

An architectural framework for flow-based IoT solutions

A study into the application of the Internet of Things paradigm and its enabling technologies
in combination with the use of flow-based programming.

K. Veldhuizen

January 2021

Master of Information Management (MSc IM)
Faculty of Economics and Business Administration
Tilburg University

Thesis author:

Karsten Veldhuizen
u544436, 2042344
k.c.veldhuizen@tilburguniversity.edu

Master's Thesis supervisor:
P.K. Medappa



Management Summary

This research was aimed to create an artefact in the form of an architectural framework for flow-based IoT solutions. Currently, research on IoT in combination with flow-based programming is scattered and scarce, while the landscape of IoT is diverse and complicated with many underlying dependencies in their enabling technologies. To help firms who are adopting flow-based IoT solutions and academics who are active within the field of IoT and flow-based programming the artefact was created. The artefact guides the user in the different visions that the flow-based IoT landscape is composed of and elaborates on the visions with different building blocks, principles, their underlying relations, and dependencies. Furthermore, the role of flow-based programming within IoT has been determined and depicted within the framework. The framework was created with a Design Science approach in which knowledge has been gained on multiple levels and iteratively implemented within the artefact. During its development literature was studied, interviews with the business and field-level experts conducted, a domain-specific proof of concept built, and an experiment held. The artefact was developed with a total of 5 iterations in which every iteration was an improvement on the previous with new lessons learned and documented changes that were implemented. The artefact was validated with the help of two different approaches which focused on multiple variables that gained insight into certain properties of the framework. The first approach was based on an experiment which validated the framework on performance & effectiveness. The experiment was deemed successful if the domain-specific goal of a higher production was achieved. The second approach focused on external knowledge coming from experts for which multiple interviews have been used. In this approach, the framework was validated on its functionalities, completeness, consistency, usability, and architectural structure. These variables were defined with the help of the literature.

The results of the experiment for the first approach where an overall increase in production by 5,67% with the treatment group, that has been aided with the flow-based IoT solution. This was possible with the competitive advantage the treatment group had in its data gathering with sensors and analysis by the flow-based solution. This was tested with three different hypotheses that all showed a significant increase in production supporting each other's results. Based on the interpretation of these results the performance & effectiveness of the framework has been proven, as the framework can be used to facilitate the development of effective IoT-solutions which can achieve their domain-specific goals (e.g., higher production). These results contribute to the statement that the framework is deemed usable in practice. In the second validation approach, the framework was presented to multiple external experts and they were asked to give their expertise on the chosen variables. The results of the interviews were then analysed and split between both lessons learned and needed changes. To strengthen the validation of the framework different experts have been consulted and the total amount of interviews were split into two rounds. In which the first round of interviews allowed changes to the framework to be made while the second round was purely to validate the final result. The experts consulted ranged from data scientist, (IoT) product architects, (IoT) hardware-engineers and automation experts working with flow-based solutions. The experts commonly agreed that the framework is a sufficient starting point when creating an IoT solution in which a flow-based approach is used. Furthermore, the experts state that the framework incorporates all the needed aspects of an IoT solution and is deemed both useable, consistent, and offering sufficient structure.

Combining both methods of validation, it can be stated that the framework has been sufficiently validated. It has proven itself as a worthy framework in creating a first architectural diagram for flow-based IoT solutions and depicting the flow-based approach within the framework. The framework has proven itself in practice with the proof of concept and held up to the critics and changes of the consulted experts. However, future research can be performed to create a more extensive framework.

Preface

Before you lies the information management thesis “An architectural framework for IoT flow-based solutions”. It was been created to finish my master graduation requirements of the study Information Management at Tilburg University. I was engaged in researching, writing, and creating the working concept from September 2020 to January 2021.

This project was initiated at the request of SMRTR, where I undertook an internship during the period. The main research question was created together with my thesis supervisor, P.K. Medappa. Performing research while at the same time creating a workable proof of concept was difficult but by putting in extensive effort has allowed me to answer the main research question.

I would like to thank SMRTR and especially Steff, Arthim and Ewout for helping me during the project also I want to thank my supervisor for their excellent guidance.

Karsten Veldhuizen,

Rotterdam, January 13

Table of contents

1.	Introduction	6
1.1.	Research position.....	7
1.2.	Problem statement and research question	7
2.	Literature review.....	9
2.1.	Internet of Things.....	9
2.2.	IoT Frameworks and the importance of architecture	9
2.1.1.	Role of Architecture frameworks for IoT.....	9
2.2.1.	Internet of Things paradigm.....	10
2.1.1.	Role of flow-based programming.....	11
2.2.2.	Main application domain for IoT	12
2.1.2.	Validation of IoT Framework within Agriculture	13
2.1.1.	Requirements & Design principles for IoT solutions within Agriculture	13
2.3.	Enabling technologies for IoT.....	15
2.4.	Findings current literature	20
3.	Research method	21
3.1.	Design science research flow	23
3.2.	Design of the artefact.....	24
3.2.1.	Role of proof of concept for creating the framework	25
3.2.2.	Collecting the functional requirements.....	26
3.2.3.	Evaluation of design.....	27
3.2.3.1.	Controlled Experiment.....	27
3.2.3.2.	Analysis of external views on the artefact	31
3.2.3.3.	Validity	32
4.	Results.....	33
4.1.	System structure of Proof of Concept.....	33
4.2.	Documented changes within iterations of the framework	35
4.3.	Proposed Framework	39
4.4.	Building blocks of IoT framework	39
4.5.	IoT framework for flow-based solutions.....	41
4.6.	Validation	44
4.6.1.	Analytical	44
4.6.2.	Experiment	52
5.	Discussion & Limitations	54
5.1.	Main new findings	54
5.2.	Findings and interpretation of results from interviews.....	54
5.3.	Interpretation of results from the experiment	55
5.4.	Theoretical and practical implications	56

5.5.	Reflection on validity.....	56
5.6.	Limitations.....	57
6.	Conclusion.....	58
6.1.1.	Future research	59
	Bibliografie.....	60
	Attachments	63
I.	Hardware Components PoC (selection & design)	63
II.	Data-Logging Equipment for PoC	65
III.	Software & design for PoC	66
IV.	Data from growth schedule	67
V.	Proposed framework to business (iteration 1, literature).....	72
VI.	Proposed framework to business (iteration 2, business requirements).....	74
VII.	Proposed framework to business (iteration 3, after Simeon).....	76
VIII.	Proposed framework to business (iteration 3, after Marijn)	77
IX.	Proposed framework to business (iteration 4)	78
X.	Requirements for PoC	79
XI.	Criteria for validating the framework	80
XII.	Requirements for developing the artifact (framework).....	83
XIII.	Interview I, questionnaire for gathering functional requirements for PoC	86
XII.	Interview I, retrieving framework validation requirements.....	87
XIII.	Interview III, validating the Framework and final changes	88
XII.	Interview III, validating the final iteration of the Framework	90
XV.	Interview functional requirements respondent B.....	91
XV.	Interview functional requirements Interview D.....	97
XVI.	Interview framework requirements respondent A	100
XVII.	Interview framework requirements respondent C	105
XVIII.	Interview framework validation respondent J.....	107
XIX.	Interview framework validation respondent E	108
XX.	Interview framework validation respondent F	112
XXI.	Interview framework validation respondent G	114
XVIII.	Interview framework validation respondent H	118
XVIII.	Interview framework validation final iteration respondent F	121
XVIII.	Interview framework validation final iteration respondent K	122
XVIII.	Interview framework validation final iteration respondent L.....	126
XVIII.	Interview framework validation final iteration respondent E	127

1. Introduction

Interest in more effective and sustainable ways of practising agriculture to feed the world is evermore growing. According to the United Nations, the global population will reach 9.7 billion people in 30 years (UN, 2020). This increase in population also means that there will be a need to provide 50% more food in 2050 (FAO, 2009). Just like many other sectors, the agrarian sector will face many changes in the coming decades. With these changes in mind, governments and the private sectors are searching for ways to improve farming methods using modern technologies. One of these technologies is the Internet of Things (Olukunle Elijah, 2018). Which is a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols consisting of many things. The basic concept is of a pervasive presence around the humanity of a great variety of things also called objects. These things can be anything: sensors, tags, phones, and these things can interact with each other with supporting communication protocols (Luigi Atzori, 2010). The industry is keeping pace to deliver more IoT solutions to the agricultural sector (Mahammad Shareef Mekala, 2017). IoT in combination with Cloud computing can provide enough resources to gather and process large amounts of sensor data which in turn can be analyzed and used to automate and guide (agricultural) processes or achieve predictive analytics on crops or livestock to achieve better efficiencies and effectiveness (Antonis Tzounis, 2017). Another major IoT enabling technology, one in which SMRTR (the company where the research is performed) wants to invest in and a technology in which global usage rose dramatically is flow-based programming (Dave Mason, 2017). This is a programming paradigm in which new solutions can be created quickly with the help of “black box” processes (a black box process is a process that transforms its input into a new output). These processes exchange data/messages across predefined connections and can be reused endlessly to form different solutions (Morrison, 2007). According to Dave Mason (2017), flow-based programming contributes to the idea that most people should have some programming capability and flow-based programming can offer this. This general programming knowledge can contribute to the development of wide IoT solutions.

This is where SMRTR takes a position, one of SMRTR categories/pillars is “Fun”. This is their pillar in which they want to create IoT & AI solutions. This pillar is ideal for SMRTR to achieve more leads, showcase their company profile, and way of working with “Fun” products/services. SMRTR wants to have an IoT flow-based solution in their portfolio which is designed with the goal of a more efficient way of producing food. The concept should include modern communication technologies, sensors, and smart products. SMRTR also wants to gain expertise & knowledge on IoT solutions in general and make a small contribution to the food production need.

1.1. Research position

SMRTR is a company that delivers tailor-made solutions and tackles complex problems. SMRTR works together, towards a smarter and better organization. SMRTR is divided into 4 main categories: better, quicker, smarter, and fun. While customer's interest for IoT is steadily growing each quarter, SMRTR does currently not have the needed knowledge on how to create flow-based IoT solutions, as well as the knowledge of what an IoT solution needs to conform to and how the different technologies are combined. IoT in combination with flow-based programming will have the central role in this research and is the main interest for SMRTR. Hence SMRTR created a research project to gain this needed knowledge. The main problem in achieving this goal is, what technologies are needed and especially to know how the needed technologies can be combined and which factors/elements need to be taken into consideration (Bilgeri, 2019).

This lack of knowledge means that before a solution can be created, new knowledge must first be gained and combined to solve design problems. This new knowledge has academic relevance as current literature seems to fall short in combining these technologies and depicting them together. This creates the opportunity to use the gained knowledge to create a currently nonexistent IoT framework for-flow based solutions with interconnected technologies, processes, and visions. This fills a currently existing knowledge gap because these technologies/solutions have not been combined like this before, are relatively new and adoption is still low (Lu Tan, 2010). Current research on the combination and application of these technologies is scattered and currently, no IoT framework for flow-based (low-code) solutions exists (Martin Wollschlaeger, 2017). Solving this problem has relevance for the Information Management discipline because it bridges a gap between traditional solutions and using modern IT tools and processes to guide these operations.

1.2. Problem statement and research question

However, the main problem in using flow-based programming for IoT solutions is to know how the needed technologies can be combined and what design elements (design elements can be defined as the different choices within components that together make up a total solution.) need to be taken into consideration on an architectural level (Bilgeri, 2019). Luigi Atzori (2010) confirms these issues and states that many challenging issues still need to be addressed and that synergetic activities are necessary to combine the technologies and different fields of knowledge needed when building IoT solutions. Currently, no architectural IoT framework for-flow-based solutions exist while IoT is composed of multiple underlying technologies/systems, and thus structure in the form of a framework is seemed very important when there are many dependencies. This fills a currently existing knowledge gap because the technologies/solutions have not often been combined like this before or are relatively new and adoption is still low. Current research on the combination and application of these technologies are scattered (Martin Wollschlaeger, 2017).

This research aims to create an architectural framework for flow-based solutions with the help of current literature and a proof of concept focused on an IoT agriculture solution. The outcome of this research can help individuals/organizations to create flow-based IoT solutions and lay certain design choices next to the architectural framework. The framework could essentially be used to create a first blueprint of the new IoT flow-based environment. The PoC will be fully created inhouse and will both have a hardware and a software part in which the hardware (IoT sensors) will share its data/status with the automation platform which in turn will automate certain actions and give insights to the user. For this multiple (technical) questions need to be answered. With the answers, the PoC can be built and the framework can be shaped as an iterative process. Eventually, the solution needs to be validated with the data that comes from the proof of concept and with the help of interviews.

Although the research focusses on agriculture and the generic problem of an increasing need in (global) food production, the solution to the above-mentioned problems could be generalizable to not only agriculture but also to other domains wanting to use IoT, flow-based solutions, and wondering how to combine certain design choices with the help of a framework.

Therefore, the following research question has been created and will be addressed:

How can the different technologies within the Internet of Things be used to create an IoT architectural framework for flow-based solutions implementing different IoT design elements?

To be able to answer the main research question sub-question have been created. The sub-questions are split into two sections theoretical and technical.

Theoretical:

- *Which design elements are available for flow-based IoT solutions?*
- *What is the role of frameworks within the IoT landscape?*
- *How can the gained knowledge be transformed to fill the framework?*
- *In what ways can the framework be correctly validated both internal & external?*
- *In which different domains can IoT be applied?*

Technical:

- *How are the needed IoT design elements combined in practice?*
- *What does the infrastructure of the total solution look like?*
- *How can the gained data be transformed and analyzed?*

This research question is composed of both a technical and theoretical part. In the theoretical part, knowledge is gained on IoT, enabling technologies and domain-specific uses. In which the gained knowledge is then used to fill in the framework and eventually create the hypotheses that are tested to be able to determine if using such flow-based solutions contributes to better performance. On the technical side information is needed on what technologies are best suited, how does the infrastructure and architecture of the proof of concept look like visually and how the needed data for the solution is gathered, stored, and analysed. Solving this problem has relevance for the Information Management discipline because it bridges a gap between traditional solutions and using modern IT tools and processes to guide these operations and gain advantages. Furthermore, an architectural IoT framework for flow-based solutions can also contribute to specific branch solutions such as agriculture solutions and help with increasing their efficiencies.

1.3. Research structure

In order to maintain a clear structure throughout the document, the research has been separated into multiple chapters. The first chapter includes the introduction, problem statement and research question. Chapter two includes the literature and a listing of all the different knowledge that can be used within the artefact. Chapter three includes the research method and method of (reliable) data collection. In chapter four all the results are displayed of both the experiment and interviews. Chapter five includes discussion and limitation of the study and chapter six concludes the study and possibilities of future research.

2. Literature review

2.1. Internet of Things

According to Luigi Atzori the Internet of Things can be defined as: "A worldwide network of interconnected objects uniquely addressable, based on standard communication protocols consisting of many objects". The basic concept is of a pervasive presence around us of a great variety of objects also called things. These objects can be anything: sensors, tags, phones. These objects can interact with each other with the help of communication protocols (Luigi Atzori, 2010). The very first definition of IoT derives from the viewpoint of "Things Oriented". The first objects considered were very basic items: Radio-Frequency Identification tags (Ashton, 2009). Nowadays the Internet of Things attempts to merge the real world with the virtual world through the use of the Internet as a way of sharing and exchanging data and information. IoT is also described as a system of interrelated computing devices, mechanical and digital machines, objects, or individuals that have unique identifiers and the ability to transmit data over a network without the need for human-to-human or human-to-computer communication (Stankovic, 2014).

2.2. IoT Frameworks and the importance of architecture

Frameworks for architecture have become a common way of coping with the complexities of today's IT solutions & technologies. By having a method for modelling and defining them, they facilitate the specification of architectures and bring guidance to those who work with these solutions. Daniel Ota (2011) defines such an architectural framework as a standard vocabulary, a collection of views focused on specific aspects of the architecture, a set of architectural styles with differing degrees of detail, and a technique for the construction and management of architecture and its views are usually specified by an architecture system. Furthermore, in IT, frameworks are particularly critical for helping to better handle dynamic processes and systems. Not only is the structure of frameworks valuable to IT, but it also helps enterprises to respond quickly to developments or introduce new solutions while having principles to guide them (Daniel Ota, 2011).

2.1.1. Role of Architecture frameworks for IoT

When a large number of things/objects are connected to the internet and each other, it is according to John A. Stankovic (2014) important to have an adequate architecture that facilitates easy connectivity, control, and different ways of communication. It is also important to give attention to the question of how these objects will interact with each other across the solution. Choices have to be made if objects need to be joined/disjoined. IoT is essentially based on underlying sensors and actuators together forming a network that acts as a utility similar to that of for example water and electricity. Different IoT solutions can be used on this utility while each solution has its problems. This can bring issues when many systems-of-systems have problems interfering with each other. This means that when bad architecture is used, the system has too many dependencies on each other (Stankovic, 2014).

To answer the main research question a framework has to be created. To be able to achieve this the framework is going to be built on current knowledge and frameworks available within literature. However, there are multiple general (architectural) frameworks available for IoT, these (architectural) frameworks can focus purely on architecture or system interactions. Architectural frameworks that are available for IoT are focused and validated on specific sectors such as healthcare, agriculture, or smart cities. There is one general framework also named the *IoT paradigm* in which a general approach to IoT is taken and it covers IoT completely with the help of three visions. This research on the IoT paradigm is considered one of the most beneficial research towards IoT because it brings a consensus to all aspects included in IoT, the paradigm has a large number of citations and has not been created with a specific sector or specification in technologies in mind. The IoT paradigm stands at the basis of IoT and thus is the best option as a starting point for creating an architectural framework for flow-based solutions.

2.2.1. Internet of Things paradigm

Internet of Things combines a lot of different aspects, technologies, and dependencies. Thus, implying that IoT can be viewed from multiple angles. The IoT paradigm mapped out these visions and explains that there are two major angles to approach IoT. The first angle drives from a network-oriented IoT view, while the second moves to incorporate the emphasis on generalized "objects" into a more shared context. In IoT opinions and variations emerge from the fact that stakeholders, alliances, research, and standardization bodies begin to address the topic from either an "Internet-oriented" or a "Things-oriented" viewpoint, based on their priorities, goals, and perspectives. This implies that the angle from which IoT is viewed can be the starting point for how an IoT solution is going to be build-up. An IoT solution can either start from the objects it wishes to use or the need for a network combining needed aspects. According to Luigi Atzori, it should not be ignored, though, that when placed together, the phrases "Internet" and "Things" assume a sense that brings a transformative degree of creativity into the IT world (Luigi Atzori, 2010). The paradigm consists of the following:

- *Things*: In terms of computing and energy power, "things" are characterized by low cost/resources and high scalability while holding quality. This could be for example NFC, Wireless Sensors, RFID, and ESP sensors.
- *Internet*: A more network-oriented vision of IoT and about internet and web of things while it connects & communicates with the "things".
- *Semantic*: The object-specific addressing, representation, storage of IoT, and building blocks to achieve a shared meaning. Addresses issues as choosing and understanding the right middleware, data models, and technologies.

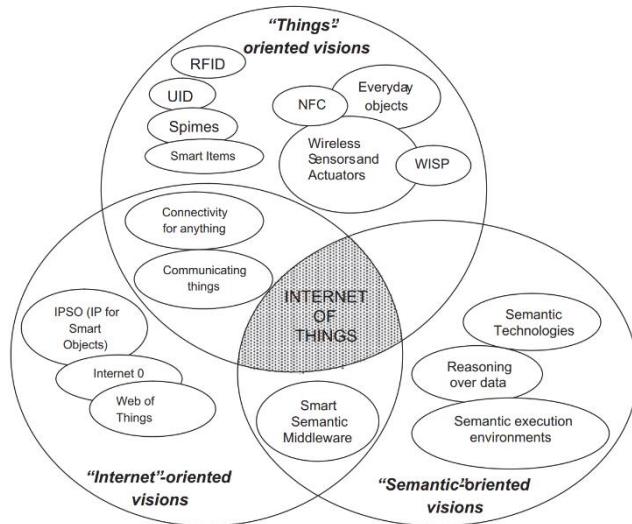


Figure 1 IoT paradigm

This paradigm clearly explains the three main visions on IoT (things, network, semantic reasoning) and how the main interest in one vision of a stakeholder influences the course of such IoT projects/solutions (Luigi Atzori, 2010). This paradigm includes all aspects of an IoT solution such as sensors/actuators, way of communicating, network, middleware solution, and semantic reasoning over data, technologies, and environment. The paradigm seems to be a necessary starting point for building an IoT framework because it starts with the way an IoT solution is approached and clearly defines the differences of these approaches and compasses all possibilities of IoT needed to create a full solution. The article also seems to be the reference article when IoT is mentioned in research. However there do appear to be limitations on the IoT paradigm, there is no architectural view on the paradigm and its vision. As well as there are no clear architectural design choices described such as rules and protocols. Thus, it cannot be directly used to create an architectural framework for flow-based solutions. But the IoT paradigm is a strong enough foundation to build the architectural framework on and is chosen to work with because of the reasons stated above.

To support the paradigm and have a fully completed reference architecture another research is used which performed a survey on domain-specific IoT architectures. The research selected certain domain-specific architectures based on an architectural structure, applicability, associativity, deployability, and incorporation measure. The outcome architecture was a Home Health Hub IoT platform (P.P., 2016). The highlighted framework is an IoT framework for healthcare and consists of all the parts from the IoT paradigm and focuses on the architecture of the IoT solution. It encompasses, objects (things), network, network types, and has semantic reasoning over the chosen technologies and network types. It encompasses the design choices needed when building such solutions in practice spread out over the different layers necessary. It makes a clear distinction between these layers such as the communication layer, sensing layer, information processing layer, internet layer, and finally user layer (Ray, 2014). The research also makes a clear distinction between the different building blocks that make up an IoT solution namely: objects, communication, services, and application.

2.1.1. Role of flow-based programming

There are multiple ways (IoT) solutions, automations, or software can be created/programmed. Flow-based programming is one of them and is increasing in popularity (Dave Mason, 2017). Flow-Based programming is being defined as a coordination language rather than a programming language. It uses both coordination and modularity. It can reuse independent components by just changing their connections. Flow-based-programming is the central node of the architectural framework and one of the main advantages of solutions with configurable modularity is that, like the chips used to build logic in hardware, you can build them from "black box" interchangeable modules. The black box principle is the ability to give the "black box" an input which it can then alter and give a new output which again can go to the next black box and perform more actions. In FBP, the fundamental building blocks that a developer uses to create an application are these black boxes also called "components" (Morrison, 2007). This gives the user the ability to create endless solutions and couplings between systems and objects which is a huge advantage in IoT solutions because of the high number of couplings, systems, and sensors/actuators. This gives the developer the ability to build solutions more quickly.

2.2.2. Main application domain for IoT

The opportunities provided by IoT allows for a large number of different solutions to be created, of which only a very limited part is currently available. There are many ways and contexts in which modern IoT applications are likely to enhance the quality of different domains. Giving different sections of these domains the ability to interact with each other and to improve the information interpreted from the environment means, providing the different domains with the possibility to deploy a very wide variety of IoT applications. These possibilities can be grouped in the following domains (Luigi Atzori, 2010):

- Transportation and logistics.
- Healthcare.
- Smart environment (agriculture, offices).
- Personal & social.

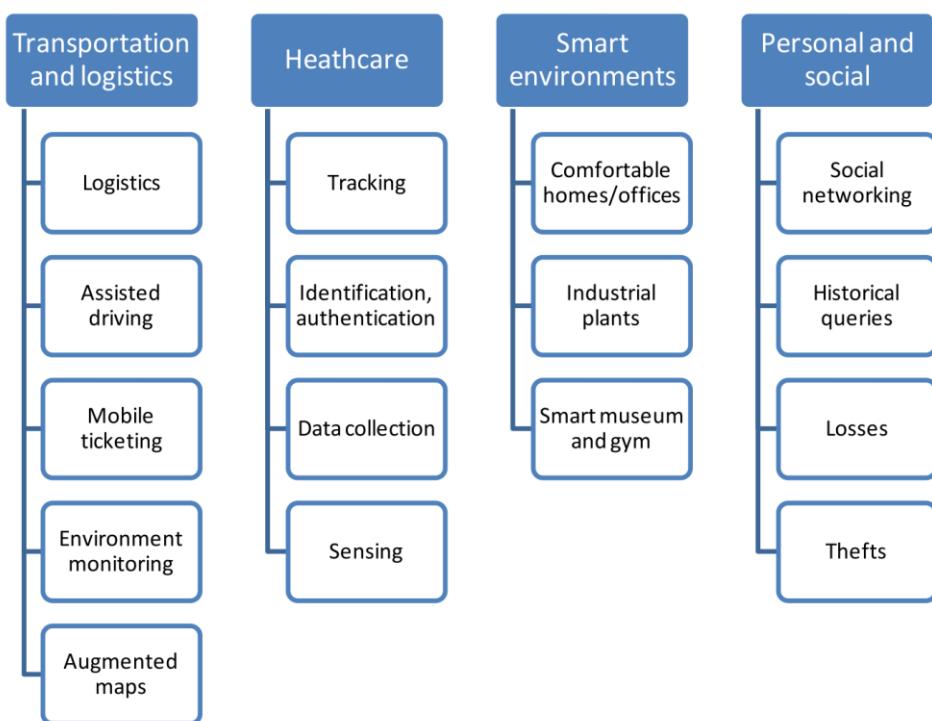


Figure 2 Application domains of IoT

2.1.2. Validation of IoT Framework within Agriculture

To be able to validate the created framework a proof of concept is going to be built according to the principles of the architectural framework. The research of P.P. Ray (2014) who surveyed multiple IoT frameworks suggests creating a branch-specific solution to be able to better validate the framework. This validation can be performed on an architectural structure, applicability, associativity, deployability, and incorporation measure (Ray, 2014). Currently, IoT is making its way to multiple different branches. One of the more common branches is Agriculture and is the subject on which the proof of concept is being built because, in the coming years, the world is turning to the use of IoT paired with data analytics to satisfy the world's food demands. In the agriculture market, IoT system installs are expected to grow from 30 million in 2015 to 75 million by 2020. The projection that the global population will be about 9.7 billion by 2050, and there will be a great increase in the need for food provided by Agriculture. This, along with declining natural resources, arable land, volatile weather patterns, makes food security a serious problem for most nations (UN, 2020). Agriculture is also an entry-level branch on which solutions can be built because setting up experiments is easier to perform than compared to healthcare because of rules, regulations, and availability of specific sensors needed. The use of the Internet of Things and data processing can therefore be used to increase organizational performance and competitiveness in the agricultural sector. There is a paradigm change from only using a wireless sensor network (WSN) to also use IoT as a driver of smart agriculture. Agricultural convergence technologies are based on IoT solutions to produce higher value by optimizing production, productivity, increasing the quality of agricultural goods in the agricultural production process, and fill a need for information to make better decisions.

2.1.1. Requirements & Design principles for IoT solutions within Agriculture

There are three different aspects of an IoT-based agricultural production system according to Meonghun Lee (2013). The Proof of Concept for Agriculture needs to conform to these requirements to be able to define the PoC an IoT agriculture solution. Literature states that an IoT agriculture system is composed out of three methods and at least one has to be incorporated into the PoC which uses a proven method for increasing efficiency:

- Statistical Prediction consists of collecting information for analysis, process information for specialization (which models to use), and information service portal which gives the user a UI to provide decision support.
- Relation Analysis consists of analysing correlations between data available such as sensors, locations to eventually achieve higher efficiency.
- IoT Service consists of a system that can collect all necessary data for analyses (data logging equipment). The IoT service needs to be reliable and within a short time and needs to be continuously operating.

The use of IoT and data analytics will allow smart agriculture with high operational efficiency and high yield expected to be delivered. As mentioned in the previous chapter IoT combines a range of existing technologies, including WSN, RF recognition, cloud storage, middleware platforms, and end-user applications (Nikesh Gondchawar, 2016). The application of IoT in agriculture is about empowering a sector with the decision tools and automation technologies that seamlessly combine goods, information & services for better productivity, efficiency, and benefits. (Meonghun Lee, 2013).

The farming IoT ecosystem is explored further in-depth based on four main components (Olukunle Elijah, 2018). These components all fall within the three IoT paradigm visions and correspond to the building blocks that have been defined by Ray P (2016). Only here they are combined and explained more in-depth to the branch solution Agriculture. The main components are:

- IoT devices: IoT systems consist of embedded devices that communicate with sensors and actuators and require wireless communication. Such IoT devices are often referred to as sensors to track and analyse various farm variables (e.g., soil nutrients, environmental data) and development factors: communication technology can be categorized based on the continuum of requirements and application scenarios. The standard of communication can be grouped into a standard of short-range communication and long-range communication.
 - Falls inside the *things* vision of the paradigm.
- Communication technology: It is possible to identify networking technologies based on standards, spectrum, and application scenarios. For example, the standard of communication can be grouped into a standard of short-range communication and long-range communication and eventually choice for a technology such as MQTT described previously.
 - Falls inside the *internet & semantic* vision of the paradigm.
- Internet: The Internet forms the central network layer, where routes are given between several subnetworks to hold and share data and network information. Connecting IoT devices to the Internet requires data to be accessible at any time and anywhere. The transmission of data through the Internet, however, requires adequate protection, real-time data support, and ease of accessibility. For cloud computing, where big data is stored for storing and retrieval, the Internet has paved the way. Cloud infrastructure encompasses user experience management, services, organization, and collaboration of network nodes, computing, and data processing. IoT middleware and networking protocols are being built to achieve the connectivity between heterogeneous systems and computers over the Internet.
 - Falls inside the *internet* vision of the paradigm.
- Data storage and processing: Data-driven farming requires the compilation of large, dynamic, complicated, and spatial data involving storage and processing. Data diversity can vary from structured to non-structured data. The data can range from historical data, sensor data, live-streamed data, industry data, and consumer data. The use of cloud IoT systems allows cloud storage of big data obtained from sensors. This entails hosting systems that are essential to the provision of resources and end-to-end IoT architecture management.
 - Falls inside the *semantic* vision of the paradigm.
- Application: The application layer is the most important in User terms as it serves as an interface that provides the necessary interfaces to manage the different facets of the IoT framework and track it. Applications allow users to visualize and evaluate the state of the device at the current level of operation, often forecasting futuristic opportunities.
 - Falls inside the *semantic* vision of the paradigm.
- Services: The IoT framework offers multiple types of functions, such as services for measurements or automation services, device monitoring data publishing, data analytics, and device discovery services.
 - Falls inside the *semantic* vision of the paradigm.

These components described for IoT within Agriculture share the same logic and components as explained in the IoT Paradigm (Luigi Atzori, 2010). This makes these two IoT types of research for Agriculture a good choice to extract certain requirements for the proof of concept to be able to call it an Agriculture IoT solution. Also, using this shared logic makes it possible to combine these requirements with the design choices from the created architectural framework. This makes it possible to be able to compare them to the framework to eventually validate the framework with the PoC because both the choices of the framework (paradigm) have been met coherently with the Agricultural requirements.

2.3. Enabling technologies for IoT

The IoT paradigm speaks about enabling technologies, in which an enabling technology can be defined as a technology that can bring radical changes to its field (V. Bhuvaneswari, 2014). Through combining many supporting (enabling) technologies, the implementation of IoT solutions into the physical world is feasible. This is done by the identification, sensing, and communication of technologies (Luigi Atzori, 2010). To be able to create an architectural framework, certain enabling technologies have to be incorporated into it. The key enabling technologies are chosen on basis of the paradigm and needed components for an IoT agriculture solution and they should work supporting flow-based programming (which is a technology on its own) and will be the protocols/rules of the framework that have to be followed. In this research, key enabling technologies will be:

Communication technologies

Communication technologies are the basis of an IoT solution and a necessary component of IoT and makes communication between objects possible. Wireless innovations have played a vital role in the communication of objects and the decrease in terms of size, weight, energy usage, and radio costs will lead us to a new age in which the usage of these objects will rise significantly. This allows developers to incorporate communication radios into almost all objects and add "everything" to the above paradigm of communicating things and thus contributing to the idea of the IoT (Luigi Atzori, 2010).

Wireless Sensor Networks

A WSN is like the name implies a wireless network containing multiple objects (Luigi Atzori, 2010). Research on Wireless Sensor Networks describes a WSN as low-cost, low-power, multifunctional sensors nodes that are small in size and communicate in short distances. These sensor nodes have a collaborative effort of many nodes (I.F. Akyildiz, 2002). In recent years, interest in Wireless Sensor Networks has grown.

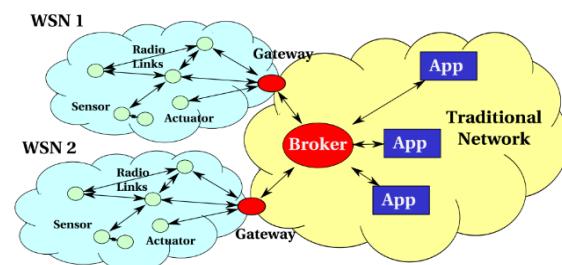


Figure 3 Structure of integrated network

This is the result of hardware components being cheap and easy to manufacture and because WSN's can offer new kinds of solutions for automation, management, and transportation. A WSN can interact with the Internet of Things (IoT) through applications and Communication Enabling Technology Standards (Luigi Atzori, 2010). In a WSNs gateways are used to connect a WSN with a traditional network. Within these WSNs multiple heterogeneous sensors can operate and collect data. These kinds of solutions/applications require transportation of data collected by connected sensors. Furthermore, these sensors are often limited when it comes to storage and processing power. So, the created data is then sent to gateways for further processing (Luigi Atzori, 2010). A so-called broker for example in the form of a Publish/Subscribe model can connect the traditional network with the WSNs (Urs Hunkeler, 2008). A Wireless Sensor Network is, therefore, a necessity in the framework and falls within the internet and things vision of the IoT paradigm under *communicating things* (Luigi Atzori, 2010).

