, or other provided OS
Instance type	3x - vCPU's: 4, RAM: 26 GB
Persistent disk	3x - 1 TB
Data location	Variable per country
Average hours up-time per day	24
Average days up-time per week	7

MongoDB	Specifications
Operating system	Debian, CentOS, OpenSUSE, CoreOS, FreeBSD, SELinux, or other provided OS
Instance type	2 x vCPU's: 4, RAM: 26 GB 1 x vCPU's: 1, RAM: 0.60 GB
Database location	Variable per country
Average hours up-time per day	24
Average days up-time per week	7

7 Software Architecture

This section outlines the software architecture of the system. Initially, Section 7.1 lists the architectural significant drivers. Subsequently, in Section 7.2 a number of architectural views are presented using the 4 + 1 view model [12]. These views include the logical, development, process and physical view. The "+ 1" of 4 + 1 is the scenarios view. The use case scenarios, described in Section 3.5, serve as the content of this view. Section Section 7.6 provides an enumeration of the most important decisions with respect to the software architecture.

7.1 Architectural Significant Drivers

Table 13 is a list of requirements that are architecturally significant and are related to the key drivers, in other words influence the design of the system and software architecture.

Type	Requirements
FR	FR-1.1, FR-1.2, FR-1.3, FR-2.1, FR-2.2,FR-2.3, FR-2.4, FR-3.2, FR-3.3, FR-3.4, FR-3.5, FR-4.1, FR-4.3, FR-4.4, FR-4.5, FR-5.2, FR-6.1, FR-7.1, FR-7.2, FR-7.3, FR-7.4, FR-7.5, FR-7.6, FR-7.7, FR-8.1
TR	TR-1.2, TR-2.1, TR-2.2, TR-2.3, TR-3.4, TR-4.1, TR-6.2, TR-6.3, TR-7.4, TR-7.6
CNR	CNR-4
TNR	TNR-1.1, TNR-1.2, TNR-1.3, TNR-1.4, TNR-2.2, TNR-2.3, TNR-2.4, TNR-3.2, TNR-3.4, TNR-4.2

Table 13: Table of architecturally significant drivers.

7.2 Logical view

The logical view allocates the functionality that the system provides to end-users. Our system is split into multiple subsystems which are described below. Afterwards the internal system dependencies are described.

7.2.1 Subsystems

The software architecture of the SFM system can be divided into multiple subsystems. The multiple subsystems are depicted in Figure 18. As seen in Figure 18 the architecture will consist of subsystems that handle the sensor communication, external communication, storage of data, flood prediction, monitoring and finally the web portal.

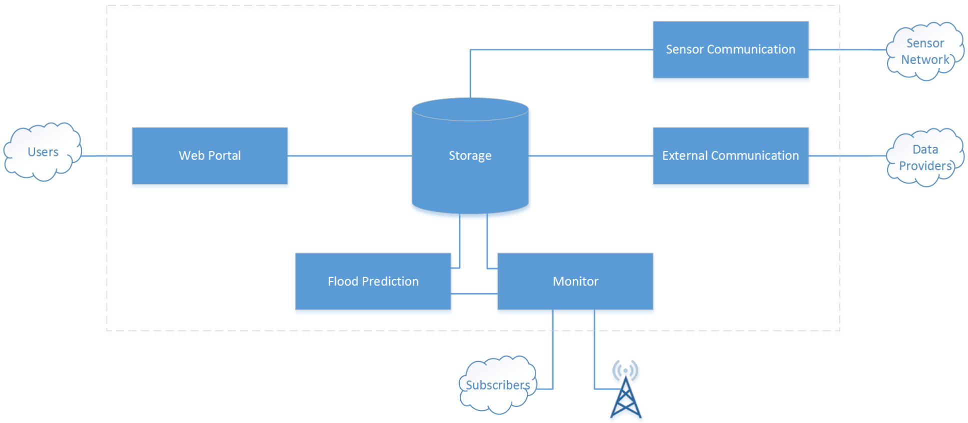


Figure 18: Subsystems overview

Figure 18 also shows a dashed rectangle. This rectangle represents the scope of our software architecture. The clouds and radio tower represent external network/clients with which the system will communicate. A more detailed decomposition can be seen below in Figure 19, which also shows the use of ActiveMQ for communication as well as MongoDB and Cassandra for storage.

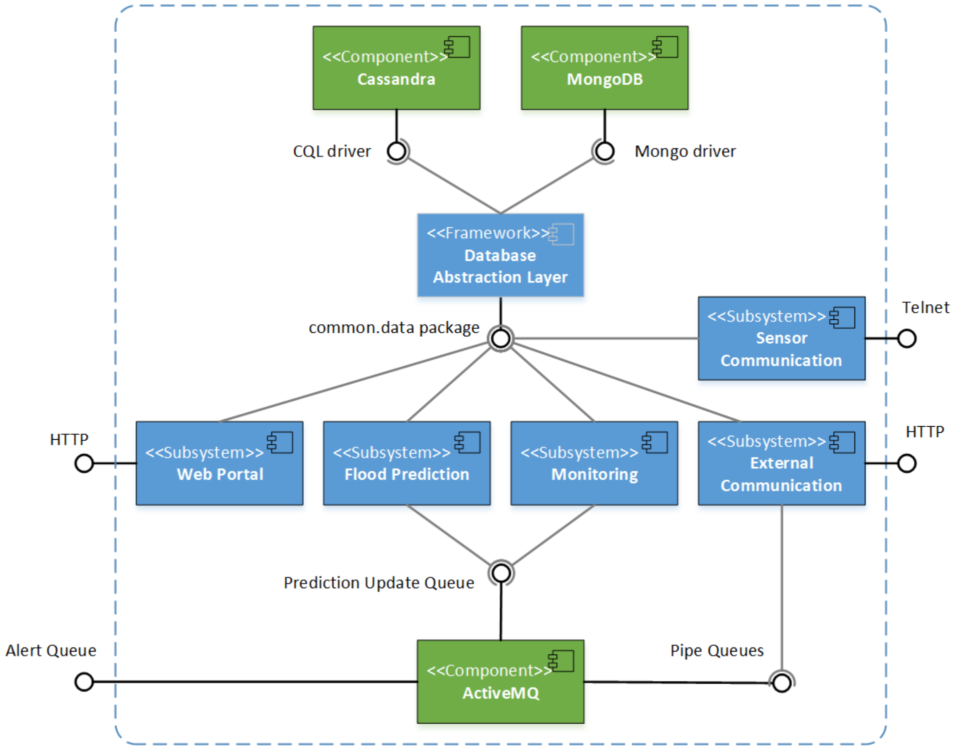


Figure 19: Component diagram

Each of the subsystems will be explained in detail below using package and class diagrams. A system-wide class diagram can be seen in appendix A2.

Sensor Communication

Figure 20 outlines the structure of the subsystem that takes care of the communication with the sensor network. The subsystem will have three components: communication, application and data. The *Layers Pattern* [20] is applied here in order to improve the adaptability of the SFM system. In the Layers Pattern, layers are only allowed to communicate with layers directly above or below themselves. For example, the Communication layer is not allowed to communicate to the Data layer directly.

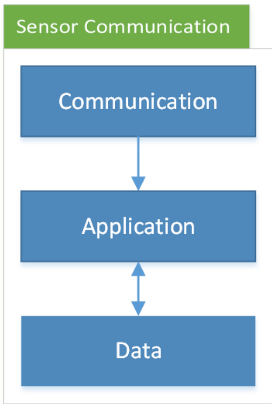


Figure 20: Sensor Communication Package Diagram

Each of the components are explained in more detail in Figure 21. The communication component will communicate with the sensor network. For this component the *Broker Pattern* [21] will be used. In the Broker Pattern, communication with multiple sources is hidden behind a broker that manages the separate sources. This abstraction increases adaptability. In our case, the *CommBroker* will have several *CommThreads* and functions to handle the communication with the sensor network. A *CommThread* will communicate with a sensor hub and request data continuously for a given interval. In order to communicate with a sensor hub the *CommThread* will need to communicate using Telnet.

The application component consists of two parts: a *DataHandler* and a *ResultFormatter*. The *DataHandler* stores the data provided by the *CommBroker* and formatted by the *ResultFormatter* in the Cassandra database. To do so the *DataHandler* will make use of the database abstraction layer *nl.sfm.common.data*, that provides functionality to communicate with the database. The *ResultFormatter* uses historic data from the sensor and data from nearby sensors to check if the data is valid, in order to detect if a sensor has failed. The *DataHandler* also handles timeouts from the *CommThreads* and marks a sensor as offline if it has not responded within an appropriate time. In both cases, communication with the sensor is halted until the maintenance team investigates the issue.

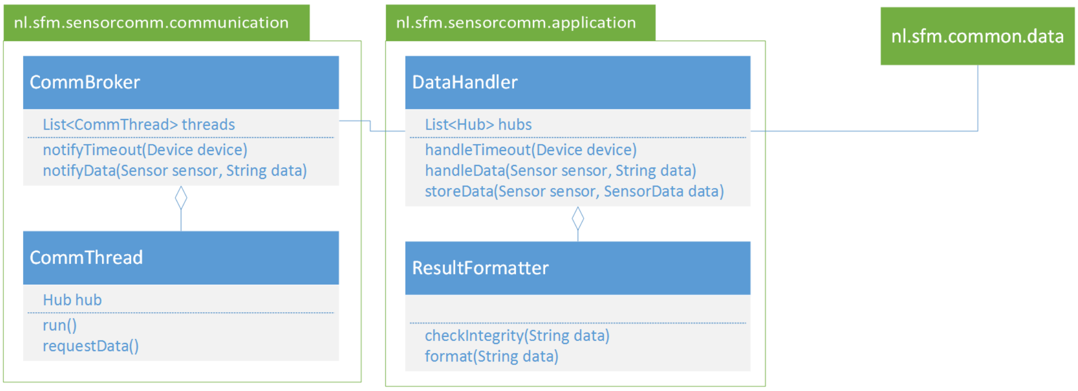


Figure 21: Sensor Communication Class Diagram

External Communication

The package diagram of the subsystem *External Communication* is shown in Figure 23. This subsystem takes care of the communication with external data sources. Similar to the subsystem *Sensor Communication*, the subsystem *External Communication* will have three components: communication, application and data. Again the *Layers Pattern* is applied here to increase the adaptability. The processing of data is separated into different steps, where some are generic and some differ for each data provider. With the *Pipes and Filters pattern* [25], the different steps are separated into discrete tasks called filters, which are then connected through pipes. The pattern is described in Figure 22. In our case, the filters are tasks running in worker processes, and the pipes are ActiveMQ queues. This allows for complete separation between the different tasks (since they are only connected through pipes), reuse of shared tasks (connecting a worker to a queue allows generic tasks to communicate with provider-specific tasks), scalability (multiple tasks can subscribe to the same queue) as well as easy replacement and modification of tasks (since new versions need only to connect to the right queues).

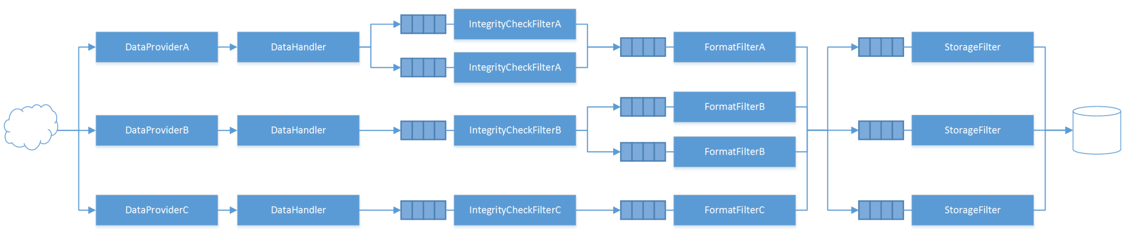


Figure 22: Pipes and Filters pattern in the External Communication subsystem

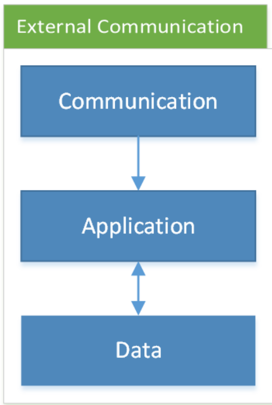


Figure 23: External Communication Package Diagram

The class diagram of the *External Communication* is shown in Figure 24. An implementation of the *DataProvider* interface is made for each data provider, which handles the communication with that provider. The raw data is sent to the *DataHandler*, which servers as the entry point to the Pipes and Filters pattern. It passes the raw data through a queue to a *IntegrityCheckFilter* implementation for that specific data provider. This filter checks to see if the data has not been corrupted. Afterwards it is sent to the *FormatFilter* which transforms the data into a format appropriate for the Cassandra datamodel. The last filter is the generic *StorageFilter*, which takes the generalized data and inserts it into the database using the *nl.sfm.common.data* database abstraction layer.

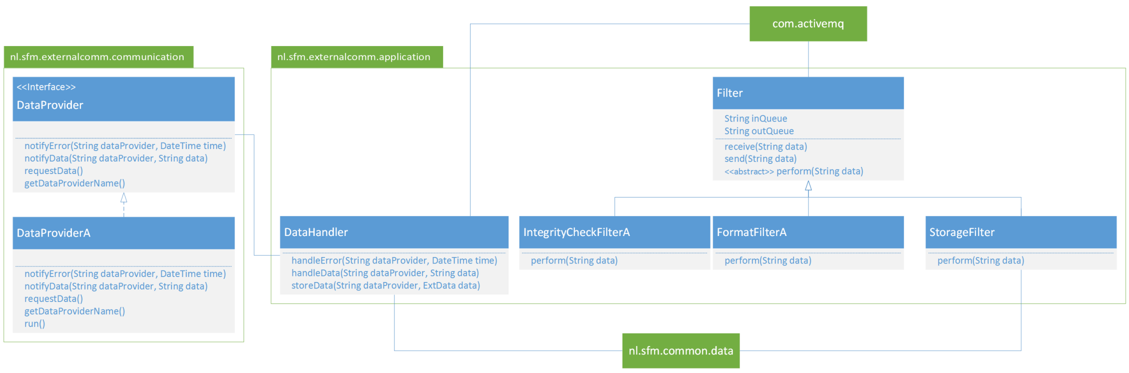


Figure 24: External Communication Class Diagram

Monitoring

Figure 25 shows the package diagram of the *Monitoring* subsystem. This subsystem is composed of a communication, application and data component. Figure 26 provides a graphical depiction of the monitoring class diagram. The monitor component contains the *FloodModelObserver*, which is responsible for monitoring the output of the flood model and it needs to take appropriate action when required. *UpdateConsumer* receives the results of each run from the flood model (i.e. the "update") and then checks whether a message should be sent. If so, for instance because the flood model calculation passes a threshold level for a sensor area, then *"AlertProducer"* will call the method *sendAlert* to alert relevant third parties (e.g. emergency services). Information regarding the transmission of the alert is then stored in the database. ActiveMQ was decided to serve as a message broker to make communication with the flood model and external parties effortless.

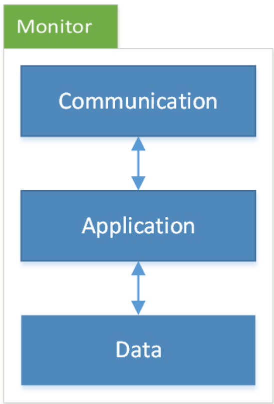


Figure 25: Monitoring Package Diagram

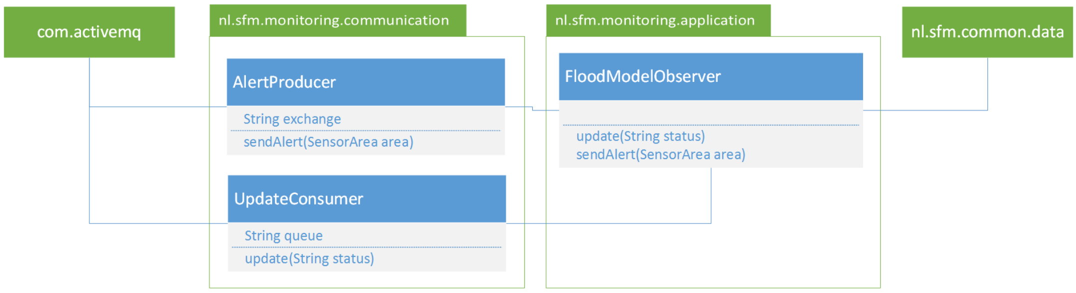


Figure 26: Monitoring Class Diagram

Flood Prediction
Figure 27 shows the class diagram of the subsystem *Flood Prediction*. This subsystem predicts the likelihood of floods. The *Strategy Pattern* [22] is applied in this subsystem. Using the Strategy Pattern, the inputs and outputs of the prediction remain the same, but a new prediction algorithm can more easily be selected. This will increase the adaptability of the system, because it is easy to add, swap and remove prediction strategies without affecting the functionality of the subsystem.

