SOFTWARE ARCHITECTURE

ARCHITECTING PROJECT

GROUP 1

Flood Monitoring Tool

Version 1.3

Authors:
Michel MEDEMA (S2396009)
Patrick VAN VREESWIJK (S2019507)
Koen ROOS (S1931741)
Daan MAURITSZ (S2586274)
Marco GUNNINK (S2170248)
Sweta SINGH (S2209381)

Supervisor:
Antonis GKORTZIS

Teacher:
Prof. Paris AVGERIOU

September 20, 2015

Revision History

Version	Author	Date	Description
1.0	Michel Medema	03-09-2015	Setting up the document
1.0	Michel Medema	06-09-2015	Introduction & Context & Stakeholders
1.0	Sweta Singh	06-09-2015	Introduction & Context
1.1	Patrick van Vreeswijk	09-09-2015	Business Information & Bibliography in Latex
1.1	Michel Medema, Sweta Singh, Daan Mauritsz, Marco Gunnink	10-09-2015	Functional Requirements & Stakeholders
1.1	Michel Medema, Sweta Singh, Daan Mauritsz, Marco Gunnink	11-09-2015	Technical non-Functional Requirements & Commercial non-functional Requirements & Key Drivers
1.1	Patrick van Vreeswijk	11-09-2015	Business Information
1.1	Michel Medema, Marco Gunnink	13-09-2015	Use-Cases
1.1	Sweta Singh	13-09-2015	Architectural Vision
1.1	Michel Medema	13-09-2015	Risk Analysis
1.1	Michel Medema, Marco Gunnink, Sweta Signh	13-09-2015	Document review
1.2	Patrick van Vreeswijk	15-09-2015	Updated Business Information
1.2	Michel Medema	17-09-2015	Stakeholders, Key-drivers, Risk Analysis, Requirements, Evolution Requirements
1.3	Michel Medema	19-09-2015	Stakeholders, Key-drivers, High-level Requirements
1.3	Michel Medema	20-09-2015	Architectural Vision, High-level Requirements, Use Cases, Requirements, Evolution Requirements, Risks, Analysis, Review
1.3	Koen Roos	20-09-2015	Business Model, Updated Product Description

Contents

Glossary

API Application Programming Interface.

NAP Amsterdam Ordnance Datum.

UAV Uninhabited Aerial Vehicle.

1 Introduction

This document describes the architecture of a Smart Flood Monitoring system that monitors various properties of dykes and waterways, which are used to detect imminent floods. When the system detects that a flood is imminent, it will issue warnings to the appropriate authorities and the local residents to help reduce the number of casualties and reduce the damage caused by floods.

The structure of the rest of the document is as follows. First, the context of the system that is to be developed is given in the *System Context*. In the *Business Information* section the architecturally relevant business information is given. The *Requirements* section states the various stakeholders along with their concerns and the requirements that are obtained from these stakeholders. In the *Analysis* section, various design alternatives will be discussed in terms of the advantages and disadvantages of these alternatives. The *System Architecture* section will give a high-level overview of the system architecture in terms of the various components of the system. The *Hardware Architecture* will describe the hardware platform, on which the system will be built. In the *Software Architecture* section, the software design will be described. The *Architecture Evaluation* will verify the described architecture based on the provided requirements and will describe possible improvements that can be made. Lastly, the *Evolution* section will describe how the system will evolve over time, to keep providing valuable services to its customers and users.

1.1 System Context

Floods are a commonly occurring phenomenon that can cause massive problems such as the destruction of homes, infrastructure and farmlands, and can even result in the loss of lives [3]. Floods are caused by a variety of different factors, including heavy rainfall, hurricanes and rising sea levels, which is especially problematic for the parts of countries that lie below sea level. The problems caused by these floods are aggravated every year due to climate change.

The early detection of floods can help mitigate the damage and casualties that result from these floods. With the emergence of new sensor technology and other smart techniques, monitoring a natural environment for potential floods in real time has become possible. The Smart Flood Monitor utilises many different technologies such as sensors and aerial and terrestrial vehicles, to monitor various environments such as dykes, waterways and rivers for potential floods. Multiple technologies are used that monitor a variety of different activities and properties to provide more accurate monitoring capabilities.

When the Smart Flood Monitor detects an imminent flood, it can issue warnings to the appropriate authorities as well as the residents in and around the area that will be flooded. Issuing early warnings gives the residents a chance to evacuate the area, which in turn could drastically reduce the loss of lives as a result of floods. Besides issuing warnings, the system is also capable of providing guidance when a flood has occurred. The system is aware of the areas that are flooded, and can provide alternative routes to the emergency services and residents avoiding the flooded areas.

2 Business Information

This section illustrates the business environment in which this system will operate and any architectural relevant business information. The vision statement outlines where we want to be and the mission directs into how we will achieve this and suffice to the customers needs. We then move on to the business rationale, in which we explain the choice of geo-region and also the type of floods we focus our environmental monitoring on.

2.1 Business Vision

Our company aims to limit the social and financial consequence of floods and most importantly avoid the loss of human lives. This can be achieved by accurately predicting floods, so that measures can be taken ahead of time.

It must be mentioned that although our focus first lies on floods, we hope that our environmental monitoring system will be the 'next generation' and that new functionalities such as monitoring water pollution or detecting extreme weather phenomena can be seamlessly added in the future.

2.2 Business Mission

In order to achieve our vision we need to have a proper mission aligned. Fullfilling our vision requires us to reduce the likelihood of flooding and reduce the impact if a flood occurs. In trying to do this, our mission is to do what we are are best at: We want to offer a pioneering (software) product that will revolutionise the market of environmental flood monitoring. We start with a pilot that offers the next generation of environment monitoring system which can be upscaled gradually with more complicated sensors and vehicles. The system will be integrated with third party systems (such as emergency services). With this collaboration we hope that also other companies, parties, universities can help us to achieve our vision.

Our team is dedicated in knowledge and skill related to software development. For that reason our focus lies on the development of software and, therefore, for the hardware components of our system we will only use tested and proven existing hardware.

2.3 Business Rationale

The company, based in the Netherlands, first intended to develop a product for the Dutch (home market). However, the Dutch government has learned its lessons in the past and increased its flood protection to an enormous extent. For example, thanks to this elaborate system the chance for the city of Rotterdam exceeding the Amsterdam Ordnance Datum (NAP) limit is reduced from once every 100 years on average to once every 10,000 years on average. The failure risk of the barrier itself around the city is only once in 10,000,000 years [4]. These striking figures have led us to believe that the market in the Netherlands is already saturated and the population and officials are ready and protected. For that reason our focus has gone to a different western European market, namely: the UK.

The recent floods in 2014 (but also the 2007 floods) in the UK show that the country has potential to grow in the field of environmental warning systems. If we look at the figures from the UK Environmental Agency report [1] and an article from The Guardian [**?**]:

- Around 5.2 million properties in England, or one in six properties, are at risk of flooding.
- More than 5 million people live and work in 2.4 million properties that are at risk of flooding from rivers or the sea, one million of which are also at risk of surface water flooding.
- 2.8 million properties are susceptible to surface water flooding.

We, for the above reasons, feel that our technology can be better put to trial in the United Kingdom.

The most common types of floods in the UK are river flood, coastal flooding, surface water flooding, sewer flooding and groundwater flooding. Our product does not take into account all types of floods. To illustrate we mentions the types of floods and why some of these are related to our system.

- River flooding - occurs when a river cannot cope with the water draining into it from the surrounding land. This is for our system the most relevant type of flood which we want to put our focus on.
- Coastal flood- is a result from a combination of high tides and stormy conditions at sea. This often leads to a tidal surge causing significant flooding. The combination of high tide and a high river output can cause a flood further up the river. This type of flood maybe should be taken into account. But in terms of further coastal floods, satelite systems and sea-bouys are already in place to track the conditions of large areas of the ocean.
- Surface water flooding - heavy rainfall in short amount of time. Difficult to pinpoint and to predict accurately. Therefore, this will not be included our system.
- Sewer flooding - Occurs when sewers get blocked from the heavy (local) rainfall in the area. Land and water can be contaminated with raw sewage which can have contaminated water in which pollution might be imminent. Althought the blockage of the sewer and the flood is not intresting for our system, the pollution is.
- Groundwater flooding - This is when water in the ground rise above the surface levels. Geogrpahic dislocation of permeable rocks, aquifers due to local sand or river gravels can be a cause of this. Also this kind of flood is not relevant and taken into account for our system.