Publish/Subscribe Systems

As a communication technology and WSN, it is chosen to go with a Publish/Subscribe system as a rule/protocol for the architectural framework. This is not only because it perfectly fits in the IoT paradigm but because loosely coupled distributed systems that have multiple interactions and many objects in large scales need increasing attention and Publish/Subscribe can provide this with open-source solutions (Patrick Eugster, 2003). The theory of the communication model publish/subscribe is that its attention is recorded to the objects that are involved in consuming/gathering information. This way of registering interest is called a subscription, which is why the involved party is called a subscriber. Through releasing their data, components that wish to generate those data are considered a publisher. The broker is the individual that guarantees that the information goes from the publishers to the subscribers. Subscriptions are organized by the broker, and customers have to directly notify the broker to subscribe. Three key types of pub/substructures are available (Urs Hunkeler, 2008):

1. Topic-based: the list of subjects is generally known in advance of topic-based programs, e.g., during an application's design process. Only on a given range of topics may subscriptions and publications be made.
2. Type-based: a subscriber specifies the type of data it is involved in (e.g., temperature data) in type-based schemes. Type-based is not a very common framework.
3. Content-based structures are the most flexible. The quality of messages it wishes to accept is defined by the subscriber. Such a subscription may be for any texts where the temperature is below a certain threshold and the light is on, containing both temperature and light readings.

Urs Hunkeler (2008) believes that topic-based systems are most appropriate for WSNs because of scalability, load-balancing, not needing to know all network details, and the subjects are known in advance. Generally, Topic-Based works as follows (see figure 2). To notify the broker of its interest in the given subject, a subscriber sends a `sub(topic)`

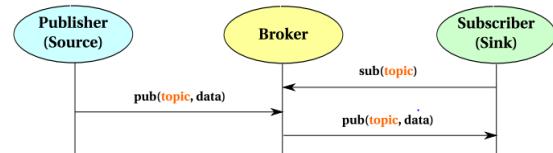


Figure 4 Publish/Subscribe communication

message, while a publisher sends a `pub (topic, data)` message that contains the data to be published along with the relevant topic. The broker passes the `pub (topic, data)` message to the subscriber if there is a match between the publisher's and the subscriber's topics. If the subject suits the topics of these subscribers, a single pub message can be transmitted to several subscribers. These features are achieved by decoupling the various communicating components from each other such that it is easy to add new data sources/consumers or to replace existing ones. This achieves a loosely coupled system (Urs Hunkeler, 2008). The public/subscribe framework masks the complexities of the underlying network for application developers and lets them rely on the architecture of the application itself which is inherent to using flow-based solutions with black-box programming. The only thing they need to know about collecting data from a certain sensor/actuator (SA) device is the subject that the SA device uses when publishing its results. Similarly, if they wish to transmit control information to a SA system, they just need to know the subjects that the SA system has subscribed to, not the exact addresses of the network. Especially if a SA system is transferred to another WSN, as long as the SA is already using the same topics for its publications and subscriptions, no change has to be made to the software and gateways. The subjects are specified by the creators of the application themselves, and not by the networks or brokers (Urs Hunkeler, 2008).

Different communication protocols

According to literature, multiple communication protocols exist for using sensors and actuators (Luigi Atzori, 2010). Below the most important communication protocols are explained (P.P., 2016).

Standard	Data Rate	Range	Energy usage	Cost
Zigbee	10kbps	10-100m	Low	Medium
Wi-Fi	1 Mb/s-6.75 Gb/s	20-100m	High	High
Bluetooth	1-24Mb/s	8-10m	Medium	Low
Lora	0.3-50 Kb/s	30km	Very Low	High
Mobile	2G: 50–100 kb/s 3G: 200 kb/s 4G: 0.1–1 Gb/s	Entire cellular area	Medium	Medium
NFC	424kbps	10m	Very Low	Low

Table I Different communication protocols for IoT

Message Queuing Telemetry transport as Publish/Subscribe model

MQTT is an open publisher and subscriber protocol. It is specifically designed for operation on low-cost and low-power devices and running over WSNs limited to bandwidth, such as ZigBee (ZigBee is an open and global WSN communication standard) or based networks. MQTT is designed in such a way that it is very easy to incorporate it on the customer side. The broker handles all the system complexities. MQTT does not define any routing or networking techniques; it assumes that the underlying network offers in-order transmission of a point-to-point, session-oriented, auto-segmenting data transfer infrastructure such as TCP/IP protocol and uses this infrastructure for message exchange. MQTT is a topic-based pub/subprotocol that provides support for hierarchical topics using character strings (Urs Hunkeler, 2008). It also makes subscribing to different subjects simpler. For example, using the hierarchical subject "wsn/sensor/F2/R248/temperature," for a temperature sensor that is located on floor "F2," room "R248" on which it could publish its data.

Middleware

Middleware is the software layer or a sub-layer placed between the operating system and the application level which includes the needed communication protocols and is included in the IoT paradigm under the *semantic* and *internet* visions. It is a necessary layer within an IoT solution, and the middleware facilitates multiple interactions between the distributed modules. However, it must be mentioned that middleware is a very diverse term and can be implemented/used in many ways (Geihs, 2001). The middleware layer hides most of the details of the different technologies used. Hiding this layer is important to exempt developers from issues that are not directly the developer's focus. Due to its major role in simplifying the implementation of new services and incorporating legacy technologies into new ones, middleware has become extremely valuable in recent years. Using middleware solutions retains developers from needing exact knowledge of all the technologies and services at lower levels used in (IoT) solutions (Luigi Atzori, 2010). It can be stated that flow-based-programming solutions fall into the middleware category because of their nature of coupling software, data, and systems with their black boxes. Also, Middleware and Flow-based programming share the same context in which underlying complex systems are hidden for the developer and that a simpler way of development and services is being achieved.

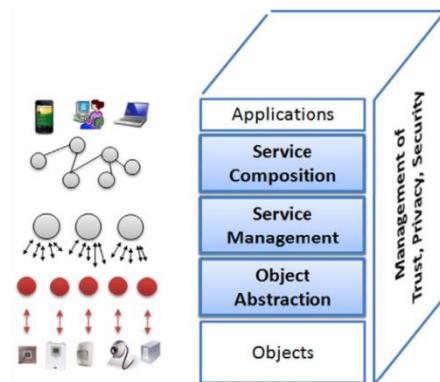


Figure 5 Example middleware lay-out

The middleware solutions consist of the following:

- Applications: are on top of the architecture and export all the systems functionalities to the final user. It exploits all the functions of the middleware layer in the form of a UI (not part of the middleware).
- Services composition is the common layer on top of the middleware architecture. It provides the functionalities for the composition of single services. Logic can be expressed in terms of workflows of business processes (e.g., black boxes).
- Service management: keeps the main functions of the objects available (status monitoring).
- Object abstraction: IoT is focused on a broad and heterogeneous array of artefacts, each of which provides unique features accessible through its dialect. There is, therefore, a need for an abstract layer capable of harmonizing access to the various devices with a shared language and mechanism.
- Objects: all the sensors, and actuators (not part of the middleware).
- Trust, privacy & security: the middleware must provide functions related to the confidence, privacy and security management of the data exchanged. The related functions can be either constructed on one layer of the previous ones or spread across the entire stack, from the abstraction of the object to the composition of the operation.

Using containers within the middleware

Containers are considered a middleware solution in which multiple software layers or sub-layers can be placed. Because as stated in the IoT paradigm, IoT solutions can bring issues when many systems-of-systems are used and have problems interfering. As well as modern applications often are assembled from multiple components and rely on other services and applications to work properly this means that when bad architecture is used, the system can have too many dependencies, and this is something you ideally want to mitigate. Using containers is a solution for this problem as containers give the user/developer the ability to package applications with all its dependencies easily and then run it smoothly in a heterogeneous development, test, and production environment. It tries to resolve the problem of dependency issues. By packaging each component and dependencies in a “container” the following issues can be resolved: conflicting dependencies, missing dependencies, platform dependencies (Merkel, 2014). A container can also help in IoT solutions combining multiple technologies.

Dialogue functionalities as a frontend communication tool

The paradigm speaks of communicating *things* in its visions and to incorporate this in the framework a chatbot which offers dialogue is used. A chatbot is a computer programmed to counterfeit a text or spoken area of intelligent communication. The user feedback is recognized by the Chatbot as well as access information to have a predefined recognition by using pattern matching. An answer from a predefined pattern is supplied to the user when the information is brought into existence in the database. Using pattern comparison, a chatbot is introduced in which the order of the sentence is remembered, and a saved answer pattern acclimatizes to the exclusive variables of the sentence. They are unable to register and respond to complicated questions and are unable to execute compound operations. The Chatbot platform is a relatively recent one. Various fields can be used in the future while applying a Chatbot (Dahiya, 2017).

Node-Red as flow-based tool and middleware solution

One flow-based programming solutions that are increasing in popularity and are frequently used in IoT solution is Node-Red. Node-Red was built by IBM's emerging technology services in 2013 and, aims to intelligently integrate hardware, APIs, and other online resources. Focused on the browser, it is a flow-based (programming) editor. It consists of two main components (Anoja Rajalakshmi, 2017). Node-Red gives an excellent visual representation of the logic that it executes and thus gives the end-user a better idea of what it is doing without writing individual lines of code.

- A. Nodes: every Node has a well-defined purpose, it receives data, does something with this data, and then passes it on to the next node (black box). The network is responsible for the correct flow of information between different nodes.
- B. Flows: by merging different nodes that are configured and stored the total combination of these Node-Red Flow Diagrams is generated.

Node-Red can act as the middleware solution because it is active on top of the architecture and export system functionalities to the front-end (or create its own UI). Furthermore, the flows can contain logic to perform actions or automation and it keeps these functions available. Node-Red supports other publish/subscribe protocols such as MQTT and relies on its connected objects. This makes it an excellent choice when building an IoT solution because it both acts as the middleware solution and supports a wide range of communication technologies (Node Red About, 2020).

2.4. Findings current literature

Below the building blocks coming from the literature review are described. These building blocks are all deemed useful for developing the artefact and will be incorporated into the framework, used to help validate the artefact or help in the creating of developing the proof of concept.

Author	Summary	IoT	Paradigm	Agri	Flow	Usability in artefact
Luigi Atzori (2010)	<p>Addresses IoT and its enabling factors, communication solutions, and technologies. In the form of a paradigm (framework). Emphasis on the synergetic approach of IoT and the combination of different fields of knowledge.</p> <p>Explains the role of middleware in technology solutions and why it is important to use middleware.</p>	X	X			<p>The paradigm visions:</p> <ul style="list-style-type: none"> • Things • Internet • Semantic reasoning <p>Enabling technologies:</p> <ul style="list-style-type: none"> • Communication technology • Publish/Subscribe • Middleware <p>Middleware lay-out</p> <p>Different components of a middleware solution are composed of.</p>
Dave Mason (2017)	A wide consensus that people should be able to program and that it can contribute to IoT. Explains flow-based-programming.	X			X	Using Node-Red as flow-based programming solution
Urs Hunkeler (2008)	Explains WSN, Public/Subscribe method and then goes further into MQTT what the protocol is and what it can mean for IoT.	X				Enabling technologies: <ul style="list-style-type: none"> • Communication technology • Publish/Subscribe
John A. Stankovic (2014)	Architecture and Dependencies address the issues and contributions to IoT research. Also mentions the role of Architecture within IoT and why it is important to set a clear architecture.	X	X			Variables for validating the artefact: <ul style="list-style-type: none"> • Architectural structure • Applicability • Deploy-ability
Meonghun Lee (2013)	Tells the role of IoT for agriculture and different roles it can full fill. Also mentions data logging.	X		X		Requirements for IoT agriculture PoC: <ul style="list-style-type: none"> • Statistical prediction • Relation analysis • IoT services
John A. Stankovic (2014)	Shows the importance and reasoning in setting up a valid Architecture.		X	X		Framework structure guidance.
Olakunle Elijah (2018)	Overview of IoT within Agriculture shows great similarity with the IoT paradigm.	X		X		Requirements for IoT PoC: <ul style="list-style-type: none"> • Devices • Communication technology • Internet • Data storage • Application • Services
Dirk Merkel (2014)	Container, introduction for consistent development and deployment.		X		X	Reason for choosing containers as a middleware solution.
Maharaja Surajmal (2017)	Introduction and definition of chatbots.					Reason for choosing chatbot as a front-end communication tool.
Ray P (2016).	A survey of Internet of Things architectures	X				Shows the different communication technologies and their differences.

Table II Elements of literature to be implemented within the framework

3. Research method

The research method described and adopted for this thesis is included in this section. Design Science will be used to organize the process of creating the framework for this research because of the exploratory and development aspect of this study. This research approach focuses on the development of solutions in the field of information systems for the needs of the enterprise and their environment. Studies in design science focus on the creation of ideas in the shape of artefacts. The goal of this research is to design a new and innovative artefact in the form of an architectural framework for the Internet of Things in combination with a flow-based approach. To help create this framework literature is going to be studied, interviews held, and a working Proof of Concept built. For the design science method, the steps of Hevner (2004) are followed. In the Design Science methodology, the goal of an artefact is to develop or solve a currently existing design problem. The Design Science approach is separated into the problem analysis, design of the artefact and validation of the artefact. In this chapter first the different steps of the process for this research are elaborated, then the role of the proof of concept depicted and last the different chosen methods of evaluating the artefact are described.

The stakeholders who will use the generated artefact in practice are consultants, developers, architects and IoT subject matter experts. As well as organizations that concentrate on providing services that achieve added value from using IoT or organizations that use IoT in combination with flow-based programming. Furthermore, the artefact can also be used by academics who are seeking knowledge on how IoT and flow-based programming are related and what their dependencies are. With the generated framework by stakeholders' architectural diagrams and blueprints of the to be developed IoT environment can be created, needed technologies and their dependencies seen. Furthermore, problems during the development and design of a flow-based IoT solution can be solved when resulting in the framework's principles. Overall, the framework should offer logic over both the usage of IoT, flow-based programming and its components when analyzed by the user.

To achieve a suitable framework, it is necessary that the solution is developed iteratively and is validated in different aspects. This is done with multiple steps taken from the Design Science approach as can be seen in figure 6. In this research great importance is given to achieving a wide and significant knowledge base which will be the input of the development phase of the framework. To further strengthen the development and knowledge base a proof of concept will be developed according to the latest iteration of the framework. While developing the PoC new input to redefine the framework will be generated as the framework is applied in practice for the first time. This process results in an iterative cycle between both the framework and the PoC in which they generate knowledge to strengthen each other. To support the development of both the framework and PoC interviews with the business will be held to retrieve basic functional requirements. The process of gathering knowledge from literature and the business environment will be ongoing.

The last step within Design Science research is adopting a strong validation method. During the framework's development, changes are added to the knowledge base of the artefact and new iterations during development created. All these changes which eventually make up the final iteration will require a method of validation. Within Design Science this can be achieved in multiple ways and for this research, it is chosen to both execute an experiment and perform an analysis on the results of interviews in which the emphasis lies on certain variables such as usability and completeness.

Furthermore, as seen in figure 6 the Design Science approach consists of understanding, applying, and evaluating knowledge which is as mentioned based upon two complementary phases in which both existing knowledge (literature) and the business needs (requirements) are used. Finally, by using the two different methods of validation the generated artefact is both validated quantitatively with the experiment and qualitatively through analyzing structured interviews with external experts.

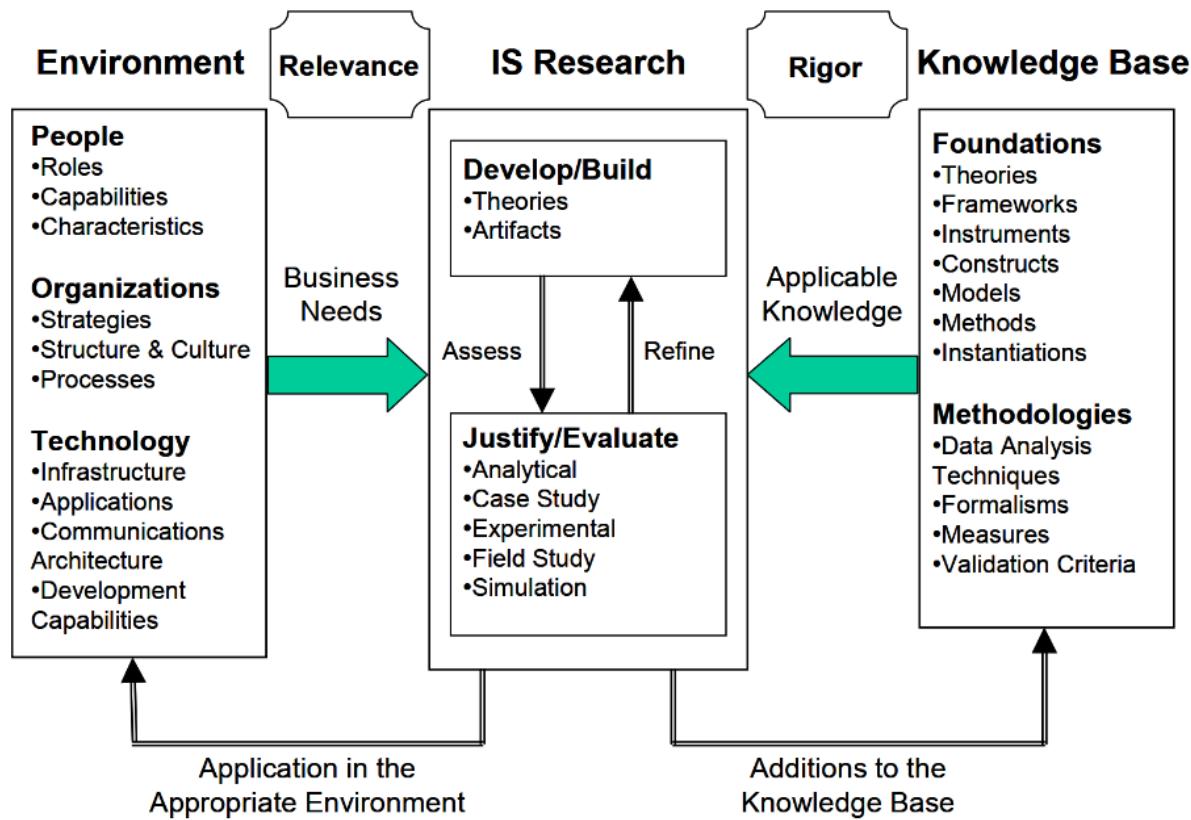


Figure 6 Design Science flow (Alan R. Hevner, 2004)

3.1. Design science research flow

The first step in design science as stated by Hevner (2004) is the problem analysis and as elaborated in chapter 1, the problem is specified for IoT flow-based solutions, in which more solutions are continued to be built-in practice while the specific role between flow-based tools and IoT is not yet clearly distinguished and no framework is available to guide developers in creating solutions. This leaves them with complex design choices on which technologies are available, how to combine these technologies and which principles need to be considered. These problems are aimed to be solved with the created artefact, which should help consultants, with their choices on which technologies are best suited or considered “building blocks” for creating their flow-based solutions.

The first step in realizing the artefact is a literature study researching the different design choices and technologies for IoT and flow-based solutions. More literature will be studied on currently existing (technology) frameworks and how these can be used to design an IoT artefact specific for flow-based solutions. Lastly, literature is also used to retrieve the different variables on which the artefact/framework can be validated on. The literature review will output different theories, frameworks, technologies, and methods in the form of a table. This knowledge base is coming from IoT research specifically and more general information management architecture, all these outputs must be structured and sorted. This step is accompanied by the retrieving of requirements in the form of structured interviews with the business for both the framework and proof of concept. While performing these interviews the business is presented with the goal of the framework and the variables on which the artefact is later examined (e.g., usability). The respondent is asked for requirements concerning these variables and general functionalities. This step is then followed by the design phase in which both the first iteration of the framework and PoC are built. Hevner (2004) elaborates that this is an iterative process as seen in figure 7. Developing the artefact iteratively will result in multiple versions of the developed artefact. Furthermore, the development of the framework and PoC will be simultaneously in which both will provide new insights and knowledge which can aid each other's development process as an ongoing cycle. The last step in Design Science is the validation of the artefact, its building blocks, and principles. In this research, this is done with both an experiment conducted on the proof of concept and an analysis of expert knowledge on the presented framework concerning the chosen variables (e.g., usability) with the help of interviews.

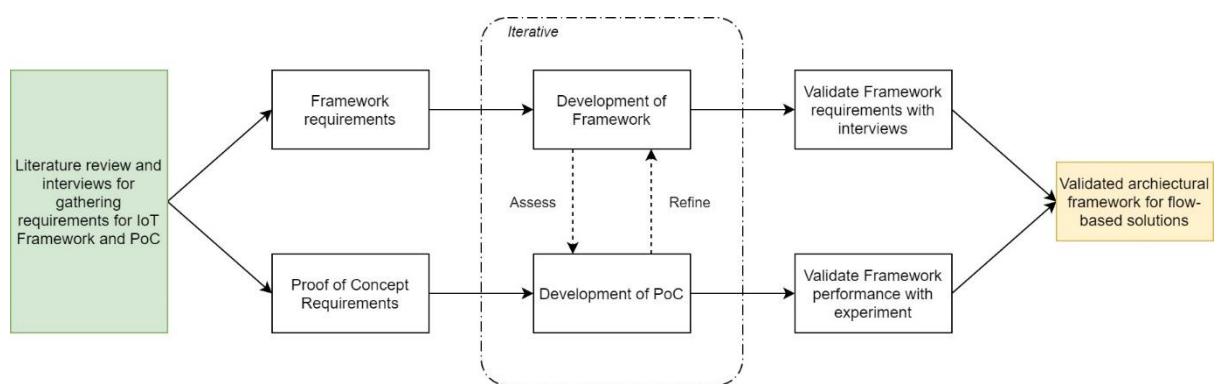


Figure 7 Research flow for artefact

N.	Main step	Outcome
1	Retrieve knowledge about elements needed to be implemented within the artefact.	Table with direct knowledge of building blocks that need to be implemented within the artefact.
2	Raw outline of the framework created with the literature and table with building blocks.	Raw draft to visualize the outlines of the artefact.
3	Retrieve functional PoC requirements with the help of interviews held within the business.	List of functional requirements that need to be implemented within the PoC according to the business.
4	Retrieve functional framework requirements with the help of interviews held within the business.	List of functional requirements that need to be implemented within the framework according to the business.
5	Development of the second iteration of the framework with the help of the functional requirements of the business and literature.	The second iteration of the framework both based on current knowledge from literature and functional requirements from the business.
5	Development of the proof of concept with the help of the second iteration of the framework and business requirements.	Functioning proof of concept housing a flow-based IoT solution.
6	Validation of the created framework with the help of interview with external experts.	Partly validated artefact on basis of the outcome of the interviews concerning the chosen variables.
7	Validation of the created framework with the experiment conducted on the proof of concept.	Partly validated artefact on basis of the outcome of the experiment concerning the chosen variables.
8	Conclusion and created artefact.	Validated artefact.

Table III Main research steps elaborated

3.2. Design of the artefact

The design of the artefact starts with the knowledge gained from the literature this will be used to create the first iteration. Following the first iteration are the interviews held with the business, the business is presented which the goal of the artefact and shown the first iteration. The business then gives input for the artefact so can be further designed and refined creating a second iteration. Parallel to these tasks the proof of concept is designed. For building the proof of concept, Agile methods will be used. This is because Agile facilitates the process of quickly creating prototypes and can handle changes in design. This means that the proof of concept will give input to the framework and vice versa to establish an iterative cycle in which knowledge is shared. Moreover, the proof of concept will also facilitate another roll in the end and that it (when it conforms to the final framework) will help validate the framework on its performance and effectiveness in an experiment. The experiment will be focused on the data that the IoT platform generates with its sensors, this data is transformed into information in the solution and can help increase the performance of the domain-specific IoT solution. The outcome of this experiment is later generalized and verified if such flow-based solutions can contribute to more effective domain-specific IoT setups. With the artefact mostly completed the first wave of interviews is followed with interviews with field-level experts. They are presented with the latest iteration of the framework which now includes lessons learned and changes from both the business as well as the proof of concept. With these interviews, external experts can help validate the framework and alternatives on current choices can be generated and analyzed. As well as that their knowledge and the information can help the architectural design to be further shaped supporting their requirements and design choices of the experts (Alan R. Hevner, 2004). With these interviews both the validation of the framework is being sought as well as new changes and lessons learned implemented to the third iteration of the framework.

3.2.1. Role of proof of concept for creating the framework

As mentioned above in the research proof of concept will be created. This proof of concept will be created according to domain-specific requirements of the business at which the research is performed and will be a fully working flow-based IoT system which composes of all the parts that make up an IoT solution from combined hardware, software, and business logic. The PoC will be fully developed in house by the writer of the thesis. The IoT proof of concept will eventually be used to validate the framework on performance and effectiveness later explained in section 3.2.1.1, as well as give input to the frameworks design principles. The components of the PoC will in the first stages come purely from literature and the same sources are used as those for the framework, as otherwise the PoC cannot be used to validate the framework when it also defines what it needs to be composed of. Furthermore, when the experiment starts the PoC is fully substantiated to the framework. This means that the final proof of concept will be built according to the rules and guidelines from buildings blocks of the framework. The IoT proof of concept needs to have a domain specification as it needs to function in practice. The domain is set by the company at which the research is performed and is going to be an agriculture solution which follows the introduction in chapter 1. Creating such a domain-specific IoT agriculture solution means that the general goal of the proof of concept is agriculture-related (e.g., increase production). This goal is independent of the framework but necessary to create a fully functioning proof of concept. The proof of concept will still be created conform the building blocks of the framework.

Therefore, the following distinction is made:

- Framework: will determine the total architecture, building blocks and enabling technologies of the solution such as communication method, things middleware as defined in the paradigm.
- Domain: will determine what the goal of the platform is. The domain will give interpretation to what the specific IoT services must deliver. Consists of a set of functional requirements which combined are the goal (which is higher production efficiency). The PoC will generate certain principles during development that have to be accounted for when using the framework.

Combined this will achieve a PoC which has a domain-specific goal and is based fully on an architecture coming from the created framework. By defining these clear boundaries between domain and framework the PoC can be correctly used to validate the framework in an experiment.

3.2.2. Collecting the functional requirements

For gathering the functional requirements for the framework and proof of concept four people within the business are asked about the requirements they have for the framework and PoC. This amount of people has been chosen based on their position and knowledge. Because the research is executed at a small start-up it was not possible to interview a larger amount of people. For gathering the requirements, the business is asked what they would like to see incorporated in the framework and which functionalities the PoC should offer concerning the variables on which the framework is later validated. When confronted with the question's respondents have already seen the first draft/outline of the framework based on current literature. To help guide these interviews a callback to the literature review is made in which clear distinctions are made about the design angle of the PoC and the importance of certain variables in a framework. This will make it easier for the second set of interviews in which independent IoT experts are asked about these variables to validate the framework and compare the results. To give structure to the requirements they are placed in a MoSCow list. With the help of the MoSCow list, individual requirements can be correctly demarcated, specification made and specify reasons if to why the certain feature is implemented or not.

Reference	Goal	Position	Current position	Knowledge of
A	Requirements	Internal	Process Automation	IoT, Agriculture, flow-based programming
B	Requirements	Internal	Automation Expert	IoT, sensor hardware, flow-based programming
C	Requirements	Internal	Automation Expert	IoT, sensor hardware, flow-based programming
D	Requirements	Internal	Founder	Lora WAN IoT projects

Table III List of people that are interviewed for the requirements

3.2.3. Evaluation of design

Using well-executed assessment techniques variables such as the usefulness, efficiency, and performance of an artefact can be demonstrated. The evaluation of an artefact is an important part of the research process. When creating the artefact, the business (SMRTR) establishes the requirements upon which the artefact is developed. This final result is then presented to external experts and evaluated. This is combined with certain variables chosen from literature based on their relationship with (validating) architectural frameworks or Design Science. In this research, the evaluation is further supported with another method of validation namely a controlled experiment. Below are the different variables presented and in which method they are applied. Further definitions of the variables and how they are going to be measured are displayed in chapter 3.2.3.1 and 3.2.3.2.

Variable	Method of validation
Performance	Controlled experiment
Effectiveness	Controlled experiment
Functionality	Interviews with expert
Completeness	Interviews with expert
Consistency	Interviews with expert
Usability	Interviews with expert
Architectural structure	Interviews with expert

Table IV List of variables and different methods chosen for validation

3.2.3.1. Controlled Experiment

By conducting an experiment, the performance/effectiveness of the framework can be validated. The experiment is executed in collaboration with the proof of concept. In this setting, the artefact and its results are studied in a controlled environment for its qualities with the help of hypotheses.

- Performance: Performance of the framework, is the framework able to facilitate better performance than its counterparts/traditional methods in term of production or created value. How performance is measured is determined by the specific domain in which the framework operates in this research the domain is Agriculture which is a requirement from within the business at which the research is executed.
- Effectiveness: Can the framework be used to facilitate the development of functioning and effective flow-based IoT solutions in which domain-specific goals can be achieved.

Hypotheses

For validating the frameworks performance/effectiveness three different hypotheses have been built. These hypotheses are created to validate if the framework can be used accordingly and successful branch-specific IoT solutions can be build when following the framework's rules, guidelines and incorporating its building blocks. These hypotheses are derived from literature on IoT within Agriculture and are based on the fact that IoT within agriculture should produce higher value by optimizing production and productivity of agricultural goods in the production process (Meonghun Lee, 2013). For these three hypotheses have been derived which focus on increasing the productivity of plant growth with three dependent variables namely, leaf length, width, and weight. The independent variable is the IoT platform itself which consists of IoT devices, communication technologies, internet, data storage and combined make up the platform (Olukunle Elijah, 2018). This platform makes use of a mediator that are the specific services that the IoT system should deliver for both automations and the specific domain. In this case, the domain is Agriculture and according to literature one of the services it should provide is relation analysis of collected data from sensors, analysing the correlation between them and give the user-specific information to perform certain actions on and thus gain an advantage compared to traditional methods (Meonghun Lee, 2013).

Furthermore, the hypotheses have been created based on the fact that a functional and working branch-specific IoT solution within Agriculture should offer better performance/effectiveness with increased production compared to traditional methods in which no automatic measurements with the help of sensors are taken and automatically processed with underlying logic (Meonghun Lee, 2013). Performance & effectiveness are also variables on which artefacts could be validated according to Hevner (2004). With combining the results of both the framework and proof of concept the framework could be partly validated on these two variables with the help of an experiment. If the proof of concept which inherits all the building blocks from the framework incorporates it's logic and design principles can achieve branch-specific IoT goals (e.g., increased production for Agriculture) it can be stated that the framework could be effectively used to create such IoT solutions and the framework is deemed as effective and can result in increased performance within branch-specific IoT solution.

The hypotheses are created to test if the framework can facilitate these goals when a solution is built according to its building blocks and principles. The following hypotheses have been created to test if the framework can be used to increase performance and effectiveness:

1. Does an IoT solution platform with agriculture domain-specific services positively influence productivity measured with plant leaf length?
The usage of an IoT automation platform using domain-specific relation analysis positively influences plant growth.
2. Does an IoT solution platform with agriculture domain-specific services positively influence productivity measured with plant leaf width?
The usage of an IoT automation platform for plants positively influences plant weight.
3. Does an IoT solution platform with agriculture domain-specific services positively influence productivity measured with plant leaf weight?
The usage of an IoT automation platform for plants positively influences plant leaf size.

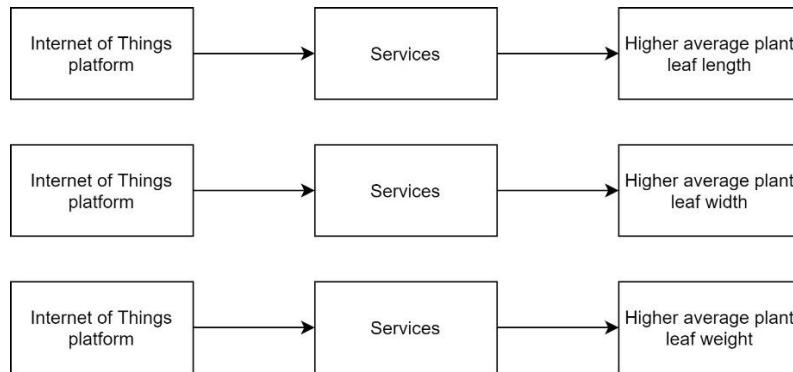


Figure 8 Hypotheses

The experiment is deemed successful when the plants in the treatment group perform better than the plants in the control group. This is tested with the three hypotheses. These hypotheses aim to measure the effects of an IoT platform with automations on plant variables such as size, weight, and growth speed. For these two identical grow-rooms for lettuce have been built. These two grow-rooms both make use of the same lighting equipment, water facilities, aeration method, and are in the same room/building. This means that the experiment will make use of a control group and treatment group to achieve a higher internal validity in which the control group will not receive any treatment. Both the treatment group and the control group will make use of the same plant growth schedule for lettuce on the number of nutrients that are supposed to be given (see table I). In this experiment, the treatment group is housed with sensors and an automation platform. These sensors can trigger automations or inform the users that it needs to perform certain actions to keep certain variables inside the right thresholds (for example add a PH lowering substance when the system notices a too high PH level). For the control group, this is not possible, the user does not know what the values of the variables are and thus can only conform to the predefined nutrient schedule. In this experiment, it is known that there are extraneous variables present. These variables can cause a historical effect when not considered. A historical effect is when events or circumstances outside of the experiment can alter the outcome of the experiment. To prevent influence from these the following measures are taken for both the treatment and control group:

- EC (nutrient level)
 - Both the TG and CG get the same schedule for nutrients as seen in table I.
- PH (acidity of the water)
 - Both the TG and CG get the same water.
- Water temperature (root temperature)
 - Both the TG and CG are set up in the same room and both have the same type of housing in which they are situated during the experiment.
- Humidity (surrounding)
 - Both the TG and CG are set up in the same room and both have the same type of housing in which they are situated during the experiment.
- Temperature (surrounding)
 - Both the TG and CG are set up in the same room and both have the same type of housing in which they are situated during the experiment.
- PPFD (Photosynthetically active radiation)
 - Both the TG and CG will get 24 hours of light daily which increases crop growth (H V Koontz, 1986).
 - Both the TG and CG will use the same light-emitting device.
 - The amount of light needed is measured with a LUX meter which has been calibrated to PPFD output by a factor of Lux * 0.023. According to literature, a PPFD output of 250 is optimal for Lettuce (Giuseppina Pennisi, 2020) (both groups are calibrated).
- Light distance to canopy (the hanging distance of the lamp)
 - Both the TG and CG will have the light 30cm above the canopy as specified by the manufacturer of the light as the ideal distance to the canopy.