In order to predict floods the SFM system will use supervised machine learning. See Decision SW7. Supervised learning is the machine learning task to learn a general rule that maps inputs to outputs. In this case the inputs are the data gathered by the sensors and provided by the external data sources and the outputs are the flood warnings.

This subsystem consists of three components: prediction strategies, flood model and data.

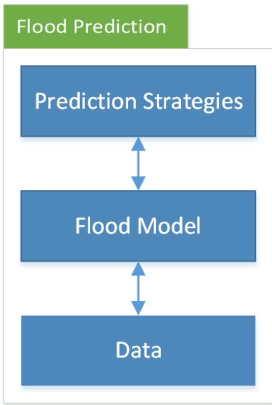


Figure 27: Flood Prediction Package Diagram

The *Flood Model* periodically loads data from the sensors and external data providers using the *nl.sfm.common.data* database abstraction layer. It then uses one of the *Prediction Strategies* to calculate a heat map, taking into account the nearby sensor nodes, their data values on a given time, and the scale of the predicted flood. Values from sensors that are marked as broken are not used in the prediction. The flood model then stores the calculated flood predictions, also including the cause and location of possible floods. The flood model is observable so that the *Monitoring* subsystem is able to receive updates and send alerts when needed. In order to communicate with the *Monitoring* subsystem ActiveMQ is used.

The *PredictionStrategyImpl* will implement the interface *PredictionStrategy* so that each *PredictionStrategyImpl* has its own implementation of the flood risk calculation.

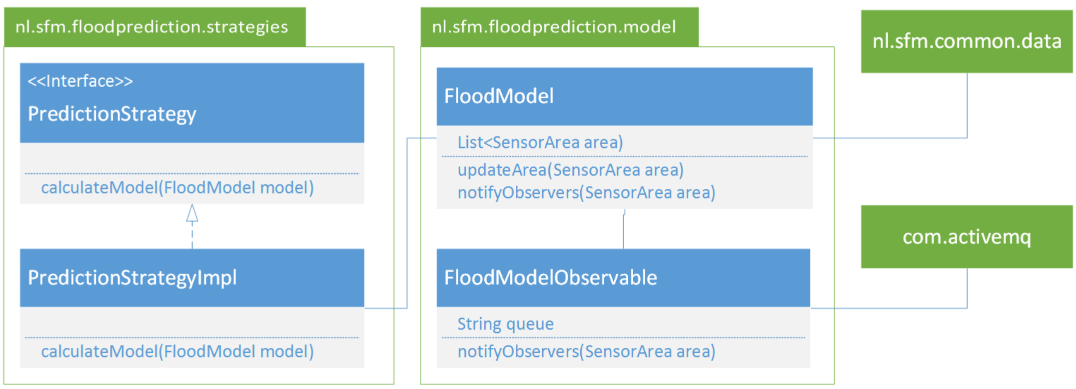


Figure 28: Flood Prediction Class Diagram

Web Portal

The package diagram of the subsystem *Web Portal* is depicted in Figure 29. This subsystem provides a portal that is accessible for the users of the SFM system. For this subsystem the *Model-view-controller(MVC) Pattern* [23] is used. The usage of this pattern enforces separation of concerns and this increases the adaptability and maintainability of the system. Since the *MVC Pattern* is used, this subsystem consists of three parts: a model, a view and a controller.

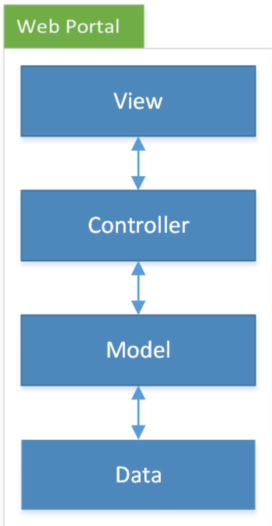


Figure 29: Web Portal Package Diagram

The *ApplicationController* is part of the controller component. It takes care of the authorization. The *ApplicationController* is extended by the *UserController, AreaController, Area::SubscriptionController, MaintenanceController* and *Maintenance::AreaController*. Each of these controllers handle the actions that each type of user can perform. The *UserController* facilitates user management, such as registering, login and editing of user details. The *AreaController* shows the different sensor areas registered in the system and allows users to view information regarding flood warnings, such as the area, time and severity. Registered users are also able to manage their subscription to alerts for given areas, using details like e-mail addresses and phone numbers. The *MaintenanceController* allows the maintenance crew to view and manage the sensor network, such as the different sensor areas, the devices present in those areas, and their status.

The model component consists of four models, *UserModel, AreaModel, SensorModel, HubModel* and *SubscriptionModel* that all inherit the properties of the class *Model*. These models store relevant data that is served to the users. The data is requested from the database using the library *nl.sfm.common.data*.

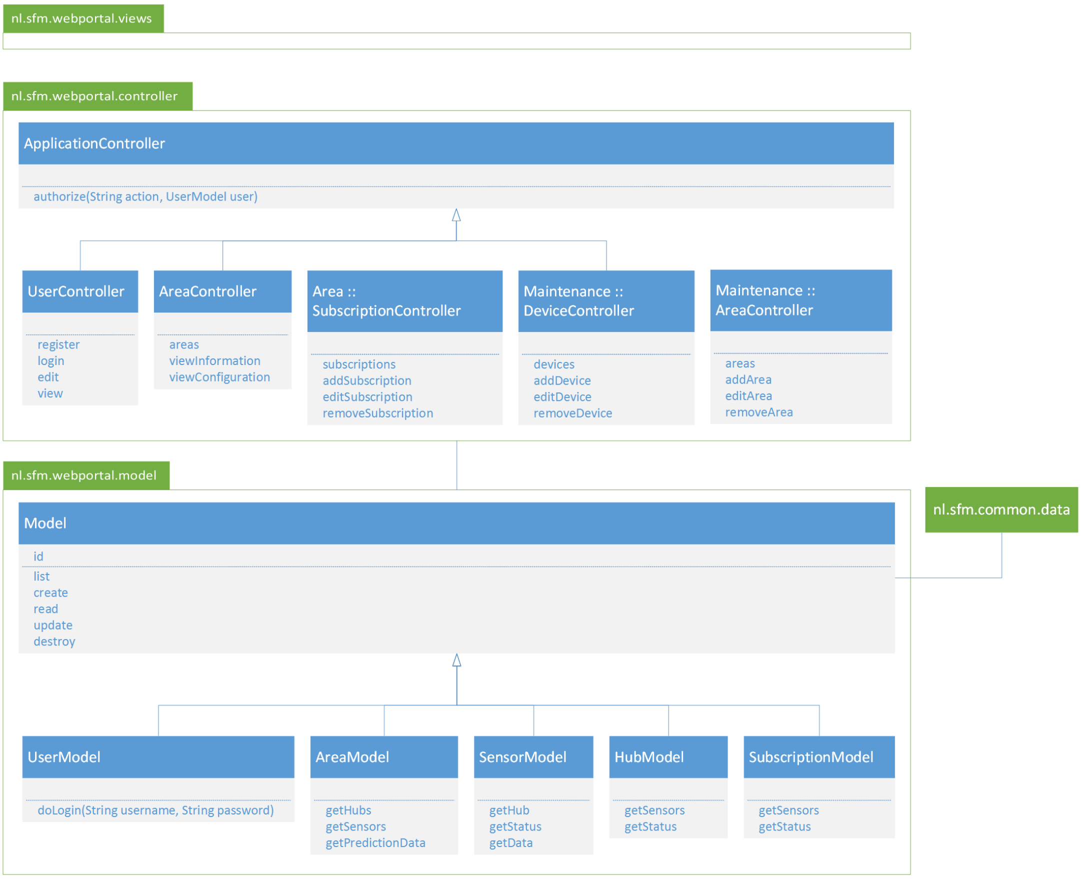


Figure 30: Web Portal Class Diagram

7.2.2 System dependencies

The system dependencies are those packages that are used across the different subsystems. They are stored in the *nl.sfm.common* package.

Database Abstraction Layer

The database abstraction layer is used by the above mentioned subsystems to communicate to the databases. Using a database abstraction layer promotes adaptability when compared to not using any shared components (since code only needs to be modified in one place) and promotes performance when compared to using an external subsystem as a wrapper around the database, which would require large amounts of serialization and deserialization to load all data into the flood model. It also acts as a *Facade* [26], hiding the fact that data is stored in two separate databases.
Figure 31 provides the class diagram of this package.

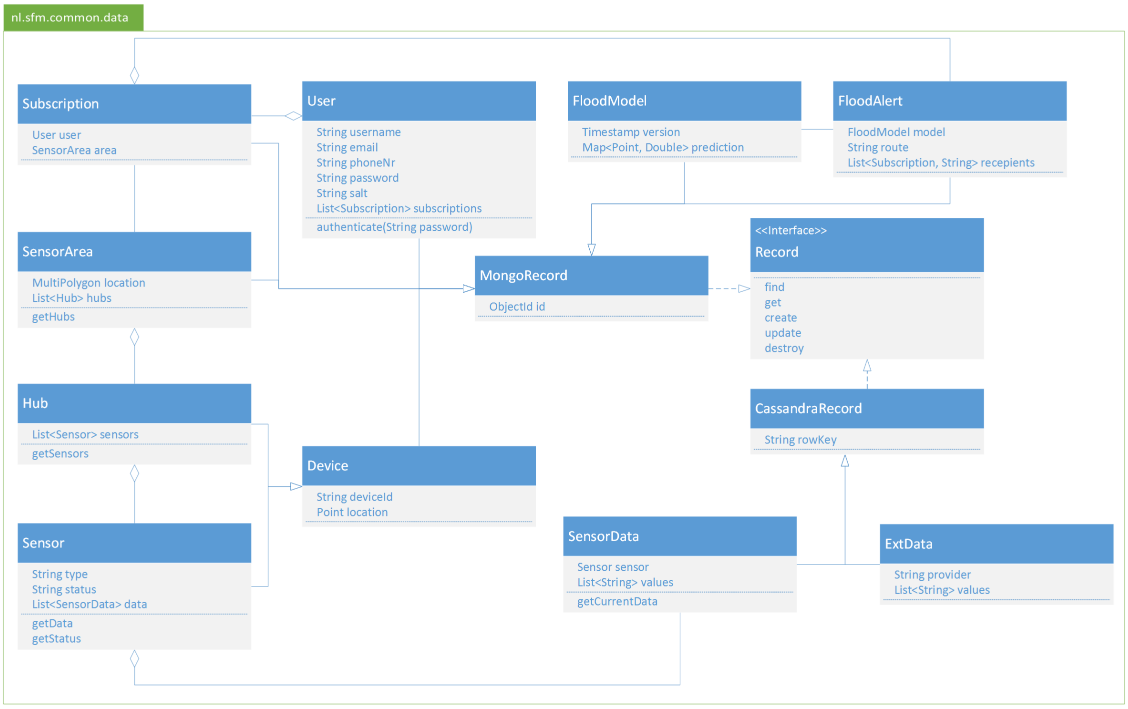


Figure 31: Database Abstraction Layer Class Diagram

7.3 Development view

The development view illustrates the system from a programmer's perspective and is concerned with software management.

Each subsystem is located in their own directory within the *nl.sfm* directory. Each subdirectory except for *nl.sfm.common* is built into a separate executable file. The packages in *nl.sfm.common* as well as other dependencies are included into the subsystems as part of the build process. This can be seen below in figure 32. The directory structure is as follows:

/ nl

  / sfm

    / sensorcomm
      / communication
      / application
      «include» nl.sfm.common.data
    / externalcomm
      / communication
      / application
      «include» nl.sfm.common.data
    / monitoring
      / communication
      / application

«include» nl.sfm.common.data
«include» com.activemq
/ floodprediction
/ strategies
/ model
«include» nl.sfm.common.data
«include» com.activemq
/ webportal
/ views
/ controller
/ model
«include» nl.sfm.common.data
/ common
/ data

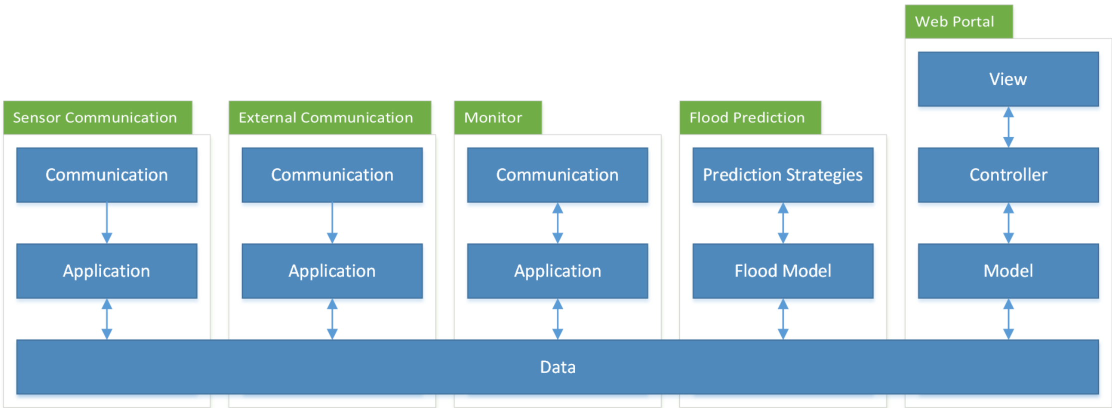


Figure 32: Integrated Package Diagram, showing the shared Data package.

7.4 Process view

The process view describes the system processes and how they interact. It focuses on the run time behavior of the system and deals with aspects such as concurrency, communication and synchronization. The process view for the SFM system consists of five components that run continuously. This comes down to all the processes that are listed in Figure 33, which are also textually described on the subsequent page.