2.4 Product Description

Our product is a system that uses a combination of radar imaging, sensor input and weather data to show flooded areas and areas that are in danger of becoming flooded. Radar imaging is done by using Uninhabited Aerial Vehicle Synthetic Aperture Radar (UAVSAR), which is specially developed to see how the state of an area changes over time. A combination of sensors measure the current state of the rivers and canals, and the weather data is used to predict future floods.

Water Management Officials are provided with a dashboard that allows them to closely monitor flood risks. When a flood is imminent, a warning goes out to the government and emergency services. The government can then warn residents by using a messaging service, and emergency services can take appropriate prepatory actions. When the flood occurs, the emergency services and residents are provided with real-time information about the flood, so that they can act accordingly, which should reduce the damage caused by the flood.

Residents can also access an API, which is a simplified version of the dashboard that the Water Management Officials receive. This gives them an overview of the flooded areas, so they do not only know the status of their own area, but also of surrounding areas, which is especially useful for people who have to travel.

2.5 Target Audience

The target customers for our product are governments who are willing to invest in preventing damage caused by floods.

Our pilot product targets people who live in areas that are prone to (river) flooding. Since floods affect all people in a specific area, our product should be usable by every citizen, regardless of age, gender, education level, etc.

2.6 Business and Domain Model

In this section, a schematic overview of both the Business and the Domain Model are given. This is done to provide the reader with a clearer view of the system. Currently, there is no Domain Model available. The Business Model is shown in Figure 1.

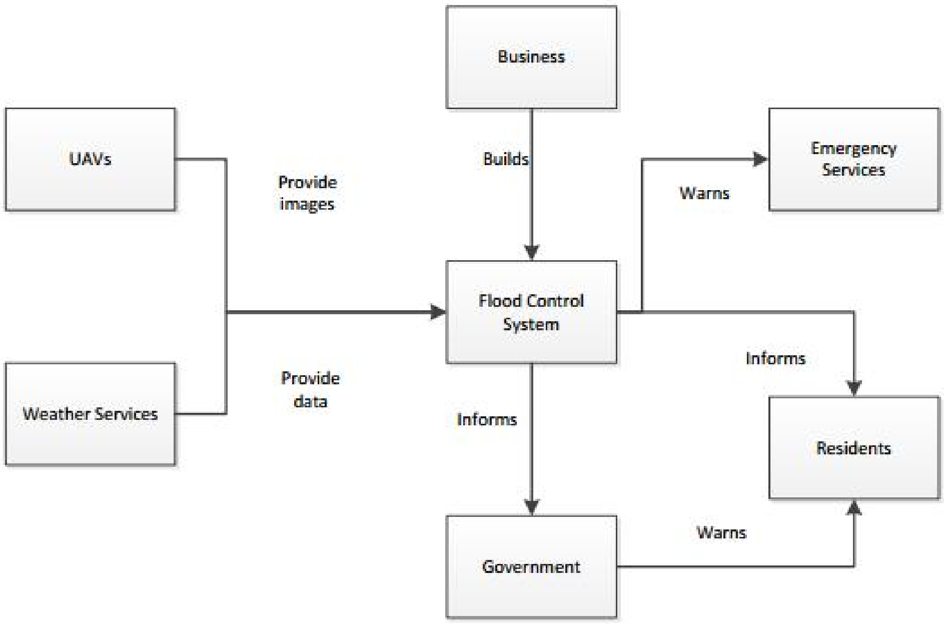

Figure 1: Business Model

As can be seen from this figure, our business builds the system. The UAVs provide images to the system, whereas the weather services provide the weather data that is used. The system informs the governments through the use of a dashboard, and informs residents through the API. The system also warns the emergency services when floods are imminent. Finally, the government warns the residents through an alert system.

2.7 Roadmap

Our system begins from ground zero. Building it up will take significant amount of skill, knowledge and of course time. In this section we illustrate how we want to achieve this. Diagram 5 gives a good illustration and summary on how we are intending to do this.

The intention is to first start collaboration on a regional scale in the UK. Finding authorities that are intrested in supporting us in creating this revolutionary system. Beta tests and close contact with all relevant stakeholders will prove whether our system is ready to go to the second phase. The first phase holds no financial benefits for the firm and has no major impact on preventing floods and reducing consequences of floods, yet. From 2020 onwards, we intend to start to divirge from the pilot model. Basic functionality should be tested and proven so that it can be delivered to all over the UK market. Collaboration is started together with third parties to add more functionality to the system. In the third 'Growing Phase' also the more complex functionality is tested, markets have the product to be sufficient enough and the selling onto larger, world wide markets is started. The benefit of this is that due to mass production also customers that have less money to spend can utilize the system. Hence, helping our final phase which is to complete our vision and reduction the consequences, social, financial and human loss of lives during floods.

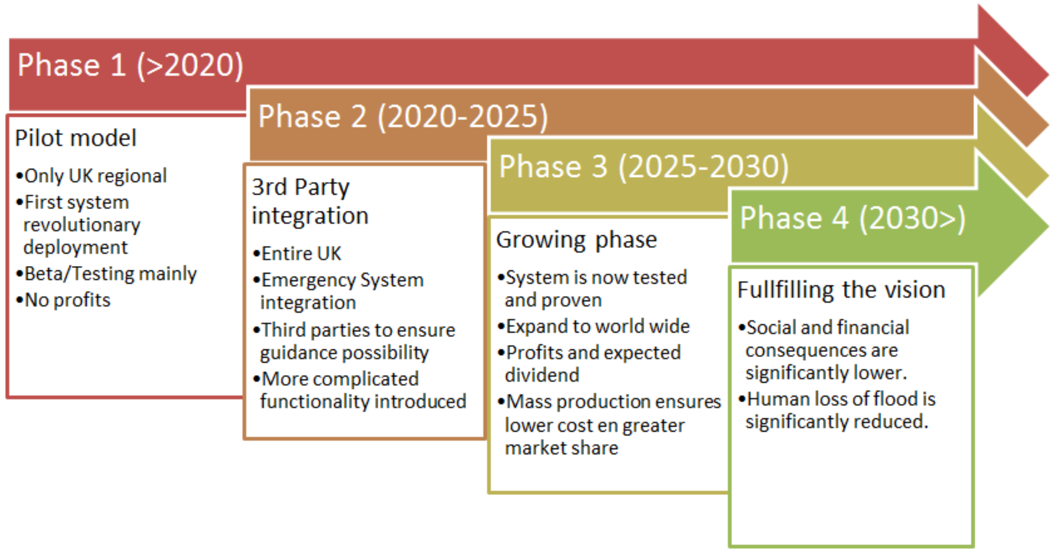

Figure 2: Roadmap for the Environmental System

2.8 Financial Model

2.8.1 Development costs

Will be updated as soon as the initial architectual model is finished.

2.9 Flood prevention costs

Repairing a house after a flood can cost (UK) between £10,000 £50,000 depending on the flood depth. More financial data is being gathered to see how much can be spend on the system.

2.9.1 Two different subscription services

The revenues from our system will be derived not only from the selling price of the product, but also from maintance and service costs. However, we want to give the customer the benefit of two different types of service models. The customer can choose to go with on a annual/yearly subscription base or a more liscense based type of service.

Subscription	Licensed
Fees (Monthly/Yearly)	Initial License
Up/down-grade flexibility	Additional Add-on License (from third-parties)
24/7 Customer support	Maintenance Fees
More benefits comparisons	?