Furthermore, at the start of the growing period, all the building blocks from the framework have to be finalized. This means that the experiment will take place at the end of the development process of the framework when the requirements/changes coming from the business, literature and the lessons learned from choosing the hardware & software for the PoC are implemented in the framework. This means that the basic structure of the framework stands at the moment when starting the experiment and the PoC conforms to all the building blocks, in the framework. Thus, the performance & effectiveness of the framework can successfully be tested without alterations.

Both the experimental and control group will make use of a growth schedule coming from the manufacturer of the nutrients. In which normal EC (electronic conductivity) of 1.4 will be aimed at. This EC level is recommended as a very successful EC level for lettuce (U.C. Samarakoon, 2006). The schedule accounts for a total of three weeks of plant growth. The total amount of litres in which the plants are situated is 20litres per 6 plants. The threshold for the PH of lettuce is between 6 and 7 (Giuseppina Pennisi, 2020).

Week	1	2	3
Phase	Growth	Growth	Growth
Nutrient A	32	36	36
Nutrient B	32	36	36
EC-level	1.4	1.4	1.4
PH	6/7	6/7	6/7

Multiple Time Series	Time series with control group	EG: O ₁ O ₂ O ₃ X O ₄ O ₅ CG: O ₆ O ₇ O ₈ O ₉ O ₁₀	Different tests possible (pre vs post / EG vs CG)
----------------------	--------------------------------	---	---

Figure 9 Example of time series experiment

Table IV Schedule for lettuce

To achieve the highest form of internal validity it is chosen to apply statistical control for the extraneous variables. All extraneous variables previously mentioned will be measured. Furthermore, all instruments will be the same during the whole experiment and the PH sensor is calibrated on the first measurement. To properly run an experiment with multiple observations and treatments a Quasi design and measurements will be taken with *multiple time series with a control group*. This means that for both the EG and CG there will be multiple measurements during the experiment. Because measurements can be taken digitally there will be a lot of observations. The experimental group will receive treatment when the automation detects a sensor value that is out of the pre-defined parameters and tells the user to alter it.

Data Collection

A total of 8 variables are measured in the lab-experiment divided into 5 sensors measuring certain values of the plant's surroundings. The other three variables are manually measured and give information about the plant and how it is growing. Sensor data is stored in an SQLite database.

Sensor data

Sensor type	Details	Frequency
Temperature	Measures the outside temperature	On state-change
Humidity	Measures the outside humidity	On state-change
Water temperature	Measures the temperature of the water in the hydroponics solution	On state-change
EC-level	Measures electronic conductivity of the hydroponics solution. EC-level gives information about the amounts of nutrients in the hydroponics solution	Daily
PH-level	Measures the acidity of the hydroponic solution. A good acidity level makes sure the correct nutrients can be taken up by the plant.	Daily

Table V List of sensors that collect data

From this table, the most important gathered data are these of the EC-level and PH-level. These two variables will influence the plant growth the most and are thus the most important sensors to monitor and act on. This is because the correct PH range lets the plant take up the right nutrients. A too high or too low PH can block out certain nutrients from being taken up by the plant and cause deficiencies (Laura Gough, 2000). The EC-level is the total amount of nutrients that are inside the solution. A too high level means there are too many nutrients, and this can burn the plant stopping growth, a too low EC-level also stops growth and shows discolouration of leaves.

Manually measured data

Plant variable	Details	Frequency
Plant height	Height of each plant for both the control group and the experimental group.	In between days
Plant weight	Total plant weight at harvest period of each plant for both the control group and experimental group.	End of the growth period
Plant leaf size	Average plant leaf size measured at the harvest period of each plant. For both the control group and the experimental group.	End of the growth period

Table VI List of manually measured data points

The data that is manually measured is all collected for testing the hypotheses. When all the data from both groups have been collected, it is going to be formatted and to be able to measure the differences in the experiment a Two-Sample t-Test: Assuming Unequal Variances is going to be used.

3.2.3.2. Analysis of external views on the artefact

Another form of evaluation design within Design Science is analytical evaluation. In this, a combination of both static and architecture analysis is performed with the help of interviews (Alan R. Hevner, 2004). This is done to fit the artefact into an information systems environment and verify the framework to certain variables. First, the variables on which the artefact is later validated are derived from literature and the specific definition of each variable are set. Then when both the PoC and framework are nearly finished and earlier requirements implemented, interviews with external experts and consultant can start to get their opinion on the variables in relation with the artefact. These answers can later be analysed to conclude whether the artefact is created conform to these variables. The experts will be external only to increase the external validity of the analysis and prevent bias. The following variables are considered and retrieved from the literature of both Ray P (2016) and Hevner (2004):

- Functionality: the (expected) functionalities that the artefact should deliver. Consists of a set of functions or capabilities which are associated with artefacts (Goodwin, 1987).
- Completeness: the expert knowledge when creating the artefact can maybe be missing needed principles or practices and can be incomplete because of missing rules (Motoi Suwa, 1982).
- Consistency: the used principles, practices and rules can be consistent or inconsistently used in the artefact (Motoi Suwa, 1982).
- Usability: a measure of the quality of a user's experience in interacting with information or services from the artefact (Liam O'Brien, 2007).
- Architectural structure: a measure to look if the architectural framework provides logic with its principles, practices and is divided into layers and views accordingly (Marley, 2003).

Interviews with experts on the proposed framework

To get an in-depth understanding of IoT experts' standpoint on the framework, structured in-depth interviews with the help of judgment sampling are used. For the judgment sampling experts have been chosen based on their expertise on IoT, flow-based programming or both. Someone is deemed an expert if they are active professionally in either the hardware or software side of IoT or for flow-based programming if they are actively developing with flow-based programming tools. The interviewer knows what information is needed and the respondents are asked the same set of questions. The respondent is presented with the final iteration of the framework, the variables that are used for validating the framework and their definition in advance of the interview. The interviews are guided by experts itself and their answers are recorded. In total eight different experts from outside the company are going to be interviewed. The respondents are asked questions about the chosen variables concerning the framework. The number of experts is based on the time available for the research, the restrictions active during the pandemic and the willingness of people to contribute to the research. The answers gained from the interviews are formatted per variable. This will result in an overview of the answers per variable. These answers can then be analysed and contribute to the discussion and the conclusion if the created artefact is deemed correctly build.

Reference	Position	Current position	Knowledge of
E	External	Senior Data Scientist	IoT domain Data Science
F	External	Product Architect	Automation Expert IoT domain Flow-based solutions IoT within Agriculture
G	External	Innovation Manager	Solutions Architect IoT domain
H	External	Hardware-engineer	IoT domain IoT hardware
J	External	Performing IoT research	IoT domain IoT paradigm
K	External	Flow-based programming consultant	Flow-based solutions
L	External	Flow-based programming consultant	Flow-based solutions IoT domain
M	External	Process automation consultant	Flow-based solutions IoT domain

Table IV Interview schedule

3.2.3.3. Validity

For the validation of the artefact, both the achievement of internal and external validity was sought. For increasing the internal validity, agreement with the business is achieved with interviews in which they were asked about their requirements and opinion on the artefact and their answers were incorporated into the framework based on the presented variables. Furthermore, the experiment was also used to increase internal validity and was executed within the business environment and accounted for multiple confounds to increase validity. External validity is sought with interviews with multiple experts with different backgrounds. Using external and diverse roles of experts means that a differs output of answers has been gathered.

4. Results

In this chapter, the results of the research are presented. The result section is split up into three different parts. First, the results of the proof of concept are presented of which only the most important part is highlighted. Following the proof of concept are the different iterations of which its changes, requirements and lessons learned are sorted per iteration and source. Then the framework is proposed together with the building blocks its principles. Then in the last part, the results of the validation process are presented for both the experiment and the expert opinions on the framework divided per variable.

4.1. System structure of Proof of Concept

In this chapter, the final (build) architecture of the Proof of Concept is presented (other elements are in the attachments under the attachments in I, II and III). The final architecture is based on the proposed framework presented in chapter 3.2 and conforms to a set of architectural requirements mentioned in chapter 2.1.2 to be defined as an Agriculture IoT system. The described system has been fully developed and at the moment of writing is functional for the business. All hardware elements have been chosen and combined based on the final iteration of the framework and the functional requirements of the business. The housing has been created with the help of 3D printing facilities and custom created 3D models to fit all the needed components.

The IoT agricultural system can measure multiple variables with external hardware sensors such as (water) temperature, PH, EC, and humidity all housed within the microcontroller. The microcontroller sends the generated data to the middleware solution with the help of the Publish/Subscribe protocol. The middleware layer can further process the data and perform actions, automations and measurements based on the type of service that is performed. This is all made possible with flow-based programming tool which is housed within the middleware solution. These outputs are then given to the integration layer which consists of a graphical user interface. The GUI is composed of a dashboard that shows all the measured variables and lets the user interact with them. The rules and automation in the flow-based programming tool include all the needed aspects of an IoT Agriculture solutions namely: *relation analysis* of data that is given by its sensors, *IoT services* that collect data continuously and perform reliable and *statistical prediction* which collects information and processes it to later give it to the user in the form of a UI to provide decision support. All aspects are in place to increase the efficiency of growing crops within the solution.

The proposed architecture uses the topic-based publish/subscribe method given as a building block from the framework as a communication solution. The specific communication technology chosen is MQTT which is an open-source Publish/Subscribe technology. Further, Mosquitto MQTT broker solution open-source software is used to deliver and receive messages. Mosquitto is the MQTT communications Protocol standard & compliant server/client solution. Mosquitto is intended for use in all situations where lightweight messaging is required, particularly on restricted, resource-constrained devices (Light, 2017). Using MQTT with a publish/subscribe method distinct both parts of the system. In the publishing part of the system, all the sensors and actuators are housed seen. These sensors and actuators are all connected to a NodeMcu microcontroller which houses an ESP8266 Wi-Fi chip which gives it the possibility to connect to 2.4Ghz Wi-Fi. The NodeMcu publishes the topics to the MQTT broker seen in figure 11.



Figure 10 Hardware of PoC

The publisher & broker part of the system is now fully described as seen in figure 11. The next highlighted part of the system structure is the container solution in which all the important systems are running. The Mosquitto broker together with the flow-based programming tool (Node-Red) is all housed within the container. The container virtualizes all the parts of the system as independent packages (e.g., Node-Red, Mosquitto) with all their dependencies. This results in greater flexibility and scalability (Docker, 2020). When the publisher part of the systems sends out its data the subscriber receives it with the help of the broker and passes it through to the flow-based solution. The flow-based solution can then execute its automations according to its logic and transform the data into information to provide the needed services. These components and services can be visualized and with the help of integrations passed through to a dashboard.

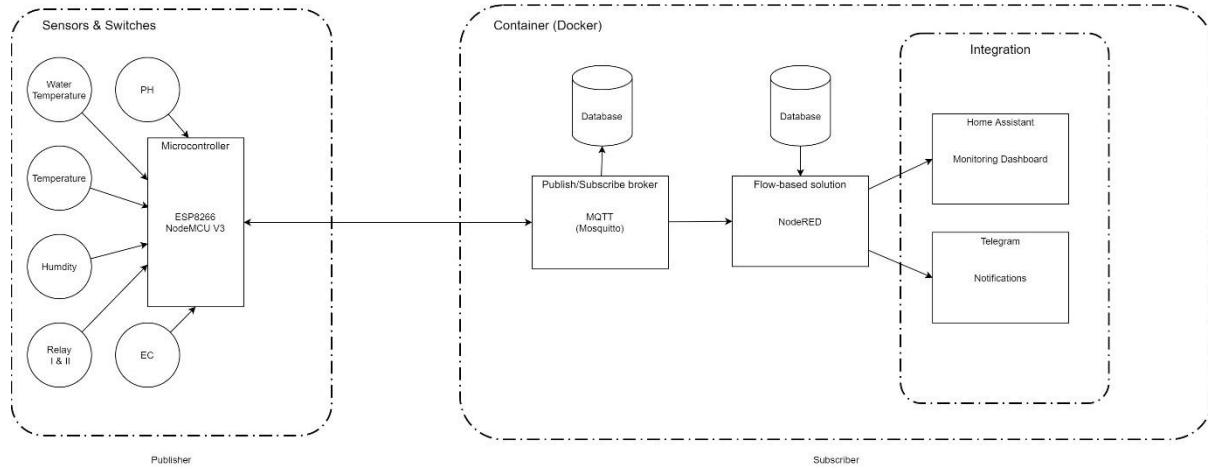


Figure 11 Architecture of the Proof of Concept

In summary, the proposed architecture incorporates all the building blocks from the framework and is created with the two visions of the framework in mind. The build of the PoC started with a top-down approach in which the variables that have to be measured were chosen first. It has *things* in which wireless sensors and actuators are used to measure the Agriculture growth space and switch on/off devices. The PoC uses the MQTT publish/subscribe protocol as the communication technology that together makes up the Wireless Sensor Network. The *middleware* takes a prominent and central place in the solution where containers are used in combination with Node-Red as flow-based solution which delivers the analysis, automation, and communication part of the solution (computation). The architecture makes use of *internet* vision in which the *things* are communicating with the middleware which eventually provides the integration and services necessary. Lastly, all the data is stored locally in the middleware solution and one database design suits all integrations. In this way, the complex sub-systems are hidden and only the necessary logic to perform system actions and automation is shown and thus can be quickly altered.

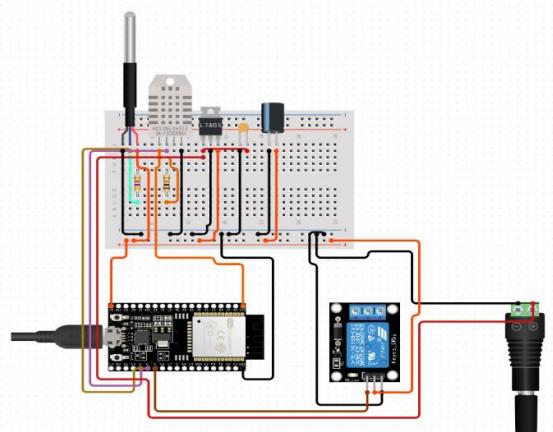


Figure 12 Hardware lay-out

4.2. Documented changes within iterations of the framework

In this chapter, all the changes and iterations that the artefact went through before its validation are presented. The first iteration was created based on literature, the second iteration on input from the business and in the third iteration compliance with external experts was sought to validate the framework. However, the external experts still had many changes and critics that they would like to see incorporated. Therefore, the framework could not be fully validated on the third iteration as changes were still being made to the framework out of necessity. To compensate this an extra iteration has been created in which these final changes from experts where implemented. After this iteration fully implemented all the extra changes from the experts the second set of validation interviews was held. This was done with both new external experts and re-occurring experts to now validate the final iteration of the framework. These results are presented in chapter 4.6.1 under validation.

The first iteration of the framework based on literature

The first iteration created purely on the knowledge gained from the literature available. For this, the table in the literature review is used to create all the building blocks mentioned in chapter 3.3.

Lessons learned:

- IoT in general has three different visions in which IoT can be approached in two different ways. These different approaches influence the result. The two approaches are a things approach (from the way you want to measure) and the internet approach (want the business wants as an end product). This means the framework needs to be able to be approached from both sides and cover the general IoT visions to be usable and familiar.
- Middleware as the name implies places the central role of the IoT solution in which all the logic is housed and data is transformed to information the is the transformation of the input. This means that the Middleware needs to be the centre of the framework.
- The importance of communication technologies, its role within the architecture and the usage of the publish/subscribe model as a communication method for different protocols.
- Different approaches to communicate with the end-user
- Different building blocks that are always present within an IoT solution such as devices, communication technology, data storage, application, and services.
- Principles for the usage of sensors and actuators such as their energy consumption, price and scale are of the utmost importance when selecting accompanying communication protocols.
- The following design elements are available for flow-based IoT solutions:
 - Paradigm
 - Things, Internet, and Semantic Reasoning
 - Enabling technologies
 - Communication technologies, Wireless Sensor Networks and Middleware
- The role of frameworks within the IoT landscape is as follows: According to John A. Stankovic, it is important to have an adequate architecture that facilitates easy connectivity, control, and ways of communication. It is also important to give attention to the question of how these objects will interact with each other across the solution (Stankovic, 2014).
- The gained knowledge can be transformed to fill the framework as follows: chosen elements need to be formatted to a single block. These single blocks have to been placed next to each other when there is a relation between them (e.g., things are connected to the microcontroller and thus placed next to each other).

The second iteration of the framework based on the Proof of Concept

The second iteration has additives based on the created proof of concept. The additives mainly come from missing principles that were found during the development of the PoC that have a relation with one of the building blocks of the framework.

Lessons learned:

- Importance of compatibility has to be better mentioned within the framework. For example, the microcontroller used in the project should support the chosen communication protocol. When using publish/subscribe certain open-source protocols as MQTT can be used these have to be supported by the microcontroller.
- Ability to deliver certain domain-specific services are fully dependent on the type of flow-based programming tool that has been selected.
- The flow-based solution (Node-Red for the PoC) should support the publish/subscribe protocol used (Mosquitto for the PoC).

Changes to the framework:

N	Specified change from interview	Implemented	Reason
1	The framework should mention differs options to communicate with the end-user and not settle for one type of integration.	Successfully implemented	When solutions are to be used within other business there may be multiple requirements on the type of integration that has to be performed.
2	The microcontroller should get a clear role in the vision of the thing of the framework	Successfully implemented	Multiple sensor and actuators can be present in such an IoT solution. However multiple sensors and actuators can also be connected to the same microcontroller. This must be made clear in the framework.
3	The middleware adopts/controls all the services such as the analyse and automate part within the flow-based solution. This role should also be distinct in the framework.	Successfully implemented	The services are domain-specific and change per application domain. It should be highlighted where the services are placed within the framework and that they are dynamically used.

Table V Changes for the second iteration

The third iteration of the architectural framework

The third iteration is based on the internal requirements from within the business. Lessons learned:

- Functionalities:
 - Functionality should offer a clear overview of its functions so people can easily use certain elements of the framework. The functions should also be in sequential order so the building blocks can succeed one another (Interview B).
 - Resemble how the data flows within the framework and at which points it is transformed to information (Interview B). Interview C agrees with (Interview B) and also implies that all data flows have to be comprehensible, so it is clear what the data in the system does and how the information is generated (Interview C).
 - Security is a factor that has to be incorporated in the framework and needs to be visible thought of for all the building blocks in the framework, just mentioning a security measure for the communication technologies used is not sufficient (Interview C).
 - Building blocks should be one-to-one transferable (Interview C).
- Completeness:
 - Regarding the dataflow, it should be clear what the input is (things) and output (services) (Interview C).
 - It could be that the framework not only delivers services with a web-application that visualizes different services in a front-end. Services could also be delivered to existing systems in the IT environment of where the IoT flow-based solution is being created. This should be included in the framework design (Interview C).
 - The IoT framework should offer all general aspects of an IoT solution known within literature and thus be all-encompassing (Interview B).
- Consistency:
 - Try to keep the same detail level of all aspects/ building blocks to prevent interference between parts of the framework (Interview B).
 - Names of elements should include logic and house the same level of detail throughout the framework (Interview C).
 - Make it clear when the building block implies that there are one or more devices/things (Interview C).
- Usability:
 - Scalable should be a requirement that has to be incorporated in all aspects of the framework to be defined usable in multiple cases (Interview C).
 - The framework should offer great visualisation of all the aspects to quickly make sure which aspects are involved (Interview B).
- Architectural structure:
 - When the framework is split into one of the visions of the paradigm It should be understandable on its own for example things, internet, or the middleware (Interview B).
 - The framework should offer logic and be understandable for the consultant who is working within tech (Interview B).
 - Clearly define a starting point of the framework (Interview B).
 - The architecture that is involved should be visible (Interview C).
 - All the data that is involved should be visible within the framework (Interview C).
 - The IoT paradigm should offer guidance within the different aspects of the framework (Interview C).

Changes to the framework:

N	Specified change from interview	Implemented	Reason
1	Not one but multiple microcontrollers have to be present in the framework (Interview C).	Successfully implemented	To better show that the wireless sensor network consists of multiple microcontrollers with multiple sensors & actuators.
2	There should be an external data block in the framework (Interview C).	Successfully implemented	Data can not only be stored locally in the middleware solution but also on external sources such as a cloud-platform.
3	The building block chatbot is too specific for the framework and should be changed to a more generic term such as "communication method" (Interview C).	Successfully implemented	The framework needs to be written down with the same detail level. No specific solutions need to be presented.
4	The name of web-application is maybe too diverse and should be more generic like integration (Interview C).	Successfully implemented	The framework needs to be written down with the same detail level. No specific solutions need to be presented.

Table VI Changes for the third iteration

4.3. Proposed Framework

In this chapter, the different iterations of the architectural framework are displayed. The framework based on literature, requirements from the and input from external experts are all implemented within different iterations. At first, the different building blocks that are incorporated in the framework are listed and explained. Next, the final architectural diagram of the framework is presented, and the different visions are explained. Following the presented framework are all the changes and lessons learned per iteration, in total four iterations have been created. The final framework is created for the technical personnel within the IT department of the business who are familiar with general IT terminology, IT landscapes and concept of IoT. The framework's goal is to support the technical/infrastructure architect and help it create a first general blueprint of their new IoT landscape in which flow-based solution is first adopted. The framework should provide a clear view of every component needed and applies logic by comparing the building block with the general known IoT paradigm to strongly depict the role of flow-based programming within IoT.

4.4. Building blocks of IoT framework

The IoT framework for flow-based solutions is based on the IoT Paradigm. Highlighted aspects of the paradigm are *internet*, *middleware*, *communication technologies* and *things* (Luigi Atzori, 2010). It is chosen to incorporate middleware and communication technologies from the paradigm as main aspects and aside from the three main visions (*internet*, *things*, and *semantic reasoning*) because they will house the most important building blocks in the architectural framework namely the flow-based solution. In the flow-based solution, all the logic is included and will eventually export all systems functionalities to the client. Around all these aspects semantic reasoning is needed to guard a working architecture that has symbiosis around its technologies. Furthermore, it is necessary to always incorporate trust, privacy and security when working with the technologies and building blocks. The building blocks elaborates on the paradigm visions and is further split out. With every building block, certain enabling technologies are laid-out. These are all necessary when using flow-based programming and combined these building blocks can make up the total system architecture. The next column gives more details over the technologies and the aspects that they cover. The last column has the framework principles that are derived from both literature and business requirements.

Paradigm		Building Block	Enabling technologies	Detail level	Principles
Semantic Reasoning & Trust, privacy and security	Internet	Applications/ Functionalities	Front-end communication	User interface	1. External system & flow-based tool should support the same integration method for compatibility.
				Communication method	
			API	Integration with external systems	2. An API strategy should be existent in the company to be able to integrate with external systems.
	Middleware	Service Composition	Flow-based programming tool	Automation	3. Should support standardized document languages to work together with external systems. 4. Business requirements should be gathered to select a suiting flow-based solution. 5. A BI strategy should be existent in the business to be able to integrate the outcoming information.
				Analyzation	
				Export of information	
		Service Management	Container	Houses all services and dependencies	6. Services should support the container to help standardization and flexibility.
	Object abstraction	Broker	Specific communication protocol	7. Support the Flow-based programming tool 8. Shares communication protocol of Wireless Sensor Network	
	Communication technologies	Wireless Sensor Network	Publish/Subscribe model	Specific communication protocol	9. Dependent on application domain and requirements which protocol is going to be used. 10. Should support the total amount of devices needed. 11. The range is a needed factor to correctly choose a suiting protocol. 12. The power consumed is a needed factor to correctly choose a suiting protocol. 13. The amount of data it needs to transport is a needed factor to correctly choose a suiting protocol.
	Things	Objects	Microcontrollers	Sensors	14. The specific type of sensors/actuators should be chosen after the application domain has been set. 15. Things chosen should have a low-energy usage. 16. Things chosen should have a low-cost (per unit). 17. The microcontroller supports the chosen communication technology. 18. Sensors chosen should be able to be easily scalable.
				Actuators	

4.5. IoT framework for flow-based solutions

In the final framework, all the different building blocks from the previous chapter, interviews and literature review are used. In the framework, two different views are present namely the *paradigm view* and the *architectural view*. The paradigm view is there to provide reasoning and logic over the different components that are listed in the architectural view. It distinct all elements in the four *views*, *internet*, *middleware*, *communication technologies*, and *things* hold together by semantic reasoning and privacy, security, and trust. This gives every user that has basic insight over the IoT tools to connect the different building blocks in the right position and compare them with each other. The choice to place semantic reasoning over these four elements comes from the goal of semantic reasoning in which all elements work in concordance with each other which is needed to be able to create a workable solution and let all the elements communicate with each other. The second view is that of the architectural view in which the building blocks are proposed that together make up the total IoT architecture for flow-based solutions. These building blocks are more technical and support each other to be able to deliver its services and features. The framework gives the user the ability to both starts from the internet vision or things. In the architectural framework, we can call this the top-down (things) or bottom-up (internet) approach. This is the general starting rule for the framework.

Paradigm view

Architectural view

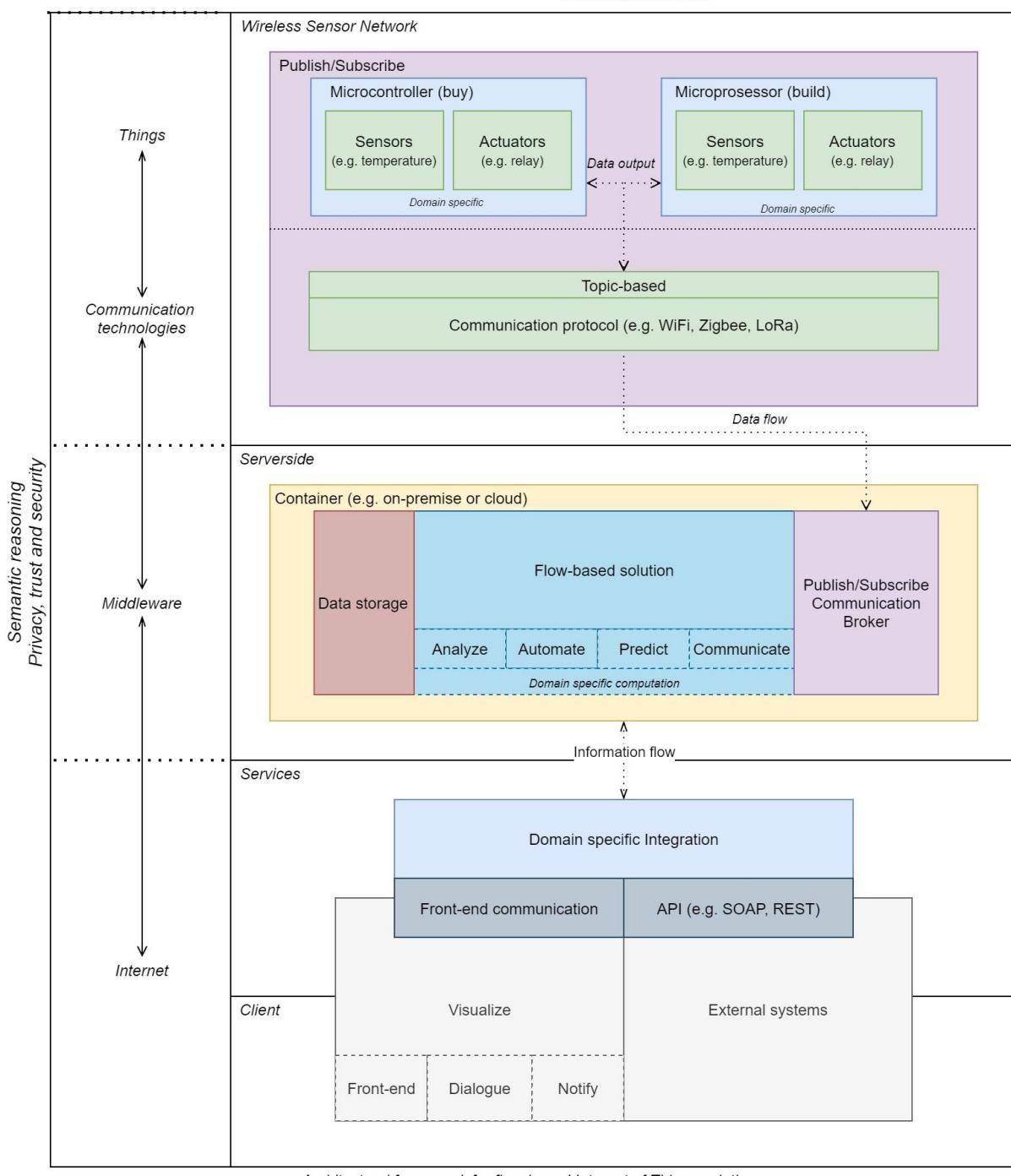


Figure 13 Final iteration/result of the flow-based Architectural framework

A. Things

Consists of the physical microcontrollers & microprocessors to which sensors or actuators are connected. The difference between these is that the controllers are pre-build logic boards, and the processors are custom logic boards. These sensors and microcontrollers are in essence endlessly scalable. The microcontroller supports the publish/subscribe communication technique used. The microcontroller can communicate with other parts of the system via its radio which is housed on the controller. The communication of the microcontroller is secured and encrypted when making use of MQTT as the communication protocol. The specific types of sensors, actuators and microcontrollers are domain-specific.

B. Communication technologies

Consists of the protocols and rules to be able to let the middleware solution communicate with the things (microcontrollers). This is done with a publish/subscribe model. The framework assumes that as publish/subscribe model a topic-based implementation is used in which the (data/sensors) topics are known in advance. By using publish/subscribe the sensors/actuators are scalable and data is more protected and can be easily sent to the right reception without knowing all network details or creating a load on the network. In the framework, the publish/subscribe model makes up both the things and communication technology. In this, all the microcontrollers and the publish/subscribe model are included in the Wireless Sensor Network and are together encrypted with its things.

C. Middleware

Is the central and connecting point of the IoT solution, the middleware houses all the logic and rules that later provide the IoT services. In the framework, the middleware contains the server-side in which docker is run, using docker gives it great flexibility, scalability, and the ability to incorporate (internal/external) data storage, flow-based programming tools (e.g., Node-Red) and perform the needed analyses, automation, and communication with the flow-based tool. This is where the raw data is transformed into information. The specific types of computation are domain-specific. The middleware receives the data sent out by the things with the help of a broker which is part of the publish/subscribe model.

D. Internet

Contains the connection/exposure to the outside and provides the client with the provided services. This is done with the help of integration with both front-end communication tools in which services as a dashboard, chatbot, and notifications through a web application are visualized. This means that the client can perform actions or ask a question to the system and can retrieve valuable information. Furthermore, it can communicate with the help of API's to connect to legacy systems. The specific types of integration are dependent on the type of domain in which the solution is built.

4.6. Validation

In this chapter, the results of the validation are presented for both the experiment and the analyses of expert opinions on the presented variables in conjunction with the framework. The results of the experts' interviews are placed under analytical.

4.6.1. Analytical

To be able to achieve the result of a correctly validated framework, multiple iterations have first been developed as seen in chapter 4.2. For this, the framework has been developed in collaboration with both the business and field-level experts and their requirements. After the final iteration of the development cycle, the analyses of the artefact with expert opinions started. This analyses of the framework by experts have been split into two different sections. First experts were asked about their opinions concerning the variables in conjunction with the framework, and if they thought the framework was sufficient enough relative to the variables. However, these first interviews with external experts resulted in more lessons learned and suggested changes to the framework which could not be ignored. The number of changes was of great enough importance to decide to create another iteration with all the changes from the external experts in it. After this final iteration has been developed the framework was again proposed to experts. This time, both new experts which have never seen the framework and the experts which were already involved have been shown this final iteration of the framework. This resulted in the second set of results on the framework. In this section first, the output of the first set of interviews with external experts are presented both with lessons learned and changes. Next, the results of the second interviews are presented in both opinions of the new and old experts have been gathered in conjunction with the variables.

Validation of the framework and final iteration of the architectural framework

After the first round of interviews with experts the final iteration has been created, in this iteration, external experts have been presented with the framework to validate it. The following variables have been presented to the experts, functionalities, completeness, consistency, usability and architectural structure. The experts where asked if the frameworks comply with their expectations concerning the variables when taking the goal of the framework into account. The results of these interviews were of such nature that important additions had to be made to the framework to be able to correctly validate it. The following this the following lessons were learned during this iteration:

- Functionalities:
 - Dependent on what the goal of the framework is the lay-out/functionalities are defined. When choosing for a business approach an input-transform-output (sideways) have to be used (Interview F).
 - Even when the framework has been created for a specific goal in which prerequisite knowledge is needed to understand the inner workings of the framework it is still better to create the framework with building blocks principles that are as generally known as possible to help the user place the framework into context (Interview H).