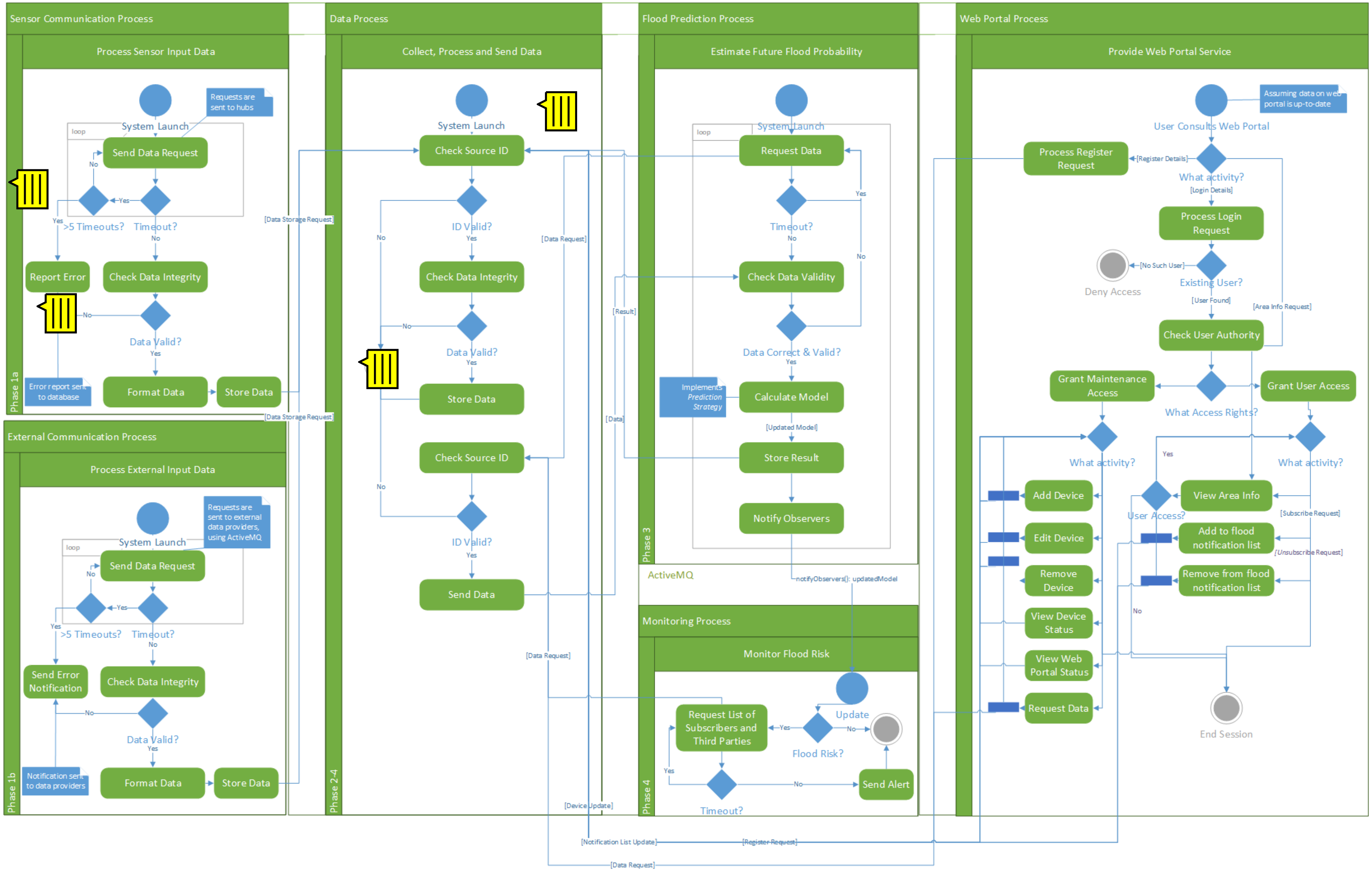


Figure 33: Activity Diagram

7.4.1 Communication Processes

Most of the processes are triggered by system launch. For instance, the sensor and external communication processes continuously pull data from the sensor hubs and external data providers, respectively. They check whether the data is valid and store the data in the common database. As depicted in 33, communication between the communication processes and the database occurs directly. Figure 34 and 35 zoom in the activities that take place during the data retrieval processes. External parties are contacted through an ActiveMQ-based message broker.

7.4.2 Data Process

The process responsible for handling all incoming and outgoing data also remains active once the system is booted. It waits for incoming data (storage) requests and processes these requests accordingly, given that the source ID was recognized. Apart from the source ID, it also checks whether the data is valid, in order to prevent that corrupted data enters the system.

7.4.3 Flood Prediction and Monitoring Process

The flood prediction process remains active after system launch. It continuously requests input data from the central database to update its model. Subsequently, the flood monitoring process is triggered by a NotifyObserver() update from the flood prediction process. The update feeds from the flood prediction process to the monitoring process are facilitated by ActiveMQ. The monitoring process determines whether the updated model indicates an expected flood risk. If so, it will send an immediate alert to relevant third parties and subscribed users. Figure 38 zooms in on the process of sending alerts when a flood has been predicted.

7.4.4 Web Service Process

The web service process commences whenever a user actively consults the web server. Such activities by users include a login or a register request. Users may also freely access flood area info without the need to login. However, upon logging in a user may be granted additional rights. For instance, a "maintenance" account permits the user to add, alter or remove devices and/or view a device's or web portal's status. A "user" account can besides viewing area information also subscribe to and *un*subscribe from a flood notification list for a particular area. Figure 36 describes the process of a user registering a new account and subscribing to flood warnings for an area.

7.4.5 Sequence Diagrams

This section describes a number of key processes (33) related to the subsystems mentioned in section 7.2.1. These key processes are illustrated using UML sequence diagrams. These processes are defined as *key* processes, because all of these processes are architecturally significant. All these processes are based on the use cases described in section 3.5.

Figure 34 describes the process of retrieving data from the sensor areas. The corresponding use cases of this sequence diagram are UC-1.1 and UC-3.1.

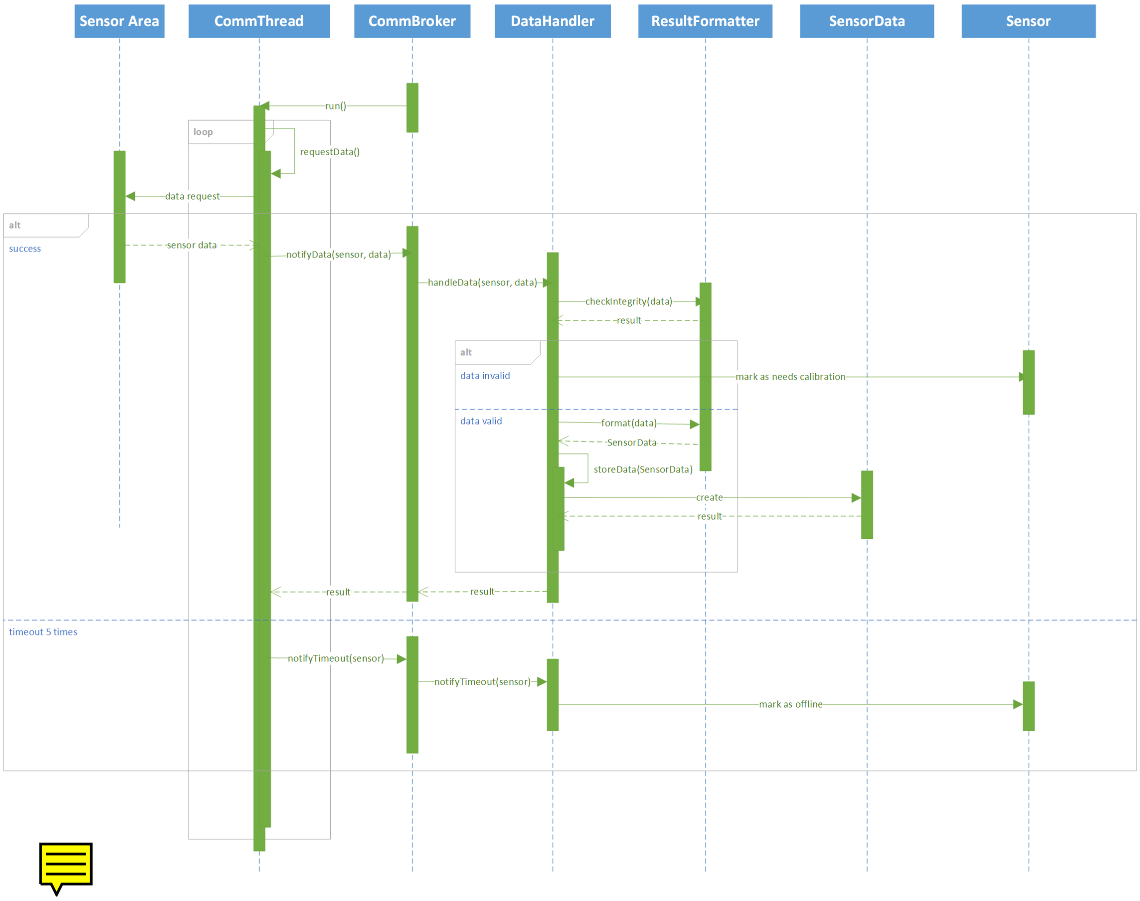


Figure 34: Sensor Communication

Figure 35 describes the process of retrieving data from external data providers. The corresponding use cases of this sequence diagram are UC-1.2 and UC-3.1.

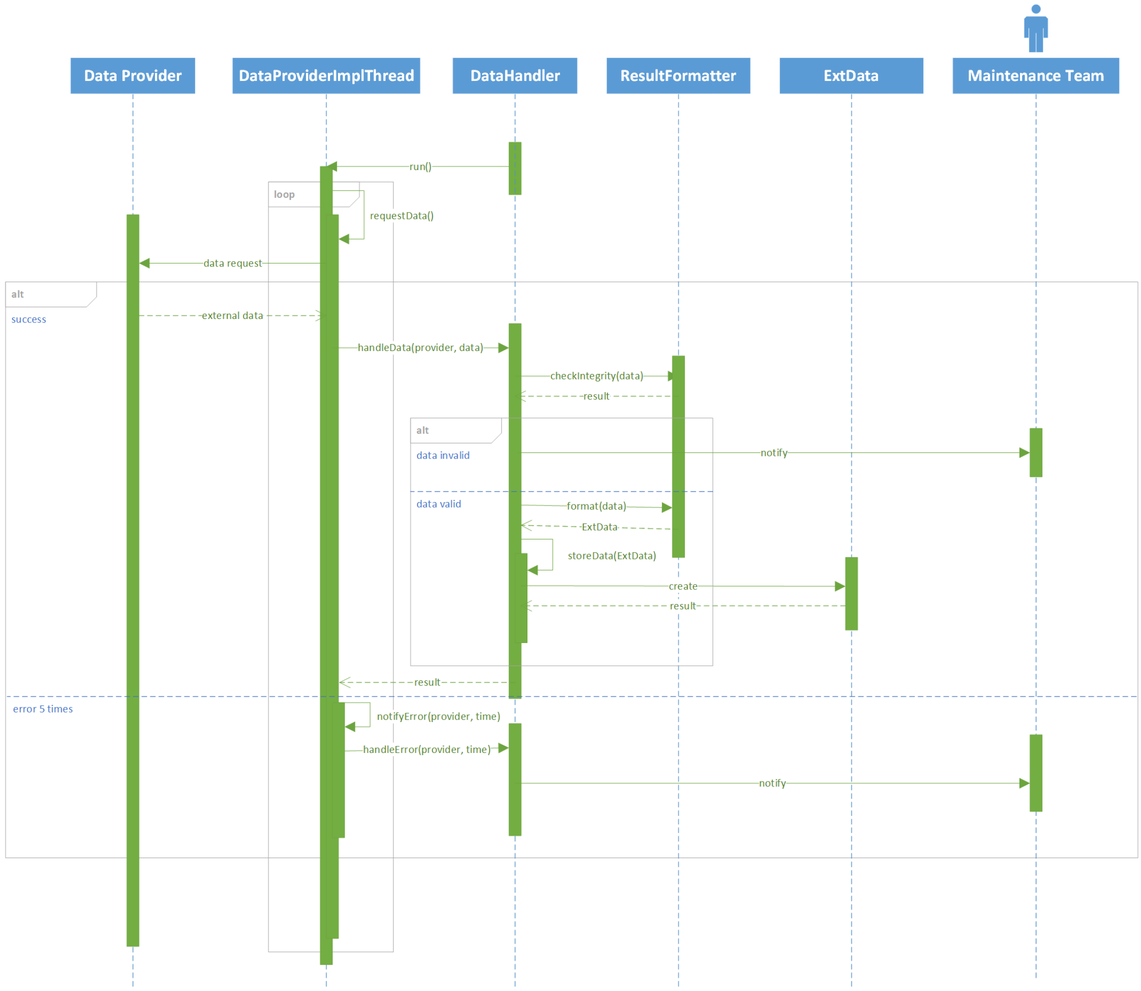


Figure 35: External Communication

Figure 36 describes the process of a user registering a new account and subscribing to flood warnings for an area. The corresponding use cases of this sequence diagram are UC-7.1 UC-7.3.

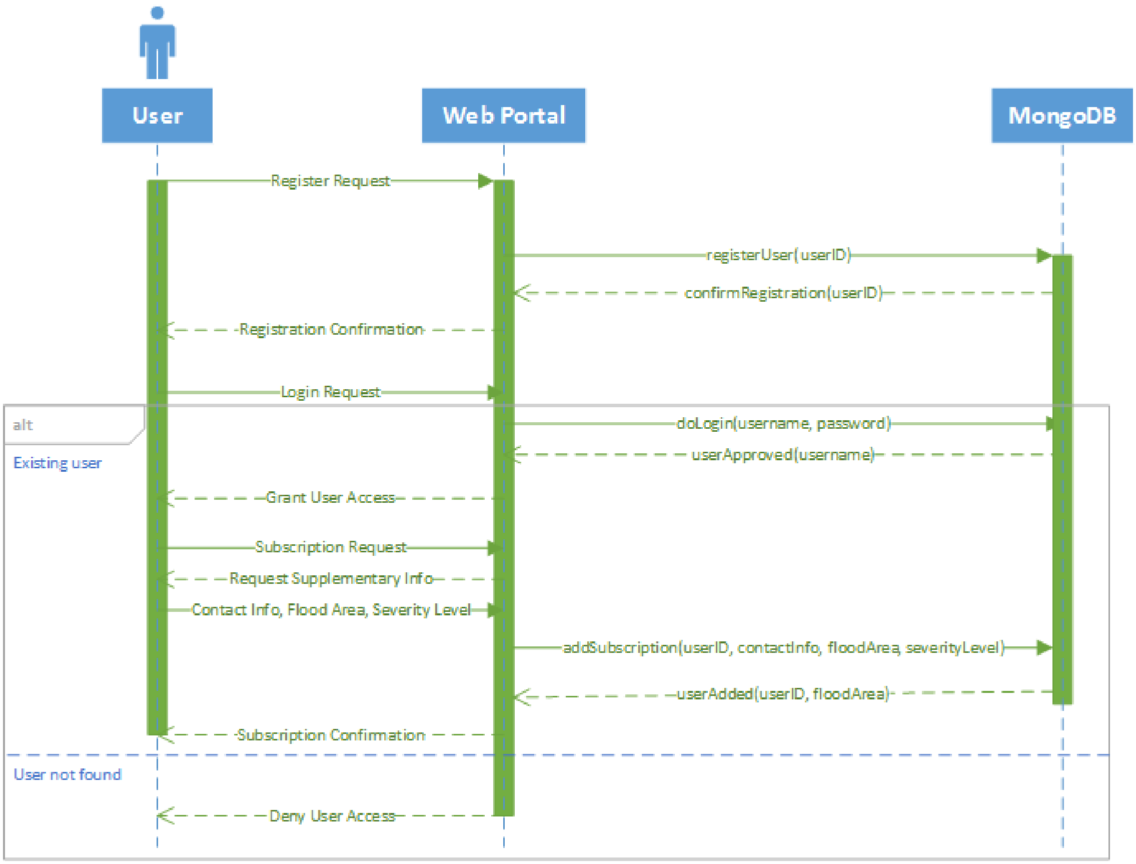


Figure 36: User Subscription

Figure 37 describes the process of making a flood prediction. The corresponding use case of this sequence diagram is UC-2.1.

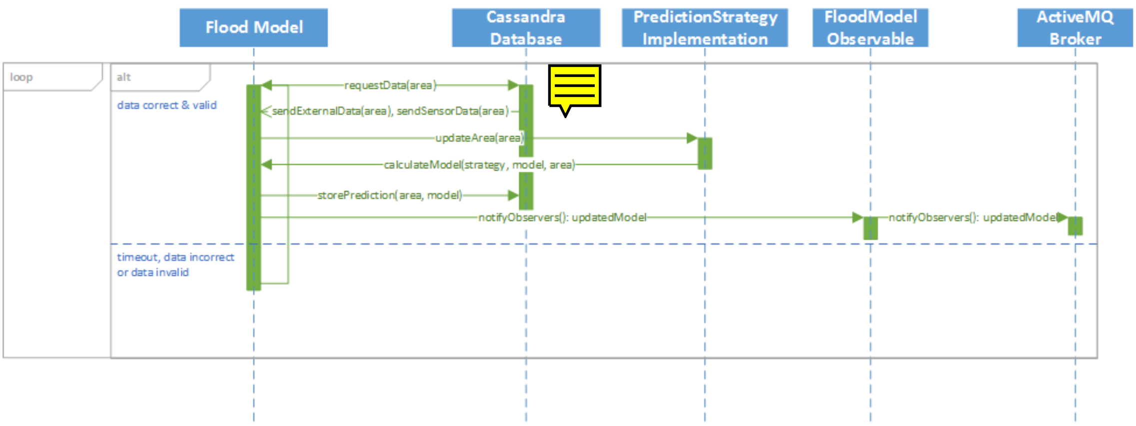


Figure 37: Flood Prediction

Figure 38 describes the process of sending alerts when a flood has been predicted. The corresponding use cases of this sequence diagram are UC-4.1 and UC-6.1.