Table 1: Comparison of the two service models

The benefit our customers can have with a monthly recurring revenue service is its pricing flexibility. In developing several entry-level services we can let specific areas upgrade or downgrade their environmental monitoring to the situation which fits best for them. An example would be in more urban areas which are more densily populated, here more better prediction with the use of (eg.) more standby UAV's. Furthermore upscaling the system to cover more area is easy. Investment is not neccersary, customers can pay upfront or spread out the payments into any interval they wish.

With the license model customers need to do an upfront investment but this will give them extra potential when they later want to upscale their system, keep their data-inhouse or make adjustments to their architecture which our company has less expertise in. The disadvantage is that the customer has to have their own servers, do regular maintanence and make sure all the sensors and the infrastructure is always running smoothly to ensure accurate predictions.

2.10 Competitors

Currently, there is only one known competitor within the UK, which is the Flood Forecasting Centre (FFC). The Flood Forecasting Centre is a partnership between the Environment Agency and the Met Office.

3 Requirements

This section gives an overview of all the requirements of the flood monitoring tool. First, a detailed description of the system and its functionality is given. After that, the stakeholders are presented along with the main concerns for each stakeholder. Based on the stakeholder concerns, the key-drivers for the system are defined. Next, the Use Cases and the functional requirements of the system are described, followed by a list of non-functional requirements along with a detailed description of how these affect our system. Finally, an overview is given of the most important risks that are accociated with the business and the product.

3.1 Architectural Vision

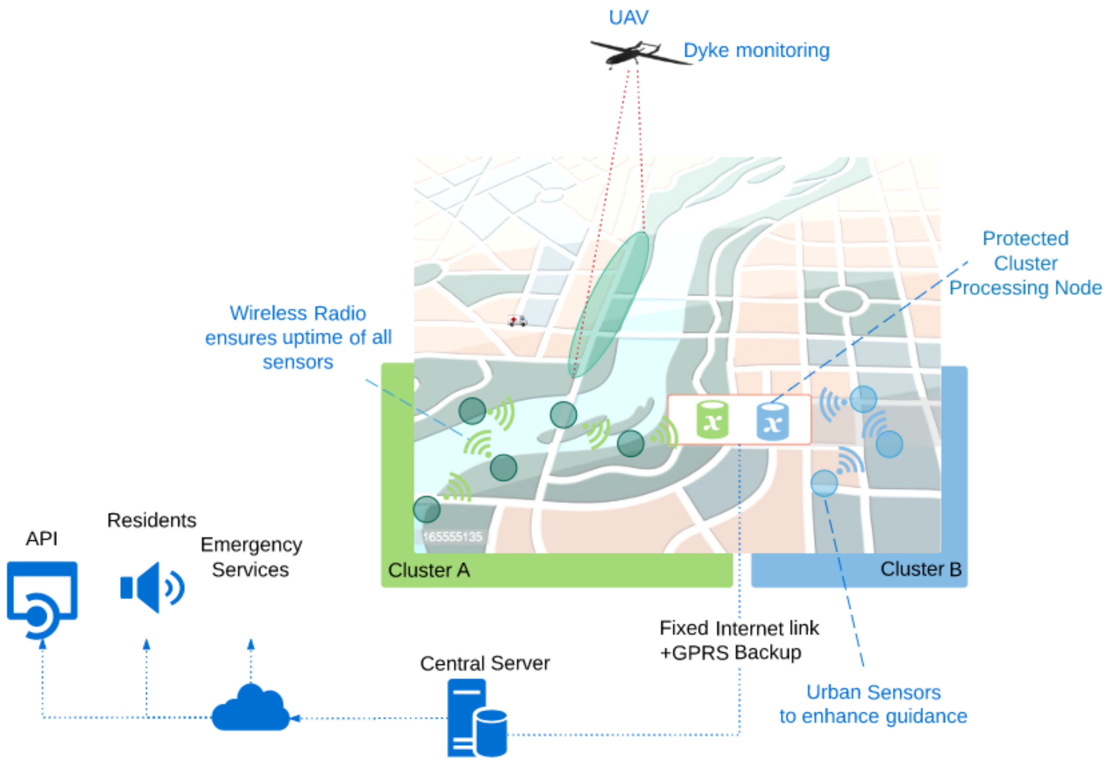

Figure 3: Initial system architecture sketch. *Background sketch taken for educational purposes from: http://www.gettyimages.nl/detail/illustratie/map-river-city-royalty-free-illustraties/165555135*

Sensors are installed in the flood sensitive areas of the UK. These areas are split up into 'clusters' (see fig 3). The sensors are interconnected with Zigbee, and the measurements of the sensors are sent to a processing node also located within the cluster. Zigbee allows the sensor nodes to sent their data to a processing node, even when the sensor node cannot directly reach the processing node. The protocal is developed in such a way, that a destination can be reached through the other nodes in the cluser. All that is required is that the cluster contains multiple processing nodes for redundancy, in the case that one of the processing nodes fails.

The processing nodes are connected with a direct wired network link (with some alternative backups) to the central server. The processed (and now filtered) information is sent to the central server where the data is being analysed. With the use of a model the probability of a flood is determined. If the determined probability exceeds a certain threshold, a warning is sent to the emergency services, which propagate the alerts through their system.

Our system does not only issue warnings in case of emergency, but it also provides guidance to the residents of the affected areas. With the use of another cluster of sensors, the system is able ot identify flooded areas. Whenever a flooded area is detected, the information is made available through an Application Programming Interface (API). We hope that with this API and in collaboration with a third party that the information from our system can be displayed on a map, or residential application. Here the guidance will be in the form of evacuation routes.

The system also provides an overview of the collected data and the results from the analysis of the data in a dashboard. The dashboard gives the water management officials the possibility to monitor the system as well as the waterways and dykes. Furthermore, the dashboard informs the water management officials about sensors that are not working properly and gives them the posibility to manually issue a flood warning to the emergency services and the local residents.

The application for residents is left to third parties, because our data can be used for guidance in so many different ways. It can be sent to the navigation in cars (similar to TomTom's traffic-jam notifications) or to Google Maps on mobile devices for pedestrians. Also, local officials who want their own application or guidance system (with their own suggested routes in combination with our information) can do so through the API.

3.2 Stakeholders

This section lists all of the stakeholders of our system and states the most important concerncs for those stakeholders. The concerns of the stakeholders are expressed as quality attributes that the system has to meet, which are defined by the ISO/IEC 25010 standard [2].

A stakeholder is a person, group of persons or an organisation that affects or is affected by the actions of product owner. The concerns of different stakeholders are varied in nature depending on their involvement in the product and process. Therefore in our architecture, they have been prioritised as High, Medium or Low. The stakeholders that are given a high priority have a direct concern for the system because they either own the product or it is critical to them that the system is operational. The stakeholders with a medium priority are the stakeholders for whom the system is not primarily intended, but that greatly benefit from the system or have to work with the system frequently and prolonged. Finally, the stakeholders that only work with the system for a short amount of time or for whom it is not critical that the system is operational are given a low priority.

High Priority Stakeholders

Product Owner The main concerns of the product owner are the usability, reliability and profitability of the system. The product owner invests in the development of the system and needs a usable and reliable product that can be sold. The product owner is responsible for maximizing the value of the product and work of the development team.

Water Management Officials The water management officials are the end users of the system. They perform the inspections of waterways and dykes, and have to monitor the dykes in case of a flood. The dashboard that the system provides can help the water management officials with their tasks. Their main concerns with respect to the data provided by the dashboard are the reliabity, accuracy and real-time performance of the system.

Emergency Services The emergency services need to be informed about where and when floods take place. They can use this information to evacuate the areas at risk and take other precautions to minimise the damage of a flood or to prevent it entirely. Therefore, the main concerns of the emergency services are reliability and real-time performance.

Medium Priority Stakeholders

Local authorities Lead in reducing risks from development in the floodplain and management of drainage. They are also leading in the emergency planning for flooding and handling of the recovery areas that have been affected by flooding. They will have to work with the system and (for example) use the prediction model to see where further improvements can be made.

Maintainers The maintainers have to maintain the system after it has been deployed, and are mostly concerned with the maintainability of the system. The maintainers have to inspect the physical part of the system, replace existing hardware, make changes to the system and possibly expand the system.