- Completeness:
 - Clarify the goal of the framework. For which group is it designed, is it for the business or the IT department and thus more technical. This defines the final layout of the framework. Make clear who has to interpret this framework and has to understand it (Interview F).
 - Clarify the detail level of the framework, is it a high-over framework which specifies all the needed elements of such an IoT solution or does it need to help with individual design choices (e.g., which flow-based solution to use) (Interview F).
 - Mention any pre-requisites needed in the design principles (e.g., when implementing IoT it should be clear that you already need to have some kind of data strategy or analysis) (Interview F).
 - It should be clear that the data & information flow in the framework are defined as encrypted flows to increase the thought of security within the framework (Interview J).
 - Better define why the principles are there, the principles itself are clear however the reason why they are needed, need to be better specified (Interview E).
 - Implement the option for serverless and on-premises to house the middleware solution. This can solve potential problems with connectivity (Interview G).
- Consistency:
 - Keeping the level of detail in the framework the same for all the building blocks. Do not specify single solutions mention the type of solution it conforms to (Interview F).
 - Keep the delivered services more generic do not specify specific protocols that can be used (Interview F).
 - Do not use non-self-explanatory terms such a legacy-system when you just mean existing systems. This helps in misunderstanding the building block (Interview E).

- Usability:
 - Define with which requirements the framework is going to help the user (Interview F).
 - The framework should be related to the environment in which it is going to be applied and this should be more prominent in the framework (Interview J).
 - The microcontrollers of today are powerful enough to incorporate some kind of logic and thus cannot be spoken of as dumb devices which do not house any logic (Interview G).
 - Within the computational side of the middleware machine learning and artificial intelligence can be added as a potential extension to the framework. This is because the industry is moving towards these aspects (Interview G).
- Architectural structure:
 - The role of things/devices should have a more central role in the framework because of the importance of the data that it gathers. It should be better highlighted better and its functions more elaborated in a text (Interview E).
 - Analyse, automate, and communicate is specifically mentioned within the middleware solution however it could be better generally specified as the computation part within the framework to better show the role of the middleware solution (Interview J).
 - The role of the different communication technologies is needed to be clear; examples can be provided to show which protocols can be used within the building block (Interview E).
 - Better elaborate that the publish/subscribe communication method also is encompassed in the middleware solution with its broker, this role has to be clearly defined (Interview G).
 - "Things" are operating within a certain context. This can be the functional context (measuring temperature) or geographical context (in Rotterdam, the Netherlands) but also Process or Application context such as (Warehouse, Silo, Motor, 3D printer etc.,). In this way, "things" can be approached from different perspectives (Interview G).

The following (final) changes were made according to recommendations of the experts:

N	Specified change from interview	Implemented	Reason
1	Docker must be changed to Container to suggest using any type of container solution (Interview F).	Successfully implemented	The detail level of the framework must be uniform, and Docker is to specify.
2	Transforming the framework to an input, change and output framework and rotate the framework sideways (Interview F).	Not implemented	The framework has a technical approach and is not only for the business. The framework is based on current literature and follows the same logic.
3	Give examples for specific solutions which can be used in the different building blocks but do not hardcode them (Interview F).	Successfully implemented	Specific solutions and technologies can differ per branch-specific application.
4	The chatbot must be renamed to a more generic term such as Conversational AI (Interview F).	Successfully implemented	The detail level of the framework must be uniform and Chatbot is to specify.
5	Implement using microservices into the framework for at least the data model (Interview F).		
6	Define the location where potential predictive analytics would take place as a service (Interview F).	Successfully implemented	Make a distinction in the different applications the generated data can go through because the branch-specific solutions can require any of these different data computations.
7	Make a distinction between the two different types of controllers that can be used within devices (Interview F).	Successfully implemented	Make clear that there are two types of controllers that can be used within the IoT landscape to give a complete picture.
8	Domain blocks from the framework which are dependent on the type of environment it is applied it should be highlighted (Interview J).	Successfully implemented	This makes it clear for the user of the framework for which building blocks specific changes are required depending on the specific domain in which it is applied.
9	Name-change of legacy systems to existing systems (Interview E).	Successfully implemented	Legacy systems are too specific to give a more generic approach it is renamed.
10	Add examples of the types of communication protocols that can be used in the communication technologies part of the framework (Interview E).	Successfully implemented	To better show the user which protocols can be used within the building block.
11	Mirror the framework so things are at the top and internet at the bottom (Interview E).	Not implemented	The framework is based on current literature and follows the same logic of both other architectural frameworks and the paradigm. These all follow the same structure.
12	Incorporate both MPU's & MCU's in the things part (Interview G).	Successfully implemented	This covers both choices of the controller which increases support of the framework.
13	Elaborate that the computational part of the framework could also incorporate functions such as machine learning and artificial intelligence (Interview G).	Successfully implemented	The feature of IoT incorporates these parts so it better to mention the place within the framework.
14	Change the usage of XML/CSV to REST/JSON as an example in the framework integration part (Interview G).	Successfully implemented	REST/JSON are considered more as a standard than XML/CSV which are more often named as outdated.

15	Mention the usage of security certificates as a principle within the different building blocks (Interview G).	Successfully implemented	Security cannot be skipped when using company critical surroundings.
16	Add more known generic examples of communication protocols within WSN (Interview H).	Successfully implemented	To make the framework better understandable extra examples of protocols have been added.
17	Create a more generic framework so everybody can understand it also non-IT personal (Interview H).	Not implemented	The framework has specifically been created for consultants who are more tech-savvy and have a basic understanding of IoT.
18	Turn the framework upside down this way it starts with things and ends with internet (Interview H).	Successfully implemented	The framework can both start from things and internet vision as stated within literature and explained in the text. It's logically a hardware-engineer want to start at things and finds it better to start with things however a data scientist/developer creating the front-end could have a different opinion.
19	Better elaborate the difference between micro & macro controllers (Interview H).	Successfully implemented	Micro & Macro controllers share allot of the same principles and confusion can occur.

Table VIII Changes final iteration

Occurrence of requirements

Certain feedback co-occurred multiple times during all interviews, the requirements that did are listed below and can be considered requirements with higher importance.

Change	Count
The framework should be mirrored	3
Include micro/macro controller	2
Give examples of technologies	4
Don't name individual technologies as set choices	4
Implement the thought of security throughout the framework	3

Table IX Occurrence of requirements

Results of experts on the defined variables after the final iteration

The answers from the experts on the questions asked about the variables are presented per variable.

Functionalities:

- It is very multi-layered, as institutions would like to approach it because it provides good functional separation and therefore allows you to easily determine where, for example, aggregation should take place (Interview G).
- The use of containers is ideal for scaling to higher coverage or load (Interview G).
- The transparent and simple three-layer overview is easy to explain to the beginner and specialist (Interview F).
- “Clear distinction between different partitions & tasks of architecture. I enjoy seeing the data-based structure that chronologically flows from one functionality to another.” (Interview E).
- To describe it in very general terms, it shows the most important characteristics of a flow-based solution within IoT. I consider the communication between the different layers to be of the utmost importance and this is well captured. I see that you use the publish/subscribe model, I think this is very direct and can help the user of the framework to design the solution. Certainly, because with publish/subscribe you can easily and quickly subscribe to topics and therefore quickly link them to subjects that benefit an IoT solution (Interview I).
- The separation of the 3 layers is a great advantage. What I find interesting is that you go from data to information through the middleware layer (Interview K).

Completeness:

- For a single-tenant environment (a factory, a household or a branch of a company) this is an excellent framework, and you don't need to go into it in more detail because it covers the main aspects (Interview G).
- Most of the principles hold up with other architectures. The principles may not be considered complete within an Enterprise environment, partly due to the lack of authorisation and authentication principles and basic security principles (Interview G).
- All the needed components are being considered in your framework and I like the way you have built it up. This makes a very clear overview of what you need (Interview E).
- Looking at the overall picture, I see all the important aspects of what an IoT solution should offer in it (Interview H).
- To bring parties together at an early stage, this framework is very complete (Interview F).
- The partitioning of the architecture into components makes for a clear and complete story, from initial sensor readings to end-user dialogue. (Interview E).
- The framework shows very well over a high-level perspective of how an IoT flow-based solution should function. Because it is so high-level, it is (Interview F).
- quite easy to see that the framework is complete. You can observe in one perspective that the different layers of an IoT solution come together again and how they are related to each other. The most important components are found in these three layers. The framework is also complete in the details it displays in the building blocks to help with implementation (Interview I).
- The framework includes many types of capabilities IoT devices, communication standards and output capabilities. I would like it to be indicated that the framework is usable up to a certain size of your system (Interview K).

Consistency:

- The different aspects complement each other well and are all a similar 'scale'. A neat balance that will function well (Interview G).
- Through the use of literature, empirical experiences and interviews from the field, I find consistency, not the most important aspect, but applicability. However, consistency within the framework is included as much as possible (Interview F).
- Through the usage of a consistent layout and flow annotation, the framework covers different parts of the solution in equal measure (Interview E).
- From how the level of detail is represented within the framework, I can see that the level is the constant throughout. It is just as I said before well made from a high-level overview and thus you have also applied this approach in the level of detail. You do not saddle the user of the framework with specific choices about which solution or techniques to use (Interview I).
- The consistency is good, it is zoomed in on all the components needed for such a system, without going too much into the matter. Everything has been made at the level of 1 device/data stream, which makes the whole thing easy to understand (Interview K).

Usability:

- In its present form, I see the best 'fit' to suit specific scenarios, where hard- and software are compatible (Interview G).
- As far as usability is concerned, it is a very good basic, let me put it this way, I was an IT person would simply understand it and be able to apply it (Interview H).
- What I think is very important is that the sensors and switches, for example, have remained very flexible. Everyone should be able to connect everything with their options. This has been made very clear in the framework (Interview H).
- Yes, I just literally see when I use the architecture that I see a kind of selection menu and that is ideal for me. Of course I want to use these sensors, use these protocols for wireless communication (Interview H).
- The applicability of the framework is as close as possible to the truth but helps in a concrete way to understand all parties that need to be involved in an IoT project (Interview F).
- The framework is utilitarian in the sense that every part has its function and utility, and no space is wasted. It is something you can pick up and start working with right away (Interview E).
- Once again, the high-level approach makes it very generic in the application and can therefore also be used for many specific purposes. It can therefore be used in several stages of a project even if there is already existing infrastructure. Besides, the use of the framework is made easy because you can approach the framework both top-down and bottom-up depending on the project (Interview I).
- As an IT professional, I can understand which components are needed in the basics to set up an IoT platform. You already give several hints about the technologies that are sometimes used, which also gives you an idea of what it will look like in a practical translation (Interview K).

Architectural structure:

- The separation from the building blocks described within this architecture and the integration is structurally fine. The MVC (Model-view Controller) model has long been a clear separation between data storage, access, and management (Interview G).
- As a good architecture often needs, this is a "building block" concept, one where the functionalities are neatly split up, the building blocks are existing, proven, widely used and modular. I consider it a good basis to start a framework and see architecture with the same building blocks returning more often (Interview G).
- What is also very important to me is that the visualisation of the framework is well put together and that it remains flexible. After all, everyone wants to see something in his or her way. I think you have applied that here too (Interview H).
- As a high-over solution architecture plate, this reference model is very suitable. The framework facilitates tremendously to map out the dependencies in advance for an IoT implementation (Interview F).
- Aligning framework with the sequences and flow of data makes for a functional and easy to read architectural structure (Interview E).
- When I look at your model, the first thing that comes to mind is a kind of pyramid in which the components follow each other logically and in hierarchical order, with the Things above and in the middle and lower layers you have the Internet. So, I would certainly say that the components in the framework interact logically (Interview I).
- At an architectural level, this is a clear guideline for actually setting up a system. A technical translation will still have to be made here, but that can certainly be done based on this framework (Interview K).

4.6.2. Experiment

For measuring the performance & effectiveness of the framework and to see if the framework can facilitate the development and functioning of flow-based IoT solutions in practice, an experiment was conducted. The experiment was held on the proof of concept which has been built with business requirements, incorporates all the elements of the framework and follows the same logic in which data is gathered and transformed into information. The



Figure 14 Setup of the experiment

resemblances of the framework in the PoC are identical as the data is gathered in the vision of the things with microcontrollers & sensors which is then with the help of communication technologies transferred to the middleware vision. In the middleware al, the computation and logic are housed which exposes the created information to the user with the help of integrations and visualization in the internet vision. This experiment is of utmost importance as the success of the experiment confirms if the framework is effective in achieving domain-specific IoT goals and performance is significant relative to more traditional ways of gathering and transforming needed data.

The performance was measured with three different hypotheses focused on the positive impact of the IoT platform with its services on plant leaf weight, leaf length and leaf width. The experiment started on 6/11/2020 and harvesting was done on 9/12/2020. In total three different treatments have been executed over the four weeks' time of the experiment based on the IoT platforms handling during the experiment. This was on 17/11/2020, 24/11/2020 and 04/12/2020. All three times the system told the user to act as higher the EC level because it was out of range. The system specifically told the user how many solutions had to be added based on the current sensor value to increase the EC level (as can be seen in attachments III). The user was directly notified when the value was out of range.

The difference in average leaf width (cm) of lettuce for the control group was ($M = 10,41$; $SD = 4,25$) and the experimental group ($M = 10,83$; $SD = 4,55$) and was significant ($t = -0,53$; $p < .05$).

	Control	Experimental
Mean	10,4081967	10,82615385
Variance	21,0544317	18,37977404
Standard deviation	4,25405786	4,550744673
Observations	61	65
Hypothesized Mean Difference	0	
df	122	
t Stat	-0,5274478	
P(T<=t) one-tail	0,29942029	
t Critical one-tail	1,6574395	
P(T<=t) two-tail	0,59884059	
t Critical two-tail	1,97959988	

A T-test on lettuce leaf width for the control & experimental group

The difference in average leaf length (cm) of lettuce for the control group was ($M = 12,01; SD = 3,54$) and the experimental group ($M = 12,65; SD = 3,83$) and was significant ($t = -0,97; p < .05$).

	<i>Control</i>	<i>Experimental</i>
Mean	12,0082	12,6461538
Variance	12,73443	14,861274
Standard deviation	3,539162	3,82526327
Observations	61	65
Hypothesized Mean Difference	0	
df	124	
t Stat	-0,964615	
P($T \leq t$) one-tail	0,168308	
t Critical one-tail	1,657235	
P($T \leq t$) two-tail	0,336615	
t Critical two-tail	1,97928	

A T-test on lettuce leaf length for the control & experimental group

The difference in average leaf weight (grams) of lettuce for the control group was ($M = 3,67; SD = 2,33$) and the experimental group ($M = 3,97; SD = 2,40$) and was significant ($t = -0,71; p < .05$).

	<i>Control</i>	<i>Experimental</i>
Mean	3,668196721	3,971384615
Variance	5,514375027	5,867455865
Standard deviation	2,328942987	2,403578023
Observations	61	65
Hypothesized Mean Difference	0	
df	124	
t Stat	-0,7132981	
P($T \leq t$) one-tail	0,238500803	
t Critical one-tail	1,65723497	
P($T \leq t$) two-tail	0,477001606	
t Critical two-tail	1,979280117	

A T-test on lettuce leaf weight for the control & experimental group

5. Discussion & Limitations

In this chapter, the findings which were presented in the previous paragraph are discussed. Following this, the limitations of the study will be addressed.

5.1. Main new findings

The general goal of this research was to find out how the different technologies and elements from both IoT and flow-based programming can be combined to bring logic and reasoning within an artefact. This answer was found while performing the research as new insights were found on multiple IoT enabling technologies and their relations to flow-based programming. During the literature study, all the elements that are considered within both IoT and flow-based programming were selected and set up within the artefact. Furthermore, the relations between these elements were studied and visual distinctions between these elements were made with the help of so-called building blocks. These building blocks with their relations were placed within a visual structure (the framework). Together these aspects hold up the newly created artefact which is a first in this combined field of knowledge. After refining the framework with the help of expert knowledge, validation was performed. This was achieved with the help of external interviews on certain variables (e.g., usability) and with the help of an experiment to test the performance of the framework. The results of both were presented in chapter 4 and the new finding and interpretation of these results are listed below.

5.2. Findings and interpretation of results from interviews.

In this chapter, the findings, and the interpretation of the answers of the external experts regarding the presented variables are highlighted. The variables that have been considered are *functionalities*, *completeness*, *consistency*, *usability* and *architectural structure*.

- For the role of functionalities experts commonly agree that the multi-layered approach is a strong foundation. With this approach, the user can easily determine which elements are located where and if aggregation should take place and that the structure. Experts mention that the framework offers the needed functionalities to create the first architectural diagram when a flow-based IoT solution needs to be developed, which enabling technologies are needed to create such a solution, to understand which hardware and software to use. Experts also state that because of the high-level overview the framework can help bring parties together in the early stages of development. These findings imply that the framework is understandable by its user and the main functionality offered by the framework to use it for creating the first architectural diagram of a new IoT landscape is possible.
- For completeness, the experts mention that the framework is deemed complete for single-tenant environments and that the chosen principles hold up with other architectures. The experts commonly agree that all the important aspects of an IoT environment are correctly displayed in the framework for both visions and that the role of flow-based programming is fully displayed. This implies that the framework is deemed complete for the role it is trying to fulfil as all the needed aspects are incorporated.
- For the consistency of the framework, the experts said that the different aspects complement each other well and that the consistency is included as much as possible throughout. Experts also mention that the framework covers all the different parts of the solution in equal measure. This implies that consistency throughout the framework is of the same level and included as much as possible.

- The variable usability was also deemed correctly present within the framework as experts commonly agree that the framework is deemed usable in domain-specific scenarios where both hard- and software agreements are sought. All the needed basic IoT components for the framework to be usable are incorporated such as *things*, *communication technologies*, *integrations*, and *computation*. Experts say the framework is easily applicable in practice as the framework has been created as closely possible to the truth. The framework helps in concrete ways to understand the concept of IoT and flow-based programming. Experts say the framework is something you can pick up and start working with right away. All these views of experts lead to the finding that the framework is correctly usable for its purpose.
- On terms of the structure of the architectural framework experts, all agree that the separation between visually displayed elements within the framework is clear and that the separation between the data and information is correctly displayed. Multiple experts mention that the visualisation and relations between building blocks offer a great reference model and the framework clearly show the dependencies of the building blocks. Furthermore, the flow of the framework is easily followed in the top-down approach as generated data is measured and transformed. All these statements contribute to the finding that the framework has a correct architectural structure as it is successful in bringing logic and structure in its design.

Combined these results can be interpreted in such a way that the framework has been correctly built-in terms of functionalities, completeness, consistency, usability and architectural structure. Because it was found that experts all commonly agreed in their output on the variables and are rather positive on the framework, its goal to provide an overview of all the components needed, their relations and to use the framework as an aid in the creation of the first architectural diagrams for flow-based solutions.

5.3. Interpretation of results from the experiment.

An experiment was conducted to validate if the framework could be used to create effective solutions which can achieve higher performance compared to more generic approaches. Both performance and effectiveness are domain-specific variables. This means that when a flow-based IoT solution is effective or achieves desirable performance differs per domain in which it is applied. For this research, the chosen domain was agriculture set by the business. Literature mentioned that for IoT within agriculture performance can be measured in term of the total productivity of for example crops. Effectiveness is achieved if the IoT solution contributes to the achievement of such goals as increased productivity. To test the productivity and effectiveness of the framework an experiment was conducted on the proof of concept to test whether the flow-based IoT platform could facilitate higher productivity compared to a traditional method with the use of a flow-based IoT solution with automations. For the experiment lettuce was grown in two different environments. The results of the experiment told that the lettuce for the experimental group which used the IoT-solution was longer, wider and heavier than its counterparts adding to the discussion that the framework can be used to create effective solutions and contributed to the overall goal of an increase in production.

This outcome of the results can be rationally explained as it is clear that the experimental group gained a competitive advantage purely from the flow-based IoT solution. This effect can be explained as the sensors in the IoT platform directly measured on-stage changes of the water quality which was immediately processed by the flow-based solution. The flow-based solution then instructed the user to apply certain actions to increase the water quality. This happened multiple times throughout the experiment. Furthermore, it is established that bad water conditions can block the intake of nutrients or cause severe burns to the crops itself (R Reid, 2003). Preventing this and keeping the water quality levels within range by the IoT platform explains the higher productivity.

With these results, it is implied that the framework could be used to increase the performance of domain-specific goals (such as a higher production). As well as that these results imply that the framework can be successfully used to facilitate the development of effective flow-based IoT solutions. This is implied because the framework facilitated all the needed/used components of the proof of concept (e.g., communication technologies and things) to achieve this result. The proof of concept conforms fully to the framework and is the direct result of following the framework building blocks and principles. These building blocks facilitated the measuring, transforming, and processing of data to information these results were achieved. Together it strengthens the argument that the performance and effectiveness of the framework have been successfully validated using this experiment.

5.4. Theoretical and practical implications

The findings have implications for both theory and practice. During this research, the artefact created is the first framework for the Internet of Things in which flow-based programming takes a central role in developing and creating such solutions. This artefact provides certain new features to both practices and in theory. One major contribution of the framework is, that it combines knowledge on technologies and their relations as this was previously available but scattered (especially when flow-based programming is involved). In the framework, different design elements are placed in which enabling technologies and their relations are visually highlighted offering a more logical view and embracing semantic reasoning over the elements.

Another contribution is that while performing the research and developing the framework new knowledge has been created. Knowledge was missing on the specific role of flow-based programming within IoT, how its role can be best defined and wherein the different visions of IoT flow-based programming are placed. The outcome of this is that the flow-based programming solution has been set to be housed within the middleware solution. This is because the flow-based solution takes part in connecting the different parts such as things, communication technologies and integrations while also executing and housing all the logic/execution from within the system. This could be subject to change but for now, seems best suited as experts commonly agree about its place and have practical implications as such consultants can now use an IoT framework which includes flow-based programming to get a high-level overview of the needed components and their relations when starting a new IoT flow-based project. It also appeared that the framework even for experts or other consultants with IoT knowledge seems to learn new things and place IoT in a more applicable context when presented with the framework.

Finally, with both methods of validation combined it could be stated that a new, usable and validated framework has been created which offers new functionalities in the field of both IoT and flow-based programming.

5.5. Reflection on validity

The methods of validation both covered internal and external validity. For internal validation, the sources for knowledge used are trustworthy, literature used is from reputable sources and have a significant number of citations. The experiment executed took a lot of extraneous variables into account and tried to expel them with the help of countermeasures based on literature. The knowledge gained from interviews is also marked as trustworthy because of the external factor, high diversity, criteria needed to be marked an expert and high level of knowledge from the people interviewed.

5.6. Limitations

The executed research and chosen method of Design Science know certain limitations. These limitations are discussed in this chapter. A number of these limitations occurred through earlier choices being made on the direct approach of the research others where solely on the amount of time and resources that were available. The following limitations are discussed for the experiment, interviews and chosen variables for the validation:

- One limitation regarding the variables for validation is that not all variables that are available in the literature for Design Science have been taken into account. The variables where chosen based on their considered worth for validation for this type of artefact. However, it could be the case that choosing extra variables would have further strengthened the validation and final design of the artefact.
- The experiment had its limitation in the resources and time that was available. Because of the limited time and resources, it was only possible to perform the experiment with 6 plants within the control group and 6 plants within the experimental group. If for these two factors, there were no constraints the experiment could be executed with more plants in both groups over a longer period to generate a more stable set of data.
- It cannot be stated the artefact is fully finished because further improvements to the artefact could still be made as experts still can give input for improvements. As well as that some improvements could not be implemented because they would change the direction of the research. However, these findings still imply that the framework could be further developed with new iterations.
- The experiment is used to validate the performance and effectiveness of the framework and although it proves to be effective, the experiment is executed for one specific domain, namely Agriculture. It could be the case that in practice for other specific domains other outcomes are observed contradicting these results.
- Answers given from experts regarding the variables could in some cases be interpreted in multiple ways resulting in a different change to the artefact thus impacting the outcome.
- More and other types of experts could be interviewed resulting in a wider set of results. This could have implications for the outcome of the final iteration.
- Incorporating privacy, security, and trust within the building blocks. Although it is mentioned within the framework correctly implementing these factors require resources that were not available during the period of the research.

6. Conclusion

Based on the literature, requirements from the business, input from interviews with external experts and the developed proof of concept the final framework has been created. While it went through multiple changes within iterations, we conclude that the final validated framework is as follows and leads to the answer of the main research question: *How can the different technologies within the Internet of Things be used to create an IoT architectural framework for flow-based solutions implementing different IoT design elements?*

We argue that the framework should incorporate four different aspects of the IoT paradigm and should be fully visualized. These highlighted aspects of the paradigm are *internet*, *middleware*, *communication technologies* and *things* (Luigi Atzori, 2010). It is chosen to incorporate *middleware* and *communication technologies* from the paradigm as main aspects and aside from the three main visions (internet, things and semantic reasoning) because they will house the most important building block in the architectural framework namely the flow-based solution. Around all these aspects semantic reasoning is needed to guarantee a working architecture that has symbiosis around its technologies. Furthermore, it is necessary to incorporate trust, privacy and security in all layers when working with these technologies. Within these paradigm aspects, multiple so-called building blocks are placed which further elaborates the paradigm visions and displays related enabling technologies.

Things consist of the physical microcontrollers to which sensors and actuators are connected. These sensors and microcontrollers are in essence endlessly scalable. The microcontroller can communicate with other parts of the system via its radio which is housed on the controller. Combined the hardware will output all the generated data. The specific types of sensors, actuators and microcontrollers are domain-specific and thus dependent on the needs of the environment in which it is applied.

Communication technologies consist of the protocols and rules to be able to let the middleware solution communicate with the things (microcontrollers). The topic-based publish/subscribe method is set as the technology to connect both aspects and for compatibility with the flow-based solution. Together the things and microcontroller make up the Wireless Sensor Network. The specific communication protocol is domain-specific and is dependent on the needs of the environment.

Middleware Is the central and connecting point within the framework, the middleware houses all the logic, computation and rules that provide the IoT services with the use of the flow-based programming tool (which is fully housed within the middleware). The middleware contains the server-side in which the data is stored, the needed analyses, automation, and communication is executed. The raw data is transformed into information. The specific types of computation the flow-based solutions provides are domain-specific.

Internet contains external integrations and provides the client with the provided services. This is done with the help of both front-end communication tools in which services as a dashboard, dialogue tools, and methods of notifications through a visualization are realised. This gives the client the ability to perform actions or retrieve information from and to the system. Furthermore, the integration can facilitate communication with the help of API's to connect to external systems. The specific types of integration are dependent on the type of domain in which the solution is built, and which legacy systems are present within the environment.

The paradigm view within the framework is supported by the architectural view which elaborates the different building blocks used within the framework. With every building block, certain enabling technologies and relations between the building blocks are laid out. This combination of building blocks is necessary when using flow-based programming in combination with IoT and combined these building blocks make up the total system architecture. The architectural view provides logic between the building blocks on a high-level and offers certain design principles for its building blocks. The fully visualized artefact/framework and its principles are displayed in chapter 4.2.

Furthermore, the presented framework has been validated with an experiment and with the help of interviews with external experts on multiple variables. The experts agreed that the framework was sufficient and providing clear functionalities for the goal to support the technical/infrastructure architect and help it create a first general blueprint of their new IoT landscape in which flow-based solution is first adopted. Furthermore, the experiment has proven that the framework could be used to achieve higher performance and be effective in achieving the domain-specific goal. According to the experts, the framework provides a clear view of every component needed and applies logic by comparing the building block with the general known IoT paradigm to strongly depict the role of flow-based programming within IoT. We thus conclude that an effective, useable framework has been created which has been built according to its goal and requirements. This framework can contribute to the creation of IoT solutions which in turn can help achieve domain-specific goals such as increased production for agriculture. Achieving these goals can for example support social changes as there will be an increase in the need for food with the growing population.

6.1.1. Future research

The framework although incorporating all aspects of the IoT paradigm in its current form could still be improved or expanded on. Future research could be performed on extending the framework. This expansion could focus on the different building blocks, the enabling technologies, and their specific choices per domain. These choices could be for example focus on which technologies to choose in different domains. This extension can also be focused on the elements of trust, privacy, and security. Because of the large and differs topic of security

More research could be performed on extending the validation of the performance and effectiveness of the framework. Because the current method of validation was only performed within a single domain (agriculture) it could be further strengthened. This could be achieved with future research performed with experiments on other domains such as logistics.

Bibliografie

Adafruit. (2020, 11 08). *Waterproof DS18B20*. Opgehaald van Adafruit:

<https://www.adafruit.com/product/381>

Alan R. Hevner, S. R. (2004). *Design Science In Information*. Tampa: MIS Quarterly.

Ammerveld, E. O. (2020, 12 11). Validation of framework. (K. Veldhuizen, Interviewer)

Anoja Rajalakshmi, H. S. (2017). *Internet of Things using Node-Red and Alexa*. San Francisco: ISCIT.

Antonis Tzounis, N. K. (2017). *Internet of Things in agriculture, recent advances and future challenges*. Greece: Elsevier.

Ashton, K. (2009). *That Internet of Things*. RFID Journal.

Bilgeri, D. G. (2019). *Driving Process Innovation with IoT Field Data*. Minnesota: MIS Quarterly.

Dahiya, M. (2017). *A Tool of Conversation: Chatbot*. India: JCSE.

Daniel Ota, M. G. (2011). *Benefits and Challenges of Architecture Frameworks*. Canada: ICCRTS.

Dave Mason, K. D. (2017). *Block-based versus Flow-based Programming for Naive Programmers*. Toronto: IEEE.

Dhall, A. (202, 12 18). Interview validation of framework. (K. Veldhuizen, Interviewer)

Dijk, D. v. (2020, 12 6). Retrieving framework requirements from the business. (K. Veldhuizen, Interviewer)

Docker. (2020, 11 7). *Why Docker*. Opgehaald van Docker: <https://www.docker.com/>

FAO. (2009). *How to feed the world 2050*. Rome: High-level expert forum.

Geihs, K. (2001). *Middleware Challenges Ahead*. Goethe: IEEE.

Giuseppina Pennisi, A. P. (2020). *Optimal light intensity for sustainable water and energy use in indoor cultivation of lettuce and basil under red and blue LEDs*. Elsevier.

Goodwin, N. C. (1987). *Functionality and usability*. ACM.

H V Koontz, R. P. (1986). *Effect of 16 and 24 hours daily radiation (light) on lettuce growth*. PubMed.

I.F. Akyildiz, W. S. (2002). *Wireless Sensor Networks: A survey*. Atlanta: Elsevier.

Ives, B. P. (2016). *Enhancing Customer Service through the Internet of Things and Digital Data Streams*. Houston: MIS Quarterly.

- Laura Gough, G. R. (2000). *Vascular plant species richness in Alaskan arctic tundra: the importance of soil PH*. Journal of Ecology.
- Liam O Brien, P. M. (2007). *Quality Attributes for Service-Oriented Architectures*. Pittsburgh: ICSE.
- Light, R. A. (2017). *Mosquitto: server and client implementation of the MQTT protocol*. JOSS.
- Luigi Atzori, A. I. (2010). *The Internet of Things: A survey*. Italy: Elsevier.
- Mahammad Shareef Mekala, P. V. (2017). *Smart Agriculture IoT with Cloud Computing*. Vellore: IEEE.
- Markus, M. (2020, 12 8). Interview for validating the framework. (K. Veldhuizen, Interviewer)
- Marley, S. (2003). *Architectural Framework*. NASA.
- Martin Wollschlaeger, T. S. (2017). *The Future of Industrial Communication*. IEEE.
- Masereeuw, E. (2020, 12 4). Interview for retrieving requirements for the framework. (K. Veldhuizen, Interviewer)
- Meonghun Lee, J. H. (2013). *Agricultural Production System based on IoT*. Korea: IEEE.
- Merkel, D. (2014). *Docker: Lightweight Linux Containers for Consistent Development and Deployment*. Linux journal.
- Mohsen Marjani, F. N. (2017). *Big IoT Data Analytics: Architecture, Opportunities, and open research challenges*. Malaysia: IEEE.
- Morrison, J. P. (2007). *Flow-Based Programming*. Rogers.
- Motoi Suwa, A. C. (1982). *An approach to verifying completeness and consistency in a Rule-Based Expert System*. California: AAAI.
- Nikesh Gondchawar, R. K. (2016). *IoT based Smart Agriculture*. India: ISSN.
- Node Red About*. (2020, 11 5). Opgehaald van Node Red: <https://nodered.org/about/>
- NodeMcu. (2020, 11 07). *NodeMcu About*. Opgehaald van NodeMcu:
https://www.nodemcu.com/index_en.html
- Olukunle Elijah, T. A. (2018). *An Overview of Internet of Things and Data Analytics in Agriculture: Benefits and Challenges*. IEEE.
- P.P., R. (2016). *A survey of Internet of Things architectures*. India: Journal of King Saud University.
- Patrick Eugster, P. F.-M. (2003). *The Many Faces of Publish/Subscribe*. ACM Computing.