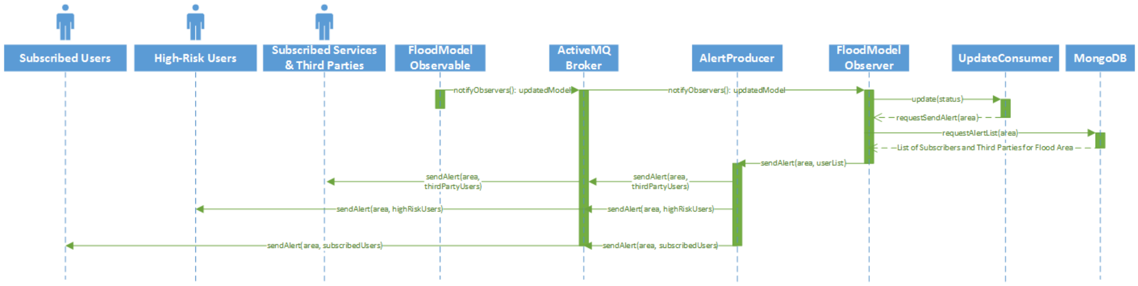


Figure 38: Alert Subscribers

7.5 Physical view

The physical view presents the systems from the perspective of the system engineer. It is concerned with the physical distribution of the hardware and software components and their connections.

The subsystems mentioned above are all located on the central server. This can be seen in Figure 39. Besides these subsystems, the devices the system communicates with are also visible in the deployment diagram.

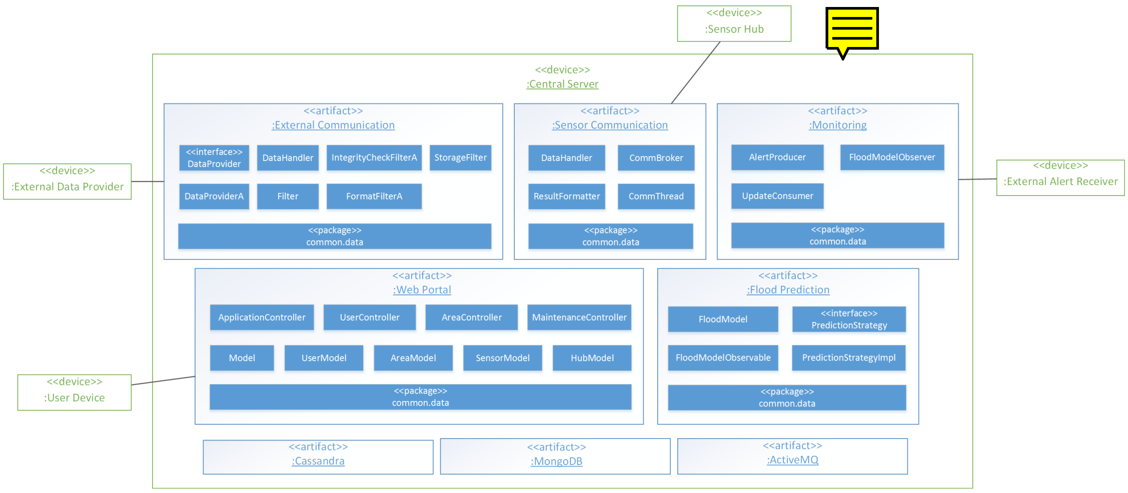


Figure 39: Deployment Diagram

7.5.1 Devices

The devices that are present in the deployment diagram are listed below, their interests or responsibilities are briefly discussed.

Device	Description
Central Server	The central server is logically the main part of the system, all of the other devices communicate with this central server. And all of the calculations are done on the central server.
Sensor Hub	The sensor hubs are responsible to collect data from nearby sensors and transporting this data to the central server
External Data Provider	External data providers provi/de data that is not gathered by our sensors, such as weather data.
User Device	User devices are devices that interact with the web portal of the central server
External Alert Receiver	External alert receivers are devices that consume from an ActiveMQ queue

7.5.2 Artifacts

The artifacts, all part of the central server, are the software parts that have responsibilities inside the system. They are listed below with their responsibilities.

Artifact	Description
Sensor Communication	The sensor communication artifact is responsible for retrieving data from sensors via sensor hubs.
External Communication	The external communication artifact is the part responsible for retrieving data from external data sources.
Monitoring	The monitoring artifact is responsible for monitoring the flood prediction and for sending alerts to our users.
Flood Prediction	The flood prediction artifact is the part responsible for correctly determining the flood risks with the available data.
Web Portal	The web portal is responsible for informing all the users with relevant information.
ActiveMQ	The ActiveMQ server is responsible for providing an interface to push alerts to external parties and also to update the Monitor with results from the Flood Prediction artifact.
Cassandra	The Cassandra Database Server is responsible for storing all the non-volatile sensor- and external data in the system.
MongoDB	The MongoDB Database Server is responsible for storing all other forms of data, such as data regarding users, subscriptions and maintenance.

7.5.3 Distribution

Two key drivers of the SFM are performance and fault tolerance. We must therefore ensure that our databases satisfy these key drivers. For MongoDB we use replication to satisfy fault tolerance. With replication, data is copied to multiple secondary databases so that in the event that the primary database fails, a secondary database can take over. To satisfy performance, we enhance scalability by enabling sharding. With sharding, data is distributed to multiple databases according to a calculated shard key. For Cassandra, clustering provides satisfies both key drivers because cluster nodes automatically divide data while maintaining multiple copies [33].

7.6 Software design decisions

This section provides an overview of the main decisions that were relevant for the software architecture.

Decision SW1	**Apache ActiveMQ is used for communication with third parties**
Status	*To be reviewed*
Problem/issue	The system and ("black box") third parties require a common platform for communication
Decision	The system will make use of ActiveMQ for communication with third parties
Alternatives	RabbitMQ - A popular AMQP-based intermediary for messaging. ZeroMQ - A socket interface that allows one to easily build a messaging system.
Arguments	ActiveMQ is the most popular message broker. ZeroMQ supports many advanced messaging scenarios and does not have the overhead of a protocol such as AMQP. To the contrary of ActiveMQ and RabbitMQ however, one needs to implement everything herself by combining various pieces of the framework, which requires lots of time and effort. ActiveMQ and RabbitMQ have numerous features that allow to balance performance with reliability, including persistence, delivery acknowledgements, publisher confirms and high availability. However, like Zmq, it can be deployed with both broker and P2P topologies, while RabbitMQ only uses a broker architecture. As a result, ActiveMQ combines best of both alternatives and is the most scalable alternative.

Decision SW2	**Apache ActiveMQ is used for communication between the monitor and flood model**
Status	*To be reviewed*
Problem/issue	The monitor and flood model require a common platform to send and receive messages
Decision	The system will make use of Apache ActiveMQ for communication between the monitor and flood model
Alternatives	RabbitMQ - A popular AMQP-based intermediary for messaging. ZeroMQ - A socket interface that allows one to easily build a messaging system.
Arguments	The same arguments apply as in the previous decision; ActiveMQ combines best of both alternatives and is the most scalable alternative.

Decision SW3	**Shared library communication with database**
Status	*To be reviewed*
Problem/issue	Communication with the database needs to proceed smoothly and common messaging scenarios should be easily altered and extended
Decision	The system will make use of a database abstraction layer in a shared library for communication with the database
Alternatives	Separate - Have each subsystem code database interaction separately Wrapper process - Write a wrapper process around the database to request data from
Arguments	Comparing the use of a shared database abstraction layer with writing repeatedly writing code for database interaction in each subsystem, the shared library is better for adaptability and maintainability, since changes are only required in one place, instead of in each subsystem. Compared to writing a wrapper process, using a shared library enhances performance, since instead of serializing and deserializing data every time a request is made, the subsystems can request the data from within their own process.

Decision SW4	**Flood model and prediction strategy**
Status	*To be reviewed*
Problem/issue	The flood prediction model needs to be easily adaptable and flexible, while leaving its core functions unaffected.
Decision	The system will make a distinction between the flood model and a prediction strategy.
Alternatives	Prediction strategy is inherent to the flood model.
Arguments	With the current decision, one can decide upon a strategy depending on the situation, with the main goal of making a more accurate prediction. The flood model does not need need to change, since one can easily alter or switch between prediction strategies to make the calculation model better fit the situation. If the prediction strategies would be inherent in the flood model, one would need to constantly adapt the model. This requires lots of effort and errors would more easily trickle into the prediction model. This could eventually lead to a faulty flood prediction. In order to mitigate this issue, a distinction is made between the flood model and the prediction strategy.

Decision SW5	**Apache Cassandra**
Status	*To be reviewed*
Problem/issue	The system needs the capacity to store and retrieve (big) data
Decision	The system will make use of Apache Cassandra for easy and scalable data storage and retrieval
Alternatives	MongoDB - One of the most popular NoSQL document stores. MySQL - An open-source relation database management system (RDMS). Oracle RDBMS - The most popular object-relational database management system in the world. SAP HANA - A relational database management system that has already been deployed in other flood monitoring systems. It is specialized in storing and processing big data.
Arguments	Apache Cassandra is a NoSQL distributed database management system. No single point of failure ensures fault and failure tolerance and 100% availability. Besides, Cassandra is considered to have the best in-class scalability of NoSQL platforms and certainly better than any RDBMS like MySQL and SAP HANA. Adding new columns to RDBMS can lock the entire database in some database, or create a major load and performance degradation in other.

Decision SW6	**MongoDB**
Status	*To be reviewed*
Problem/issue	The system needs to be able to user and management data
Decision	The system will make use of MongoDB for scalable storage
Alternatives	Apache Cassandra - Use the same database as for the sensor data. MySQL - An open-source relation database management system (RDMS). Oracle RDBMS - The most popular object-relational database management system in the world.
Arguments	Cassandra stores data using keys and sets of keys and while this is a good fit for large amounts of sensor data identified by few keys, it is not efficient for storage of user-, management- and configuration data. MySQL and Oracle both provide well known database systems, but as mentioned above relational databases are less scalable. MongoDB document storage is a good fit for the type of data and also scales well.

Decision SW7	**Supervised machine learning**
Status	*To be reviewed*
Problem/issue	The system needs to be able to detect possible floods
Decision	Supervised learning will be used to determine upcoming floods.
Alternatives	Algorithm - a fixed algorithm could be used in order to determine floods. There are several algorithms like ANN-GA and ANDFIS [24].
Arguments	Supervised learning will be used since it will give the most accurate results in the long run. The system will learn from previous events. Current algorithms are very accurate as well. However they have been mostly tested in China. The SFM system will be used in different natural environments. Furthermore these algorithms require certain input parameters whereas the supervised learning method does not have this requirement.

Decision SW8	Layers Pattern
Status	*To be reviewed*
Problem/issue	The External and Sensor Communication subsystems need to be easily adaptable
Decision	The Layers Pattern will be used in the External and Sensor Communication subsystems
Alternatives	Relaxed Layers Pattern - This variation on the layers pattern allows that higher layers can directly call lower layers, in order to help offset the performance hit. No Pattern - I.e. communication shortcuts are allowed. This option comes at the costs of adaptability with the benefit of a (minor) performance boost.
Arguments	The External and Sensor subsystem will consist of three layers: communication, application and data. Using the Layers Pattern [20], each of these layers is only allowed to communicate with layers directly above or below themselves. The pattern is applied here in order to improve the adaptability of the SFM system. Since the components only use a three-layer structure, the performance hit is expected to be small. Hence, we reject the option of using no nor the Relaxed Layers Pattern.

Decision SW9	Model-View-Controller Pattern
Status	*To be reviewed*
Problem/issue	The Web Portal needs to be easily adaptable and maintainable
Decision	The Model-View-Controller design pattern will be used in the Web Portal's design
Alternatives	Model-View-Presenter Pattern - View is more loosely coupled to the model. The presenter is responsible for binding the model to the view. No pattern - Use a self-invented design solution tailored to fit the requirements of the web portal.
Arguments	The MVC pattern [23] enforces separation of concerns, by separating the design into the data model (model), data representation (view) and application logic (controller). As such, for instance, changes in the user interface do not impact the data model and vice versa. This structure seamlessly allows to build many different views without needing to change the underlying model. Also, breaking up of responsibilities makes the code more readable and re-usable. The MVC pattern is preferred over the MVP pattern, because the Controller is based on behaviors and can be easily shared across views. Accordingly, the MVC pattern is anticipated to be the best alternative for increasing the adaptability and maintainability of the web portal.

Decision SW10	Broker Pattern
Status	*To be reviewed*
Problem/issue	Communication between the sensor communication subsystem and sensor hubs should be easily adaptable
Decision	The sensor communication subsystem will use the Broker Pattern for communication between the sensor communication component and the sensor hubs
Alternatives	No pattern - Distributed sensor hubs use own application functionality for communication with the sensor communication subsystem.
Arguments	In the Broker Pattern [21], communication with multiple sources is hidden behind a broker that manages the separate sources. This abstraction helps to separate system communication functionality from the application functionality. This makes communication more transparent to the developer. Besides, components do not need to comply to a single programming language. All these elements help to make the system more adaptable.

Decision SW11	Pipes and Filters pattern
Status	*To be reviewed*
Problem/issue	Tasks in processing of external data should be easily identifiable and adaptable to promote flexibility
Decision	The Pipes and Filters pattern will be used in the External Communication subsystem
Alternatives	Monotholic module - Usage of a monotholic module for performing all tasks required in processing the external data.
Arguments	Deployment of the Pipes and Filters pattern [25], decomposes the external data processing steps into a set of discrete components (or filters), each of which performs a single task. These filters are then connected through pipes. In our case, the filters are tasks running in worker processes, and the pipes are ActiveMQ queues. This allows for complete separation between the different tasks (since they are only connected through pipes), reuse of shared tasks (connecting a worker to a queue allows generic tasks to communicate with provider-specific tasks), scalability (multiple tasks can subscribe to the same queue) as well as easy replacement and adaptation of tasks (since new versions need only to connect to the right queues). The pattern is highly adequate for the external data process, since the external data processing can easily be decomposed into a set of discrete, independent steps (see Figure 22). Building a monotholic module for the processing, however, makes it hard to adapt or reuse code if parts of the same processing functionality is required elsewhere in the application.

Decision SW12	Strategy pattern
Status	*To be reviewed*
Problem/issue	The flood prediction determination model should be easily adaptable in order to be able to adapt quickly to environmental changes
Decision	The Strategy pattern will be used in the Flood Prediction subsystem
Alternatives	No pattern - The flood prediction strategy is inherent to the flood model
Arguments	The *Strategy Pattern* [22] is applied in the Flood Prediction subsystem (section 7.2.1). Using the Strategy Pattern, the inputs and outputs of the prediction remain the same, but a prediction algorithm can more easily be altered on short notice. This will increase the adaptability of the system, since one can seamlessly add, swap and remove prediction strategies without affecting the functionality of the subsystem. As such, the pattern allows the system to respond fast to environmental changes which helps to guarantee accurate, timely and valid (i.e. high performance) flood predictions, even in highly unstable and uncertain environments.

7.7 External data providers

In order to increase the accuracy of the predictions of the SFM system, we will make use of external data providers to provide the SFM system with more data and provide the system with types of data that are not measured by the sensors of the SFM system. Per country there should be external data providers that provide the system with data such as wind speed, rainfall and temperature.