Residents The residents living in the area that is being monitored by the system. Their primary concern is reliability, since they want to be assured that they will be warned in time when a flood is imminent. Another concern of the residents is the accuracy of the system, because they do not want to evacuate the area when there is no flood imminent. Finally, in the case of a flood, residents that still remain in the flooded area also want to be informed about the parts of the area that have been flooded.

Insurance industry Insurance companies (in the UK the*Association of British Insurers*) and its members are vital in providing cover and handling claims for damages caused by a flood. They have an agreement with the government of the UK to commit to continue insurance coverage for most properties, even when they are at high risk. In return, the government is undertaking action to reduce this risk. For that reason, our system can provide the neccersary mapping of flooded areas, where claims can therefore be expected. Furthermore, using our predicition model the insurance industry can force the government to take more corrective action. Hence fulfilling the vision and mission of our project.

Low Priority Stakeholders

Third Party Users The third parties that want to make use of the data that is being provided by the system through an API. They are mostly concerned with the reliability, compatability, accuracy and real-time performance of the system. Compatability is a concern, because the API has to be integrated into their own system. Reliability is important, because the third party users cannot obtain any data when the API is not available. Accuracy and real-time performance affect the data that is being provided to the third parties and how often new data becomes available.

Developers The developers are mostly concerned with the modularity of the system, since this makes it possible to build the system by developing the separate components. The modularity also affects the testability and maintainability of the system.

Testers The main concerns of the testers are the reliability, real-time performance and security of the system, because these attributes affect the testability of the system.

The key-drivers of the system are identified based on the concerns of the stakeholders. The key-drivers indicate the most important quality attributes that the system has to meet and they have a significant influence on the architecture of the system. Table 2 shows the importance of the concerns, defined in the ISO/IEC 25010 standard [2], for all of the stakeholders.

A number of points is assigned to the concerns that are of the most importance to the stakeholder. The different priorities assigned to the stakeholders above influence the number of points that can be divided. Stakeholder with a high priority can assign a total of 15 points, stakeholders with a medium priority can assign 10 points and the low priority stakeholders can assign 5 points in total.

Stakeholders	Concerns: Functional Suitability	Performance Efficiency	Compatibility	Usability	Reliability	Security	Profitability
Product Owner	2	1			2		10
Water Management Officials	3	3		2	6	1	
Emergency Services	4	3		2	6		
Local Authorities	3	2			5		
Hardware Maintainers					10		
Residents	3	2			5		
Insurance Industry	3	2			5		
Third Party Users	1	1	1		2		
Developers	1	1	1	1	1		
Testers		2			2	1	
Total	**20**	**17**	2	5	**44**	2	10

Table 2: Key Stakeholder Concerns

3.3 Key-drivers

Table 2 gives an overview of the importance of every non-functional requirement for the various stakeholders. For the high-priority stakeholders, 15 points are distributed over the quality attributes. For the medium-priority 10 points are divided. And the low-priority stakeholders get 5 points. The points indicate the amount of concern that stakeholder has for that attribute. The sum of the points for each quality attribute indicates the priority of that attribute for our system.

From the table it is clear that, according to the stakeholder concerns, reliability is by far the most important quality attribute of the system. Besides reliability, both *Functional Suitability* and *Performance Efficiency* are also considered to be key-drivers for the system. These attributes are the next highest ranking and have been assigned at least some points by almost all of the stakeholders. Therefore, these attributes are also considered as key-drivers for the system, which results in the following key-drovers for the system:

1. **Reliability**
2. **Functional Suitability**
3. **Performance Efficiency**

No additional key-drivers are selected, because the system has to meet the requirements of all of the key-drivers. Ideally, all of the stakeholder concerns should be considered as key-drivers, however, since this would greatly complicate the architecture of the system and due to time constraints it is not possible to consider all of the conerns.

3.4 High-level Requirements

This section describes all of the high-level requirements of the system, which specify the main functionality that the system will provide. The high-level requirements presented here will be used to determine the functional requirements in the *Functional Requirements* section.

HL-01	**Must**	**Gathering Sensor Data**

The system has to be able to monitor various properties of waterways and dykes. Monitoring these properties shall be done with the use of sensors. These sensors will take measurements periodically and send the obtained data to a processing node. The processing node will forward the data to a central server, where the data is stored for later processing. Besides the sensor data, the system also needs data that indicate the future weather conditions, which is obtained from external third parties.

HL-02	**Must**	**Inspecting Dykes**

The system shall be able to detect irregularities in dykes. The dykes shall be monitored by UAVs that will periodically fly over a dyke and take photographs of that dyke. These photographs are then sent to the central server where they can be processed.

HL-03 **Must** **Analysing Obtained Data**

The system shall use the obtained sensor data to periodically calculate the probability of a flood. The data that is obtained is used in a mathematical model that can calculate the probability of a flood based on the obtained measurements.

Besides the sensor data, the system shall use the weather data to predict possible floods in the near future. When heavy rain is expected and the measurements obtained from the sensors show that the water level of a waterway is already quite high, the system is able to notify the water management officials of this, which allows them to take precautions if necessary.

Finally, the photographs of the dykes are used to detect possible irregularities in a dyke. An image processing algorithm can analyse these photographs and can detect such irregularities.

HL-04 **Must** **Issuing Warnings**

After the system has calculated the probability of a flood, the calculated probability is compared to two predefined thresholds:

> *definite-flood*: The probability of a flood is high enough to warrant immediate evacuation. When the determined probability exceeds this threshold the system sends warnings to emergency services and local residents.
>
> *possible-flood*: There is a possibility of a flood, but not high enough to warrant immediate evacuation. When the determined probability exceeds this threshold, but not the *definite-flood* threshold, warnings are only sent to Water Management Officials. They can then issue a warning manually.

In all cases, the system shall also store a record of the warnings that have been issued.

HL-05 **Must** **Detecting Flooded Areas**

In case of a flood the system shall identify the parts of the area, which is being monitored by the system, that have been flooded. Indentification of the flooded areas is done with the use of sensors that are placed in the area that has to be monitored. The system does not provide an overview of the flooded areas, instead the data is made available for third parties to use.

HL-06 **Must** **Exposing Data through an API**

The system shall make the data that is gathered from the various sensors available through an API. The calculated probability of a flood as well as the flooded areas that have been detected by the system shall also be made available here. This allows third parties to make use of some of the data that the system gathers.

HL-07 **Must** **Information Dashboard**

A dashboard shall be made available to the water management officials where they are provided with various kinds of information. The dashboard shall show a history of all the calculated probabilties of floods, a history of all of the warnings that have been issued, and current and previous sensor failures. Furthermore, the dashboard shall also allow the water management officials to manually issue a flood warning to both the emergency services and the local residents.

3.5 Use-Cases

This section gives an overview of all the architecturally significant Use Cases. For each Use Case, a short description is given, followed by the basic flow of the Use Case. The alternative flow specifies the alternative steps that can be performed at a certain point during the basic flow. The preconditions state the conditions that have to be met before the use case can be performed and the post conditions describe the conditions that apply after the Use Case has been performed. Finally, the related requirements indicate which requirements are fulfilled by the Use Case.