- R Reid, J. H. (2003). *Mechanisms and control of nutrient update in plants*. International review of cytology.
- Ray, P. P. (2014). *Home Health Hub Internet of Things: An Architectural Framework for Monitoring Health of Elderly People*. IEEE.
- Stankovic, J. A. (2014). *Research Directions for the Internet of Things*. IEEE.
- U.C. Samarakoon, P. W. (2006). *Effect of EC of the Nutrient Solution on Nutrient, Uptake, Growth, and Yield of Leaf Lettuce in Stationary Culture*. Sri Lanka: PGIA.
- UN. (2020, 9 7). *Population*. Opgehaald van United Nations: <https://www.un.org/en/sections/issues-depth/population/index.html>
- Urs Hunkeler, H. L. (2008). *MQTT-S - A Publish/Subscribe Protocol For Wireless Sensor Networks*. Zurich: IEEE.
- V. Bhuvaneswari, R. P. (2014). *The Internet of Things Applications and Communication Enabling Technology Standards*. India: ICICA.
- Velde, S. t. (2020, 12 7). Interview with experts on validating the framework. (K. Veldhuizen, Interviewer)
- Vinke, C. (2020, 12 9). Validating the IoT framework. (K. Veldhuizen, Interviewer)

Attachments

I. Hardware Components PoC (selection & design)

To support the proposed architecture displayed in chapter 4.1 the following hardware components are highlighted. All these hardware components are going to be used in the proof of concept:

- NodeMcu V3:
 - These microcontrollers are low-cost, consume less power and are easy to scale. The NodeMcu makes use of open-source development kits which makes it easy to create IoT products or prototypes (NodeMcu, 2020).
- Raspberry Pi 4 2GB:
 - The Raspberry Pi is a small computer with a pocket-size used to do small operations for computing and networking. It is the predominant one, A component in the Internet of Things sector. It provides access to the internet and it is also possible to connect the automation system to the remote location control unit. There are different models of the Raspberry Pi available in various versions. Here, model Pi 2 model B is used, and it has quad-core ARM Cortex-A53 CPU of 900 MHz, and RAM of 1GB. It also has 40 GPIO pins, Full HDMI port, 4 USB ports, Ethernet port, 3.5mm audio jack, video Camera interface (CSI), the Display interface (DSI), and Micro SD card slot.
- DHT22:
 - The DHT11 is a basic, low-cost digital temperature and humidity sensor. It gives out the digital value and hence there is no need to use conversion algorithm at ADC of the microcontroller and hence we can give its output directly to data pin instead of ADC. It has a capacitive sensor for measuring humidity. The only real shortcoming of this sensor is that one can only get new data from it only after every 2 seconds.
- DS1820B:
 - A pre-wired and waterproofed version of the DS18B20 sensor. When measuring something far away, or in wet conditions. The sensor is good up to 125°C the cable is jacketed in PVC, so it is suggested keeping it under 100°C. The sensor outputs are digital, so there is no signal degradation over long distances. These 1-wire digital temperature sensors are fairly precise ($\pm 0.5^\circ\text{C}$ when used in long-range) and can give up to 12 bits of precision from the onboard digital-to-analogue converter. They work great with any microcontroller using a single digital pin, and you can even connect multiple ones to the same pin, each one has a unique 64-bit ID burned in at the factory to differentiate them. Usable with 3.0-5.0V systems (Adafruit, 2020).
- Relay 3V:
 - A relay is a switch actuated by an electromagnet that can open or close any number of switching contacts. The relays used in this system operate at low voltage to control the electromagnet (3V).

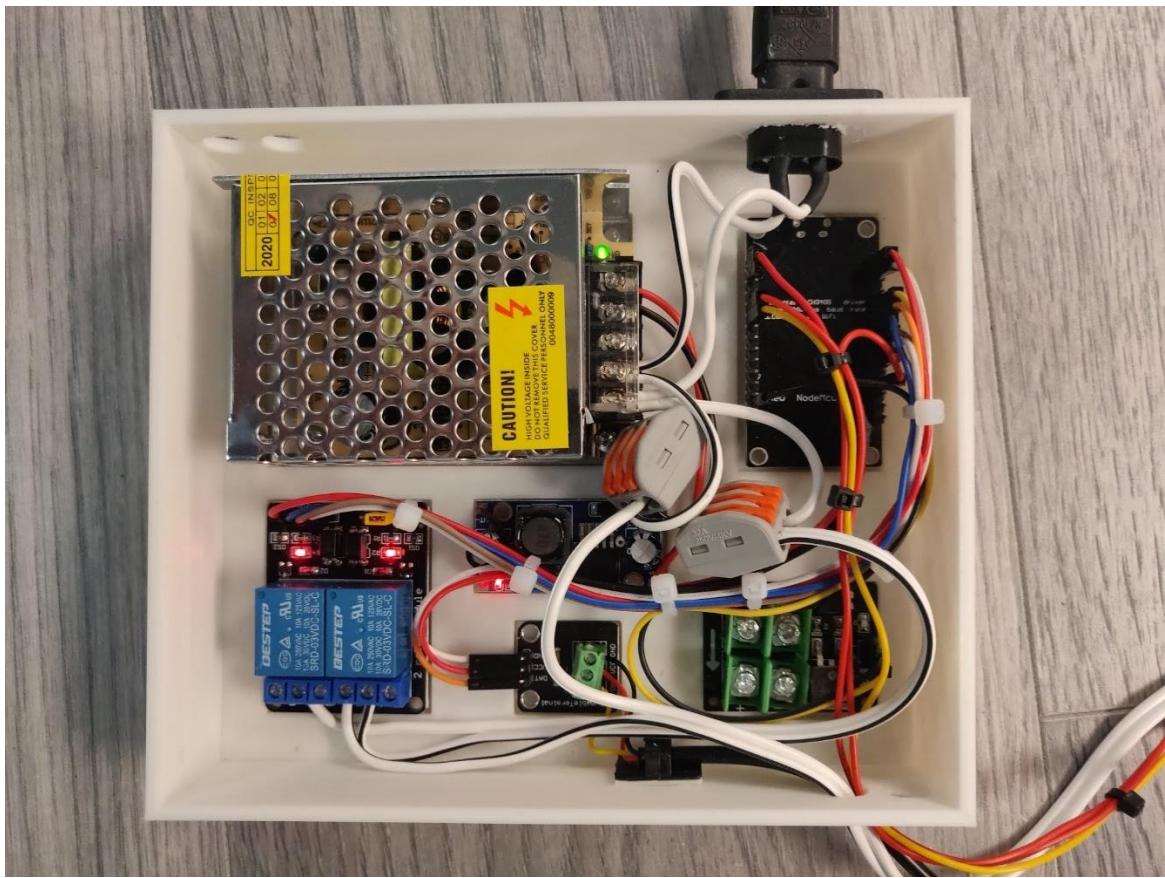


Figure 15 Hardware components from Proof of Concept

II. Data-Logging Equipment for PoC

For data-logging and database design, two tables are going to be used. These two tables are going to house the data for both automations and sensors. The first table is the state table. This table houses the entities, its state and information about when this state was changed. The second table is for events this table can be used to “fire” events (automations).

```
CREATE TABLE states (
    state_id INTEGER NOT NULL,
    domain VARCHAR(64),
    entity_id VARCHAR(255),
    state VARCHAR(255),
    attributes TEXT,
    event_id INTEGER,
    last_changed DATETIME,
    last_updated DATETIME,
    created DATETIME,
    old_state_id INTEGER,
    PRIMARY KEY (state_id),
    FOREIGN KEY(event_id) REFERENCES events
    (event_id)
)
```

```
CREATE TABLE events (
    event_id INTEGER NOT NULL,
    event_type VARCHAR(32),
    event_data TEXT,
    origin VARCHAR(32),
    time_fired DATETIME,
    created DATETIME,
    context_id VARCHAR(36),
    context_user_id VARCHAR(36),
    context_parent_id VARCHAR(36),
    PRIMARY KEY (event_id)
)
```

Sensor	Function	Specifications.
<i>DSB1820B</i>	Communicates temperature over a one-wire bus.	Operating voltage: 3.0V to 5.5V Operating temperature: -55°C to 125°C Accuracy +/-0.5°C with a range of -10°C to 85°C
<i>DHT22</i>	Send a digital signal of the temperature and humidity with using a capacitive humidity sensor and a thermistor for the temperature.	Operating voltage: 3.0V to 5.5V Temperature range -40°C to 80 °C Accuracy +/-0.5°C and +/-1%
<i>PH</i>	The meter that measures the acidity of a solution. With the use of a glass electrode.	Accuracy +/-0.05ph
<i>EC</i>	The meter that measures the electronic conductivity of a solution. With two electrodes which measure the conductivity.	Accuracy +/-2%

Table X Chosen sensor and attributes

III. Software & design for PoC

For Software & Design, it is now from the system structure that Node-Red & HASSIO as software systems have been selected. Node-Red will fulfil the role of building the automations, houses all the logic and pass-throughs the data to HASSIO which provides the user-interface. These two individual software solutions are running within a docker and can interact with each other.

Node-Red:

- Flows:
 - Notifications: Node-Red gives the possibility to supply the client with notifications with integration with Telegram. In Telegram, a bot has been created which can be asked question when there is a need for information or sends the user a message when input is needed. One flow is dedicated to all notifications.
 - Nodes for sending answers when the chatbot is asked a question.



Figure 17 Example node of the dialogue flow

- Automations: Node-Red receives the inputs from the sensor values and passes them through. One flow is dedicated to all the automations.
 - Nodes for turning on and off the lights.

HASSIO:

- Dashboard:
 - All the sensors values that are passed through MQTT are displayed on the front-end dashboard of HASSIO, furthermore, all automations can be seen and be disabled/enabled.

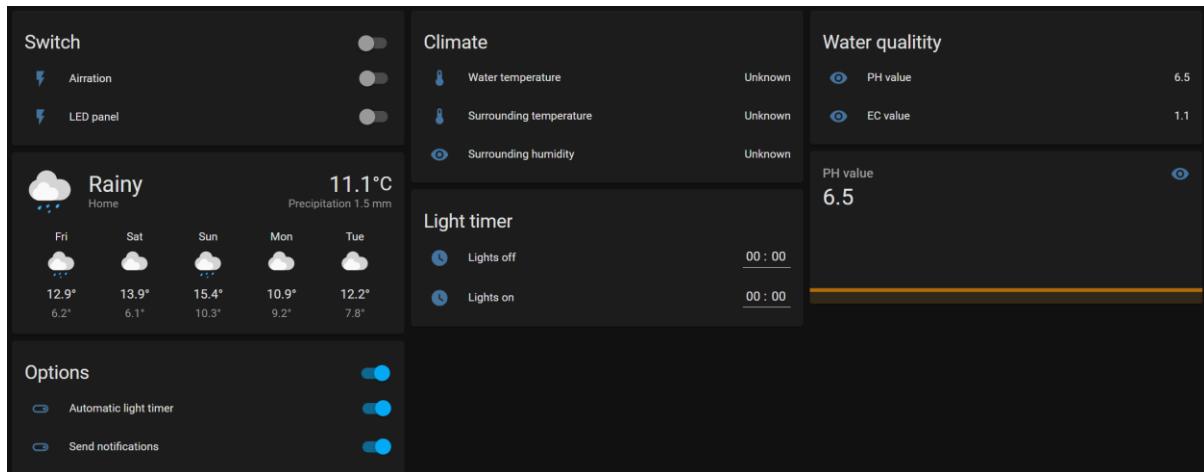


Figure 18 Example dashboard of Proof of Concept

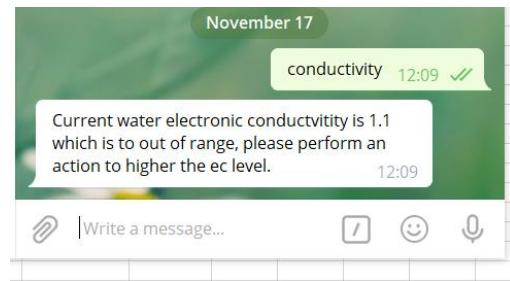


Figure 16 Example of dialogue within PoC

IV. Data from growth schedule

Control										Growth schedule
Day	1	2	3	4	5	6	PH	EC		
06/11/2020									seeds planted	
07/11/2020										
08/11/2020										
09/11/2020										
10/11/2020	Sprouted	Sprouted	Sprouted	Sprouted	Sprouted	Sprouted				
11/11/2020										
12/11/2020										
13/11/2020										
14/11/2020										
15/11/2020										
16/11/2020										
17/11/2020	0,9cm	1,1cm	1cm	1,1cm	1,4cm	1,2cm	6,5	1		
18/11/2020										
19/11/2020										
20/11/2020								1,3		
21/11/2020										
22/11/2020										
23/11/2020										
24/11/2020	2,2cm	3,8cm	4,9cm	3,4cm	2,2cm	2,4cm	6,2	1,3	Add nutrients + 04,mm voor lininaal	
25/11/2020										
26/11/2020										
27/11/2020										
28/11/2020										
29/11/2020										
30/11/2020										
01/12/2020								1,2	Add nutrients + 04,mm voor lininaal	
02/12/2020										
03/12/2020										
04/12/2020								1,1		
05/12/2020										
06/12/2020										
07/12/2020										
08/12/2020										
09/12/2020										
10/12/2020										

Table XI Data from growth schedule control

Experiment		1	2	3	4	5	6	PH	EC	System interaction
Day										
06/11/2020	eds planted									
07/11/2020										
08/11/2020										
09/11/2020										
10/11/2020	Sprouted	Sprouted	Sprouted	Sprouted	Sprouted	Sprouted		6,5	1,1	
11/11/2020										
12/11/2020										
13/11/2020								6,6	1,3	
14/11/2020										
15/11/2020										
16/11/2020										
17/11/2020	1,5cm	1,3cm	1,7cm	1,3cm	1,1cm	1,2cm		6,4	1,1	Current water electronic conductivity is 1.1 which is to out of range, please perform an action to higher the ec level.
18/11/2020										
19/11/2020										
20/11/2020								6,3	1,5	
21/11/2020										
22/11/2020										
23/11/2020										
24/11/2020	4,9cm	5,3cm	4,5cm	3,4cm	3,4cm	3,4cm		6,4	1,3	
25/11/2020										
26/11/2020										
27/11/2020										
28/11/2020										
29/11/2020										
30/11/2020										
01/12/2020							1,3			
02/12/2020										
03/12/2020										
04/12/2020							6,5	1,2	Current water electronic conductivity is 1.2 which is to out of range, please perform an action to higher the ec level.	
05/12/2020										
06/12/2020										
07/12/2020										
08/12/2020										
09/12/2020										
10/12/2020										

Table XII Data from growth schedule treatment

Controle groep 1 lengte					
Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
17	14	16	7,6	14,4	14,7
13	6,9	14,5	14,6	16,2	15
14	12,4	15	14,4	12,9	14,9
13,9	11,6	14	5,8	15,1	16,9
13	13,5	8	5	16,2	15,6
11,5	14,7	13	13,7	15,3	13,5
11,4	14	10,2	16,4	13,8	5,7
8,4	9,8	13,1	14,4	12,3	12,7
8,5	6,4	4,1	12	9,8	8,9
4,5	13,5		14,9	5,9	
6,5	11,7		5,8		
11,52	11,68	11,98888889	11,88	13,19	13,1
					12,22648148

Table XIII Data from growth schedule treatment length

Controlgroep 1 breedte					
Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
14,8	9,2	14,8	6	15	14,5
14,3	4,6	10	19	16	16,6
11	12	16	12	10,3	15,5
16	11	14,5	4,8	19	14
10	12	4,5	3,5	14	11,2
6,7	14	10	17	11	8,5
10	6	15,5	16,5	14,3	6,8
6	5	7,5	13	8	4
4,5	15,5	2,5	7,8	7,3	6,2
3	11,5		11	4	
3	8,2		4,5		
9,63	10,08	10,58888889	11,06	11,89	10,81111111
					10,67666667

Table XIV Data from growth schedule control width

Experimentele groep 1 lengte					
Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
8	7,8	16,3	12	6,4	16,5
11,5	13,9	15,8	15,8	16,2	16
13,1	15,6	15,8	13,7	15,9	17,5
16,6	17,3	12,5	11,6	14,5	12,7
15,5	15,2	14,4	16,5	17,2	15,1
16,2	14,8	13,9	16	9,8	16,2
14,9	11	15,4	15,3	16,7	13,4
16,6	13,4	11,9	18,4	10,1	14,6
11,5	7,8	6,7	5,9	14,7	7,2
7,5	10,9	9,6	7,4	10	10,1
	5,4	7,7	4,6		4,4
					5,1
13,76666667	12,97777778	13,63333333	13,91111111	13,5	14,35555556
					13,69074074

Table XV Data from growth schedule treatment length

Experimentele groep 1 breedte					
Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6
5,5	4,9	12,5	8,9	6,3	16,1
7,8	15,7	14,7	13,4	17,2	14,8
10,8	14,8	16,5	11,4	13,1	16,3
14,8	14,6	11,6	7,4	15,5	8,4
15,5	15,4	16	14,5	15,7	15,8
12,5	11,1	11,1	14,6	9,5	13,1
17,4	7,1	10,8	12,4	13,9	12
17	10	7,8	16,9	5,7	11,1
9,8	8,4	4,9	4,2	11,9	5,4
6,4	5,5	6,3	6,2	6,9	6,2
	4	5,6	4,7		3,8
					3,6
11,75	10,75	11,22	10,99	11,57	12,55555556
					11,47259259

Table XVI Data from growth schedule control length

Weight					
Control 1	Control 2	Control 3	Control 4	Control 5	Control 6
3,46	2,65	6,75	6	4	3
7	3,49	5,43	7,76	9	3
5	2,68	3,89	5,6	8	6
5,6	3,99	5,91	8	6	7
3	4,94	2,85	7	3	7
2	4,59	2,62	4	1,31	6
1	1,08	1,87	1,22	0,97	4
1,67	1,17	5,6	1,36	1,13	1
0,54	0,3	1,24	3,42	6	1,13
0,58	1,81	2,12	2,87	2	0,32
0,27	2,7				
2,738182	2,672727	3,828	4,723	4,141	3,845

Table XVII Data from growth schedule control weight

Weight					
Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
5,4	1,39	0,97	6	1,01	1,85
6,2	1,5	1,39	5	3,47	8,45
5,8	2,64	3,45	5,78	2,49	5,79
8,3	3,98	3,33	6,46	6	6,63
9,2	2,8	4,49	0,7	2,4	7,48
5,08	0,61	6,04	3,21	6	3,16
2,25	5	3,36	1,46	5,37	0,43
0,58	8	8,14	1,41	6,23	1,71
1,6	4,43	6,2	2,52	8,07	4,56
2,51	6	1,03	2,46	1,53	3,02
	4,3	7,42			1,92
4,692	3,695455	4,165455	3,5	4,257	4,090909

Table XVIII Data from growth schedule treatment weight

V. Proposed framework to business (iteration 1, literature)

The final framework is a framework in which two views are present, the paradigm view consisting of the internet, middleware, communication technologies, and things hold together by semantic reasoning. The choice to place semantic reasoning over these four views comes from the goal of semantic reasoning in which all views work in concordance with each other and this is needed to be able to create a workable solution and let these views communicate correctly with each other. The second view is that of the architectural view in which the building blocks are proposed that together make up the total IoT architecture for flow-based solutions. These building blocks are more technical and support each other to be able to deliver its services and features. As mentioned in the IoT paradigm an IoT solution can both start from within the internet of things vision. In the architectural framework, we can call this the top-down (internet) or bottom-up (things) approach. This is the general starting rule for the framework.

E. Things

Consists of the physical microcontrollers to which sensors or actuators are connected. The microcontroller supports the communication technique used and can communicate via a radio which is housed on the controller.

F. Communication technologies

Consists of the protocols and rules to be able to let the middleware solution communicate with the things. This is done with a Publish/Subscribe model. Both the things and communication technology make up the Wireless Sensor Network in the architecture.

G. Middleware

Contains the server-side in which a docker is run, this gives it the ability to incorporate data storage, flow-based tools (e.g., Node-RED and perform analyses, automation, and communication with the flow-based tool. The broker for the publish/subscribe model is also running inside the docker.

H. Internet

Contains the connection to the world wide web and provides the client with the provided services, these services are visualized in the form of a dashboard, chatbot, and notifications through a web application. This means that the client can perform actions or ask question to the system and is able to retrieve valuable information.

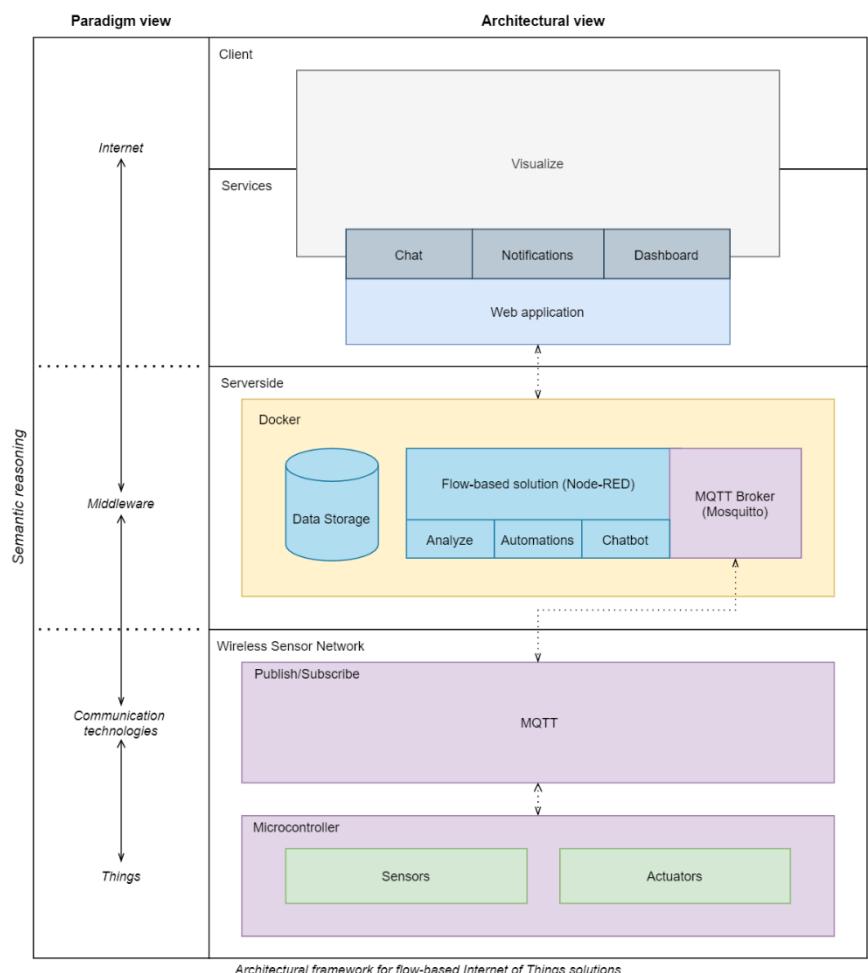


Figure 19 Architectural framework

Vision	Building Block	Rule	Addition
<i>Things</i>	Sensors	Low-energy Low-cost	Microcontrollers supporting radio firmware.
	Actuators	Radio	
	Communication things	The ability of sensors/actuators to support publish/subscribe. Support encryption protocol	
<i>Internet</i>	Communication Technologies	Publish/Subscribe model <ul style="list-style-type: none"> • Topic-based • Type-based • Content-based 	Wireless Sensor Network Protocol is encrypted
	Web of things	Client interface	E.g., dashboard/application
		Client communication	E.g., notifications and or chatbot
<i>Semantic Reasoning</i>	Middleware	Docker	Gives the possibility to install multiple additions
		Flow-based solution tool	Placed inside the docker could be e.g., Node-Red
	Reasoning over data	Data storage solution should be connected to a flow-based solution	
	Semantic Technologies	Technologies used should support the Middleware flow-based solution	

VI. Proposed framework to business (iteration 2, business requirements)

The final artefact is a framework for all the different components from the previous chapter and literature review table is used. In the framework, two different views are present namely the *paradigm view* and the *architectural view*. The paradigm view is there to provide reasoning and logic over the different components that are listed in the architectural view. It distinct all elements in the four views, internet, middleware, communication technologies, and things hold together by semantic reasoning. This gives every user that has basic knowledge over IoT the tools to quickly connect the different building blocks in the right position and compare them with each other. The choice to place semantic reasoning over these four elements comes from the goal of semantic reasoning in which all elements work in concordance with each other which is needed to be able to create a workable solution and let all the elements communicate correctly with each other. The second view is that of the architectural view in which the building blocks are proposed that together make up the total IoT architecture for flow-based solutions. These building blocks are more technical and support each other to be able to deliver its services and features. The framework gives the user the ability to both starts from the internet vision or things. In the architectural framework, we can call this the top-down (internet) or bottom-up (things) approach. This is the general starting rule for the framework.

I. Things

Consists of the physical microcontrollers to which sensors or actuators are connected. These sensors and microcontrollers are in essence endlessly scalable. The microcontroller supports the publish/subscribe communication technique used. The microcontroller can communicate with other parts of the system via its radio which is housed on the controller. The communication of the microcontroller is secured and encrypted when making use of MQTT as the communication protocol.

J. Communication technologies

Consists of the protocols and rules to be able to let the middleware solution communicate with the things (microcontrollers). This is done with a publish/subscribe model. The framework assumes that as publish/subscribe model a topic-based implementation is used in which the (data/sensors) topics are known in advance. By using publish/subscribe the sensors/actuators are scalable and data is more protected and can be easily sent to the right reception without knowing all network details or creating a load on the network. In the framework, the publish/subscribe model makes up both the things and communication technology. In this, all the microcontrollers and the publish/subscribe model are included in the Wireless Sensor Network and are together encrypted with its things.

K. Middleware

Is the central and connecting point of the IoT solution, the middleware houses all the logic and rules that later provide the IoT services. In the framework, the middleware contains the server-side in which docker is run, using docker gives it great flexibility, scalability, and the ability to incorporate (internal/external) data storage, flow-based programming tools (e.g., Node-Red) and perform the needed analyses, automation, and communication with the flow-based tool. The middleware receives the data sent out by the things with the help of a broker which is part of the publish/subscribe model.

L. Internet

Contains the connection to the outside and provides the client with the provided services. This is done with the help of integration with both front-end communication tools in which services as a dashboard, chatbot, and notifications through a web application are visualized. This means that the client can perform actions or ask a question to the system and can retrieve valuable information. Furthermore, it can communicate with the help of API's to connect to legacy systems.

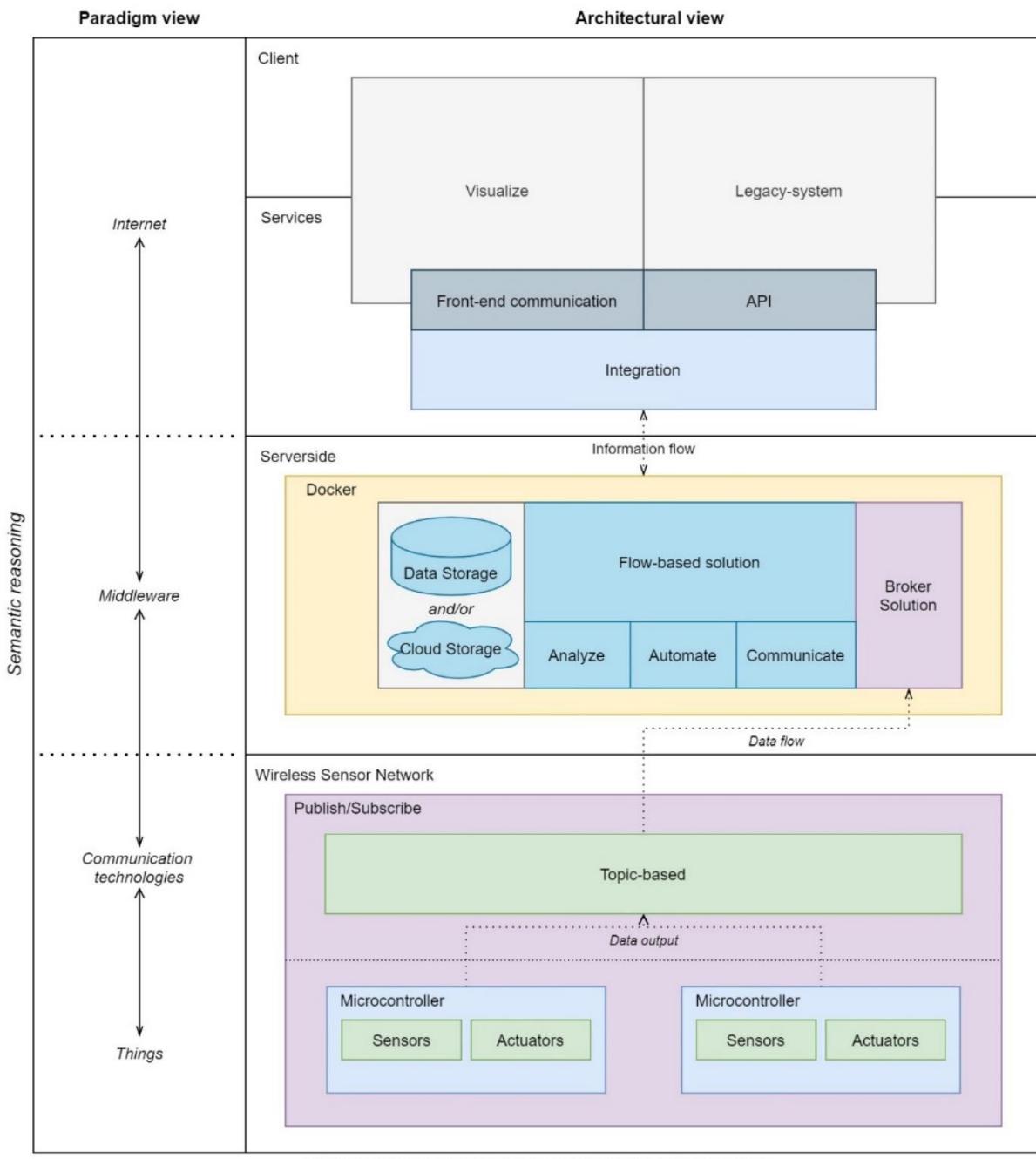
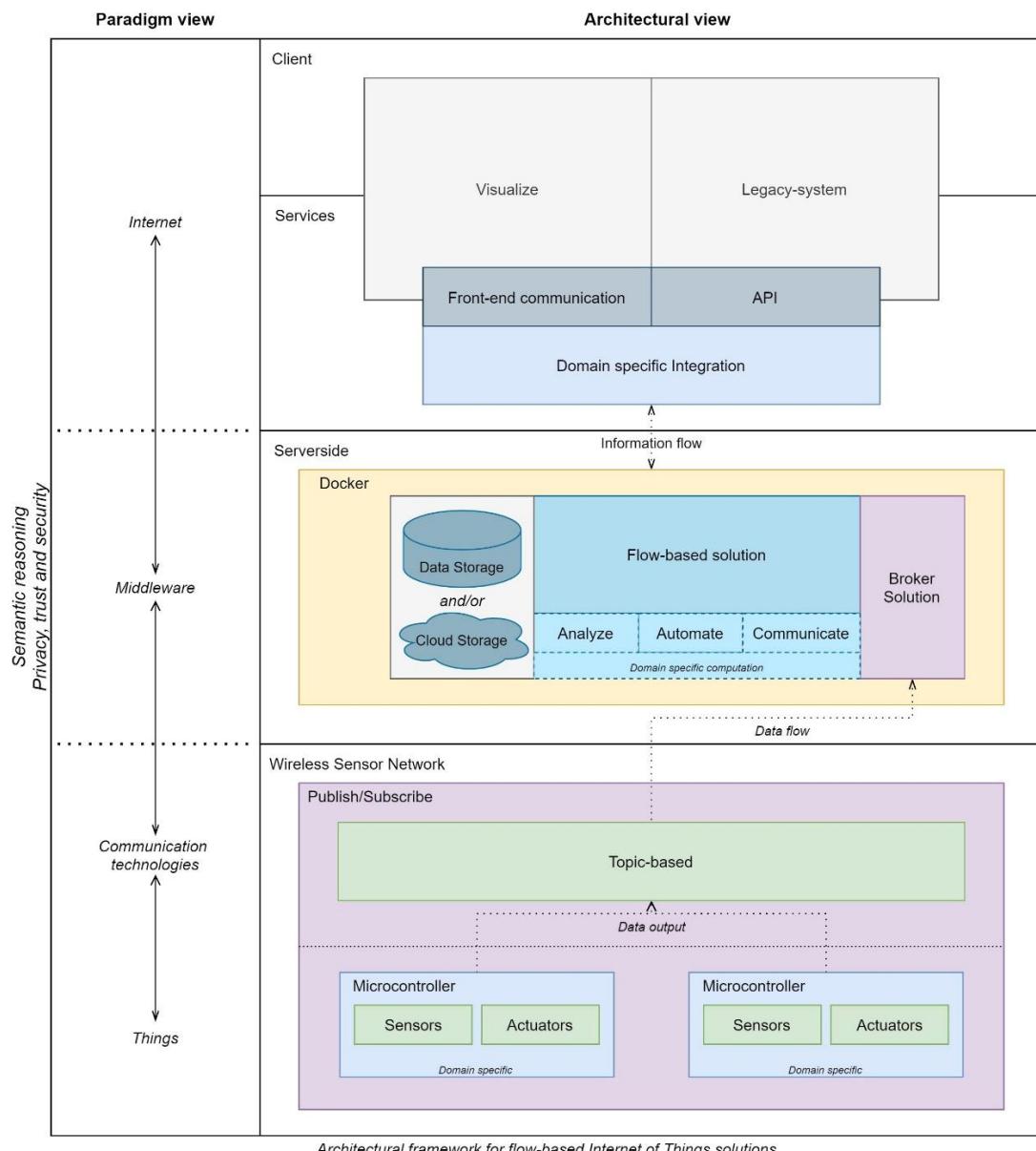
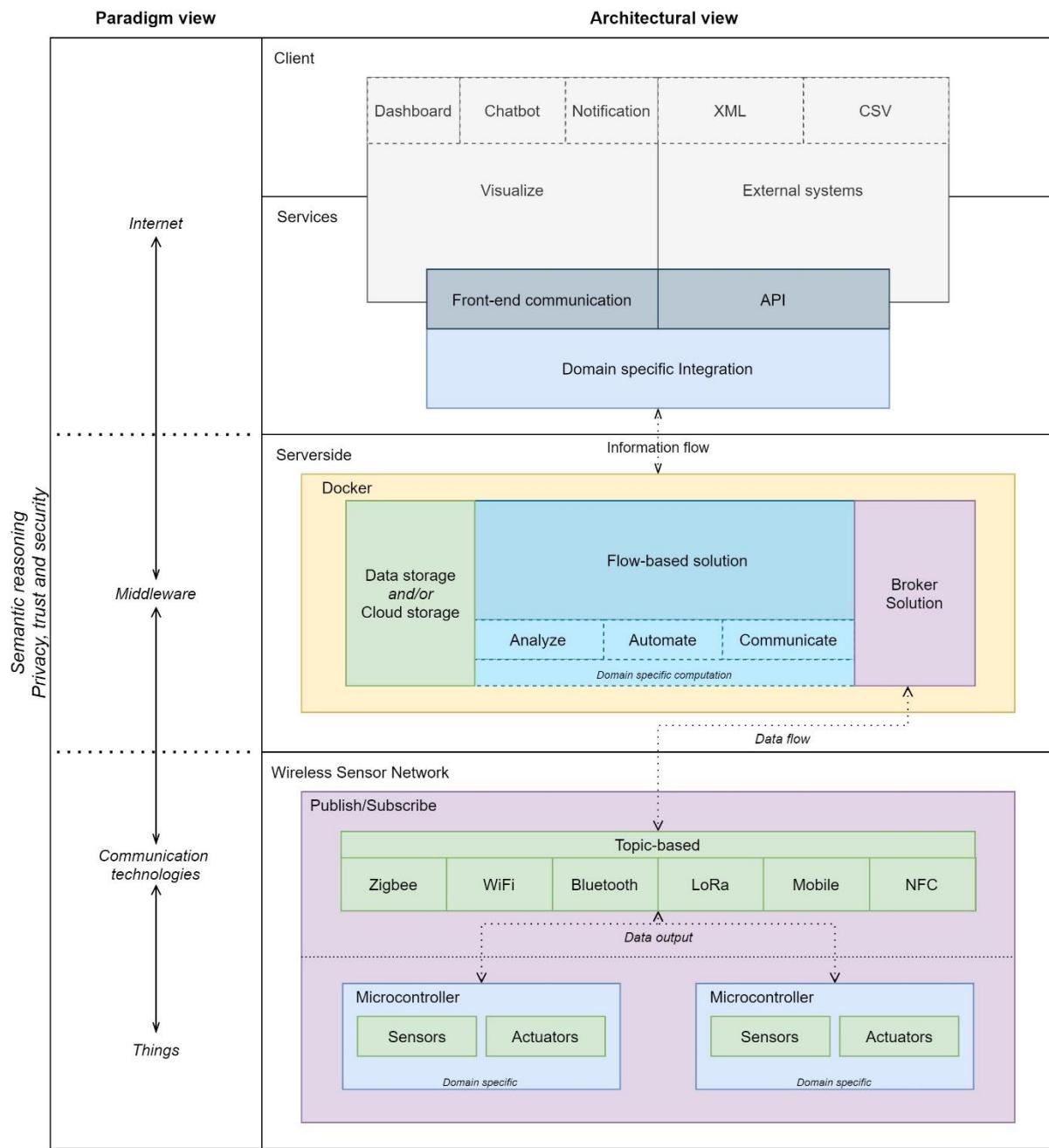