Since the SFM will be deployed in several different countries, the external data providers per country will differ. These data providers will be determined in consultation with each government. To explain this in a bit more detail; for some countries the discharge rate of water could be important whereas other for other countries this is not that important. However there are some external data providers that cover multiple countries. So in practice the SFM system for country X makes use of external data provider A, B and C whereas country Y makes use of external data provider B, C and D.

To give an indication of which data providers should/could be used we will list a few examples of data providers in Table 14 and Table 15.

External data provider	**Country specific**	**Weather data**
OpenWeatherMap	No, worldwide	Temperature, wind speed, cloudiness, pressure, humidity, precipitation.
Rijkswaterstaat	Yes, the Netherlands	Discharge rate of the water and speed of the water in rivers
ShootHill	Yes, United Kingdom	River levels inland, river levels tidal, river flow level and ground water levels.
Speedwell	No, worldwide	Temperature, precipitation, sea pressure levels, snow depth on ground, etcetera

Table 14: Overview of external data providers

Some of the external data providers, like OpenWeatherMap, ShootHill and SpeedWell, provide weather forecasts as well. Using these forecasts the system might be able to detect earlier. Furthermore OpenWeatherMap and Speedwell have large data sets consisting of historical data. This data could potentially serve as a training set.

External data provider	**Found at**
OpenWeatherMap	http://openweathermap.org/
Rijkswaterstaat	http://www.rijkswaterstaat.nl/apps/geoservices/rwsnl/awd.php?mode=html&project
ShootHill	http://www.shoothill.com/environment-agency-liveapi/environmental-data-api/
Speedwell	http://www.speedwellweather.com/Default.aspx

Table 15: Sources of external data providers

8 Architecture Evaluation

This chapter will evaluate the architecture created in the previous chapters. In the first part the requirements are checked if all of the important requirements are fulfilled and which decisions are related to specific requirements. The second part the ATAM (Architecture Tradeoff Analysis Method) technique is used to evaluate the architecture.

8.1 Requirements verification

The architecturally significant drivers (see Section 7.1) are checked against the architecture to see whether these requirements are fulfilled in the architecture. The structure of the requirements is taken from Section 3 and the identifiers correspond with the tables in Section 3. To improve the readability of the table, the sections in which the requirements appear are listed in the "Traced in" column, and small remarks may be provided to argue why the requirement is or has not yet been fulfilled.

8.1.1 Functional requirements

The functional requirements are listed as found in Section 3.6.

ID	Priority	FF*	Decisions	Traced in
FR-1.1	Must	Yes	-	Section 7.2.1
Remarks:	Sensor data can be requested by the system via telnet commands to the sensor hubs			
FR-1.2	Must	Yes	-	Section 7.2.1
Remarks:	The data from the sensors will be stored in a Cassandra database			
FR-1.3	Must	Yes	-	Section 7.2.1
Remarks:	The External Communication subsystem will collect data from external sources			
FR-2.1	Must	Yes	-	Section 7.2.1
Remarks:	Provided by the FloodModel			
FR-2.2	Must	Yes	-	Section 7.2.1
Remarks:	Provided by the FloodModel			
FR-2.3	Must	Yes	-	Section 7.2.1
Remarks:	The prediction is done with the PredictionStrategy			
FR-2.4	Must	Yes	-	Section 7.2.1
Remarks:	The calculations are stored by the FloodModel			
FR-3.2	Must	Yes	-	-
FR-3.3	Must	Yes	-	-
FR-3.4	Must	Yes	-	Section 7.2.1
FR-3.5	Must	No	-	-

FR-4.1	Must	Yes	-	Section 7.2.1
Remarks:	Warnings are sent by the AlertProducer via the ActiveMQ queue			
FR-4.3	Must	Yes	-	Section 7.2.1
Remarks:	A list of SensorAreas is kept by the FloodModel to quickly know which areas should receive warnings			
FR-4.4	Must	No	-	-
Remarks:	Not yet in the architecture			
FR-4.5	Must	No	-	-
Remarks:	Not yet in the architecture			
FR-5.2	Must	Yes	-	Section 7.2.1
Remarks:	The web portal model will store the data in the MongoDB via the common data package			
FR-6.1	Must	No	-	Section 7.2.1
Remarks:	For now only the message broker for warning messages is in the architecture			
FR-7.1	Must	Yes	-	Section 7.2.1
Remarks:	Provided by the web portal's AreaController			
FR-7.2	Must	Yes	-	Section 7.2.1
FR-7.3	Must	Yes	-	Section 7.2.1
FR-7.4	Must	Yes	-	Section 7.2.1
Remarks:	Provided by the web portal's UserController and MaintenanceController			
FR-7.5	Must	Yes	-	Section 7.2.1
Remarks:	Provided by the web portal's AreaController			
FR-7.6	Must	Yes	-	Section 7.2.1
Remarks:	Provided by the web portal's AreaController			
FR-7.7	Must	Yes	-	Section 7.2.1
Remarks:	Part of the web portal's UserModel			
FR-8.1	Must	Yes	-	Section 7.2.1

∗) Fulfilled

8.1.2 Technical requirements

The technical requirements are listed as found in Section 3.7.

ID	**Priority**	**FF∗**	**Decisions**	**Traced in**

TR-1.2	Must	Yes	H-500XL Data Logger, Line-of-Sight Radio series (H-4400)	Section 6.4
TR-1.4	Must	Yes	Apache Cassandra	Section 7.2.2
TR-2.1	Must	Yes	Apache Cassandra	Section 7.2.2
TR-2.2	Must	Yes	MongoDB	Section 7.2.2, Section 7.2.1
TR-2.3	Must	?	-	-
TR-3.4	Must	Yes	-	Section 7.2.1
Remarks:	Part of the sensor communication's ResultFormatter			
TR-4.1	Must	Yes	-	Section 7.2.1
Remarks:	For now only the message broker for warning messages is in the architecture			
TR-6.2	Must	No	-	-
Remarks:	Not yet in the architecture			
TR-6.3	Must	No	-	-
Remarks:	Not yet in the architecture			
TR-7.4	Must	Yes	-	Section 7.2.1
TR-7.6	Must	Yes	-	Section 7.2.1

*) Fulfilled

8.1.3 Commercial non-functional requirements

The commercial non-functional requirements are listed as found in Section 3.8.

ID	Priority	FF*	Decisions	Traced in
CNR-4	Must	Yes	-	Section 6.2

*) Fulfilled

8.1.4 Technical non-functional requirements

The technical non-functional requirements are listed as found in Section 3.9.

ID	Priority	FF*	Decisions	Traced in
TNR-1.1	Must	No	-	Section 6.2
Remarks:	Wired sensors are connected only to one hub, wireless sensors are able to be connected to multiple hubs			
TNR-1.2	Must	No	-	-

Remarks:	Not yet in architecture			
TNR-1.3	Must	Yes	Private cloud hosting,	Section 5.2.5
Remarks:	The private cloud can be scaled up in cases of high demands			
TNR-1.4	Must	Yes	Private cloud hosting,	Section 5.2.5
TNR-2.2	Must	Yes	Most of the decisions in Section 6.1	Section 6.4
TNR-2.3	Must	Yes	Most of the decisions in Section 6.1	Section 6.4
Remarks:	The variety of sensors with the own sensors and the external data sources is high enough to fullfill this requirement			
TNR-2.4	Must	Yes	-	Section 2.8.4
Remarks:	Each sensor is given 1 man-hour a year for maintenance			
TNR-3.2	Must	Yes	SDI-12 protocol, ActiveMQ	Section 5.2.2, Section 7.2.1
TNR-3.4	Must	Yes	-	Section 7.2.1
TNR-4.2	Must	Yes	-	-

∗) Fulfilled

8.2 Architecture trade-off analysis method

In this section, we will discuss the steps of the Architecture trade-off analysis method (ATAM), how the method has been used during the elaboration of the architecture, and the results of these steps of the ATAM will be described.

8.2.1 Evaluation steps

The Architecture trade-off analysis method (ATAM) is a risk-mitigation process, that is used early in the software development life cycle to discover trade-offs and sensitivity points, which help provide the basis for choosing a suitable architecture. The ATAM consists of the following steps according to its formal description: [30]

Step	Description	Initiator	Documented in
1	Present ATAM	Moderator	Section 8.2.1
2	Present business drivers	Business stakeholder	Section 2
3	Present the architecture	Architect	Section 3.1, 5, 6, 7
4	Identify architectural approaches	Architect	Section 8.2.2
5	Generate quality attribute utility tree	All	Section 8.2.3
6	Analyze architectural approaches	Architect	Sections 8.2.3, 8.2.4
7	Brainstorm and prioritize scenarios	All	Sections 8.2.4
8	Analyze architectural approaches	Architect	Sections 8.2.4
9	Present results	Moderator	All sections

Table 20: Steps of the ATAM and their elaboration in this document

Moderator refers to the person who is assigned to lead the ATAM meetings, discussions and brainstorm sessions. Architect refers to the person or group of persons that are responsible for designing proposals and elaborated models of the architecture. Steps 1 to 3 have already been covered in earlier sections. In the next sections, the results of the rest of the steps will be given.

8.2.2 Architectural approaches

In the architecture, as described in sections 5 up to 7, several decisions are taken that support the key drivers of the system. Some of the main high level architectural structures are listed here together with an description to identify what has driven the architects to choose that structure.

For storing the sensor data and computing the flood detection a cloud based solution is chosen. In this case a private cloud will be deployed, which is able to scale quite easily. Due to this scalability the algorithm used to determine whether or not there will come a flood is able to be more precise, this will increase the accuracy of the flood detection greatly. This cloud based solution doesn't only impact the performance of the system but also impacts the fault tolerance. The private cloud can be deployed on multiple physical locations so that in case of faults in one data center the rest of the system will still be able to operate.

The communication protocol used by the sensors is chosen to be SDI-12, which is a widely used protocol for sensors. By using such a general protocol with a large range of sensors using that protocol the adaptability of the system is quite high. The prediction model naturally has to be changed in order to add new kinds or new types of sensors but the system will not get an end of life status when the current sensor supplier disappears.

Most of the software architecture patterns described in the architecture are used to support the adaptability of the system. The patterns that are used are: the layers pattern, the broker pattern, the strategy pattern and the MVC pattern. All these patterns deliver a software system that can be easily adapted with new information sources and new prediction strategies.

The sensor data is stored on the central server instead of locally on the sensors or sensor hubs. This increases the performance of the system because the data is located where, or at least nearby where, the calculations are performed.
Putting these approaches together results in the following list with the most important architectural approaches:

- Using cloud infrastructure to be able to perform more compute heavy algorithms to increase the accuracy of the system.
- Using cloud infrastructure to be fault tolerant in terms of the central server
- Using the SDI-12 protocol to support the addition of new sensors
- Using several software patterns to be adaptable
- Storing data in the central server to decrease the communication lines for the data used by the calculations.

8.2.3 Utility tree and scenarios

We now present the quality attribute utility tree of the architecture. The utility is defined by the quality factors of the system, which have been defined as the key drivers in Section 3: performance, fault tolerance and adaptability. These three main quality factors are refined down towards the requirements of the system as defined in Section 3.

QA*	Attribute Refinement	Scenario
Performance	Validity	**(H,L)** TNR-2.3 (3)
		(M,L) TNR-2.4 (2)
	Accuracy	**(M,L)** TNR-2.2 (2)
		(H,H) TNR-2.5 (9)
	Timeliness	**(H,L)** TNR-2.1 (3)
		(H,H) TNR-2.6 (9)
		(M,L) TNR-2.7 (2)
Fault tolerance	Central Server failures	**(M,M)** TNR-1.3 (4)
		(M,L) TNR-1.4 (2)
	Sensor Hub failures	**(H,H)** TNR-1.1 (9)
	Sensor failures	**(H,L)** TNR-1.2 (3)
	Communication failures	
Adaptability	Sensor adaptability	**(M,L)** TNR-3.1 (2)
		(M,L) TNR-3.2 (2)
		(L,L) TNR-3.3 (1)
	Prediction strategy adaptability	**(H,M) (6)** TNR-3.4
	External data sources adaptability	**(M,M) (4)** TNR-4.1

*) Quality Attribute

Table 21: Quality attribute utility tree

The scenarios that are displayed in the utility tree are given an indication of the contribution to overall success (= Importance) and an indication of the cost of implementation (= Cost), displayed as a pair of $(Importance, Cost)$ in the utility tree, where both Importance and Cost can take any value from the set consisting of $\{Low, Medium, High\}$. To prioritize the scenarios in relation to each other, we propose the following table where a score is calculated for each possible pair of $(Importance, Cost)$, the resulting priority is also visible in the utility tree:

	Cost		
Importance	High	Medium	Low
High	3	6	9
Medium	2	4	6
Low	1	2	3

Table 22: Table used for quality attribute prioritization

The following is a list of scenarios that we have distilled from the quality attribute utility tree (keeping in mind the aforementioned prioritization):

Scenario	**Requirement**	**Description**
Scenario 1	TNR-1.2	Resilience to partial sensor outage
Scenario 2	TNR-1.3	Resilience to failures in the cloud infrastructure.
Scenario 3	TNR-3.1	The maintenance team wants to add a new sensor and succeeds within 20 minutes.
Scenario 4	TNR-4.1	A new API, with relevant data for the system, is added as new external data source to the system.

8.2.4 Analysis of scenarios

We include the ATAM performed on two of the described scenarios:

Scenario 1	**Resilience to partial sensor outage**				
Quality attribute:	Fault tolerance				
Environment:	Normal operation				
Stimulus:	One or more components in the sensor network power down or fail to communicate with hubs				
Response:	The system as a whole is able to maintain a functional state, although risk analysis in the areas which include the failing sensors may be limited until the power/network error is resolved or the sensors are replaced.				
Design decision	**Reqrmt.**	**Sensitiv.**	**Trade-off**	**Risk**	**Non-risk**
Redundancy hubs	...	S1	TO1	-	NR1
Central server	...	-	-	R1	-
Self-sustaining	...	-	TO2	R2	NR2
Wired and wireless communication	...	S2	-	R3	NR3
S1:	If the redundancy is too high, then the communication between the sensors and the central server might take too long, which degrades performance.				
S2:	If too large a portion of the network is either wired or wireless, when more than 75% of the communication is done either wired or wireless, than the costs of a failure may be too high to compensate for.				
TO1:	Affordability vs. fault tolerance, the cost goes up with the amount of sensor hubs, but so does fault tolerance.				
TO2:	Affordability vs. fault tolerance, the cost of self-sustained sensors is higher than sensors connected to power grid.				
R1:	If the central hub fails, then the entire system is affected.				
R2:	If the sensor has no power due to lacking production of energy, the sensor becomes unusable.				
R3:	In weather conditions under which a flood is more likely, wireless sensors may have decreased performance.				
NR1:	If one hub fails, another can fetch the sensor data.				
NR2:	If there is a power outage, self-sustaining sensors can still function.				
NR3:	Diversity of communication will lead to better availability of sensors, when one communication method fails other methods will still work.				
Conclusion:	The sensor network and the hubs are fairly resilient to faults, due to redundancy in both sensors and hubs, and the fact that sensors overlap with multiple hubs (so there are multiple paths to the server). The central server approach makes the system vulnerable to faults in case the central server is hit.				

Scenario 2	**Resilience to failures in the web service**
Quality attribute:	Fault tolerance
Environment:	Normal operation
Stimulus:	One or multiple instances of the web service or database handlers fail to maintain functionality
Response:	The total amount of traffic that the web service can handle is temporarily reduced until the failing instances can be brought back online or bring themselves back up, or additional instances are added to the web service. The web service remains functional during the partial failure of instances.