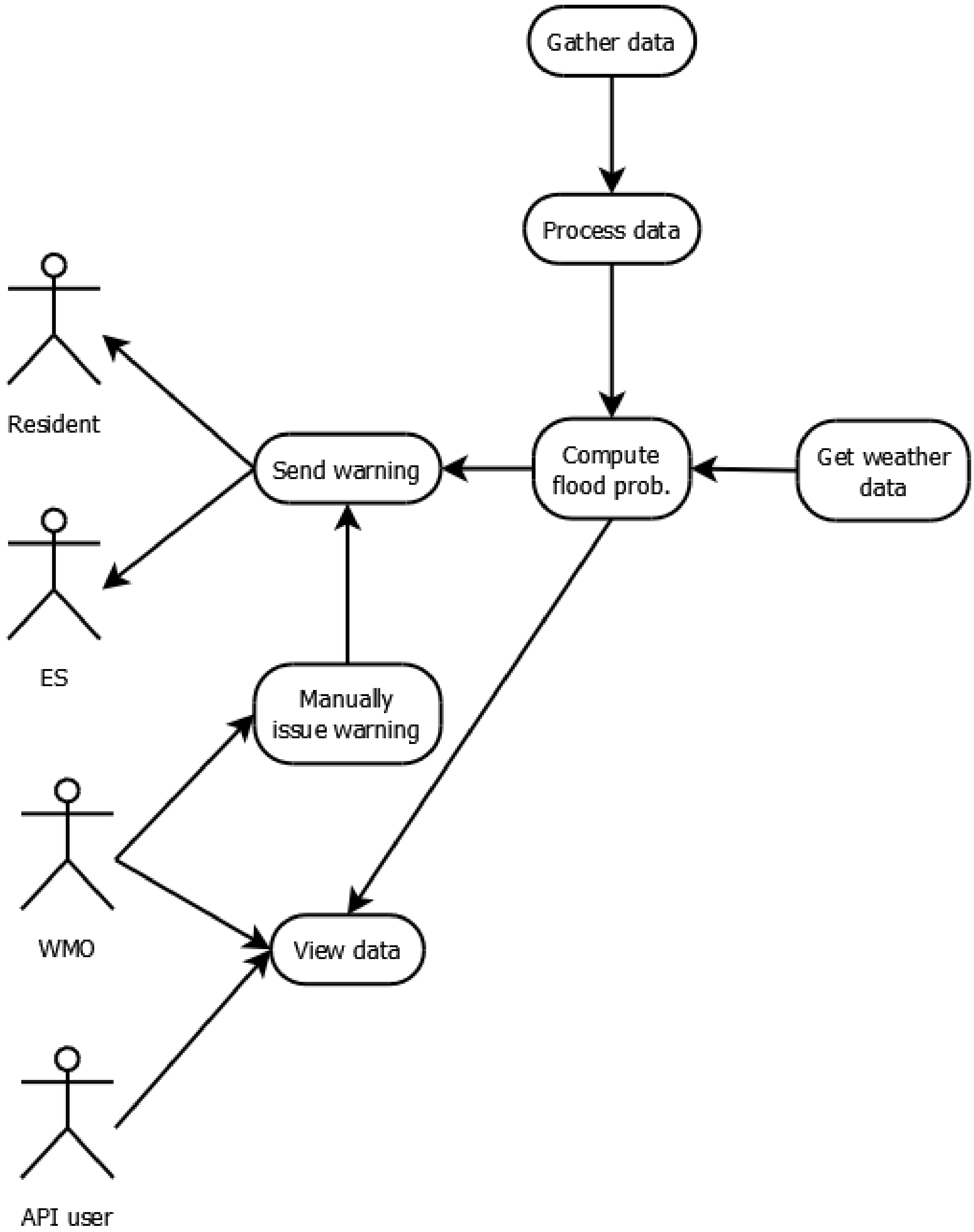

Figure 4: Use Case diagram

3.5.1 UC-01: Gathering Sensor Data

There are different types of sensors installed in the coastal areas and in the urban areas, that monitor the various properties of the dykes and waterways, and are also used to detect flooded ares. A specified interval determines how often the sensors perform a measurement. After the measurement has been performed the data is normalized by the sensor before it is sent to a processing node. The processing node collects the data of multiple sensors and sends this data to the central server, where it is stored for later processing. The stored data is also made available through the API.

Primary actor: The system

Basic flow

1. A specified interval has elapsed.
2. The sensor performs a measurement.
3. The obtained data is normalized.
4. The normalized data is sent to a processing node.
5. The processing node combines the data from its connected sensors.
6. The processing node sends the data to the central server.
7. The central server stores the data for later processing and to make it accessible through the API.

Preconditions

- The sensor is operational.

Postconditions

- The internal state of the system is updated with the latest environmental information.

Related requirements: HL-01, HL-06, FR-01, FR-02, FR-03, FR-08, FR-10

3.5.2 UC-2: Detecting irregularities in dykes

Once every so often, a UAV will fly over a dyke looking for any irregularities, by taking photographs of the different parts of the dyke. The photographs are sent to the central server where they are being analysed. The results of the analysis are stored and in the case that any irregularities are found, the system will notify the water management officials.

Primary actor: The system

Basic flow

1. A specified interval has elapsed.
2. The system lets a UAV take off and fly over a dyke.
3. The UAV takes photographs of the dyke.
4. The UAV lands.
5. The UAV sends the photographs that were taken to the central server.
6. The photographs are received by the central server.
7. The central server analysis the photographs.
8. The central server detects an irregularity in the dyke.
9. The central server sends a notification to the water management officials through the dashboard.

Alternative flow

(8a) The system does not detect any irregularities.

  (a) The system does not send a notification.
  (b) The use case ends.

Preconditions

- A UAV is available to perform the inspection.

Postconditions

- When an irregularity has been detected, the water management officials are notified.

Related requirements: HL-02, FR-07

3.5.3 UC-3: Manually issue a flood warning

When the system has predicted some possibility for flooding, but not high enough to issue a warning automatically (*possible-flood*), a Water Management Official may want to issue a warning manually.

Primary actor: Water Management Official

Basic flow

1. The Water Management Official opens the administration dashboard.
2. The Water Management Official selects *Issue warning*.
3. The system confirms the issue-warning request.
4. The system issues a flood warning.

Alternative flow

(3a) The Water Management Official cancels the issue-warning request.

   (a) The system does not issue a warning.
   (b) The use case ends.

Preconditions

1. The system has predicted a possibility of flooding below the threshold for an automatic warning: *possible-flood*.
2. The Water Management Official has authorized access to the administration dashboard.

Postconditions

1. The system has sent a warning.

Related requirements: HL-04, FR-16, FR-19

3.5.4 UC-4: Determining probability of a flood

Based on all of the obtained data from the various sensors and third parties, the system determines what the probability is of a flood. The probability is tested against two predefined thresholds:

> *definite-flood*: The probability of a flood is high enough to warrant immediate evacuation. Warnings will be sent to local residents and the emergency services.
>
> *possible-flood*: There is a possibility of flooding, but the probability is not high enough to warrant immediate evacuation. The system sends a warning to Water Management Officials. They can choose to issue a warning manually through the web interface.

Primary actor: The system

Basic flow

1. A specified interval has elapsed.
2. The system retrieves the latest sensor data that is relevant for determing the probability of a flood from the central database.
3. The system retrieves the precipitation forecast from a third party.
4. The system retrieves the wind speed forecast from a third party.
5. The system determines the probability that a flood will occur.
6. The system stores the calculated probability.
7. The system compares the determined probability with the *definite-flood* threshold.
8. The system detects that the determined probability exceeds the *definite-flood* threshold.
9. The system sends a warning to the emergency services via a warning system.
10. The system sends a warning to the local residents via SMS.

Alternative flow

(7a) The system detects that the probability does not exceed the threshold.

1. No warnings are sent.
2. The Use Case ends.

(7b) The flood probability is below the *definite-flood* threshold, but above the *possible-flood* threshold.

1. A warning is sent only to Water Management Officials.
2. The Use Case ends.

Preconditions

- Sensor data has been gathered and stored.

Postconditions

- The calculated probability is stored.
- If the probability exceeds the *definite-flood* theshold, the system has sent a warning to the emergency services and the local residents.

Related Requirements HL-03, FR-04, FR-05, FR-09, FR-17, FR-18

3.5.5 UC-5: Detecting Flooded Areas

Part of the system consists of a number of sensors that monitor an area and have to detect the flooded parts of that area in case of a flood. The Use Case *Gathering Sensor Data* describes how the data from these sensors is obtained. Once every so often, the system anaylises the obtained data and looks for flooded areas. The results of the analysis are stored at the server and are made available trough the API.

Primary actor: The system

Basic flow

1. A specified interval has elapsed.
2. The system retrieves the sensor data relevant for detecting flooded areas from the central database.
3. The system starts the analysis on the data.
4. The system detects areas that have been flooded.
5. The system records the areas that have been flooded in the database and makes the information accessible through the API.