Figure 20 Architectural framework

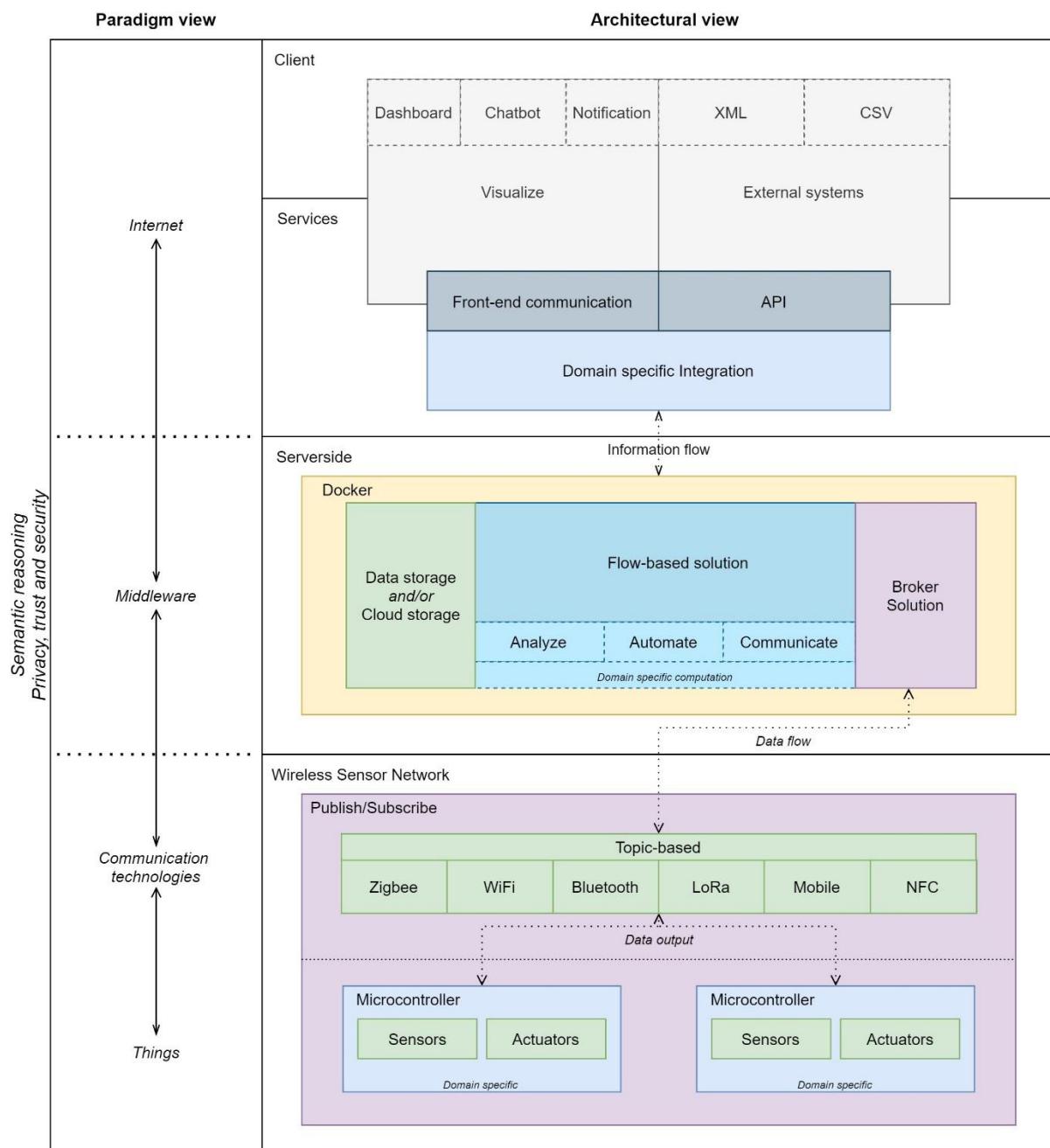
VII. Proposed framework to business (iteration 3, after Simeon)



VIII. Proposed framework to business (iteration 3, after Marijn)



IX. Proposed framework to business (iteration 4)



X. Requirements for PoC

Below are the business requirements of the proof of concept. If a requirement is not going to be implemented a reason will be placed in the comment section.

ID	Source	Priority	Original Requirement	Comment
2	Interview B	Must	Digitally see the progression	
3	Interview B	Must	Ability to receive notifications.	
4	Interview B	Must	Retrieve updates about statuses.	
5	Interview B	Must	Show graphs on certain values.	
6	Interview B	Must	Have a sensor that measures water temperature.	
7	Interview B	Must	Ability to draw conclusions based on the data.	
8	Interview B	Should	Ability to talk to a chatbot.	
9	Interview B	Should	Ability to connect it to smart home applications.	
10	Interview B	Must	The solution needs to be secure	
11	Interview B	Must	Ability to measure the temperature	
12	Interview B	Will not have	Ability to measure the oxygen level	Measuring the oxygen level will need expansive and out of scope measuring devices.
13	Interview B	Could have	Ability to measure the level of light	
14	Interview B	Will not have	Self-repairing system	Out of scope.
15	Interview B	Should	Conclusions with the analyzed data	
16	Interview B	Should	Pen-testing	
17	Interview B	Could have	Hands-off	Because of time constraints, a fully hands-off system is not possible
18	Interview B	Must	Provide a dashboard for the client	
19	Interview D	Must	Provide monitoring	
20	Interview D	Must	Give insights	
21	Interview D	Must	Historic data	
22	Interview D	Must	Compare data	
23	Interview D	Must	Communication with the user	
24	Interview D	Must	Automate certain actions	
25	Interview D	Must	Solution needs to be secure	

XI. Criteria for validating the framework

Below are the business requirements for the framework. These are requirements that the business finds useful to be implemented inside of the framework. All the requirements are categorized from one of the variables that are later used to validate the framework with experts.

ID	Source	Original Requirement	Variable from literature
2	Interview A	Duidelijk als een overzicht samenkommen	Functionalities
3	Interview A	Er is nauw beschreven wat alle functies doen.	Functionalities
4	Interview A	Overzichtelijkheid zodat mensen het een op een over kunnen nemen.	Functionalities
5	Interview A	Sequentieel de verschillende aspecten in elkaar over laten lopen.	Functionalities
6	Interview A	Verwacht dat dezelfde technieken gebruikt worden over het gehele framework.	Completeness
7	Interview A	Verwacht dat gekozen zaken met elkaar overweg kunnen.	Completeness
8	Interview A	Bouwblokken van het framework komen uit de literatuur.	Completeness
9	Interview A	Aan literatuur gecontroleerd op compleetheid van aspecten IoT.	Completeness
10	Interview A	Bouwblokken van het framework hebben een onderlinge correlatie.	Consistency
11	Interview A	Genoemde technieken hebben overal dezelfde benaming.	Consistency
12	Interview A	Genoemde technieken worden hetzelfde benaderd als in de literatuur	Consistency
13	Interview A	Duidelijk aangeven hoe het framework moet worden gebruikt/gestart	Consistency
14	Interview A	Leidraad zijn wanneer er in de praktijk een oplossing wordt gebouwd.	Usability
15	Interview A	Het wordt zo beschreven dat iedereen het kan oppakken en de informatie op zo manier wordt gepresenteerd dat leken het begrijpen.	Usability
16	Interview A	De verschillende fasen moeten duidelijk gevisualiseerd zijn in het framework en het moet meteen duidelijk zijn wat je nodig hebt om het te gebruiken.	Usability
17	Interview A	Het moet logica bieden en voor een consultant makkelijk te lezen zijn.	Architectural structure
18	Interview A	De lagen moeten duidelijk verdeeld zijn.	Architectural structure
19	Interview A	IoT zou splitsbaar moeten zijn in verschillende fasen waarin de duidelijk de verbindende tools (bouwblokken) terugkomen.	Architectural structure
20	Interview A	Wat is nou eigenlijk het startpunt.	Architectural structure
21	Interview A	Allesomvattend zijn voor flow-based IoT oplossingen	Architectural structure

ID	Source	Original Requirement	Variable from literature
1	Interview B	Algemeen toepasbaar op verschillende onderwerpen (wel flow-based).	Functionalities
2	Interview B	Componenten moeten vervangbaar/uitgebreid kunnen worden	Completeness
3	Interview B	Framework moet volledig op zichzelf staand zijn geen andere input nodig.	Completeness
4	Interview B	Consistentie in de verwerking, inhoudelijk en het framework moet altijd doen wat er verwacht wordt door de gebruiker.	Consistency
5	Interview B	Consistentie in de toepassing het framework moet duidelijk zijn beschreven en opgesteld.	Consistency
6	Interview B	Algemeen toepasbaar en wendbaar in de toepassing	Usability
7	Interview B	Wendbaar	Usability
8	Interview B	Eenduidig overzicht	Usability
9	Interview B	Uitgebreid door componenten te vervangen of toe te voegen	Usability
10	Interview B	<i>Duidelijk omschreven wat de kern van het framework is en wat die moet bieden</i>	Architectural structure
11	Interview B	Duidelijk zijn waar je het framework op kan toepassen	Architectural structure
12	Interview B	Niet heel erg groot zijn	Architectural structure
13	Interview B	Componenten moeten uitwisselbaar (per component input-proces-output beschrijven)	Architectural structure

ID	Source	Original Requirement	Variable from literature
1	Interview C	Architectuur moet zichtbaar zijn	Functionalities
2	Interview C	Datastromen inzichtelijk hebben en ook wat je doet met die data.	Functionalities
3	Interview C	Hoe wordt data-informatie	Functionalities
4	Interview C	Plek zijn in het framework waar de data output/input gevisualiseerd wordt	Functionalities
5	Interview C	Wat gebruik je waar voor (specificeren)	Functionalities
6	Interview C	Welke securityprotocollen gebruik je	Functionalities
7	Interview C	Niet enkel kijken naar webapplicatie (ERP/Legacy)	Functionalities
8	Interview C	De gehele datastream moet erin staan van sensor (input) tot Use case (output)	Completeness
9	Interview C	Structuur zien die consequent is	Consistency
10	Interview C	Voorkomen van ruis met detail level van aspecten	Consistency
11	Interview C	Paradigma moet begeleidend zijn voor de gebruiker om specifieke aspecten te visualiseren	Consistency
12	Interview C	Geholpen worden in gelaagdheid	Consistency
13	Interview C	Voor datastromen generieke termen gebruiker	Consistency
14	Interview C	Meerdere microcontrollers laten terugkomen in het framework	Consistency
15	Interview C	Publish/Subscribe en microcontroller eventueel samenvoegen	Consistency
16	Interview C	Gekozen onderdelen moeten makkelijk te schalen zijn.	Usability
17	Interview C	Begeleiding zodat het duidelijk is op welk niveau je opereert in het framework	Usability
10	Interview C	Welke data gaat nu waarnaartoe	Architectural structure
11	Interview C	Welke veiligheidsprotocollen worden gebruikt	Architectural structure
12	Interview C	Welke data is beschikbaar	Architectural structure
13	Interview C	Data integratie is die extern of intern	Architectural structure

XII. Requirements for developing the artifact (framework)

ID	Source	Implemented	Original Requirement	Paradigm Building block
1	Interview A	Yes	Geen specifieke apparaten meegeven in het framework	Devices
2	Interview A	Yes	Duidelijke positie van de devices in het framework aantonen	Devices
3	Interview A	Yes	Rol in combinatie met communicatie duidelijk meegeven	Devices
4	Interview A	Yes	Aangeven dat hier alle data vandaan komt	Devices
5	Interview A	Yes	Duidelijke plek in het framework	Communication Technologies
6	Interview A	Yes	Koppeling met andere aspecten zoals objecten en middleware	Communication Technologies
7	Interview A	Yes	Schaalbaar	Communication Technologies
8	Interview A	Yes	Dynamisch	Communication Technologies
9	Interview A	Yes	Services moeten voor de client zichtbaar zijn	Services
10	Interview A	Yes	Laagdrempelig	Services
11	Interview A	Yes	Meerdere manieren om informatie te ontvangen	Services
12	Interview A	Yes	Afgesloten van de middleware oplossing	Services
13	Interview A	Yes	Dynamisch	Middleware
14	Interview A	Yes	Data moet kunnen worden opgeslagen en verwerkt	Middleware
15	Interview A	Yes	Data omzetten naar informatie	Middleware
16	Interview A	Yes	Makkelijk uit te breiden	Middleware
17	Interview A	Yes	Breed support voor functionaliteiten	Middleware
18	Interview A	Yes	Schaalbaar	WSN
19	Interview A	Yes	Dynamisch	WSN
20	Interview A	Yes	Duidelijke connectie met Middleware en de sensoren	WSN
21	Interview A	Yes	Geen specifieke protocollen noemen	WSN

ID	Source	Implemented	Original Requirement	Paradigm Building block
1	Interview B	Yes	Things: schaalbaar zijn met zijn tijd mee kunnen geven zoals uitbreidbaar met nieuwe sensoren. (Door snelle ontwikkelingen)	Devices
2	Interview B	Yes	Componenten moeten onderling communiceren over beschreven protocollen maar ook dit moet aanpasbaar zijn in de toekomst.	Communication Technologies
3	Interview B	Yes	Rol van de componenten vastleggen	Communication Technologies
4	Interview B	Yes	Communicatielagen dienen ook als componenten/elementen die uitwisselbaar zijn en zodanig gedocumenteerd.	Communication Technologies
5	Interview B	Yes	Verwachtingen goed beschreven	Communication Technologies
6	Interview B	Yes	Vastgelegd wat de output van het framework is	Services
7	Interview B	Yes	Duidelijk vastgelegd wat de voordelen van het framework zijn	Services
8	Interview B	Yes	Snel kunnen zien wat het doet/levert	Services
9	Interview B	Yes	Communicatie laag e.g.	Services
10	Interview B	Yes	Geen expert niveau nodig om het te begrijpen	Services
11	Interview B	Yes	Aanpasbaar en bruikbaar als losse componenten kunnen.	Middleware
12	Interview B	Yes	Database staat bijvoorbeeld niet vast	Middleware
13	Interview B	Yes	Koppeling met bestaande infrastructuur organisatie	Middleware
14	Interview B	Yes	Breed support	Middleware
15	Interview B	Yes	Robuust in de kern maar wel aanpasbaar en uit te breiden	Middleware
16	Interview B	Yes	Schaalbaar	WSN

ID	Source	Original Requirement	Paradigm Building block
1	Interview C	Duidelijk maken dat er meerdere devices aanwezig zijn	Devices
2	Interview C	Datastromen van devices uitleggen	Devices
3	Interview C	Publish/Subscribe & microcontroller samenvoegen	Devices
4	Interview C	Schaalbaar	Communication Technologies
5	Interview C	Veilig	Communication Technologies
6	Interview C	Niet specifiek op chatbot maar meer op de communicatiemethode in het algemeen	Services
7	Interview C	Dashboard hoeft geen verplichting te zijn	Services
8	Interview C	Visualiseer moet ook automate kunnen zijn of in combinatie met integratie.	Services
9	Interview C	Eventueel legacy systemen koppelen	Services
10	Interview C	Eventueel toevoegen API	Middleware
11	Interview C	Data storage kan zowel intern als extern zijn	Middleware
12	Interview C	Nadenken over de rol van analyse automations & chatbot t.o.v. integratie	Middleware
13	Interview C	Schaalbaarheid	WSN

XIII. Interview I, questionnaire for gathering functional requirements for PoC

Description: This interview has been built to retrieve functional requirements for the PoC that has to be built. The owner of the product is SMRTR. This interview is guided on employees of SMRTR.

Generieke informatie en introductie: voor SMRTR wordt er een hydrocultuur (kweken op basis van water zonder aarde) automatisch kweeksystem gebouwd. Dit systeem heeft als uitgangspunt zoveel mogelijk automatisch te doen en de eindgebruiker tijdig te informeren over wijzigingen van de situatie van het systeem doormiddel van notificaties en een bijbehorend dashboard.

Vragen:

1. Wat zijn jouw verwachte functionaliteiten die passen bij de hierboven beschreven introductie.

De literatuur kenmerkt een Internet of Things oplossing standaard gekenmerkt als een oplossing die bestaat uit drie visies:

- *Internet*
 - *Communicatie met de gebruiker*
 - *Objecten*
 - *Sensoren*
 - *Schakelaar*
 - *Semantisch (overeenstemming)*
 - *Middleware (verbindende software laag)*
 - *Datagebruik*
 - *Technieken*
2. Met deze visie in het hoofd welke algemene functionaliteiten zou je nog toevoegen aan je vorige antwoord?
 - a. Internet
 - i. Dashboard
 - ii. Communicatie met gebruiker (chatbot)
 - b. Middleware
 - i. Flow-based programming (Node-Red)
 - ii. Home Assistant
 - iii. Data bijhouden
 - iv. Data-logging-equipment
 - c. Communication technologies
 - i. Wireless Sensor Networks
 1. MQTT
 - d. Things
 - i. Sensoren
 - ii. Schakelaars
 3. Welke data vind je belangrijk om te verzamelen?
 4. Welke informatie zou het systeem je moeten geven?
 5. Welke automatiseringen wil je zien?
 6. Hoe zou je geïnformeerd willen worden?

XII. Interview I, retrieving framework validation requirements

Description: This interview has been created to retrieve the requirements of the framework to later validate the created framework qualitatively with experts in the field of the Internet of Things. These questions are asked about what the expectations are of the business when presented with certain variables that are used in literature to validate artefacts and frameworks.

IoT oplossingen kunnen erg divers zijn en bestaan uit velen verschillende communicatietechnieken, technologieën en objecten. Een manier om hiermee meer harmonie te creëren en sneller oplossingen te kunnen verwezenlijken is door gebruik te maken van flow-based programmeren hiervoor bestaat nog geen framework. Het doel van dit onderzoek is een artefact te maken voor een architecturaal framework op basis van Flow-Based programmeren. In de literatuur worden architectuur frameworks voor IoT gevalideerd door middel van 5 variabelen en bestaan uit enkele functionele bouwblokken (Ray, 2014) en enabling technologies (Luigi Atzori, 2010). Dat zijn:

- Functionality: the (expected) functionalities that the artefact should deliver. Consists of a set of functions or capabilities which are associated with artefacts (Goodwin, 1987).
- Completeness: the expert knowledge when creating the artefact can maybe be missing needed principles or practices and can be incomplete because of missing rules (Motoi Suwa, 1982).
- Consistency: the used principles, practices and rules can be consistent or inconsistently used in the artefact (Motoi Suwa, 1982).
- Usability: a measure of the quality of a user's experience in interacting with information or services from the artefact (Liam O'Brien, 2007).
- Architectural structure: a measure to look if the architectural framework provides logic with its principles, practices and is divided into layers and views accordingly (Marley, 2003).

Ten opzichte van IoT functionele blokken (rol van de aspecten in het framework) (P.P., 2016):

- Devices
- Communication
- Services

Ten opzichte van de enabling technologies (Luigi Atzori, 2010):

- Middleware
- Publish/Subscribe (WSN)

De vragen die gesteld gaan worden nemen elk van deze middelen door en verwachten een passend antwoord waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen.

Vragen:

1. Welke req. zou u kunnen geven over de functionaliteiten van het framework?
2. Welke req. zou u kunnen geven over de compleetheid van het framework
3. Welke req. zou u kunnen geven over de consistentie van het framework?
4. Welke req. zou u kunnen geven over de bruikbaarheid van het framework?
5. Welke req. zou u kunnen geven over de architecturale structuur van het framework?
6. Welke req. zou u kunnen geven over de rol van devices in het framework?
7. Welke req. zou u kunnen geven over de communicatie technieken in het framework?
8. Welke req. zou u kunnen geven over de geleverde services in het framework?
9. Welke req. zou u kunnen geven over de rol die de middleware vervult?
10. Welke req. zou het Wireless Sensor Network aan moeten voldoen?

XIII. Interview III, validating the Framework and final changes

Description: This interview has been created to validate the created framework qualitatively with experts in the field of the Internet of Things. For these questions are asked about certain variables and the architectural framework has been studied before the interview by the respondent. In this interview changes to the artefact may still be made as a result of the answers

IoT oplossingen kunnen erg divers zijn en bestaan uit velen verschillende communicatietechnieken, technologieën en objecten. Een manier om hiermee meer harmonie te creëren en sneller oplossingen te kunnen verwezenlijken is door gebruik te maken van flow-based programmeren waarbij door black-box principe van input & output wordt gebruikt om zo low-code oplossingen te creëren. Het doel van dit onderzoek is een artefact te maken voor een architectuur framework op basis van Flow-Based programmeren. In de literatuur worden architectuur frameworks voor IoT & artefacten in design science gevalideerd door middel van verschillende variabelen (Ray, 2014) en enabling technologies (Luigi Atzori, 2010). Dat zijn:

- Functionality: the (expected) functionalities that the artefact should deliver. Consists of a set of functions or capabilities which are associated with artefacts (Goodwin, 1987).
- Completeness: the expert knowledge when creating the artefact can maybe be missing needed principles or practices and can be incomplete because of missing rules (Motoi Suwa, 1982).
- Consistency: the used principles, practices and rules can be consistent or inconsistently used in the artefact (Motoi Suwa, 1982).
- Usability: a measure of the quality of a user's experience in interacting with information or services from the artefact (Liam O'Brien, 2007).
- Architectural structure: a measure to look if the architectural framework provides logic with its principles, practices and is divided into layers and views accordingly (Marley, 2003).

Ten opzichte van IoT functionele blokken (P.P., 2016):

- Devices
- Communication
- Services

Ten opzichte van de enabling technologies (Luigi Atzori, 2010):

- Middleware
- Wireless Sensor Network

De vragen die gesteld gaan worden nemen elk van deze middelen door en verwachten een passend antwoord waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen.

Vragen:

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
2. Wat zou u kunnen zeggen over de compleetheid van het framework
3. Wat zou u kunnen zeggen over de consistentie van het framework?
4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?

6. Wat zou u kunnen zeggen over de rol van devices in het framework?
7. Wat zou u kunnen zeggen over de rol van de communicatietechnieken in het framework?
8. Wat vindt u van de rol van de geleverde services?
9. Wat vindt u van de rol van de webapplicatie?

10. Wat zou u kunnen zeggen van de rol die de middleware vervult?
11. Overall feedback over het framework

12. Wat zou u kunnen zeggen over de beschreven principles?

XII. Interview III, validating the final iteration of the Framework

Description: This interview has been created to validate the last iteration of the created framework qualitatively with experts in the field of the Internet of Things like an extra justification for the answers given in the previous interview. For this interview questions are asked about certain variables and the architectural framework has been studied before the interview by the respondent. In this interview, no changes to the artefact may be made after the answers have been given.

IoT oplossingen kunnen erg divers zijn en bestaan uit velen verschillende communicatietechnieken, technologieën en objecten. Om hierin meer harmonie te creëren is een IoT framework voor flow-based oplossingen gecreëerd. Het doel van dit framework is consultants met een technische achtergrond te helpen bij het creëren van hun eerste architecturale omgeving of blauwdruk wanneer voor het eerst een flow-based IoT oplossing in de domein specifieke omgeving moet worden gebouwd.

In de literatuur worden architectuur frameworks voor IoT & artefacten in design science gevalideerd door middel van verschillende variabelen (Ray, 2014) en enabling technologies (Luigi Atzori, 2010). Dat zijn:

- Functionality: the (expected) functionalities that the artefact should deliver. Consists of a set of functions or capabilities which are associated with artefacts (Goodwin, 1987).
- Completeness: the expert knowledge when creating the artefact can maybe be missing needed principles or practices and can be incomplete because of missing rules (Motoi Suwa, 1982).
- Consistency: the used principles, practices and rules can be consistent or inconsistently used in the artefact (Motoi Suwa, 1982).
- Usability: a measure of the quality of a user's experience in interacting with information or services from the artefact (Liam O'Brien, 2007).
- Architectural structure: a measure to look if the architectural framework provides logic with its principles, practices and is divided into layers and views accordingly (Marley, 2003).

De vragen die gesteld gaan worden nemen elk van deze middelen door en verwachten een passend antwoord waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen. Neem hierbij het doel van het framework in acht (bijvoorbeeld, heeft het framework voldoende functionaliteiten om een eerste blauwdruk te kunnen creëren en is het hierin compleet genoeg).

Vragen:

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
2. Wat zou u kunnen zeggen over de compleetheid van het framework
3. Wat zou u kunnen zeggen over de consistentie van het framework?
4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?

XV. Interview functional requirements respondent B

Geïnterviewde: Arthim de Vogel

Interviewer: Karsten Veldhuizen

Datum: 12-11-2020

Location: Vanuit huis

I: Welkom bij het eerste interview over het verzamelen van de functionele requirements. Ik zal eerst wat generieke informatie vertellen zoals in de introductie maar, waarschijnlijk weetje al dat we een kweeksysteem gaan bouwen op hydrocultuur (kweken op basis van water in plaats van aarde) waarbij we zoveel mogelijk automatisch willen doen en de gebruiker altijd tijdig wordt geïnformeerd over de kweek doormiddel van notificaties of een dashboard. Wat zijn jouwe verwachte functionaliteiten die passen bij de hierboven beschreven introductie? Dus als jij dit zo hoort, wat zou jij verwachten van zo een soort systeem?

R: Ik heb het meerdere keren getracht om in het klein hier te kunnen kweken bij mij in huis. Daarbij als ik ook het systeem een beetje probeer te bedenken hoe het uiteindelijk zal zijn, is dat je uiteindelijk ergens in de hoek een kastje hebt staan in je kamer of dat je dan **digitaal de hele voortgang kan zien**, dus dat je misschien ook **notificaties krijgt** van je sla is nu klaar of groot genoeg om te eten. En wat me ook heel erg leuk lijkt is dat je tussendoor **updates zou kunnen vragen dat je dus inderdaad via een communicatie laag**, dus stel ik ben ergens anders en ik heb trek in sla dat ik zou kunnen vragen hoe staat het ervoor. En dat je dan informatie krijgt ervoor misschien in combinatie met een afbeelding van camera beelden ofzo., Dus het is eigenlijk kijken dat je **digitaal kan kijken hoe het met je sla gaat** zonder dat je er bij hoeft te zijn. Dus dat denk ik.

I: Zoals we net al even hebben doorgenomen hebben we bepaalde dingen van IoT benoemd zoals internet, objecten en je overeenstemming tussen al die middelen die je daarvoor gebruikt. Dan ga ik nu nog een keer dezelfde vraag stellen maar dan terwijl we het paradigma doornemen. Dus op het gebied van internet dus bijvoorbeeld dashboarding over requirements en functionalisten zou je dan graag willen zien?

R: In wezen dat dashboard dan zou ik gewoon ten alle tijden **voortgang moeten zien**. Voor de bioloog is het dan misschien ook leuk om alle waardes erbij te hebben en voor de niet bioloog is het vooral leuk om de voortgang te zien dus in je metingen dan van inderdaad hoe groot is het hoe goed gaat en hoe groot was het voorheen en misschien een boogje met wat modelletjes en **grafiekjes** en aan de andere kant is misschien ook leuk om aan **de zijkant de watertemperatuur** te zien en dan ook **conclusies te kunnen trekken op basis van wat er gemeten is**.

I: Dus eigenlijk een soort logboek aan de kant met ik heb op die dag geplant en de voortgang en dan rechts eventueel de wat meer geeky sensor waardes.

R: Ja maar goed dat is heel persoonlijk de echte leek of digibeet willen dat allemaal niet. Wat misschien ook leuk is misschien de **verwachting** ik weet niet of dat te doen is want dan heb je eerst een hele berg data nodig. Maar dat je iets plant en dan dat je weet over een week of zoveel **is het klaar** of misschien dat je dat inhält.

I: Ja inderdaad maar dan zou je wel wat data eerst moeten verzamelen maar dat zou wel mogelijk moeten zijn? En verder als we denken aan communicatie met de gebruiker dus dan zou je bijv. moeten denken aan notificaties op de telefoon chatbot, wat voor functionaliteiten zou je dan graag willen zien?

R: Nouwjah bijvoorbeeld van die notificaties dan er zijn natuurlijk een hoop dingen die de sensoren op zichzelf kunnen regelen maar stel nou dat er inderdaad iets bij moet of het water is bijna op dat je dan inderdaad een **melding krijgt** dat hey om goed door te groeien heb ik weer even wat nodig want ik zit nu op 70% van de bak dus dat je daardoor **getriggerd wordt om dingen te doen**. Dus heel stom want hier in huis hebben we best wat planten staan maar ze gaan hier nog wel eens dood door te veel of te weinig water geven dus als we daardoor geholpen worden door een seintje te krijgen van hey ga je planten eens water geven dan gaan ze langer leven. En zoals net als even de mogelijkheid hebben om ook vragen te stellen, stel je zit op kantoor en zit lekker te pochen met collega's dat je de vraag kan stellen van hey zo ziet mijn sla er nu uit. Maar je te vragend en bijvoorbeeld als je een stel je hebt nog een enthousiaste collega of iemand of vriend van je die dit ook aan het doen is en je bent daar en je bent het even aan het bespreken en hij vraagt, joh waarom is dat van jou 20 graden dat je dan via je dashboard actief informatie kan ophalen.

I: Oké goed om te horen dus eigenlijk zeg je dat in dit geval een chatbot interessanter is dan puur notificaties omdat je soms informatie wil opvragen als je met een vraagstuk zit.

R: Ja maar, wel als een **combinatie van notificaties en chatbot** want soms een indicatie ook een reminder en soms is het ook als je iet heel lang wil laten groeien dat je om te de paar dagen even een update krijgt.

I: Gaan we naar het volgende stukje toe dat is de middleware dat is waar alle logica zit dus ook je node-red. Dus ja stel we beginnen daarbij flow-based programmeren (Black-box) principe wat voor functionaliteiten zou je daar ongeveer voor willen zien?

R: Uhm, ja in wezen knoopt het alles aan elkaar dus alles wat ik net ook zei dat bij notificaties en communicatie met het dashboard dat gaat via in dit geval denk ik node red van a naar b en dat wordt op een manier getoond. Dus ja in wezen als ik in zover kijk dat zal het waarschijnlijk bij de technische en enthousiaste groep horen dat ik het ook zelf zou willen zien dus at het opensource is dus zelf **extra dingen zou kunnen toevoegen** of inhaken maar dat is persoonlijk denk ik.

I: En als flow-based programma, in dit geval heb ik node red genoemd maar lijkt jou dat ook de sterkste keuze daarvoor of heb jij ook nog een andere tool op het oog?

R: Zelf heb ik wel wat ervaring met flow-based programmeren node-red is denk ik wel de enige die op deze manier dit kan uitvoeren en die open-source is. Heb wel eens andere opties gezien maar dan zit je meteen aan een abonnement van hier tot om de hoek. Dus ik ben heel positief over node-red.

I: Ja en het is ook nog eens ontwikkelt door IBM.

R: Ow dat wist ik nog niet eens een.

I: Ja het komt uit een onderzoekslaboratorium van ze. Voor middleware heb ik ook kleine stukje home assistent benoemd als dashboard ik weet niet of je daar nog bepaalde dingen aan zou willen zien, niet per se dashboard maar ja wat zou je connectiviteitslaag nog meer moeten kunnen behalve je chatbot, sensor of automatiseringen. Dat je denkt van nou aanvullend.