Design decision	**Reqrmt.**	**Sensitiv.**	**Trade-off**	**Risk**	**Non-risk**
Private Cloud	-	S1	-	-	NR1
Database replication	-	-	TO1	-	NR2
Multi purpose web portal	-	-	TO2	R1	-
Pipes and Filters pattern	-	-	TO3	-	NR3

S1:	If the amount of sensors for one customers is too low to create a large enough private cloud at multiple physical locations, the system can be shutdown due to failures at one data center.
TO1:	Affordability vs. Fault-tolerance, replicating the databases will double the costs of the databases but the system is much more fault tolerant.
TO2:	Complexity vs. Fault-tolerance, creating a portal for each subsystem will increase the complexity but also increases the fault-tolerance.
TO3:	Performance vs. Adaptability, adding the necessary communication overhead will decrease the performance, but the adaptability and reusability will increase by using this pattern.
R1:	In case too many web portal instances fail the system will not be reachable anymore. For instance when the system is overloaded with user requests and the maintenance team tries to fix sensors they can have connection errors.
NR1:	The estimated scale of the system will be large enough to distribute the private cloud over multiple data centers.
NR2:	The costs of replicating the databases are relatively low in comparison with the total costs of the complete system, and fault-tolerance is a key driver for this system.
NR2:	The pipes and filters pattern makes it possible to use multiple instances for collecting data from external data sources, so in case one instance fails the external data retrieval will still be possible.

Conclusion:	The web service is quite well protected to failures inside it. By deploying the system on a private cloud with machines on different physical locations the chances that all of the data center will is very minimal. Using the private cloud also helps in creating good database replication, data can be stored on instances across all different locations making that the stored data is highly fault-tolerant. The multi purpose web portal is the only risk for failures in the web service, however the chance is fairly small that too much web portal instances will fail at the same time.

9 System Evolution

The SFM system should keep evolving over time to keep its position in the market, due to the fact that software naturally degrades in quality over time, which is caused by changing user needs and execution environments. This section will discuss how and what the system should evolve in order to maintain quality for its users.

9.1 Dike sensors

Currently research is being performed with respect to dike sensors, which use fiber optics to measure water leakage in dikes [28]. This solution has already been used on experimental dikes to prove that this method is capable of the early detection and localisation of both leakage and settlement. When this system is in production, this detection system can be implemented into the SFM system to give warnings about breakthroughs of dikes located in the area of the sensor network.

The high-level requirements affected by this extension is HL-4, as warnings should be sent when the dike sensors detect water leakage, and HL-7 is also affected, because the web portal has to be able to show the status of the dikes to maintenance personnel.

9.2 Mobile application

Mobile devices are a great tool for reaching out to a large group of people at once, in our case to inform these people about upcoming floods. A mobile application will be developed for Android, iPhone and Windows Phone. With this mobile application, users can perform all possible actions which are possible using the web portal, but adapted for mobile use. Furthermore, the mobile application could be used for guidance of citizens in case of a flood. This software will be available for customers and citizens for free. The mobile application will be deployed by the Google Play, Apple App Store and Microsoft App Store content providers.

The high-level requirements affected by this extensions are HL-4 and HL-6 as warnings should be pushed to the mobile devices that have the mobile application installed. HL-10 is also affected by this extensions as guidance for citizens is part of this high-level requirement.

9.3 Providing guidance to smart (Android) vehicles and route planners

There are a few companies that are currently researching and developing so-called smart vehicles. For example this year, 2015, the Open Automotive Alliance is planning to bring the Android platform to cars. [29] We can work together with Google to extend the functionality of the Google Maps application. The Google Maps application will be supplied with information about floods by the SFM system in order to avoid flooded or to be flooded areas when planning routes. Furthermore, it should be possible that our mobile application, discussed in Section 9.2, will be available on these smart cars.

The high-level requirements affected by this extension are HL-4, HL-6 and HL-10. HL-4 and HL-6 are affected, because the system needs to send flood warnings to the smart vehicles. Since we provide guidance to citizens by extending the functionality of the Google Maps application, HL-10 is also affected.

9.4 Solar energy for hubs

In order to be less dependent on the energy network and being able to deploy sensor networks in more remote location, we will use solar energy to power the hubs of the sensor network. This extension would make it possible for the system to be deployed in more diverse locations. This extension would make the system cost more, but this will come with an increase of the performance of the system.

Moreover, as we have shown for sensors, components will become more resilience to failure of power providers if they can depend on one of multiple sources of power. If one of the energy sources fails, there is always a backup power source that will make sure that the hub stays up and running.

The high-level requirement affected by this extension is HL-5, because it involves placing new hubs in power-sensitive areas.

9.5 Interacting instances of the SFM system

In a future update it should be possible that several instances of the SFM communicate with each other. This could prove very useful to predict floods earlier. If we take the river Rhine for example. It originates in Switzerland, goes through Germany and ends in the Netherlands. So say that the river Rhine causes a flood in Switzerland, it is likely that Germany and the Netherlands will also suffer from floods or at least have a higher chance of getting a flood as well.

The high level requirement affected by this extension is HL-1, because the SFM system will have to request the data from other instances.

9.6 Drones

To extend the possibilities of the warning systems, drones could be deployed in case flood warnings are issued, so that they may be used to distribute these warnings to areas where no network is present or no network is currently available. Drones could also be used to supply users and emergency services with the information that is available on the web portal, and can supply networked route planning devices in vehicles in the area with the information that is needed to plan around the flooded area.

9.7 Real-time high-resolution image processing

Using the real-time high-resolution images from either satellite images or drone images, we can process them to see if areas are still flooded. People have developed algorithms using these images to predict floods in rural and urban floods. [31] This method can be used to give better guidance to users who are traveling in, through or out of the flooded area, and it can be used to verify that a flood is currently happening automatically.

The high-level requirement is affected by this extension is HL-10, because the guidance for users and emergency services are enhanced using this extension.

9.8 Versions

Table 25 shows the upcoming versions of the SFM system with each version implementing one of the extensions mentioned in this section Section 9.

Version	**Description**
1.0	Current version
1.1	Dike sensors
1.2	Mobile application
2.0	Providing guidance to smart (Android) vehicles
2.1	Rerouting of vehicles
2.2	Solar energy
3.1	Interacting instances of the SFM system
3.2	Drones
3.3	Real-time high-resolution image processing

Table 25: Version evolution of the SFM system

10 References and Acknowledgements

We would like to extend our gratitude to our coach, Antonis Gkortzis, for reviewing our iterations and providing feedback on our drafts of this document.

We thank the students of Team 1 (Michel Medema, Marco Gunnink, Patrick van Vreeswijk, Daan Mauritsz, Sweta Singh and Koen Roos) for providing feedback and recommendations on version 0.3 of this document.

We thank the students of Team 3 (Gerrit Eedema, Guntur Putra, Aurélie Fakambi, Joris Schaefers, Jeroen Brandsma, Wouter Menninga and Guus Klinkenberg) for providing feedback and recommendations on version 0.7 of this document.

Furthermore, we refer to the following literature:

[1] Ignasi U. (2014), *SWOT Table - how to convert to an article?* Retrieved from TeX - LaTeX Stack Exchange: http://tex.stackexchange.com/questions/177167/swot-table-how-to-convert-to-an-article

[2] *SWOT analysis* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/SWOT_analysis

[3] *ISO/IEC 9126* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/ISO/IEC_9126

[4] *ISO 25010* Retrieved from Wikipedia: https://nl.wikipedia.org/wiki/ISO_25010

[5] *2013 European floods* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/2013_European_floods

[6] Luo, T., Maddocks, A., Iceland, C., Ward, P., Winsemiu, H. (2015, March). *Worlds 15 Countries with the Most People Exposed to River Floods.* Retrieved from World Resource Institute: http://www.wri.org/blog/2015/03/world's-15-countries-most-people-exposed-river-floods.

[7] European Union - ECMWF (2015). *Global Flood Awareness System in a nutshell.* Retrieved from European Commission: http://www.globalfloods.eu/

[8] European Commission Joint Research Centre - 2014. *Global Flood Detection System - Version 2.* Retrieved from Global Disaster Alert and Coordination System: http://www.gdacs.org/flooddetection/

[9] RDS Forum - 2010. *RDS Technical Specification R14/020_2* Retrieved from RDS Forum: http://www.rds.org.uk/2010/RDS-Specification.htm

[10] Dataloggers (September 8, 2014). *SDI-12: The Technology Behind Smart Sensors.* Retrieved from CAS Dataloggers: http://www.dataloggerinc.com/content/resources/white_papers/760/sdi-12_the_technology_behind_smart_sensors/

[11] Private Cloud Ranking by Synergy Research Group Retrieved from: http://www.crn.com/slide-shows/cloud/300076702/pole-position-ranking-the-top-5-iaas-paas-and-private-cloud-providers.htm

[12] Kruchten, P. (1995), *Architectural Blueprints - The "4+1" View Model of Software Architecture.* Retrieved from: http://www.cs.ubc.ca/ gregor/teaching/papers/4+1view-architecture.pdf

[13] *H-500XL Data Logger.* Retrieved from: http://www.waterlog.com/productsdetail.php?XL-Series-Data-Loggers-3

[14] *Line-of-Sight Radio series (H-4400).* Retrieved from: http://www.waterlog.com/productsdetail.php?Line-of-Sight-Radio-Series-H-4400-11

[15] *SDI-12 Rain Gauge (H-3401).* Retrieved from: http://www.waterlog.com/productsdetail.php?SDI-12-Rain-Gauge-H-3401-6

[16] *Bubbler/Pressure Sensor (H-3553T).* Retrieved from: http://www.waterlog.com/productsdetail.php?Bubbler-Pressure-Sensor-H-3553T-5

[17] *Nile Radar Series.* Retrieved from: http://www.waterlog.com/productsdetail.php?Nile-Radar-Series-27

[18] *H-3123 Submersible Pressure Transducer.* Retrieved from: http://www.waterlog.com/productsdetail.php?Submersible-Pressure-Transducer-H-3123-21

[19] *H-2191 Telephone modem* Retrieved from: http://www.waterlog.com/productsdetail.php?Voice-Data-Modem-2191-9

[20] *Layered Application* Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/ff650258.aspx

[21] *Broker* Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/ff648096.aspx

[22] Gamma, Helm, Johnson, and Vlissides (1995). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1995

[23] *Model-View-Controller* Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/ff649643.aspx

[24] C.L. Wu and K.W. Chau (2006). *Evaluation of Several Algorithms* Retrieved from: http://www.academia.edu/240924/Evaluation_of_several_algorithms_in_forecasting_flood

[25] *Pipes and Filters* Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/dn568100.aspx

[26] *Facade Pattern* Retrieved from MSDN: https://msdn.microsoft.com/en-us/library/orm-9780596527730-01-04.aspx#Anchor_0

[27] *Watchdog timer* Retrieved from: https://en.wikipedia.org/wiki/Watchdog_timer

[28] *Fiber optics monitoring solution for canal dykes* Retrieved from: http://zhtrip.com.cn/emea/Images/pianc%20mmx%20liverpool%20217%20fiber%20optic%20based%20n 7618.pdf

[29] *Open Automotive Alliance* Retrieved from: http://www.openautoalliance.net/

[30] Kazman, R., Klein, M., & Clements, P. (2000). ATAM: Method for architecture evaluation (No. CMU/SEI-2000-TR-004). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[31] *Flood Identification Using Satellite Images* Retrieved from: http://www.ijareeie.com/upload/2014/january/4_Flood.pdf

[32] *Sharding Introduction* Retrieved from MongoDB Manual: http://docs.mongodb.org/manual/core/sharding-introduction/

[33] *Understanding How Cassandra Stores Data* Retrieved from Hakka Labs: https://www.hakkalabs.co/articles/how-cassandra-stores-data

Appendices

A1: Risk Assessment

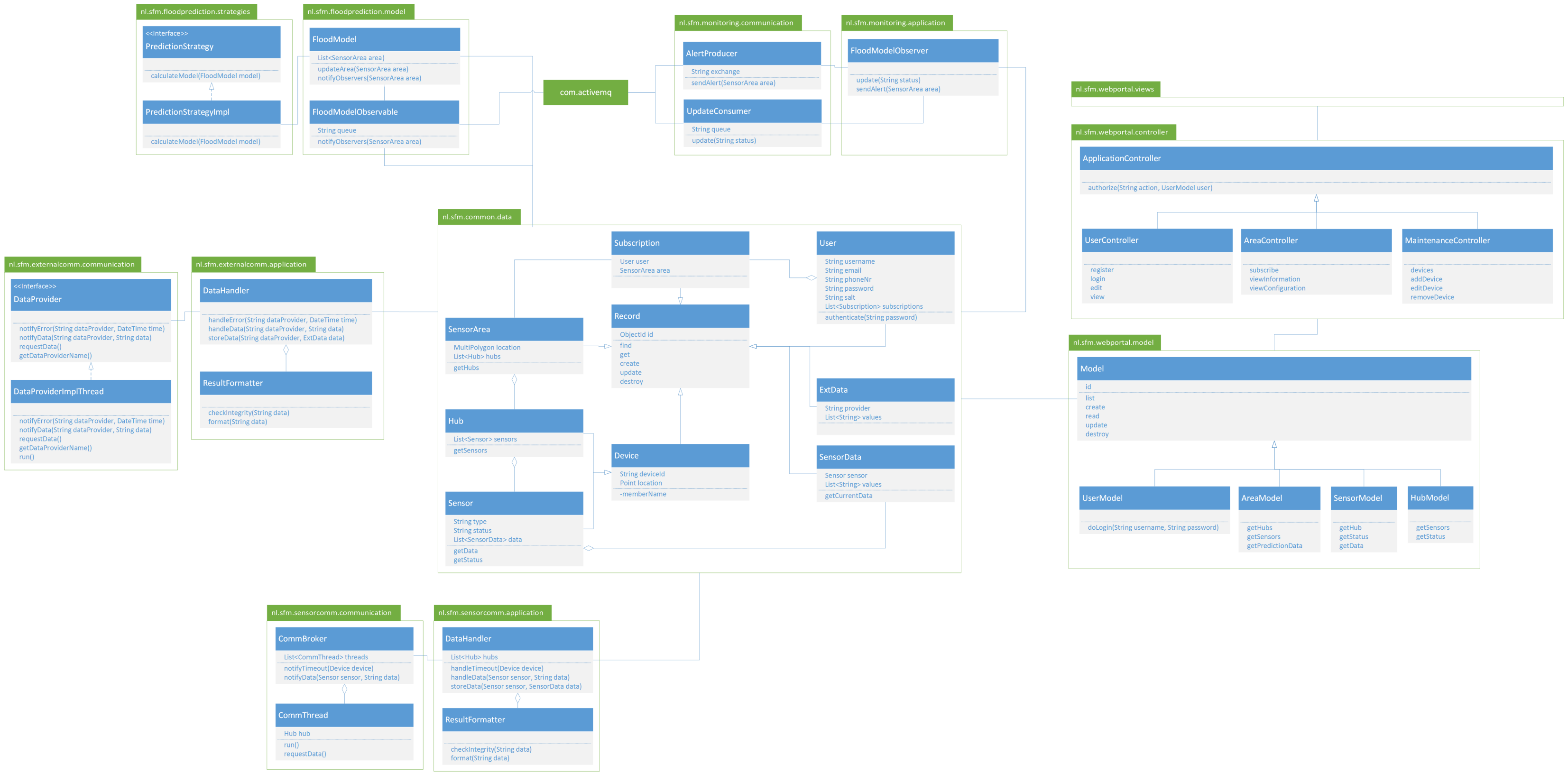
	Risk	Probability	Impact	Severity	Owner	Indicator	Prevention	Reaction
Business								
RB-1	Lack of customer interest	medium	high	high	Product owner	Too few orders in a fiscal year to maintain profitability	Market research	Think of other uses for product
RB-2	Too much competition to maintain profitability	low	high	medium	Product owner	Too few orders in a fiscal year to maintain profitability	Market research	Increase unique features and product quality
RB-3	Too large project scope	low	medium	low	Architect	Inability to reach milestones	Market research to find key features	Cut features to core requirements
Technology								
RT-1	Incorrect weather prediction	low	medium	low	Product owner	Forecast matches actual weather less than 70% of the time	Find reliable weather data provider	Switch to a different weather data provider
RT-2	Incorrect flood prediction	medium	high	high	Architect, Developer	Flood predicted but not occurring	Design reliable algorithm	Revise algorithm and retrain flood detection model