Alternative flow

(4a) The system detects that no areas have been flooded.

   1. The Use Case ends.

Preconditions

- Relevant sensor data is available.

Postconditions

- The system has detected the areas that have been flooded, if there are any.

Related Requirements HL-05, HL-06, FR-06, FR-10

3.5.6 UC-6: View administration dashboard

The Water Management Official can view the state of the system on the administration dashboard. The dashboard shows a history of all the warnings that have been issued, the latest and previous flood predictions and the current status of the sensors.

Primary actor: Water Management Official

Basic flow

1. The Water Management Official opens the administration dashboard web interface.
2. The administration dashboard presents a login screen.
3. The Water Management Official enters their authentication credentials.
4. The Water Management Official gets full access to the system administration dashboard.

Preconditions

1. None

Postconditions

1. The Water Management Official can monitor the system from the administration dashboard.

Related requirements: HL-07, FR-13, FR-14, FR-15, FR-16, FR-11, FR-19

3.6 Functional Requirements

The requirements of the system are presented below. This describes the main functionality that is required from the system.

FR-01	Must	The system shall monitor the water level.
FR-02	Must	The system shall monitor the water pressure.
FR-03	Must	The system shall monitor the water flow.
FR-04	Must	The system shall gather the precipitation forecast.
FR-05	Must	The system shall gather the wind speed forecast.
FR-06	Must	In case of a flood the system shall identify the areas that are affected by the flood.
FR-07	Must	The system shall detect irregularities in dykes.
FR-08	Must	The system shall store the gathered sensor and weather data.
FR-09	Must	The system shall determine the probability of a flood, based on all of the gathered sensor data and the obtained weather data.
FR-10	Must	The system shall expose the sensor and weather data to third parties.
FR-11	Must	The web interface shall display the current sensor values.
FR-12	Must	The web interface shall display past sensor values.
FR-13	Must	The web interface shall display a history of previous warnings that have been issued.
FR-14	Must	The web interface shall display the current predictions.
FR-15	Must	The web interface shall display previous predictions.
FR-16	Must	The web interface shall allow the water management officials to manually issue a flood warning.
FR-17	Must	The system shall issue a warning to local residents when a flood is imminent.
FR-18	Must	The system shall issue a warning to the emergency services when a flood is imminent.
FR-19	Must	The administration interface shall only be accessible by authenticated users.

3.7 Commercial non-functional Requirements

Below are the non-functional requirements that are important to the business side of system.

CNF-01	Must	The system should be profitable. Although the initial investment of creating the system isn't completely covered by the costprice. The system will generate profit through the maintainance contract after 5 years.
CNF-02	Must	The sensors lifetime of the sensors is important to the system. The sensors should be able to power itself by the use of solar power. The lifetime of a sensor should not be less then 5 years.

3.8 Technical non-functional Requirements

This section defines the technical non functional requirements of the system. The requirements are defined by the ISO/IEC 25010 standard [2].

3.8.1 TNF-01 Functional suitability

Functional completeness	Must	The system should be functionally complete, it should be able to complete all the tasks defined by the use cases.
Functional correctness	Must	The system has to be precise in both the measurements of the sensors and the flood probability calculations.
Functional appropriateness	Must	The system should be functional appropriate. It should only do what is required of the system (as described in this document) and nothing more in order to improve the performance and minimize overload of the system.

3.8.2 TNF-02 Performance efficiency

Time behaviour	Must	The system should have good performance and almost real-time data of the sensors. Under normal circumstances the delay should be under five seconds.

3.8.3 TNF-03 Compatibility

Interoperability	Must	It is required of the system to be able to communicate with third parties. The system communicates to third parties by the use of an API, this enables the parties to access the gathered data and the flood predictions of the system.

3.8.4 TNF-04 Usability

Learnability	Must	The system user interface is only used by a select amound of proffesionals. These should be able to learn the systems with training in about two days.
Operability	Low	An end user who had training on the system should be able to make use of all the functionalities of the user interface (dashboard).
User error protection	Low	The system does not provide much protection for user error. Because the system should only be used by authorized personnel that is familiar with the system.

3.8.5 TNF-05 Reliability

Availability	Must	The system is critical in the process of flood detection and therefore should have a high uptime. It should have an availability of at least 99.99%
Fault tolerance	Must	The system should be able to continue functioning and detect when sensors or other parts of the hardware experience failures. Although the system keeps functioning by defects of sensors the flood prediction model will become less accurate.

3.8.6 TNF-06 Security

Integrity	Must	The system contains functionality (manually issue flood warning) wich should never be allowed to be accesssed by unauthorized personnel. The system requires strong authentication and the communications will be encrypted.
Accountability	Low	The system should log all actions taken by the end users so these can be traced back to the users.

3.8.7 TNF-07 Maintainability

Modularity	Must	The system should be modulair in order to be fault tolerant, modifiable and testable.
Modifiability	Must	The flood prediction component (actual prediction with the data) should be modifiable. When there are new insights or requirements for the prediction the system should be able to be modified to meet these new requirements.
Testability	Must	The system should aim for a high test coverage as errors of this system could have a huge impact. The system should have at least a test coverage of 99.99%.

3.9 Evolution Requirements

In this section requirements for the future are described.

ER-01	Must	When new sensor technology becomes available, or when the prediction of additional types of floods or other environmental hazards is required, additional sensors have to be added to the system. The integration of these new sensors should be possible without disrupting the existing system.
ER-02	Must	The system should be able to monitor additional envorinmental hazards, which requires different models from the model used for the flood prediction. It should be possible to add new models to the system to allow monitoring of these additional types of hazards.
ER-03	Must	The existing mathematical models that are being used for the prediction of an environmental hazard should be replaceble. When a better alternative becomes available, it should be possible to replace an existing model with the alternative model without disrupting the system in the mean time. While the model is being replaced, the system should still be able to calculate predictions with the use of the old model.
ER-04	Must	The flood prediction component should be modifiable. While the system is operational, the gathered data and the calculated probabilities from previous predictions can be used to change the prediction model, making it more accurate.
ER-05	Must	Important third parties should be able to integrate more with the system. Therefore changes in the system should be made in order to promote interopablity and have a better intergration with other systems.

3.10 Risk Assessment

This section gives an overview of the major risks associated with the system. The risks are divided into three different categories. The business risks are risks that are related to the business model and the development of the system, technical risks are related to the system itself, and the operational risks are related to the system during the time that the system is operational. Each risk is assigned a probability, severity and priority respectively. The priority is derived from the product of the probability of the risk and the severity of the risk. Table 3 shows the possible priorities that can be assigned to a risk. Furthermore, the overview shows who is responsible for taking action to prevent the risk from impacting the system or to react upon a certain problem, the threshold for the occurrence of the risk, the preventive action and the reaction that a risk requires upon occurrence.

Severity	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Very Low	Low	Medium
		Low	Medium	High
	Probability			

Table 3: Risk Priorities

Risk	Priority	Responsible	Threshold	Prevention	Reaction
Business					
Scope creep	Low, Medium, Low	Architect	Milestones not met or too late.	Determine the requirements of the system and do not deviate from those requirements.	Review all the requirements and drop some of the requirements.
Too much competition	Medium, High, High	Marketing	Not enough sales to reach the return on investment goal in time.	Analyse the products offered by the competition and include functionality in the system that is not offered by them.	Come up with new functionality that will improve the position in the market or become more competing with regard to the price of the product.
Final product is too expensive	Medium, Medium, Medium	Architect	Final price of the product exceeds the intended sales price.	Carefully choose the hardware that will be used for the system.	Look for cheaper alternatives for the chosen hardware.
Regulations restricting the use of UAVs	Medium, Medium, Medium	Architect	N/A	Research potential regulations before hand and choose an alternative if necessary. Design the system in such a way that this can be easily replaced.	Switch to an alternative that is not restricted by regulations.