R: Ik ben zelf geen expert van home assistent ik heb het zelf ook allemaal nog niet draaien, ik heb wel eens wat gezien van andere mensen en hiermee en dan doe ik even de aanname dat dit dus ook een manier kan zijn dat je hiermee een dashboard zou kunnen krijgen en het ook kunnen **afstemmen met de rest van je slimme huis** Zie ik hier alleen maar voordeel in omdat je het koppelbaar kan maken met andere dingen dat je dus bijvoorbeeld in je huis ook de airco erop kan aansluiten zodat in die of in de ruimte of de verwarming omdat er een temperatuur verschil is.

I: Dus eigenlijk zeg je dan het moet de mogelijk hebben om veel integraties te ondersteunen.

R: Ja **koppelbaar** met ander dingen.

I: Ja dan komen we ook terug op het docker verhaal dat je wat dat betreft schaalbaar kunt zijn met plugins. Nou dan gaan we het naar het data bijhouden. En ja of hoe zie je daar nog speciale dingen die data moet natuurlijk kunnen worden bijgehouden en opgeslagen heb je daar nog wat voorwaarden aan of zeg je oké op dit moment wordt het wel oké opgeslagen.

R: Noujah doemdenken zou ik liever niet willen dat de Chinese overheid meekijken ook al is het maar met een krop sla die ik aan het groeien ben, het moet **gewoon goed beveilig** zijn. Het moet afgesloten worden van ongewenste personen en verder denk ik dat de data op zich van belang is om misschien een **stukje projectie** te doen van stel nou je plant nu verwacht ik dat ie over een week of 2 klaar is of dat soort dingen. Dat is denk ik de kracht van de data die je verzamelt als je er mee bezig bent. En natuurlijk de andere data de aanvragen en de reacties dat dat ook gewoon netjes opgeslagen wordt al kan ik me indenken dat bij telegram je natuurlijk gewoon naar boven kan scrollen in je chatbot maar nee het is vooral denk ik van belang dat het netjes gaat en netjes opgeslagen wordt.

I: Dus eigenlijk lokaal opgeslagen zou dan de voorkeur hebben in plaats van data opslaan in een niet beveiligd.

R: In ieder geval beveiligd opgeslagen, ik weet dat er genoeg beveiligde manieren en verbindingen zijn om echt niet per se lokaal op te hoeven slaan het heeft misschien ook wel een beetje te maken met hoe uitgebreid je lokale user dan wordt als je daar opslag in nodig heb moet je daar ook weer in dingen doen. Als je je data direct veilig kan sturen naar je Cloud dan lijkt me dat ook nog wel goed denk ik.

I: Oké top dan hebben wij deze twee nu gehad, dan gaan we door naar de volgende de communicatietechnologie hoe zijn de sensoren aan elkaar gekoppeld en hoe werkt als in zijn geheel samen. Met wireless sensor network dat zal ik ook even uitleggen met publish. Subscribe heb je daar nog bepaalde voorwaarden of zeg je hey juist omdat je werkt met die onderwerpen is dat misschien wel het sterkste.

R: Uhm, nee en daar ga ik dan even volledig in op de introductie die je gegeven hebt, hier zelf wist ik nog niet heel veel van deze technieken en wireless sensor network. Dus nee wat je hebt laten zien en wat je uitlegde klinkt en dat het al staat en bewezen is lijkt me het alleen maar goed om op die manier verder te gaan.

I: Want dat vervult op dit moment wel de andere requirements dat het schaalbaar moet zijn. Want dat is in de hoofdlijnen wel belangrijk

R: Ja want door hier gebruik van de maken is het beter schaalbaar?

I: ja 100%. En voor de rest kwa sensoren en schakelaars wat wil je dan allemaal zien wat moet die allemaal kunnen meten wat moet die allemaal kunnen schakelen

R: Ja uhm dat is natuurlijk ja lastig te zeggen wat ik ben zelf geen expert of bioloog. Dus dat is heel erg van belang van wat je erin gaat groeien want misschien is het voor de een belangrijker dat de ph waarde stabieler blijft dan voor de andere of dat water **temperatuur** of **zuurstofgehalte** en **licht waardes** van belang is. Waardoor groeit een plant waardoor groeit een stuk sla in dit geval die waardes moet je wel kunnen meten en daar ook wat mee kunnen. Denk dat dat een beetje over die schakelaars gaat.

I: Ja hey, ik wil gewoon kunnen zeggen dat he zo laat en uitgaat of als er iets mis gaat dan ie uit zichzelf uit gaat of meet hoe het water niveau is of het temperatuur of grote of dat soort functionaliteiten.

R: ik denk dat dat natuurlijk een beetje samenhangt van wat hij meet en dingen die hij moet kunnen doen en ik bedoel als ik nou echt er helemaal niets van snap dan **zou hij zichzelf moeten kunnen fixen en dat hij meldingen geeft wat hij nodig heeft met precisie aantallen**. Hoeveel van wat je doet hoe krijg je ph waarde weer terug moet je bepaald aantal ml van iets erbij doen dat het wat zuurder wordt of wat zachter wordt. Dus dat soort dingen het moet eigenlijk dummy proof zijn voor de eindgebruiker of misschien kun je niveaus ervan selecteren en misschien ben je echt een dummy en moet alleen reageren of lekker zelf doen op basis van wat ik meet.

I: Nog wat generieke vragen is er nog data dat je echt belangrijk vindt dat dat wordt verzameld. Dan gaan we vragen naar informatie dan moeten we daar nog even onderscheid in maken, eerst losse data.

R: In principe data hangt samen met welke sensoren heb je en wat krijg je daaruit en op basis van de data die je dan verzamelt denk ik ja blijf toch steeds een beetje terugkomen bij een soort **penanalyse** de vorige keer is het zo hard gegroeid en stel je hebt 10x wat gedaan dan kun je redelijk wel wat dingen conclusies eruit trekken en dus is het dan bij 4 6 8 keer is hij heel goed gegroeid en twee keer wat minder dan kun je dus kijken wat hebben die 2x gedaan wat anders dan die een keer. En staan er wat algemene dingen in dan kun je dus je hele systeem daarop aanpassen. Dat je ph niet bijvoorbeeld verandert.

I: Gaan we even verder naar de informatie verder, welke informatie moet het systeem verzamelen dat dit gebeurt of het is nu te warm.

R: Net hadden we dan ff die ph waarden maar dan ook temperatuur en zuurstofgehalte en dat daar dan ook de **conclusie aan de hand van de data** getrokken kan worden dus over die 8 keer dat het goed ging dat je de verschillende zuurstofgehaltes gemeten.

I: Er moet dus ook worden vertaald naar direct begrijpbare informatie voor de gebruiker dat het systeem zegt het is nu te warm dus x of y.

R: Ja dus inderdaad die inschatting van of het te warm of te koud is op basis van lokaal gewonnen data en nou ja en als je hem verder trekt en iedereen hangt aan een gedeelde Cloud omgeving en alles is geanonimiseerd gedeeld met elkaar dan bij alle 1K kroppen sla is er dit en dit gebeurt en dit de beste aanpak geweest. Maar goed dan heb je ook nog andere technische connecties nodig.

I: Dan gaan we naar welke automatiseringen zou je willen zien, wat moet het systeem automatisch doen. Zoals lamp moet uit en aan kunnen gaan of luchttoevoeging aan het water aan en uit. Automatisch dimmen van de lamp.

R: Alles wat je manueel zou toe kunnen voegen zoals lamp of extra water erbij als dat ook automatisch gedaan kan worden dan is dus de melding van de persoon van jo hey het niveau is te laag dus ik heb de kraan even aangezet dat hij het zelf dus ook reguleert. Dan denk ik aan compartimentjes. Bijv. 10ml van het ph neutralisatie tool zit en dan hij het zelf dan kan toevoegen met 1ml. Dus echt nog meer **hands off** is. Dat lijkt mij wel heel tof.

I: Vraag 6 een generieke op welke manier je geïnformeerd wil worden zoals we net al even over hadden zowel notificaties als vragen.

R: Op zich is je **chatbot** dan je actieve uitvraag en je **dashboard** is dan je passieve vraag. Dus als je zondag avond op de bank zit dat als je even wil kijken hoe je krop sla gaat dat je dan naar je dashboard gaat en anders voor de chatbot gaat. Daarmee heb je het in principe redelijk afgevangen. En als er echt iets aan de hand is dat je dan ook je notificaties krijgt.

I: Oké cool, goed dat waren de vragen voor de functionele requirements denk dat daarmee het mooie antwoorden waren, dan ga ik nu de opname weer even uitzetten.

Interview – Arthim de Vogel op 04-12-2020

Samenvattend antwoorden op de vragen:

1. Welke req. zou u kunnen geven over de functionaliteiten van het framework?
 - a. *Algemeen toepasbaar verschillende (wel flow-based) topics.*
2. Welke req. zou u kunnen geven over de compleetheid van het framework
 - a. *Componenten moeten vervangbaar/uitgebreid kunnen worden*
 - b. *Is in zijn geheel op zichzelfstaand*
3. Welke req. zou u kunnen geven over de consistentie van het framework?
 - a. *Consistentie in de verwerking, inhoudelijk en moet dus altijd doen wat er verwacht wordt door de gebruiker.*
 - b. *Consistentie in de toepassing het framework moet duidelijk zijn beschreven en opgesteld.*
4. Welke req. zou u kunnen geven over de bruikbaarheid van het framework?
 - a. *Algemeen toepasbaar en wendbaar in de toepassing*
 - b. *Uitbreidbaar door componenten te vervangen/ toe te voegen*
 - c. *Eenduidig overzicht*
 - d. *Wendbaar*
5. Welke req. zou u kunnen geven over de architecturale structuur van het framework?
 - a. *Duidelijk omschreven wat de kern is en wat die moet bieden*
 - b. *Snel doorhebben waar je het op kan toepassen*
 - c. *Niet heel erg groot zijn*
 - d. *Componenten moeten uitwisselbaar (per element input-proces-output beschrijven) dus stel je gebruikt geen node-red dan zouden de input en outputs nog steeds moeten gelden.*
6. Welke req. zou u kunnen geven over de rol van devices in het framework?
 - a. *Things: schaalbaar zijn met zijn tijd mee kunnen geven zoals uitbreidbaar met nieuwe sensoren. (Door snelle ontwikkelingen)*
7. Welke req. zou u kunnen geven over de communicatie technieken in het framework?
 - a. *Componenten moeten onderling communiceren over beschreven protocollen maar ook dit moet aanpasbaar zijn in de toekomst.*
 - b. *Rol van de componenten vastleggen*
 - c. *Communicatielagen dienen ook als componenten/elementen die uitwisselbaar zijn en zodanig gedocumenteerd.*
 - d. *Verwachtingen goed beschreven*
8. Welke req. zou u kunnen geven over de geleverde services in het framework?
 - a. *Vastgelegd wat de output van het framework is*
 - b. *Duidelijk vastgelegd wat de voordelen van het framework zijn.*
 - c. *Snel kunnen zien wat het doet/levert*
 - d. *Communicatie laag e.g.)*
 - e. *Geen expert niveau nodig om het te begrijpen*

9. Welke req. zou u kunnen geven over de rol die de middleware vervult?
 - a. *Aanpasbaar en bruikbaar als losse componenten kunnen.*
 - b. *Database staat bijvoorbeeld niet vast*
 - c. *Koppeling met bestaande infrastructuur organisatie*
 - d. *Breed support*
 - e. *Robuust in de kern maar wel aanpasbaar/uitbreidbaar*

10. Welke req. zou het Wireless Sensor Network aan moeten voldoen?
 - a. Schaalbaar

XV. Interview functional requirements Interview D

Geïnterviewde: Tim van Uden

Interviewer: Karsten Veldhuizen

Datum: 16-11-2020

Location: Vanuit huis

I: Introductie weer omdat ik dus een kweeksysteem ga bouwen op hydrocultuur (water) moet zoveel mogelijk geautomatiseerd worden met sensoren en de gebruiker moet natuurlijk tijdig worden geïnformeerd over wat hij moet doen en het moet ook zo makkelijk mogelijk zijn. Ga ik eerst de vraag stellen heel globaal als je zo systeem hebt welke functionaliteiten zou je dan verwachten zonder er nog dieper over na te denken.

R: Ik zou in ieder geval verwachten [dat er monitoring plaats vindt](#), ik denk dat dat het alle belangrijkste is. Dat je een [inzicht](#) hebt in wat er allemaal nu plaatsvindt uhm dat er [historische data](#) opbouwt van eh eh de afgelopen tijd. Waardoor je en dat je die dus kunt koppelen aan dus eh wat er nu dus eigenlijk gebeurt waardoor je naar de toekomst toe daarop kan inspelen dus een verbetering kan doorvoeren of dingen weglaten die niet goed gaan.

I: Dus we kijken naar sensoren, die vertellen jou wat en misschien vertaalt het systeem wel iets dat je direct input krijgt wat dat dan betekent.

R: Ja vergelijkbaar dat je een plant een [sensor](#) erin duwt dat ie aangeeft dat hij te droog is of te nat of hij staat te veel in de zon.

I: Oké, klinkt top nou ja dan ga ik even kortleggen waar IoT voor staat, waarschijnlijk weet je het al maar in de literatuur wordt het ongeveer gekenmerkt uit drie visies zeg maar dus je hebt de visie van internet dus dat je alles met elkaar wilt verbinden, visie van objecten dus dat je alle sensoren en schakelaars hebt en natuurlijk het semantische dus dat alles samen moet komen en dat er gelijk spraak moet zijn. En dan heb je middleware je software die alles bij elkaar brengt en hoe je je data opsluit en enabling technologies dus een beetje de technieken die je gaat gebruiken. Dus stel deze visies welke functionalisten zou je dan nog verwachten. Dus dan kijken we naar de visie van internet of dashboarding of hoe er gecommuniceerd wordt. Hoe zou je graag dat willen zien?

I: Nouwjah ik zou ook verachten dat je dat er bijvoorbeeld informatie is over online over wat normale groeipatronen zijn van bepaalde gewassen die je dan zou groeien op deze manier en dat je daar in ieder geval [vergelijking](#) mee kan zien wat is nou het voordeel van dat ik dit gebruik in plaats van de traditionele manier laat maar zeggen. Van waar kun je wel is dat er in space station nu ook allemaal groente verbouwen zijn. En ja [communicatie met gebruiker](#) ik denk dat hij moet signalen geven dus dat hij aanhaakt je moet [nu ingrijpen of je hoeft niet in te grijpen](#). En ook misschien kunnen aangeven van uhm, wat de verschillen zijn tussen de verschillende groei stations die je hebt.

I: Dus dat je ook een baseline hebt met alle bestaande data en daar een vergelijking mee kan zien en kwa communicatie zouden we kunnen denken aan een chatbot die je een vraag kan stellen of notificatie kan geven waarmee je kan ingrijpen.

R: Ja dat je ook kan roepen van eh hey google hey weet ik veel *plant give me an update on my plants* en dat je dan krijgt te horen van eh, deze gaan goed, deze staan zoveel aan zou je niet eens een raam openzetten want het stinkt hier. Dat soort dingen want ik denk dat het heel interessant is want mensen zijn gewoon lui, mensen willen [volledig voorzien worden](#). Ik kan zelf geen plant laten leven in huis dus dat ja, dat een denk ik en ik denk ook dat je moet een geven moment ook gaan nadenken registreren van wat de effecten zijn wat kost het je aan energie, wat kost het je eh alles daaromheen

je setup en dat soort dingen. Dat je kan gaan **monitoren** hoe lang gaan je onderdelen mee. Het is leuk dat je kan zien wat je gewassen doen maar je moet ook monitoren wat je apparatuur doet. Ik denk met name dat de data heel interessant is want daarmee ga je ook bepalen of het werkt of niet. Dat is uiteindelijk je sluit punt.

I: Dus je wilt zoveel mogelijk monitoren zoveel mogelijk sensoren plaatsen en daarop moet worden geacteerd door het systeem zelf en als je zelf met een vraag zit dan zou je het ook aan het systeem moeten stellen en hij zou je tijdig de invloed moet geven.

R: Ik denk ook nog een ding en dat zit dan meer op het traditionele hoe maak je het zichtbaar? Dus als jij wat je nu hebt staan kun je eigenlijk niet inkijken toch?

I: Nee dat klopt. Dan moet je de deksel optillen

R: Ik zou verachten dus dat je uiteindelijk misschien wel wat elementen met doorzicht of een raam in willen maken waardoor je ook kan zien wat er gebeurt. Ik weet niet of daar dat moet natuurlijk ook kunnen dat dan het UV-effect de boel verstiert. Allemaal dingen waar ik over na denk als je kijkt naar de toekomst hoe je het niet alleen in een app visueel kan maken men vindt het ook fijn om steekproeven te doen door met de ogen te kijken. Of bijvoorbeeld een camera.

I: Inderdaad en dan kun je ook werken met alle libraries die dan ook de groei kunnen monitoren, daar heb ik al naar gekken, dat zou mooi zijn voor de toekomst. Om even terug te komen op de middleware, je weet dat we met node-red bezig gaan dus flow based, dus heb je daar nog requirements of iets voor of zeg je van nou ik ken node-red of ik vind het tof want?

R: Ik gebruik node-red natuurlijk voor de universiteit van Tilburg in combinatie met dialoog flow en node red. Eh die chatbot die we erop gezet hebben en het werkt gewoon goed, prima, inventief. Als er iets van een infrastructuur opzet dan moet er altijd een goudpoets zijn die ergens bij houdt wat je nu doet. Die chatbots die we bouwen het moet helder zijn wat ie doet dat hele loggen van de data en dat is heel erg belangrijk.

I: Ja want node-red is eigenlijk in staat om alles met elkaar te koppelen dus in principe is het je middleware oplossing. En ook voor communicatietechnologieën i kweet niet of je daar iets over hebt te zeggen maar ja je hebt natuurlijk je wireless sensor network dat zijn je sensoren die dan of draadloos met elkaar communiceren naar je (Wi-Fi Zigbee). Wat zou je daarvan willen zien of zeg je nou.

R: Ja dat zou ik niet nee daar heb ik te weinig expertise van denk ik. Ik zou niet weten wat je hier dan per se met wireless sensors zou willen.

I: Je data moet natuurlijk van het systeem naar de gebruiker op zijn telefoon moeten komen.

R: Ja je moet zorgen dat het **secure** is het maakt niet uit wat voor data is maar op het moment dat je data niet secure is kunnen andere mensen daar heel veel dingen mee doen. Dus ik weet niet wat de protocollen zijn die daarop zitten maar die protocollen moeten wel voldoen aan bepaalde certificering.

I: Ja tuurlijk en schaalbaarheid natuurlijk voor als je lekker groot wil gaan.

R: Ja want het is een beetje als je kijkt naar die eh, die vriend van mij heeft gedaan met die brandstof treinen van shell met die sensoren met die beacons war hij lang ging want dat werd op een gegeven moment niet meer **schaalbaar** omdat er zoveel data kwam dat er zoveel acties moeten worden uitgevoerd om te achterhalen waar die nou langs kwam. Die dat signaal al hadden uitgestuurd dat het bijna gewoon onmogelijk werd daar moet je wel rekening mee houden.

I: Dan had ik nog een laatste objecten wat voor dingen zou je dan gemeten willen zien worden? Dus denk bijvoorbeeld temperatuur of luchtvochtigheid of wat zou jij interessant vinden dat gemeten wordt?

R: Uhm, even denken ja inderdaad **temperatuur luchtvochtigheid**, uhm, even kijken. Licht het **licht** zou ik willen meten van want dat is tuurlijk in deze situaties is er sprake van je moet gewoon een bepaald soort licht hebben dat werkt voor je planten je moet bepaalde lampen hebben. Dat is belangrijk denk ik. Ik denk ook **uitstoot**, kan je dat meten?

I: Uitstoot als in? Dat zou je kunnen meten in theorie als je een gesloten systeem hebt we hebben nu een systeem die niet helemaal dichtzit. Maar als die dichtzit zou je het in theorie kunnen meten.

R: **Zuurstofgehalte** natuurlijk ook van het water ik weet niet ik heb de opstelling van binnen niet gezien hoor maar als men er wordt gegroeid in water ja dan moet je ook het zuurstofgehalte van het water misschien wel kunnen meten. En andere stoffen die in het water zitten of in gewoon de tijd verstrijkt of daar dan bepaalde zaken door deze manier van werken toenemen in het water dus eh. Of bijvoorbeeld kalk niveau toe neemt of magnesium toeneemt. Mineralen en de **PH** waarde testen. Verder zou ik het niet echt weten nee.

I: Dan hebben we precies alles wat nu ongeveer voor gekozen hebben dus dat is goed. Zoals PH de zuurgraad. Wanneer welke voedingstoffen worden opgenomen en EC voor als er genoeg mineralen in zitten. Dan heb ik nog wat kleine korte algemene dingetjes. Welke data je zou willen verzamelen hebben we net al eigenlijk beantwoord voor sensoren hebben we al beantwoord. Welke informatie zou je willen verzamelen hebben we ook al beantwoord met een chatbot. Dus notificaties en ook vragen kunnen stellen en doe dit anders gaat het niet goed. Zijn er verder nog automatiseringen die je zou willen zien, dus wat zou hij automatisch willen doen.

R: Ik zou in ieder geval verwachten dat hij automatisch kan bijsturen, dus als hij denkt dat er te weinig licht is dat hij dan automatisch dat regelt dat dat erbij komt of eraf komt. Of dat nou het openzetten is van een klep of dat een aanzetten is van een speciale lamp. Hetzelfde met het water dat als er niet genoeg mineralen inzitten dat hij de mogelijkheid heeft om de mineralen aan te vullen of water te verversen met een soort. Ik voer het water af en voer nieuw water in.

I: Of een pompje die aan een flesvoeding zit inderdaad, dus los van informatief ook gewoon bepaalde acties dus voeding geven of licht de dimmer misschien sterk zetten. Dan zijn we alweer bij de laatste die heb je eigenlijk ook al beantwoord, hoe zou je eigenlijk geïnformeerd willen worden dan gaat het over de chatbot met notificaties met die vraag en over dashboarding. Dat je wel een dashboard zou hebben waar je alles met een overzichtje zou kunnen zien.

R: Wat ook cool is dat je mensen vinden het altijd leuk dat jezelf om **achievements** te krijgen om een beetje een competitie-element te krijgen als jij zegt wij hadden op kantoor bij Deloitte elke week kregen we fruitmanden en die fruitmanden het geld daarvan werd dan gebruikt om bomen te planten en elke keer kregen we een mailtje met u heeft nu zoveel bomen geplant en dat werd dan een wedstrijdje. Je hebt zoveel bijvoorbeeld zoveel gegroeid zonder bepaalde uitstoot of je hebt zoveel al voor elkaar gekregen of dat soort dingen als je daar een soort in een element aan kan koppelen is dat voor mensen natuurlijk interessant dat vinden ze leuk.

I: Nou dat waren eigenlijk dan al de vragen, kort en bondig maar wel compleet en komt overeen met wat we voor ogen hebben.

XVI. Interview framework requirements respondent A

Gesprek:

I: Nou dan gaan we zo een interview houden over de requirements waar een framework aan moet doen om deze later te kunnen valideren aan IoT experts. Dus ik ga vragen wat je verwacht wat IoT framework een beetje moet kunnen bieden. Welke functionaliteiten er allemaal zien om inderdaad later te kunnen valideren. We gaan hierbij kijken naar een aantal variabelen zoals consistentie, bruikbaarheid en architectuur structuur. Zoals je weet heb ik eerder besproken dat er drie grote functionele blokken zijn in IoT. Houdt die even in gedachten dat zijn devices, communicatie en services. Daarbij hebben we twee major enabling technologies waar je niet omheen kan als je IoT gebruikt dat is je Middleware oplossing en je Wireless Sensor Network. Dan gaan we nu door na de vragen. Dus nu vraag 1 welke requirements zou je kunnen geven over de functionalisten die het framework moet bieden:

R: allereest om het een beetje generaal aan te geven moet het duidelijk en overzichtelijk te zijn. Hiermee bedoel ik dat mensen in een oogopslag moeten kunnen zien wat voor betrekkingen een framework moet kunnen hebben in een IoT oplossing. Dan zou je kunnen wat valt er dan binnen communicatie devices, wat voor protocollen vallen er onder WSN en dat dat eigenlijk wordt aangekaart. Los van het feit dat je natuurlijk nauw moet beschrijven wat er natuurlijk gebeurt in zo een framework. Dus de mensen weten waar ze als ze het hebben gelezen weten waar ze het over hebben. In dit geval zouden leken moeten begrijpen wat het framework kan doen voor hen en wat het kan betekenen. In dit geval misschien eigenlijk 1 op 1 kunnen overnemen. Los van dat feit is het denk ik punten vanuit zo framework sequentieel over elkaar moeten kunnen overlopen. Dat je begint bij punt A en dan punt B zodat er een natuurlijke flow ontstaat over het framework. Dus misschien heb je meerdere start punten zodat je het op meerdere manieren kunt opvatten. Maar er moet eigenlijk geen natuurlijk flow zijn in het framework. Dat zou ik als belangrijk ervaren.

I: Duidelijk, dan gaan we door naar vraag 2 welke requirements zou je kunnen geven over de compleetheid van het framework dus wat verwacht je dat er sowieso in zit om het compleet te maken.

R: Oké daar zou je dan dus die aspecten van IoT kunnen opnemen. Dus de basis van de sensoren tot aan data verwerking tot aan de analyse ervan tot het uiteindelijk kunnen analyseren en presenteren van die data naar bijvoorbeeld een dashboard of chatbot. Zodat mensen meteen kunnen zien wat die IoT oplossing kan bieden. Natuurlijk zitten er meerdere aspecten in ook met middleware etc. Het moet ook kunnen beschrijven wat elk punt van het IoT proces inhoudt. Los van het feit dat je er overkoepelende takken zijn moet er wel in vallen dat de punten en bouwblokken die erin vallen ook specifiek worden aangekaart in het framework. Dus dan zou ik als ik zou moeten kijken voor de sensoren om maar wat te zeggen dat je daadwerkelijk kan beschrijven welke sensoren betrekking hebben op het framework niet te specifiek omdat het een framework is maar wel in je thesis zou moeten aangeven welke keuze je hebt gemaakt op welke devices en welke communicatietechnologie je hebt gebruikt en welke manier van analyse je hebt gebruikt. Dat zou de compleetheid van het framework genereren. Nogmaals als je het nog theoretisch model is zou de visualiseert van het framework enigszins generaliseer zijn. En dan in depth jouw thesis de in depth informatie komen. Dat is een beetje hoe ik denk erover

I: duidelijk onderscheiding dacht ik zelf, gaan we naar de volgende vraag welke requirements kunnen je geven over de consistentie van het framework daar bedoelen van of overall hetzelfde wordt genomen een alles een geheel is.

R: Wat ik weet over consistentie is dat het belangrijk is dat de genoemde technieken die je doet door het hele traject op dezelfde manier worden behandeld. Dan denk je bijvoorbeeld protocol A en protocol B en dat je er maar een gebruikt. Het gaat erom dat je je exact vasthoudt aan welke technieken je gebruikt en ook niet los van je framework extra componenten gaan benoemen. Er mag dus geen nieuwe informatie worden genoemd los van het feit dat je mag specifiëren. Los daarvan denk ik ook dat in het framework zeker in de zin van consistentie er consistentie moet zijn. Ik verwacht dat als je een framework opbouwt dat er in dat framework correlatie heerst. Dat kan uit eerdere literatuur worden opgevat of dat kan in feite worden getest maar dat je in zekere zin kan onderbouwen waarom deze componenten met elkaar kunnen bouwen. Je beroept je natuurlijk op het paradigma maar je specificeert enorm dus waarom zou dit samenhang kunnen hebben. Dus kijk naar de bron van de consistentie van het onderzoek eigenlijk.

I: Oké goed, ja dan hebben weh et over de volgende vraag alweer dat is welke requirements zou je kunnen geven over het bruikbaarheid van het framework. Dus stel je bent consultant en je wilt een IoT oplossing bouwen en je wilt dat framework ervoor gebruiken wat zou je daarover kunnen zeggen.

R: Consultant presenteren enorm vaak dus als een model goed visualiseert is, is het enorm sterk dus dat is het leidraad in het gesprek. Dus een goede visualisatie van de componenten is enorm belangrijk. Dus daarom zou i kalle aspecten goed uitgeschreven willen zien en ook呈teerbaar. Dus dat het logisch ontwikkeld kan worden en mensen snappen waarom deze componenten aan elkaar worden gekoppeld. Dus dan zou ik in dit geval een framework willen zien waarbij je bovenaan internet aspect zou hebben en onderaan helemaal het devices. Er is namelijk een grote extreme te zien. Wat heerst hier nu eigenlijk tussen. Dat zou in theorie middleware kunnen zijn. Zo een visualisatie had ik dan voor ogen. En dan ook het communiceert met elkaar want IoT is heel dynamisch. Dus dan zou je ook moeten kunnen aantonen dat het met elkaar kan communiceren en dat het een continue eh, traject. Dus dat zou ik ook onder bruikbaarheid kunnen zeggen. Los daarvan, ja dat is het eigenlijk wel.

I: Oké goed duidelijk, gaan we naar de volgende welke requirements zou je kunnen geven over de architecturale structuur van het framework. Ik kan wel even wat context daaraan geven. Dan kijken we naar de logica van het framework en de principes die zijn gekozen en de technieken die zij gekozen. Zijn de lagen goed onderscheiden en kan het normaal bekeken worden.

R: Nou ja, op basis van zo een oplossing daadwerkelijk goed in beeld laten komen, dus dat je het spitst op vooral voordat je natuurlijk zo een framework opstart spits je op de verschillende fasen die daadwerkelijk in het traject aan bod komen. Die heb je nu net al beschreven dus dat is goed, ik ga ervan uit dat vanuit daar het frame wordt opgehaald en ook laat zien waar precies de afscheiding is dus er zijn natuurlijk componenten die enigszins overflow hebben met elkaar en daar zou je een goede afweging moeten maken hoe en wat welke positie neemt het aan en welk proces dus ik zou als enigszins zeggen dat middleware zou niet als communicatie tool moeten gelden. Daar zou een aparte tool voor moeten worden ingesteld en daar eigenlijk kan worden opgeleverd. Los als we het dan over devices hebben welke koppen zitten er precies in en de sfeer tussen die twee en de middleware. Dat zou eigenlijk de structuur een beetje kunnen bepalen.

I: Dan gaan we naar de volgende alweer, welke requirements zou je kunnen geven over de rol van devices en schakelaars. Welke requirements daarover eigenlijk.

R: Requirements zou zijn dat je niet specificeert in je model welke apparaten je daadwerkelijk hebt gebruikt maar het gewoon omvat als sensoren dus generaliseren in termen en ook natuurlijk dat je de rol met de communicatie technologies of wat dan ook. Dus welke rol communicatie speelt. Dat is nou eigenlijk de bron de oorsprong van jouw data. Die halen alle data op en dat zou eventueel goed gespecificeerd worden en goed aangegeven worden in je model hoe het eigenlijk communiceert met elkaar.

I: Oké duidelijk naar de volgende dus welke requirements zou je kunnen geven over de communicatietechnieken in het framework dus we hebben onze sensoren dus hoe zou jij dat in zo framework graag terug willen zien.

R: Als ik het vanuit consultancy aspect kijkt moet het schaalbaar en dynamisch zijn. Dus schaalbaar dus het zou eigenlijk uitgebreid kunnen worden zonder het feit dat het de huidige flow van informatie aantast. En moet natuurlijk extreem duidelijk worden aangegeven wat de positie van de communicatietechnologie is. Dat is natuurlijk heel erg interessant wat is nou eigenlijk de rol wat voeren ze uit en wat is de urgentie. Dat is wel logisch dat het aangekaart wordt. En dat verklaart ook eigenlijk waar ik naartoe wil en het moet duidelijk een koppeling kunnen zijn tussen de objecten en devices en de middleware. Wat dat is eigenlijk waar het naartoe wordt gecommuniceerd lijkt. Dus het moet wel duidelijk laten zien hoe de koppeling is tussen die twee componenten.

I: Gaan we snel door naar de volgende welke requirements zou je kunnen geven over de geleverde services van het framework. Dat moet een specifieke rol hebben hoe zou je dat neerzetten in het framework.

R: Dit zou ik persoonlijk apart zien van de middleware want de services zijn een heel ander componenten is meer gericht op de client. Dus ik zou er een apart stuk van maken. Waar ik eerst aan zit te denken is dat het laag drempel moet zijn en leken die moeten begrijpen waar het voor staat en wat het betekent. Dat is natuurlijk logisch en betekent in feit dat er een laagdrempelige oplossing moet zijn. Ook dat je het schaalbaar maakt voor grotere oplossingen en dat mensen zich bezig kunnen houden met enkel de informatie en niet het onderhoud. Dat zou de rol van services zijn.

I: Welke requirements zou je kunnen geven over de rol die de middleware vervult.

R: Oké middleware is een apart stuk an sich. Ook daar ga ik weer terug naar een requirements dat het enigszins makkelijk uit te breiden moet zijn. Want wat hier eigenlijk gebeurt is dat je de data ophaalt uit de sensoren en het eigenlijk omzet van data naar informatie. Dus bruikbare punten. Mijne naar goud als het ware. Het moet natuurlijk ook worden opgeslagen. Ja het verwerken van het proces start om naar informatie te gaan en het communiceren staat daar los van. De data staan er opgeslagen en kan worden opgehaald door de tool voor de client maar het staat daar wel los van in feite zie ik als een soort van meer storage tool. Wat natuurlijk extreem dynamisch moet zijn.

I: Dan gaan we naar de laatste vraag welke requirements zou je eigenlijk aan het WSN geven dus alle sensoren die samen het netwerk opmaken dat moet met elkaar praten en met de middleware praten.