RT-3	Incorrect sensor data	medium	medium	medium	Architect, Developer	Sensor reading does not match nearby sensors and physical observation	Test sensor accuracy and fault tolerance	Replace faulty hardware or find new sensor hardware
RT-4	Failure to alert end users	medium	high	high	Architect, Developer	Alert requested but not sent	Use redundant communication methods	Re-engineer communication protocol
RT-5	Insufficient sensor communication range	medium	medium	medium	Architect	Sensors unable to communicate with sensor hubs	Research long-distance communication standards	Add signal repeaters for long distances
Implementation								
RI-1	Incompatible hardware components	medium	high	high	Architect	Components can not work together	Research compatible components	Change hardware components for compatibility
RI-2	Wrong time estimation for development	medium	medium	medium	Project manager	Inability to reach milestones	Spend time on planning, limit project or milestone scope	Cut features to reduce workload
Operational								
RO-1	Low sensor lifetime	medium	medium	medium	Architect, Developer	Sensors need replacement within 2 years	Research durable sensor hardware	Change hardware components for durability
RO-2	Lack of maintenance staff in remote areas	low	medium	low	Product owner	Inability to service remote equipment	Hire contractors in remote areas	Hire contractors in remote areas
RO-3	External weather source is unavailable	medium	medium	medium	Architect	Not receiving remote data or receiving corrupted data	Include support for multiple weather data providers	Switch weather data provider

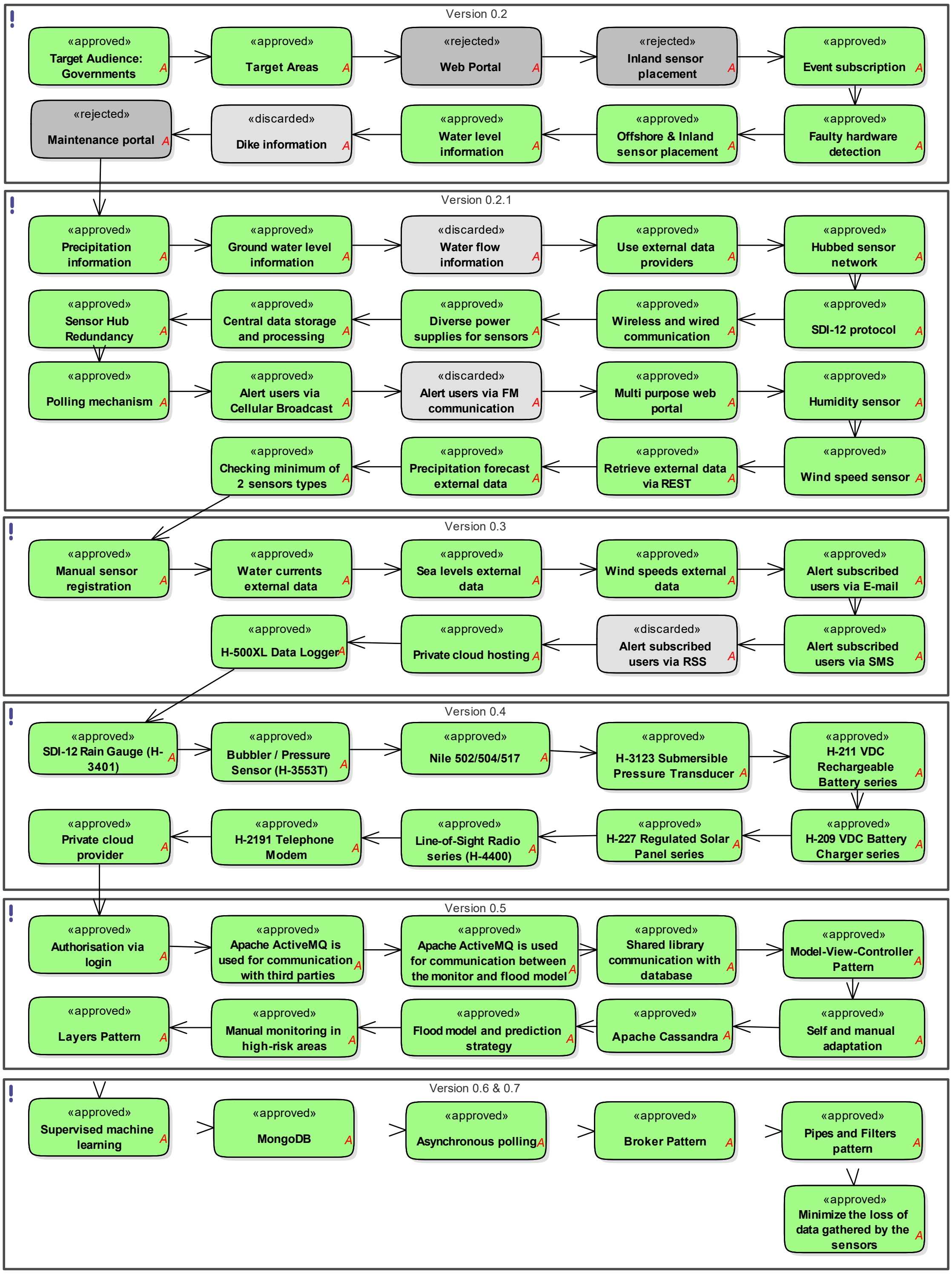
RO-4	Phone line or power outage	low	high	medium	Architect	Not receiving data from a large group of hubs	Make Service Level Agreements (SLAs) with utility providers	Manual inspection during situations with a high flood probability

Table 26: Risk severity

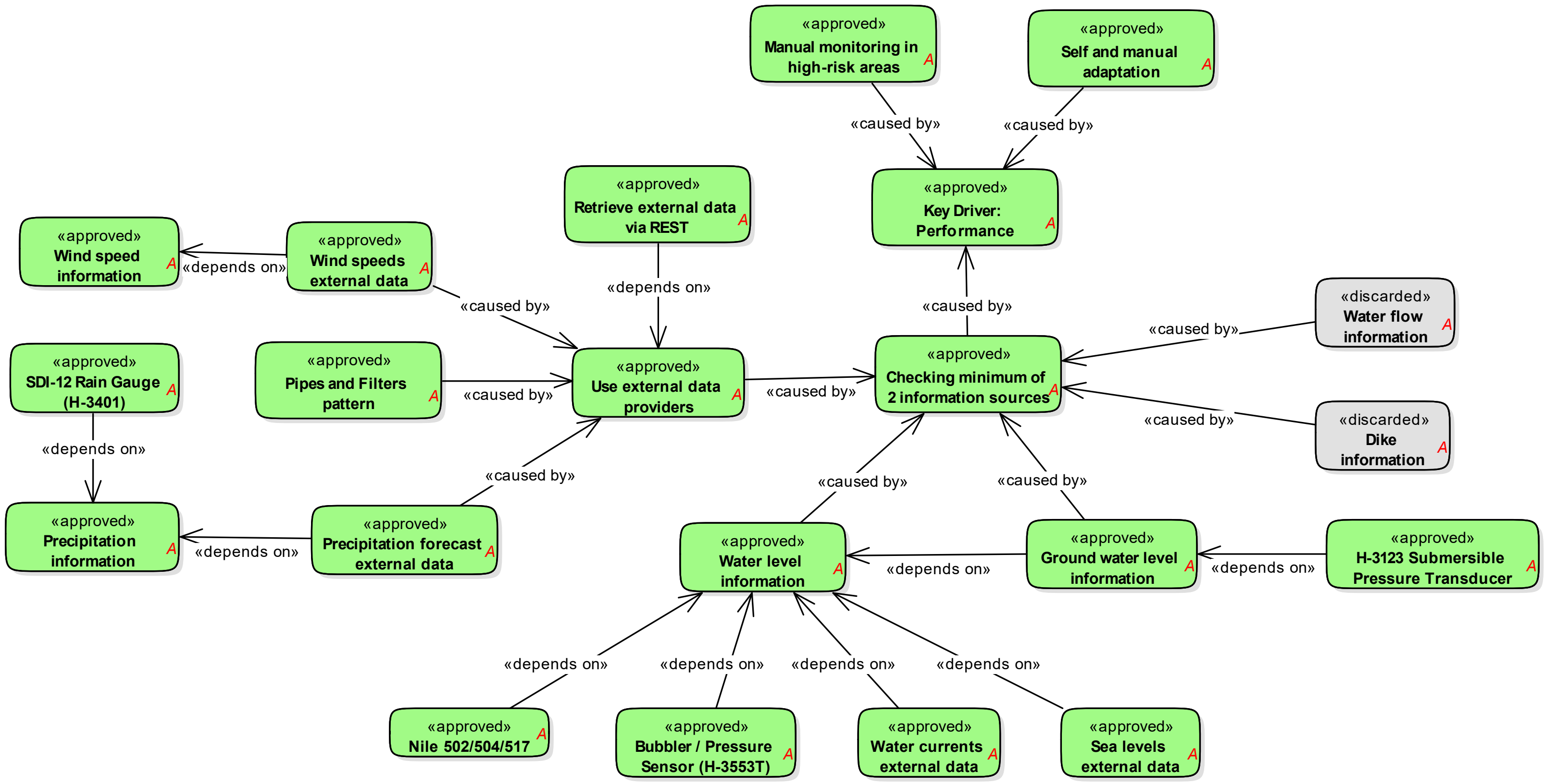
A2: Class Diagram

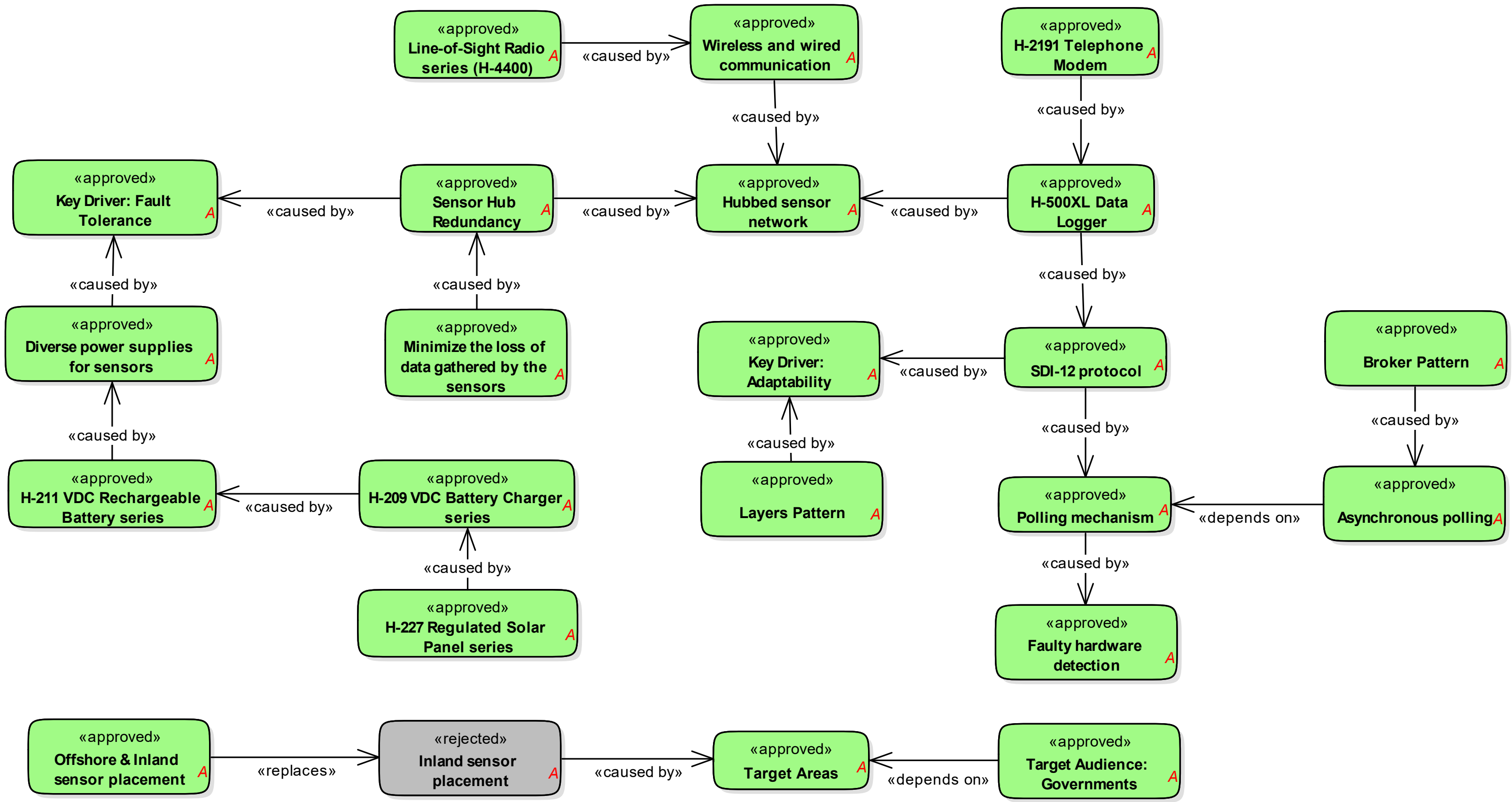


A3: Chronological decision view

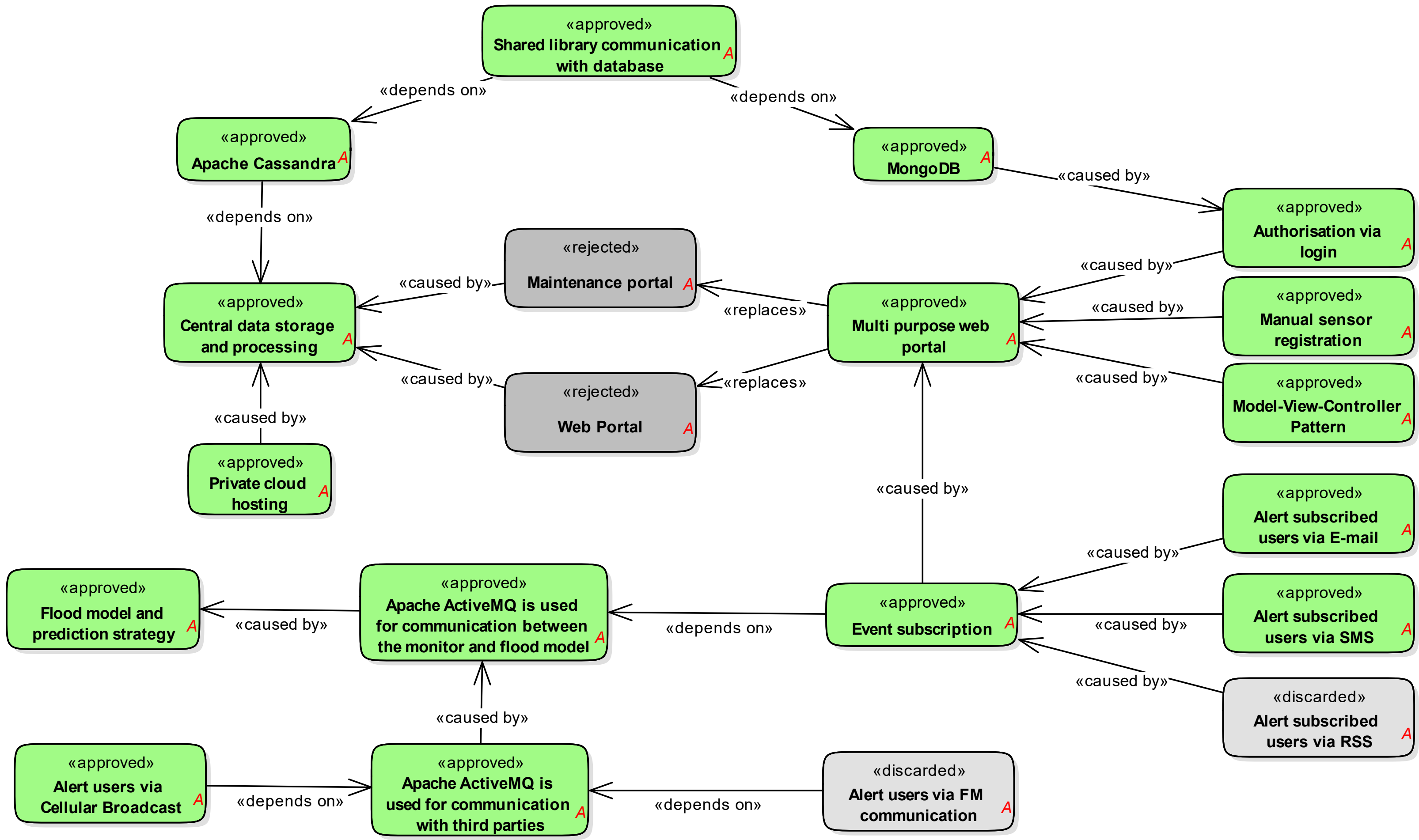


A4: Relational decision view





«approved»
Line-of-Sight Radio series (H-4400)
«caused by»
«approved»
Wireless and wired communication
«approved»
H-2191 Telephone Modem
«caused by»
«caused by»
«approved»
Key Driver: Fault Tolerance
«caused by»
«approved»
Sensor Hub Redundancy
«caused by»
«approved»
Hubbed sensor network
«caused by»
«approved»
H-500XL Data Logger
«caused by»
«caused by»
«caused by»
«approved»
Diverse power supplies for sensors
«approved»
Minimize the loss of data gathered by the sensors
«approved»
Key Driver: Adaptability
«caused by»
«approved»
SDI-12 protocol
«approved»
Broker Pattern
«caused by»
«caused by»
«caused by»
«caused by»
«approved»
H-211 VDC Rechargeable Battery series
«caused by»
«approved»
H-209 VDC Battery Charger series
«approved»
Layers Pattern
«approved»
Polling mechanism
«depends on»
«approved»
Asynchronous polling
«caused by»
«caused by»
«approved»
H-227 Regulated Solar Panel series
«approved»
Faulty hardware detection
«approved»
Offshore & Inland sensor placement
«replaces»
«rejected»
Inland sensor placement
«caused by»
«approved»
Target Areas
«depends on»
«approved»
Target Audience: Governments