Technical					
No suitable model for calculating the probability of a flood	Low, High, Medium	Architect	N/A	Find a suitable model early on in the project.	Find an alternative way to calculate the probability of a flood.
Operational					
System requires too much maintainance	Low, High, Medium	Architect	Maintainance costs exceed set budget.	Choose hardware components that requires little maintaince.	Replace the existing hardware with alternative hardware that requires less maintainance.
Large number of hardware failures	Medium, High, High	Architect, Maintainers	Multiple hardware components fail before they exceed the mean time to failure.	Choose hardware components that have a large mean time to failure.	Replace the existing hardware components with an alternative that has a higher mean time to failure.
Warnings issued when no flood imminent	Medium, High, High	Architect	N/A	Setting the threshold, which needs to be exceeded before a warning is issued, to a high enough value. Using accurate prediction models.	Increase the chosen threshold. Replace the current flood prediction mode with a more accurate model.
Sensors cannot withstand weather conditions.	Medium, High, High	Architect, Maintainer, Developers	N/A	Choose hardware components that can operate in severe weather conditions and build the hardware in such a way that critical components are protected from the impact of the weather.	Improve the hardware in such a way that it is better capable of withstanding severe weather conditions.

4 Analysis

This section presents the analysis of the system. The analysis gives an overview of the assumptions that have been made about the flood monitoring tool and also about the environment in which the system will be used. Furtermore, it presents the technology roadmaps that indicate how the system will be improved over time. Finally, the most important design decisions are presented here. The rationale behind the choice is given, as well as at least two alternative options.

4.1 Assumptions

The following provides an overview of all of the assumptions that have been made about the system and the environment in which the system will be used.

- A wired internet connection is available in the area in which the cluser of sensors is placed. This wired connection is only assumed for the processing nodes.
- If the above assumption is not possible due to, for instance, environmental or cost related factors the assumption is made that a GPRS connection can be made for the processing nodes.
- While there are models that can be used for the prediction of floods, none of the models have been tested or emperically proven. It is assumed that there is a suitable prediction model that can be used for the system.
- A location is available where the Uninhabited Aerial Vehicle (UAV), that is intended for the inspection of the dykes, can take off and land again.
- The emergency services have a system where the flood monitoring tool can send warnings to, in order to inform the emergency services about an imminent flood.
- The security of the comminucation between the nodes is provided by Zigbee.

We make the assumption that an authority in the region will be part of the product placement. For the UK region a relevant party that could be involved in our system would be the *Environmental Agency*. Which is the principal flood risk management authority in England and Wales. Responsible for forecasting, mapping flood risk, providing warnings, advising on development in the floodplain, building and keeping defences in good order and taking part in emergency planning and response. Also manages central government grants for capital projects.

4.2 Technology Roadmaps

The first version of the flood monitoring tool will include all of the functionalitiy described in the 3 section. The next generation (Beta) of the tool shall include additional flood monitoring capabilities such as the monitoring of coastal areas and the detection of flash floods, which occur due to heavy rainfall in an urban area. Also important is that third party integration is offered from this point. A good working API is up and running. A third generation (Charlie) of the tool shall be able to detect other types of natural disasters as well, such as earthquakes. Thanks to the world wide market expansion big data algorithms will ehnance the prediciton rate and mass production will enable that our hardware software costs can be spreak over more products. This will enable more investment and thanks to the next generation track in the previous genertion we can develop some of the beta projects which started from there. In the last phase our system is indeed constantly being updated by the hardware and software components we have developed year on end. This will ensure our vision goal.

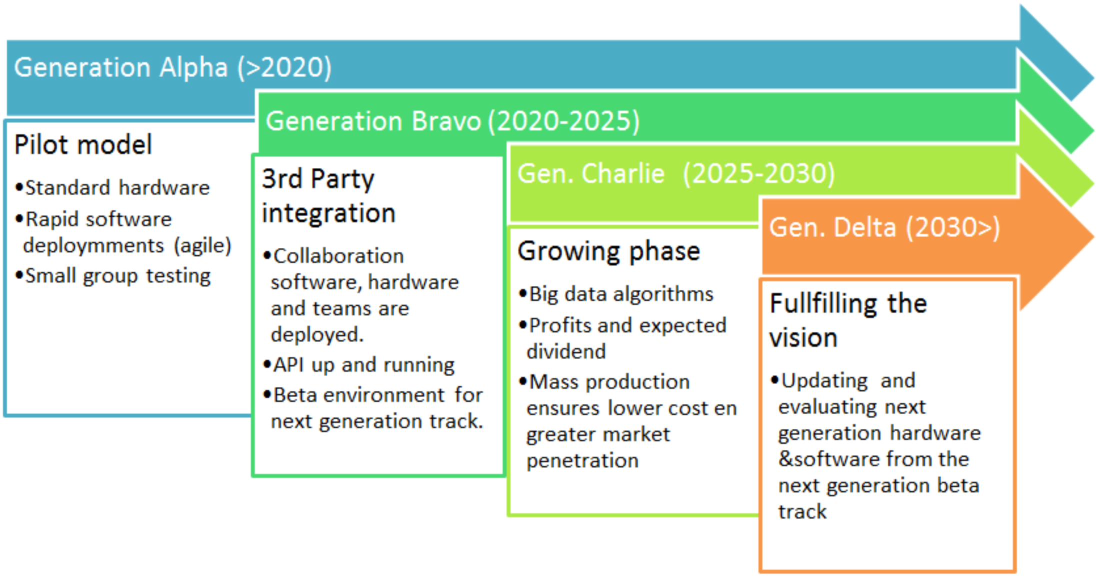

Figure 5: Technological Roadmap for the Environmental System

4.3 Design Alternatives

Name	Sensor Network Topology
Status	***Approved***
Issue	The system should be able to obtain the data from the different sensors and use all the different measurements for the flood monitoring. This requires that the sensors can communicate with a server where the data can be sent to.
Decision	A grid based approach will be used. The different sensors are nodes in the grid and can communicate with each other. Furthermore, the grid contains a number of processing nodes that collect the data and send the data to a central server.
Alternatives	**Local Processing** A local network is set up that include all the sensors and a server that processes all the data. **Central Server** Every sensor is capable of communicating with a central server, which can be located anywhere, and send the obtained data directly to this server.
Arguments	With the grid based approach not every sensor has to be equipped with a long distance communication module. The nodes in the grid can communicate with one another via a short distance communication technique, which is cheaper. Only a few processing nodes need to be able to communicate with the central server. Furtermore, the nodes can communicate with each other, which makes it easier to detect broken nodes. The nodes can regularly send a heartbeat and when this is no longer received by the other nodes, it is known that one of the nodes is no longer operational. This can then be reported to the maintainers of the system.

Name	Communication Between Grid Nodes
Status	***Approved***
Issue	The nodes that are part of the grid (sensor and processing nodes) should be able to communciate with each other. The nodes have to exchange data and have to be able to send a heartbeat to the other nodes, which allows the detection of nodes that are no longer operational.
Decision	Zigbee will be used for the communication between the nodes within the grid.
Alternatives	**Z-wave** Z-wave is a proprietary wireless standard that allows devices to communicate with each other. **Cellular** The use of GSM/GPRS for the communication between the devices.
Arguments	Zigbee has a range of about 100 meters, which is succificent since the sensors don't have to be that far apart. Furthermore, it does not consume too much power, which is very benificial for the power consumption and thus the battery lifetime. Cellular communication does offer a greater range, but the modules are far more expensive. Z-wave on the other hand is a proprietary wireless technique and adding or removing devices from the network requires a special process.

Name	Issuing Warnings to residents
Status	***Approved***
Issue	When the system detects that a flood is imminent a warning should be sent to the local residents. It should also be possible to issue a warning manually. The warning that is being issued by the system should reach as many people as possible.
Decision	The system will issue the warning via SMS.
Alternatives	**Mobile Application** The warning can be sent to a smartphone application, which can in turn display a notification on the phone. **Web Interface** A web interface could show the warnings to the users if applicable.
Arguments	A web interface requires people to navigate to the web interface, before they are informed of an imminent flood. This requires too much effort from the residents and would not reach that many people. A mobile application on the other hand, does allow the system to send a warning to the user and the phone can in turn automatically display a notification. While this succificient, it does require that the local residents install the application and have a device that supports the application. Issuing the warning automatically via SMS does not require any effort from the residents, since it can be pushed to their phones automatically.

Name	Storage of Sensor Data
Status	***Approved***
Issue	When the sensors perform measurements, the data that is obtained has to be stored somewhere.
Decision	The data will be sent to the central server, where it will be stored in a database.
Alternatives	**Local Storage** The data is stored at the processing nodes that aggregate all the data from the sensor nodes. **Mixed Storage** The data is stored both at the central server and at the processing nodes.
Arguments	When the system performs the analysis on the data to calculate the probability of a flood, the data from all of the sensor nodes is required. Since the analysis is computational expensive it is performed at the central server, which means that all of the data needs to be available at the central server. Furthermore, local storage requires that the processing nodes have the capacity to store a certain amount of data. Since this would increase the costs of the node and also increase the power consumption, all of the data is sent to the central server.

Name	Information Dashboard
Status	***Approved***
Issue	The water management officials should be given access to the data and analysis results of the system. They should also be able to view any sensor failures that have occured.
Decision	The water management officials are given a web interface (dashboard) where they are provided with all of this information.
Alternatives	**Mobile Application** The water management officials can access the information from an application. **Native Application** The water management officials can access the information from a native application running on a PC.
Arguments	A web interface can be accessed relatively easy from all devices and platforms. Furtermore, a web application can be made in such a way that it gives the same experience as a native application. To make a mobile or native applications run on all devices and platforms requires a great deal of effort. Another disadvantage of a native application is the lack of portability. The water management officials will not be able to access the dashboard when they are not near a PC that has the application installed on it.

5 System architecture

5.1 Initial model

5.2 Elaborated model

5.3 Verification

6 Hardware architecture

7

8 Software architecture

8.1 Architectural design

8.2 Architectural views

8.3 Components

8.4 Design views

9 Architecture evaluation

9.1 Requirement evaluation

9.2 Architecture evaluation

10 System evolution

References

[1] Environmental Agency. Flooding in england: A national assessment of flood risk. page 33, 2007.

[2] International Organization for Standardization (ISO). Iso/iec 25010.

[3] Danny Hughes, Jo Ueyama, Eduardo Mendiondo, Nelson Matthys, Wouter Horré, Sam Michiels, Christophe Huygens, Wouter Joosen, Ka Lok Man, and Sheng-Uei Guan. A middleware platform to support river monitoring using wireless sensor networks. *Journal of the Brazilian Computer Society*, 17(2):85–102, 2011.

[4] PHAJM Van Gelder. Risks and safety of flood protection structures in the netherlands. In *Proceedings of the Participation of Young Scientists in the Forum Engelberg*, pages 55–60, 1999.

Appendices

A Time Tracking

Table 4: Week 1 (31-08-2015 - 06-09-2015)

Name	Task	Hours
Michel Medema	Meeting (03-09-2015)	1 Hour
	Literature	1 Hour
	Setting up the document structure	1 Hour
	Section 1: Introduction	0.5 Hour
	Section 2: Context	1.5 Hours
	Section 4.2: Stakeholder	0.5 Hour
		5.5 Hours
Marco Gunnink	Meeting (03-09-2015)	1 Hour
	Reading & minor reviewing	1 Hour
		2 Hours
Daan Mauritsz	Reading Project Description	1 Hour
	Section 2: Context	1 Hour
		2 Hours
Koen Roos	Meeting (03-09-2015)	1 Hour
	Reading Project Description	1 Hour
	Section 2: Context	1 Hour
		3 Hours
Sweta Singh	Meeting (03-09-2015)	1 Hour
	Reading Project Description	0.5 Hour
	Section 2: Context	2.5 Hour
		4 Hours
Patrick van Vreeswijk	Meeting (03-09-2015)	1 Hour
	Literature	1 Hour
		2 Hours

Group Total: 18.5 Hours

Table 5: Week 2 (07-09-2015 - 13-09-2015)

Name	**Task**	**Hours**
Michel Medema	Literature	1 Hour
	Setting up layout for Requirements section	0.5 Hours
	Coaching session (07-09-2015)	1 Hour
	Establishing system scope + Literature	3 Hour
	Working Session (requirements)	2.5 Hours
	Working Session (requirements)	2 Hours
	Fixing glossary and references	0.5 Hours
	Processing feedback	0.5 Hours
	Use Cases	1.5 Hours
	Risk Assessment	1 Hour
	Reviewing	1 Hour
		14.5 Hours
Marco Gunnink	Coaching session (07-09-2015)	1 Hour
	Work session (requirements)	2 Hours
	Work session (requirements)	2 Hours
	Requirements & use cases	2 Hours
	Reviewing	1 Hour
		8 Hours
Daan Mauritsz	Coaching session (07-09-2015)	1 Hour
	Working Session (requirements)	2,5 Hours
	Working Session (requirements)	2 Hours
	Non functional requirements	1,5 Hours
		7 Hours
Koen Roos	Coaching session (07-09-2015)	1 Hour
	Working Session (requirements)	2 Hours
	Working Session (requirements)	1 Hour
		4 Hours
Sweta Singh	Coaching session (07-09-2015)	1 Hour
	Processing feedback	0.5 Hour
	Working Session (requirements)	2 Hours
	Working Session (requirements)	2 Hours
	Architecture Vision	2 Hours
	Reading Literature	2 Hour
	Use Cases	0.5 Hours
	Reviewing	0.5 Hours
		10.5 Hours
Patrick van Vreeswijk	Coaching session (07-09-2015)	1 Hour
	Literature & Business Information (09-09-2015)	4 Hour
	business Information & Working Session (11-09-2015)	3 Hour
		8 Hours

Group Total: 52 Hours

Table 6: Week 3 (14-09-2015 - 20-09-2015)

Name	**Task**	**Hours**
Michel Medema	Analysis	2 Hour
	Coaching session (14-09-2015)	1 Hour
	Discussing system requirements	1 Hour
	Finding flood prediction models	3 Hour
	Working Session	3 Hour
	High-level Requirements	1 Hour
	Requirements	0.5 Hours
	Stakeholders + Key-drivers	1 Hour
	Risk Analysis	1 Hour
	Evolution Requirements	0.5 Hours
	Stakeholders + Key-drivers + High-level Requirements	2 Hours
	High-level Requirements + Use Cases + Requirements	3 Hours
	Evolution Requirements + Risks	1 Hour
	Architectural Vision, Analysis, Use Cases, Review	4 Hour
		24 Hours
Marco Gunnink	Coaching session (14-09-2015)	1 Hour
	Requirements	2 Hours
	Working session (17-09-2015)	2 Hours
	Use cases	1 Hour
	Reviewing	2 Hours
	Use cases	1 Hour
		9 Hours
Daan Mauritsz	Coaching session (14-09-2015)	1 Hour
	Discussing system requirements	1 Hour
	Working Session (17-09-2015)	2,5 Hours
	Technical Non-Functional Requirements	3,5 Hours
	Commercial and Evolution Requirements	1 Hour
		9 Hours
Koen Roos	Coaching session (14-09-2015)	1 Hour
	Work and Discussion (14-09-2015)	1 Hour

	Working Session (17-09-2015)	2 Hours
	Business Information	4 Hours
		8 Hours
Sweta Singh	Coaching session (14-09-2015)	1 Hour
	Work and Discussion (14-09-2015)	1 Hour
	Group Session and use cases, review doc (17-09-2015)	4 Hour
		6 Hours
Patrick van Vreeswijk	Coaching session (14-09-2015)	1 Hour
	Work and Discussion (14-09-2015)	1 Hour
	Work, finishing business information (15-09-2015)	3 Hour
	Business Information, review document (16-09-2015)	2 Hour
	Group Session and system sketch, review doc (17-09-2015)	4 Hour
	Technology Roadmap, Roadmap, review and updating document (20-09-2015)	6 Hour
		17 Hours

Group Total: 73 Hours