«approved»
Shared library communication with database
«approved»
Apache Cassandra
«approved»
MongoDB
«approved»
Authorisation via login
«rejected»
Maintenance portal
«approved»
Central data storage and processing
«approved»
Multi purpose web portal
«approved»
Manual sensor registration
«rejected»
Web Portal
«approved»
Model-View-Controller Pattern
«approved»
Private cloud hosting
«approved»
Alert subscribed users via E-mail
«approved»
Flood model and prediction strategy
«approved»
Apache ActiveMQ is used for communication between the monitor and flood model
«approved»
Event subscription
«approved»
Alert subscribed users via SMS
«discarded»
Alert subscribed users via RSS
«approved»
Alert users via Cellular Broadcast
«approved»
Apache ActiveMQ is used for communication with third parties
«discarded»
Alert users via FM communication
«depends on»
«caused by»
«replaces»

A5: Time Tracking

Week 1

ID	Date	Task	Hours	Sum
TH	31/08/15	Setting up LaTeX environment and elaborate template document	3	**8**
	01/09/15	Refining LaTeX template and commands for future versions of document	1.5	
	03/09/15	Meeting	1	
	04/09/15	Preface and A5 template definition	1	
	06/09/15	Finalisation and submission of first draft and submission	1.5	
MK	03/09/15	Meeting	1	**2**
	05/09/15	Refining draft of section 1	1	
TA	03/09/15	Meeting	1	**4**
TA	04/09/15	First draft for sections 1, 2.4 and 3.2	3	
MV	03/09/15	Meeting	1	**6**
MV	06/09/15	First draft for sections 1 and 2.	5	
YT	03/09/15	Meeting	1	**1**
			Total:	**21**

Week 2

ID	Date	Task	Hours	Sum
TH	07/09/15	Meeting	3	**12**
	09/09/15	Research on sensor systems and networks	1.5	
	12/09/15	Applying feedback of version 0.1	1.5	
	13/09/15	Work and discussion on sections 1-3 and creating LaTeX templates for section 3	6	
MK	07/09/15	Meeting	3	**11.5**
	10/09/15	Research on sensors and sensor techniques	2	
	13/09/15	Refining draft versions of sections 3.2, 3.3, 3.5	5	
	13/09/15	Initial draft of 3.4	1.5	
MV	07/09/15	Meeting	3	**11.5**
	10/09/15	Working on sections 1 and 2	2	
	13/09/15	Working on section 2 and 3.6	6.5	
YT	07/09/15	Meeting	3	**10**
	12/09/15	Research on sensors and sensor techniques	4	
	13/09/15	Research on sensors and sensor techniques	2	
	13/09/15	Started with section 4	1	
			Total:	**45**

Week 3

ID	Date	Task	Hours	Sum
TH	14/09/15	Meeting	4	**23**
	15/09/15	Refining section 1, 2, reworking 3.2	2	
	16/09/15	Validation of 3.2 in discussion with team	0.5	
	16/09/15	Reworking 3.2, 3.3, 3.4, LaTeX templates	3	
	17/09/15	Reworking 3.4 - 3.9	6.5	
	19/09/15	Applying 0.2.1 feedback on sections 1 and 2	0.5	
	20/09/15	Discussing requirements of system and feedback of version 0.2.1, and reviewing section 3	1	
	20/09/15	Refining sections 2 and 3	3	
	20/09/15	Refining document structure, table and figure layouts and definitions	0.5	
	20/09/15	Final review and minor refining to sections 1-4	2	
MK	14/09/15	Meeting	4	**13.5**
	15/09/15	Working on use cases	0.5	
	16/09/15	Discussion	0.5	
	19/09/15	Processing feedback of version 0.2.1	1	
	19/09/15	Reviewing 3.5, 3.6, 3.7	2	
	19/09/15	Initial draft of 3.10	0.5	
	20/09/15	Discussing requirements of system and feedback	0.5	
	20/09/15	Refining 2.6 (domain model)	1.5	
	20/09/15	Initial draft of 2.8	2	
	20/09/15	Refining 2.7, 4.1, 4.2	1	
TA	14/09/15	Meeting	4	**14**
	15/09/15	Working on sections 3.5 and 3.9	1	
	16/09/15	Discussion, review, working on 3.6, 3.11 and 4.1	4.5	
	17/09/15	Working on 3.11	1	
	19/09/15	Finishing 3.11, review	1.5	
	20/09/15	Editing sections 3, 4 and 5	2	
MV	14/09/15	Meeting	4	**14**
	15/09/15	Searching for and setting up modelling tools	1	
	15/09/15	Working on section 2.4, 2.6 and 2.7	2.5	
	16/09/15	Working on section 2.3, 2.6, 2.7 and 2.9	3	
	17/09/15	Initial draft and diagram for section 3.1	2	
	19/09/15	Applying feedback of version 0.2.1	0.5	
	20/09/15	Editing diagram of 3.1, refining section 2 and 4	1	
YT	14/09/15	Meeting	4	**8**
	17/09/15	Initial draft of section 4.3	2	
	20/09/15	Working on 4.3	2	
			Total:	**72.5**

Week 4

ID	Date	Task	Hours	Sum
TH	21/09/15	Meeting	1	**13**
	21/09/15	Reviewing document of group 3 and report	3	
	22/09/15	Helping prepare for presentations	1	
	23/09/15	Finishing review report for document of group 3	2	
	25/09/15	Skype meeting to discuss review of team 1 and next iteration	1.5	
	26/09/15	Processing feedback and reviewing chapter 1-4	2.5	
	27/09/15	Discussion about section 5 and initial version	2	
MK	21/09/15	Meeting	1	**13**
	21/09/15	Reviewing document of group 3 and report 3	3	
	22/09/15	Helping prepare for presentations (financial model & practice)	2	
	25/09/15	Skype meeting to discuss review of team 1 and next iteration	1.5	
	26/09/15	Updated financial model	1	
	27/09/15	Started on section 5.2	2.5	
	27/09/15	Discussion about section 5	2	
TA	21/09/15	Meeting	1	**15**
	21/09/15	Presentation images, research into sensor communication and capabilities	2.5	
	22/09/15	Helping prepare for presentations	1.5	
	25/09/15	Skype meeting to discuss review of team 1 and next iteration	1.5	
	26/09/15	Rework decisions after feedback	2.5	
	27/09/15	Work on section 6.1, 6.2, 6.3, improve A1	6	
MV	21/09/15	Meeting	1	**14.5**
	21/09/15	Building presentation and updating diagrams	4	
	22/09/15	Prepare for presentation	1.5	
	25/09/15	Skype meeting to discuss review of team 1 and next iteration	1	
	26/09/15	Start working on section 6.1	2.5	
	27/09/15	Proceed working on section 6.1	5.5	
YT	21/09/15	Meeting	1	**13**
	21/09/15	Building presentation and preparation	2.5	
	22/09/15	Preparing for presentations	1.5	
	25/09/15	Skype meeting to discuss review of team 1 and next iteration	1.5	
	26/09/15	Processing feedback and reviewing chapter 1-4	3.5	
	27/09/15	Working on section 6.1	3	
			Total:	**68.5**

Week 5

ID	Date	Task	Hours	Sum
TH	28/09/15	Meeting	3	**14**
	30/09/15	Applying feedback of version 0.4	3	
	02/10/15	Applying feedback of version 0.4	1	
	03/10/15	Adjusting some requirements w.r.t. log-in	1	
	03/10/15	Working on sections 3-5	1	
	04/10/15	Refining sections 2 and 3	1	
	04/10/15	Working on and discussing section 5	3	
	04/10/15	Reviewing section 7	0.5	
	04/10/15	Layout for section 8	0.5	
MK	28/09/15	Meeting	3	**14**
	01/10/15	Rewriting section 5.1 and 5.2	2.5	
	02/10/15	Working on section 5	1.5	
	04/10/15	Working on section 5	4.5	
	04/10/15	Refining financial model	0.5	
	04/10/15	Working on section 7	2	
TA	28/09/15	Meeting	3	**13.5**
	02/10/15	Starting on section 7	6	
	04/10/15	Working and discussion on section 7	4.5	
MV	28/09/15	Meeting	3	**15.5**
	01/10/15	Preparatory reading for and starting on section 7	0.5	
	02/10/15	Working on section 7.2	4	
	03/10/15	Working on section 7.6	2.5	
	04/10/15	Working on section 7.2 and 7.4	5.5	
YT	28/09/15	Meeting	3	**17.5**
	02/10/15	Working on section 7	6	
	03/10/15	Working on section 7	2.5	
	04/10/15	Working on section 7	6	
			Total:	**74.5**

Week 6

ID	Date	Task	Hours	Sum
TH	05/10/15	Meeting	1	**16**
	05/10/15	Searching for information on ATAM and architecture evaluation, obtaining paper for ATAM	0.5	
	08/10/15	Setting up Enterprise Architect	0.5	
	08/10/15	Reading parts of ATAM manual	1.5	
	09/10/15	Working on layout and content of section 8	1.5	
	09/10/15	Modifications to section 3 and review section 7	0.5	
	10/10/15	Fixing installation of Enterprise Architect and getting familiar with Decision Architect	1.5	

	10/10/15	Review of sections 5-7	2	
	10/10/15	Working on section 8.1	0.5	
	11/10/15	Discussion and review of sections 6-7	1	
	11/10/15	Working on section 8, looking into examples in ATAM paper and lecture on evaluation	5	
	11/10/15	Initial layout of section 9	0.5	
MK	05/10/15	Meeting	1	**16**
	05/10/15	Creating initial Enterprise Architect project with all present decisions	1.5	
	07/10/15	Reading parts of the ATAM manual	2	
	09/10/15	Starting on section 8.1.1	2	
	10/10/15	Review of sections 5-7	2	
	10/10/15	Update on section 5	0.5	
	11/10/15	Working on section 8.1, 8.2.2, 8.2.3	6	
	11/10/15	Discussing scenarios for section 8	1	
TA	05/10/15	Meeting and updating diagrams	2.5	**14**
	09/10/15	Working on sections 6.2 and 7	3.5	
	11/10/15	Adding legend to hardware overview, working on section 7, review section 8.2	8	
MV	05/10/15	Meeting, applying feedback of version 0.5 and updating activity and process diagrams	2	**13**
	09/10/15	Working on section 7	4	
	11/10/15	Working on section 7, creating activity and sequence diagrams, review	7	
YT	05/10/15	Meeting	1	**11.5**
	11/10/15	Preparatory reading	1	
	11/10/15	Working on section 6.4	5	
	11/10/15	Refining chapter 7	1.5	
	11/10/15	Reviewing section 2, 3.1 and 5-7	3	
			Total:	**70.5**

Week 7

ID	Date	Task	Hours	Sum
TH	12/10/15	Minor layout refinements throughout document	1	**21**
	13/10/15	Meeting	1	
	14/10/15	Meeting	1.5	
	15/10/15	Modifying LaTeX templates for cross referencing	1	
	16/10/15	Discussion on software architecture and patterns	1	
	16/10/15	Setting up cross referencing (including templates) for all traceable items throughout the document (clickable references), fixing spacing/-formatting issues for all templates	3	
	17/10/15	Refining decisions, creating new decisions, working on decision views	3	

	17/10/15	Discussions about software patterns and decisions	0.5	
	17/10/15	Reviewing, refining and adding requirements for correctness and consistency	3	
	18/10/15	Working on decision views and discussion	1.5	
	18/10/15	Refining section 3	1	
	18/10/15	Refining section 8	3	
	18/10/15	Reviewing section 9	0.5	
MK	13/10/15	Meeting	1	**19**
	14/10/15	Meeting	1.5	
	17/10/15	Small refinements on section 8	1	
	17/10/15	Refining decisions, creating new decisions, working on decision views	3	
	17/10/15	Creating chronological decision view	1.5	
	17/10/15	Reviewing, refining and adding requirements for correctness and consistency	3	
	18/10/15	Creating relational decision view and refining chronological decision view	3.5	
	18/10/15	Refining section 8.1	2.5	
	18/10/15	Refining section 8.2	2	
TA	13/10/15	Meeting	1	**21.5**
	14/10/15	Meeting	1.5	
	16/10/15	Updated section 7.2.1, reworked diagrams in section 7 (add MongoDB, change External Communication architecture, change package diagram style)	6	
	17/10/15	Refining section 7.2	4	
	18/10/15	Refining section 7.2, working on relational decision view, refining section 8.1	9	
MV	14/10/15	Meeting	1.5	**13**
	16/10/15	Incorporating notes: Pattern decisions + activity diagram + sequence diagrams + architectural vision	4	
	17/10/15	Updating section 7.6 + architectural vision + reviewing/refining section 7	2.5	
	18/10/15	Refining sections 7.2 + 7.4 + 7.6 + reviewing/refining section 7	5	
YT	13/10/15	Meeting	1	**14**
	17/10/15	Applying feedback and refining section 6	6	
	18/10/15	Refined section 3.10 and worked on section 7.7 and section 9	7	
			Total:	**88.5**

Week 8

ID	Date	Task	Hours	Sum
TH	00/00/00	...	0	...
MK	00/00/00	...	0	...
TA	00/00/00	...	0	...
MV	00/00/00	...	0	...
YT	00/00/00	...	0	...
			Total:	

Week 9

ID	Date	Task	Hours	Sum
TH	00/00/00	...	0	...
MK	00/00/00	...	0	...
TA	00/00/00	...	0	...
MV	00/00/00	...	0	...
YT	00/00/00	...	0	...
			Total:	

Week 10

ID	Date	Task	Hours	Sum
TH	00/00/00	...	0	...
MK	00/00/00	...	0	...
TA	00/00/00	...	0	...
MV	00/00/00	...	0	...
YT	00/00/00	...	0	...
			Total:	

Total logged time

ID	Author	Hours
TH	Thomas Hoeksema	107
MK	Maarten Kollenstart	89
TA	Toon Albers	81
MV	Marijn Vonk	87.5
YT	Ynte Tijsma	75
	Total:	**439.5**