R: Ik weet dat je binnen WSN-gebruik maakt van verschillende protocollen zoals MQTT. Belangrijk is dat je niet specifiek benieuwd welke protocollen er aan bod komen. Dat het een beetje een opvatting is van de client zelf. Ook daar schaalbaarheid behoorlijk belangrijk en misschien extra punten aan het hele proces kan worden toegevoegd. Dynamisch is in feite dat het kan communiceren met je middleware en ook weer terug kan opvatten. En de sensoren moeten duidelijk aanwezig zijn (connectie).

I: Dat was hem, bedankt voor je tijd en moeite.

Samenvattend antwoorden op de vragen:

1. Welke req. zou u kunnen geven over de functionaliteiten van het framework?
 - a. *Duidelijk in een overzicht*
 - b. *Nauw beschreven hebt wat alle functies doen.*
 - c. *Overzichtelijkheid zodat mensen het 1 op 1 over kunnen nemen.*
 - d. *Sequentieel de verschillende aspecten in elkaar over laten lopen.*
2. Welke req. zou u kunnen geven over de compleetheid van het framework
 - a. *Alle aspecten bevatten die een IoT flow-based oplossing nodig heeft.*
 - b. *Beschrijft elk punt van het IoT proces.*
 - c. *Alle aspecten meegenomen van sensoren tot aan dataverwerking.*
 - d. *Verwacht een koppeling aan bestaande literatuur.*
3. Welke req. zou u kunnen geven over de consistentie van het framework?
 - a. *Verwacht dat dezelfde technieken gebruikt worden*
 - b. *Verwacht dat gekozen zaken met elkaar overweg kunnen*
 - c. *Bouwblokken van het framework komen uit de literatuur.*
 - d. *Bouwblokken van het framework hebben een onderlinge correlatie.*
 - e. *Genoemde technieken hebben overal dezelfde benaming.*
 - f. *Duidelijk aangeven hoe het framework moet worden gebruikt/gestart*
4. Welke req. zou u kunnen geven over de bruikbaarheid van het framework?
 - a. *Leidraad zijn wanneer er in de praktijk een oplossing wordt gebouwd.*
 - b. *Het wordt zo beschreven dat iedereen het kan oppakken en de informatie op zo manier wordt gepresenteerd dat leken het begrijpen.*
 - c. *De verschillende fases moeten duidelijk gevisualiseerd zijn in het framework en het moet meteen duidelijk zijn wat je nodig hebt om het te gebruiken.*
5. Welke req. zou u kunnen geven over de architecturale structuur van het framework?
 - a. *Het moet logica bieden en voor een consultant makkelijk te lezen zijn.*
 - b. *De lagen moeten duidelijk verdeeld zijn.*
 - c. *IoT zou splitsbaar moeten zijn in verschillende fases waarin de duidelijk de verbindende tools (bouwblokken) terugkomen.*
 - d. *Wat is nou eigenlijk het startpunt.*
 - e. *Allesomvattend zijn voor flow-based IoT oplossingen.*
6. Welke req. zou u kunnen geven over de rol van devices in het framework?
 - a. *Geen specifieke apparaten meegeven in het framework*
 - b. *Duidelijk de positie van de devices in het framework aantonen*
 - c. *Rol in combinatie met communicatie duidelijk meegeven*
 - d. *Aangeven dat hier alle data vandaan komt*
7. Welke req. zou u kunnen geven over de communicatie technieken in het framework?
 - a. *Duidelijke plek in het framework*
 - b. *Koppeling met andere aspecten zoals objecten en middleware*
 - c. *Schaalbaar*
 - d. *Dynamisch*
8. Welke req. zou u kunnen geven over de geleverde services in het framework?
 - a. *Services voor de client moeten zichtbaar zijn.*
 - b. *Laagdrempelig*
 - c. *Meerdere manieren om informatie te ontvangen*
 - d. *Afgesloten van de middleware oplossing*
9. Welke req. zou u kunnen geven over de rol die de middleware vervult?

- a. *Dynamisch*
 - b. *Data moet kunnen worden opgeslagen en verwerkt*
 - c. *Data omzetten naar informatie*
 - d. *Makkelijk uit te breiden*
 - e. *Breed support*
10. Welke req. zou het Wireless Sensor Network aan moeten voldoen?
- a. *Schaalbaar*
 - b. *Dynamisch*
 - c. *Duidelijke connectie met middleware en de sensoren*
 - d. *Geen specifieke protocollen noemen*

XVII. Interview framework requirements respondent C

Ewout Masreeuwop 04-12-2020

Samenvattend antwoorden op de vragen:

1. Welke req. zou u kunnen geven over de functionaliteiten van het framework?
 - a. *Architectuur zichtbaar moet zijn*
 - b. *Datastromen inzichtelijk hebben wat doe je met die data*
 - c. *Hoe wordt data informatie*
 - d. *Plek zijn in het framework waar de data naartoe (output/input data) visualiseren*
 - e. *Wat gebruik je waar voor (specifiëren)*
 - f. *Welke securityprotocollen je gebruikt.*
 - g. *Web applicatie of ERP voor legacy systems.*
2. Welke req. zou u kunnen geven over de compleetheid van het framework
 - a. *De gehele datastream moet erin staan van sensor (input) tot use case (output)*
3. Welke req. zou u kunnen geven over de consistentie van het framework?
 - a. *Structuur zien die consequent is*
 - b. *Voorkomen van ruis met detail level van aspecten*
 - c. *Paradigma moet begeleidend zijn voor de gebruiker om specifieke aspect te visualiseren.*
 - d. *Gelaagdheid moet je in geholpen worden*
 - e. *Datastromen generieke term bedenken*
 - f. *Meerdere micro controllers in het framework*
 - g. *Publish/Subscribe en Microcontroller eventueel samenvoegen*
4. Welke req. zou u kunnen geven over de bruikbaarheid van het framework?
 - a. *Makkelijk op te schalen zijn*
 - b. *Begeleiding zodat het duidelijk is op welk niveau je opereert*
5. Welke req. zou u kunnen geven over de architecturale structuur van het framework?
 - a. *Welke data gaat nu waar heen*
 - b. *Welke veiligheidsprotocollen*
 - c. *Welke data is beschikbaar*
 - d. *Data integration intern of extern*
6. Welke req. zou u kunnen geven over de rol van devices in het framework?
 - a. *Duidelijk maken dat er meerdere devices aanwezig zijn*
 - b. *Datastromen van devices uitleggen*
 - c. *Publish/Subscribe & microcontroller samenvoegen*
7. Welke req. zou u kunnen geven over de communicatie technieken in het framework?
 - a. *Schaalbaar*
 - b. *Veilig*
8. Welke req. zou u kunnen geven over de geleverde services in het framework?
 - a. *Niet specifiek op chatbot maar meer communicatiemethode*
 - b. *Dashboard hoeft geen verplichting te zijn*
 - c. *Visualize moet ook automate kunnen zijn visualize/integrate*
 - d. *Legacy system naast web app*
9. Welke req. zou u kunnen geven over de rol die de middleware vervult?
 - a. API
 - b. Data storage int. Ext.
 - c. Nadelen over de rol van analyse automations & chatbot t.o.v. integration/services

10. Welke req. zou het Wireless Sensor Network aan moeten voldoen?
 - a. Schaalbaar

Algemene opmerking voor huidige framework:

Denk na over de positie van de webapplicatie misschien hernoemen naar integration. Denk ook na over de rol van API, Legacy & communication (chat, notifications, and dashboard).

XVIII. Interview framework validation respondent J

op 08-12-2020 IoT expert

De vragen die gesteld gaan worden nemen elk van deze middelen door en verwachten een passend antwoord waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen.

Vragen:

11. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
 - a. *Betrekking op de omgeving waarin je het wilt toepassen.*
 - b. *Duidelijk moet zijn in welke omgeving je iets aan het maken bent.*
 - c. *Omgevingsfactoren specifiëren of noemen, dat ontbreekt in het framework.*
12. Wat zou u kunnen zeggen over de compleetheid van het framework
 - d. Analyze, automate communicatie. Keuze maken wel of niet onder computate zetten.
 - i. Situatie afhankelijk
 - e. Data flow/information flow encrypted laten maken
13. Wat zou u kunnen zeggen over de consistentie van het framework?
 - f. Lijkt niets te missen
 - g. Rol van internet visie met services en client
14. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
 - h. Waar komen de requirements vandaan bijvoorbeeld de business perspectief of IT innovatie perspectief ten opzichte van top down of Bottom up.
15. Wat zou u kunnen zeggen over de architecturale structuur van het framework?
 - i. Gebaseerd op de IoT stack, framework laat correct de aspecten van IoT zien en manier hoe ze met elkaar werken zien.
16. Wat zou u kunnen zeggen over de rol van devices in het framework?
 - j. Staan in de perceptie laag. Ze nemen data op en verwerken deze niet. Het staat goed.
17. Wat zou u kunnen zeggen over de rol van de communicatietechnieken in het framework?
 - k. Pub/Sub is een sterke en wel gebruikte techniek om data uit te wisselen.
 - l. Goed dat er geen specifiek protocol wordt gebruikt zodat de gebruiker van het framework daar aan der hand van business req. Zijn eigen keuze in kan maken.
18. Wat vindt u van de rol van de geleverde services?
 - m. Ziet er prima uit misschien in de tekst laten weten wat onder die API & Legacy-system valt zoals ESB/SOA.
19. Wat vindt u van de rol van de (front-end communication) webapplicatie?
 - n. Waarborgen van de IoT processen
20. Wat zou u kunnen zeggen van de rol die de middleware vervult?
 - o. Duidelijk maken hoe de flow-based tool zich verhoudt tot standaard communicatie zoals CSV, XML?
21. Overall feedback over het framework.
 - p. Internet view misschien een aparte combinatie met client en services. Building blocks worden goed onderscheiden. Omvat noodzakelijke IoT aspecten.

XIX. Interview framework validation respondent E

De vragen die gesteld gaan worden nemen elk van deze middelen door en verwachten een passend antwoord waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen.

Vragen:

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
 - a. Ondersteboven zetten
 - b. Flow op basis van triggers
 - c. Wat bedoel je met legacy-system
 - d. Begin/einde van het framework
 - e. Visualize = analytics, chatbot, dashboard side.
 - f. Legacy system (externe systemen noemen i.p.v. legacy) = voorbeeld in de tekst
2. Wat zou u kunnen zeggen over de compleetheid van het framework
 - a. Aanpasbaarheid van sensoren = state-change
 - b. Analyze Automate Communicate zet er een pijl bij
3. Wat zou u kunnen zeggen over de consistentie van het framework?
 - a. Wel of geen iconen gebruiken en niet enkel voor data storage
 - b. Visualize en Legacy kan meer over worden uitgeweid (detail-niveau)
 - c. Tweeziijdig maken van de pijlen voor dataflow en info flow loopt niet gelijk
4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
 - a. Aanpasbaarheid van sensoren = state-change
5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?
 - a. Zie comment over volgorde
6. Wat zou u kunnen zeggen over de rol van devices in het framework?
 - a. Onderbelicht alleen sensors & actuators
 - b. Niet duidelijk of het wel of geen zelfde types zijn.
 - c. Example sensoren
 - d. Microcontrollers
7. Wat zou u kunnen zeggen over de rol van de communicatie technieken in het framework?
 - a. Laat de verschillende communicatie technieken er zijn: LoRa, Wifi Blauwe tand.
8. Wat vindt u van de rol van de geleverde services?
 - a. Chatbot, dashboard en notificaties
9. Wat vindt u van de rol van de (visualize) web applicatie?
 - a. Chatbot, dashboard en notificaties
10. Wat zou u kunnen zeggen van de rol die de middleware vervult?
 - a. Ziet er compleet uit
11. Overall feedback over het framework
Compartimenten goed opgedeeld/uitgevoerd
Goed de hele rotzooi opgedeeld
12. Wat zou u kunnen zeggen over de beschreven principles?
 - a. Benoem wat en waarom voor elke principle
 - b. Domain specific wel benoemen waarom
 - c. Docker benoem waarom (standariseren)

I: Oké super nou, bedankt dat je wil meedoen aan mijn interview over het valideren van mijn IoT framework. We hebben 12 vragen en gaan direct door naar de eerste vraag en die is wat zou je kunnen zeggen over de functionaliteiten van het gepresenteerde framework?

R: Als ik kijk naar het gepresenteerde framework vraag ik me af of je het vanuit een user of business perspectief hebt geschreven want om te begrijpen wat er gebeurt kijk ik eigenlijk eerst naar de onderkant om naar de sensors en de microcontrollers te zien en te kijken. En te zien hoe die via analytics naar het dashboard output geven. Dus ik zou adviseren om de plaat onderste boven te zetten en om de jouw meest unieke punt het gebruik van sensoren daarmee centraal te stellen. Daarnaast vanuit het business perspectief zou ik meer uitweiden over het visualiseer gedeelte. Uhm zodat de lezer begrijpt wat de eindgebruiker allemaal te zien krijgt. Uhm, daarmee krijg je een flow op basis van triggers wat gebeurt er chronologisch want een sensor pikt iets op, het gaat naar een analytische laag en dat wordt door gestuurd naar een presentatie laag en dat leest duidelijker en makkelijker weg. Verder zou ik mij meer uit gaan wijden over legacy systemen, mogelijk moet je dat gewoon external systems noemen want bijna alle externe systemen zijn legacy systemen. Wijd ook uit of je het daar hebt over SAP in combinatie met Node Red of automation anywhere integraties. Je hebt er genoeg ruimte en het voegt heel veel toe voor de lezer om zich een beeld te vormen om wat voor soort integraties hier beoogd worden. Hetzelfde geld voor visualiseren, want voor soort integraties beoog jij hier is van belang om daar te noemen. Want het is zonde om heel veel werk te doen kwa sensoren en analyse en dat goed te lichten maar het niet vervolgens te presenteren. Visualize zou ik dus samenvatten tot verschillende sub kopjes zoals analytisch, chatbot en dashboard kant en manier van presenteren.

I: Precies, ja volgende vraag is inderdaad over wat zou je kunnen zeggen over de compleetheid van het framework?

R: Ik vraag me af, uhm. Of de manier waarop. Ik vraag me af wat voor soort sensoren het over gaat. Momenteel presenteert twee microcontrollers. Ik maak hier niet uit of het om 2 of 200 gaat en of het allemaal dezelfde soort sensoren betreft of verschillende. Als die licht is en temperatuur en vochtigheid benoem dan al die drie de sensoren apart want dat verbetert de compleetheid van je sensor laag maar ook de compleetheid van je analytische laag want dan weet je wat de input is van je analytische laag.

En als we het toch over de middellaag hebben, zet een extra pijl bij analyse, automate en communicatie want dat is redelijk logisch.

I: Dan gaan we naar vraag drie en dat is wat zou je kunnen zeggen over de consistentie van het framework.

R: Als je iconen gebruikt zoals je heel leuk doet in de middelste laag, niet opvallend maar gewoon de vorm van de blokken en vaag aanpassen aan de context doe dat dan overal wat mijn voorkeur is of doe dat nergens. Maar doet dat niet alleen in het midden. Verder over visualiseer en legacy dat kan verder worden uitgebreid, dit omdat de andere lagen wel gedetailleerd zijn. Verder zie ik dat de data flow een kant op gaat en de information flow twee kanten op gaat. Dit impliceert dat er data vanuit sensoren verstuurd naar de middleware en de interne analytisch laag heen en weer communiceert met de eindgebruiker maar nooit iets terugstuurt naar de sensoren. Dit terwijl ik mij kan voorstellen dat er juist informatie terug wordt gestuurd omdat de eindgebruiker commando's geeft. Dat zou dat net zo goed bij datastream vanuit de analytisch integratie ook weer terug naar beneden moeten. Dus kies of voor eenzijdig of tweezijdig.

I: Oké duidelijk, dan gaan we naar de volgende vraag en dat is wat zou je kunnen zeggen over de bruikbaarheid van het framework.

R: Mijn eerdere benoemde punt over de aanpasbaarheid van de sensoren en de state changes ik vraag me heel erg af wat de tijdintervallen zijn voor de sensoren elke 30 seconde hoe vaak wordt de analytiscs gedraaid. Dit gaat bepalen hoe real time werkt. Je kan het meest perfecte dashboard ooit bouwen maar als het maar 1x per jaar refresht is het geen goed dashboard. Dus wil ik heel graag hoe vaak het dashboard hoe snel dashboard geüpdateert wordt en omdat te begrijpen wil ik ook heel graag weten hoe snel de lagen eronder zich aanpassen.

I: Dan gaan we door naar vraag 5, dat is wat je zou kunnen zeggen over de algemene architecturele structuur van het framework?

R: Zoals eerder benieuwd zou ik hem omdraaien en zou ik beginnen met hem chronologisch maken en je begint met jouw meest unieke selling point en dat is het gebruik van sensoren bij deze planten en dan naar analytsich en dan naar presenteren.

I: Wat vind je van de rol die devices hebben op het framework. Dat is eigenlijk al een beetje beantwoord in de vorige vraag.

R: Nouwjah, zet je sensoren meer in de spotlights ik wil weten wat voor soort sensoren het zijn, benoem een paar types om duidelijk te maken dat het niet alleen om een vochtigheidssensor gaat want daar doe je jezelf te kort mee. Want het is het coolste en meest unieke gedeelte van je hele project. Iedereen gelooft dat je data kan pushen en T-toetsen kan uitvoeren en dat kan presenteren in een app. Zet je belangrijkste stuk niet onderaan. En maak explicet wat verschillende type sensoren. Noem daar voorbeelden van. En zeg gewoon microcontrollers met een waarde daarnaast zoals 20 dat je weet dat het 20 sensoren zijn.

I: Dus eigenlijk de rol van de sensoren en schakelaars moeten beter naar voorkomen en misschien voorbeelden van wat dat in de praktijk zou kunnen betekenen

R: Ja, je mag best opscheppen over het coolste stukje.

I: Nice, dan gaan we door naar vraag 7, wat zou je kunnen zeggen over de rol van communicatietechnieken in het framework dus het publish/subscribe network.

R: Ik zie hier WSN staan en benieuwd of je bluetooth, wifi of Zigbee aan het gebruiken bent en ik ook heel erg benieuwd naar jouw overwegingen daarbij. Waarom je wel voor de een en niet de ander gaat. Dit is ook weer een punt waar jij je expertise toont. Dit door die uit een te zetten. Dat zou ik wel beter in de tekst doen dan hier doet dat alsjeblieft wel.

I: Dan gaan we door naar vraag 8 en 9 die eigenlijk samen al zijn beantwoord dat is wat je vindt van de rol van geleverde services en de rol van het visualisatie gedeelte dus helemaal de bovenkant van het framework.

R: Ik begreep van je dat je daaronder chatbot functionaliteiten voorziet, dashboard en notificaties en pushmessages. Wijk daarover uit want dat geeft mij een veel beter beeld over het eindresultaat van je framework voor een gebruiker. En uiteindelijk ziet de gebruiker alleen het eindresultaat en daar gaat hij je keihard op afrekening dus ik ben heel erg benieuwd wat dat resultaat dan is.

I: Dan gaan we naar vraag 10, de rol van middleware de centrale plek en de plek waar alle logica zit. Vind je dat die goed is gepresenteerd in het framework.

R: Nouwjah vanuit mijn achtergrond in data analytisch ziet dit er voor mij er duidelijk uit en logisch uit. Ik zou alleen die pijl toevoegen.

I: Oke top dan gaan we naar de laatste twee vragen waarvan er 1 heel algemeen is die is wat is je overall feedback als je in totaliteit naar het framework kijkt. Wat zou je dan hierover nog kunnen zeggen.

R: Nou je bent allemaal componenten aan het beschouwen in je framework en ik vind het heel fijn hoe je dit opgebouwd hebt. Dit maakt een heel duidelijk overzicht, duidelijker dan mijn woning om zo te zeggen.

I: Oké super, dan gaan we naar het laatste onderdeel toe en dat zijn eigenlijk de principles en de tabel die ik hiervoor even heb laten zien. Wat zou je daar eigenlijk over kunnen meegeven.

R: De principles zijn goed maar ik zie bij een aantal bij elk principle uhm moet je een wat en een waarom vermelden. Van oke leuk dat dit een principle is maar geef aan waarom. En bij sommige is dat heel duidelijk als er low-cost of low-energy staat dan kan ik mij goed voorstellen dat het is om geld te besparen maar zeker bij aan je meer internet en je middleware kant maak meer duidelijk waarom je het principle hebt. Als er staat "services should support docker" wil ik erachter hebben staan voor of because voor standaardisatie en niet omdat je docker nou heel erg cool vindt.

I: Oke super, dat waren de vragen dus hartstikke bedankt voor je tijd dan ga ik de opname nu stoppen.

XX. Interview framework validation respondent F

Vragen:

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
 - 1.
2. Wat zou u kunnen zeggen over de compleetheid van het framework
 1. Ik denk dat hij vrij compleet is maar niet logisch is opgebouwd. Je hebt input er gebeurt iets en je hebt output. Maak duidelijker wat de input/output en wijzingen worden gemaakt. Ligt eraan wat je doelgroep is meer technisch of meer business dan zou je links naar rechts moeten. Technisch, of business. De business is je control dus dan zal ik zeggen maak hem landscape. Maak hem ook simpel. Hij lijkt vrij compleet maar maak het simpel.
3. Wat zou u kunnen zeggen over de consistentie van het framework?
 1. Inconsistent met technology dus docker is heel specifiek maar flow-based totaal niet. Zelfde met docker dat is container. Verander docker naar container. Ga voor een gelaagdheidslevel. Geen zigbee maar protocols. Het hele framework moet op dezelfde laag zitten. Chatbot moet op conversational AI chatbot is daar een onderdeel van. Dashboard. Noem eventueel. API helpt met de zwarte box dat kan SOAP, REST of RPA of SAP PO zijn. Dat zijn technieken.
 2. Je datamodel als microservices. Pak de microservices gedachte en hak alles erin op.
 3. Wat ik nog wil doen is een andere laag dat is iets met je data doen. En daarna kan je nog predicten. Kijk naar het bpm-proces want eigenlijk met image reg en de sensor en de data die je genereert dat is een onderdeel.
 4. Predictive analytics
 5. Wat is je doel is het een hulpmiddel.
4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
 1. Geeft nog niet genoeg antwoorden of antwoorden. Beslis goed wat je doel is van het framework. Gebruik je het framework aan de voorkant als hypothese om hem te maken of gebruik je hem ook om te testen daar is het framework niet voor.
 2. Misschien kan het helpen in het prototyping stage van een startup van lean. Misschien is het framework geschikt voor prototyping in
 3. Applied flow-based IoT framework toegepast op de realiteit en deze vragen. Het is voor als je een idee hebt en je eerste plan die je wilt invullen.
5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?
 1. Draai hem om again, voor wie maak je hem en welke flow.
 2. Kijk naar BPM

6. Wat zou u kunnen zeggen over de rol van devices in het framework?
 1. Kijk micro/macrocontroller. Specificeer daarin.
7. Wat zou u kunnen zeggen over de rol van de communicatietechnieken in het framework?
 1. Haal de specifieke protocollen weg maar noem gewoon protocols. Misschien bij principes voorbeelden geven.
 2. Wat vindt u van de rol van de geleverde services?
 3. Niet specificeren in specifieke services. Dus visualize en external systems pas op met rest en soap api en hou het algemeen.
8. Wat vindt u van de rol van de – applicatie

Voeg principles toe als:

1. Heb je al api strategie
2. Heb je al power bi
3. heb je al containers
4. Moet je dingen opnieuw opzetten

10. Wat zou u kunnen zeggen van de rol die de middleware vervult?

11. Overall feedback over het framework:

1. Framework is heel vet voor toepasbaarheid op alle componenten internet middleware things allemaal invullen en dezelfde vraag stellen op die drie assen en dat daaruit komt of het duur of niet duur is. Dus kijk naar tijd geld o
2. Maak duidelijk waarom je het framework gebruikt. Voor wie het is en in welke situatie je het gebruikt. Dit is meer solution design high-over. High-level solution
3. Form follows function.
4. architecture framework. Dit moet er minimaal staan in het document. Het framework gaat je helpen met requirements opzoeken in de business om je oplossing te bouwen.
5. Bij grote bedrijven heb je enterprise/business en iedereen moet dit plaatje begrijpen. Business, Architecten en Infra (hosting). Alle stakeholders wil je met deze plaat een taal laten praten. Iedereen spreekt een andere taal en met deze laat je de rolverdeling goed zien. Met andere brillen naar hetzelfde product bekijken. Gebruik het praktisch nut van het framework. Stel je publiceert dit werk in een paper dan kun je zeggen dat dit een praktische manier is om met flow-georiënteerde IoT te werken. Het is een soort cheat sheet. Welke dependencies zijn er. Wie heb je nodig doorlooptijd per afdeling je kan er user stories ontwikkelen. Doorlooptijd staat los van ontwikkeltijd. De delta op die twee.

12. Wat zou u kunnen zeggen over de beschreven principles?

1. Voeg handvatten toe
2. Voor de toepassing maak inzichtelijk wat je ist soll en gap is dus heb je bij de analyse. Kan een template zijn voor een gap analyse.
3. Why what how miro. Ingevoerde cursus met je team of met arthim en ewout kan je voor dit framework dit even invullen.

XXI. Interview framework validation respondent G

De vragen die gesteld gaan worden nemen elk van de eerder gepresenteerde variabelen door of een van de elementen die in het framework worden gebruikt. Verwacht wordt dat per vraag wordt toegelicht waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen.

Vragen:

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?

Voor een eerste opzet ziet het er dekkend uit.

Things kunnen bestaan uit MCU's; Microcontrollers (ESP/STM/ATMEL/etc) maar ook MPU's; Microprocessors (Raspberry Pi ARM/ x86 of x64 based platforms), de laatste wordt niet genoemd maar ook Edge devices behoren tot IoT .

2. Wat zou u kunnen zeggen over de compleetheid van het framework?

Het is erg gelaagd, zoals scholen het graag willen benaderen omdat dit een goede functionele afsplitsing verzorgt en je dus prettig kunt bepalen waar bijvoorbeeld aggregatie moet plaatsvinden. Kijkend naar wat de computertechniek de afgelopen decennia heeft gedaan is er stuiterbal effect geweest van thick, naar thinclient tot waar we nu staan; een soort hybride staat met cloud in een bijzondere centrale rol. Kijk bijvoorbeeld naar Amazon Greengrass waar Cloud functionaliteit naar de EDGE wordt gepushed en een IoT infrastructuur in staat is om -zonder- cloud te kunnen functioneren.

In deze huidige vorm zie ik de beste "pas" om een goede invulling te geven voor specifieke scenario's, waar hard- en software op elkaar zijn afgestemd. Dit is niet per se de meest generieke benadering. Waar je soms sensoren als domme dingen wilt beschouwen kun je deze ook inzetten om lichte aggregaties te doen zodat je bandbreedte en databundels bespaart. Dat houdt dus wel in dat je wat intelligentie daar neer legt, iets wat de meeste MCU's al prima aankunnen.

Functioneel zou ik Machine Learning en Artificial Intelligence toevoegen bij de uitbouw van dit framework om te laten zien waar het naartoe gaat (krachtigere doch specifieke sensoren). Neurale structuren worden meer en meer onderdeel van de sensoring kant. (Camera's met OCR, QR-code of gezichtsherkenning etc.) Devices worden hierdoor meer autonoom en dus in staat om zelfstandig of binnen een EDGE-structuur hun ding te doen.

Misschien interessant om te kijken naar de verschillende architectuurmodellen die de lead architect van NGINX heeft beschreven rond 2016.

<https://www.nginx.com/blog/microservices-reference-architecture-nginx-fabric-model/>

*Met name het **fabric model** beschrijft eigenlijk een zelfstandige architectuur welke vanuit een abstract template beschreven wordt en zelfstandig de interne security toepast.*

Waar dit misschien een communicatiestructuur van het internetservices beschrijft is er qua architectuur niet zoveel verschil met fysieke communicatie anders dan de flexibiliteit in de "instantiëring" van dergelijke devices.

3. Wat zou u kunnen zeggen over de consistentie van het framework?

Het moment dat de stroom uit valt aan de middleware kant is het de vraag of de compleetheid van de berichtenstroom geborgd is.

Er wordt gesproken over data storage en afhankelijk van wat je met de data wil gaan doen zal deze mogelijk nog een voorbereidende handeling moeten ondergaan. (Denk aan real-time aggregaties naar GraphDB of soortgelijke grafen database).

Event sourcing is voor IoT een veelgebruikt mechanisme dat in veel gevallen afdoende dekkend is. Het hebben van een concrete database in de cloud biedt schaalbaarheidsvoordelen.

Het gebruik van containers is ideaal voor schalen naar hogere dekking of belasting. Cloud “alternatieven” zoals serverless zijn binnen handbereik en mijns inziens een goedkoper en nog beter schaalbaar. Een goede voor -next level- IoT.

4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?

Voor een single tenant omgeving (een fabriek, een huishouden of een vestiging van een bedrijf) is dit een prima framework en hoef je het verder niet uit te diepen omdat het de voornaamste aspecten bedient.

5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?

Kijkend naar de schalings en onderhoudsaspecten kan het zijn dat het framework qua beheersbaarheid te wensen over laat zodra er honderden devices in zitten. De abstractie van de functionaliteiten welke de things “alleen” of als groep realiseren zie ik graag terug in het framework.

“Things”, zijn bedienend binnen een bepaalde context. Dit kan de functionele context zijn (temperatuur meten) of geografische context (in Rotterdam, Nederland) maar ook Proces of Applicatiecontext zoals (Warehouse, Silo, Motor, 3D printer etc.,). Daarmee kunnen “things” vanuit verschillende perspectieven benaderd worden.

Security zie ik binnen deze architectuur niet geïmplementeerd waar het geautomatiseerd beheer van certificates binnen een professionele of bedrijfskritische omgeving niet kan worden overgeslagen. Je wil immers niet dat dergelijke informatie zomaar in handen van anderen komt.

6. Wat zou u kunnen zeggen over de rol van devices in het framework?

Dit antwoord dekt ook een deel van vraag 7 af.

‘Low-level’ zijn er veel bestaande communicatietechnieken waar ook niet iedere MCU of MPU -out of the box- mee om kan gaan. De generieke protocollen zoals Serieel (RS232 / UART) veelal wel, maar waar MCU’s communiceren met sensoren via I²C en SPI gebruiken industriële netwerken veelal RS-485, CANbus, Profinet, SCADA etc.

Het hebben van een veelzijdig IoT device en het kunnen toepassen van sensoren is essentieel om aan te kunnen sluiten bij de stuur/informatievraag welke de IoT oplossing in wil voorzien.

7. Wat zou u kunnen zeggen over de rol van de communicatietechnieken in het framework?

REST/JSON is eigenlijk de “Things” standaard; dit zou ik ook als zodanig noemen. Er wordt API vermeld waar dit niet direct het protocol aanwijst en zijn oorsprong kent in XML/WSDL structuren. Structuren die ik te allen tijde zou vermijden. Naast API’s zal in de nabije toekomst worden overgegaan op GraphQL als flexibel protocol, misschien toegepast op een hogere abstractielag.

8. Wat vindt u van de rol van de geleverde services?

Ze vullen elkaar goed aan en zijn allen van een gelijksoortig "schalingsniveau".

Een nette balans die goed zal kunnen functioneren.

Er zit hier geen Porsche 2 Liter motor op een fiets.

9. Wat vindt u van de rol van de webapplicatie?

Het los zien van de bouwblokken welke binnen deze architectuur beschreven zijn en de webapplicatie is structureel prima. Het MVC-model is al geruime tijd een duidelijke afsplitsing tussen gegevens opslag, inzage en beheer. In een zekere mate zal je zien dat de protocol scheidingslijn tussen de componenten van MVC "dunner" wordt. Websockets in IoT kunnen immers vrijwel direct "praten" met een dashboard.

Dit patroon is goed voor de snelheid van ontwikkeling; web applicaties kunnen sensoren dus vrijwel direct en bijna zelfstandig "interpreteren".

Waar je meer "orchestratie" verlangt (bijv. via NodeRED) is dit via het Event protocol mooi in te regelen; alles en iedereen kan acteren op sensors en het in elkaar zetten van een Dashboard met geaggregeerde en geharmoniseerde informatie is meer een kwestie van design en het onderliggend model begrijpen dan van de techniek.

10. Wat zou u kunnen zeggen van de rol die de middleware vervult?

Het publish/subscribe model vind ik onderdeel van de middleware, niet van de communicatie laag. Afstemming of LoRa / Wifi oid gebruikt wordt en de security van deze laag TLS / SSL etc. is meer onderdeel van deze communicatielaag.

Brokerage; de rol die MQTT speelt en NodeRED ; de orchestratierol is typisch middleware en een essentieel breid rond de werking van het IoT Framework.

11. Overall feedback over het framework

Zoals een goede architectuur vaak behoeft is dit een "bouwblokken" concept, een waar de functionaliteiten netjes en duidelijk zijn opgesplitst, de bouwblokken bestaand, bewezen, veelgebruikt en modulair zijn. Ik beschouw het als een goede basis om een framework te starten en zie een architectuur met dezelfde bouwblokken vaker terugkomen.

Op maintenance vlak mis ik nog het een en ander; denk aan het kunnen -upgraden- van devices (OTA), de alhele governance van het IoT landschap.

Het omgaan met onvoorziene situaties en risico's mitigeren door failsafe statussen te governen. Het governen van IoT devices en de data welke ze harvesten en het uitvoeren van translaties. Het laatste is meer een vraagstuk dat zijn oplossing in software kent maar een belangrijk aspect aan de beheerkant. Environment "health"; het weten of alles goed werkt en dat dus ook kunnen borgen. Electrische apparaten houden er allemaal een keer mee op, het vervangen moet een happy flow zijn en is een beheersaspect.

12. Wat zou u kunnen zeggen over de beschreven principles?

Het gros van de principles houden stand met andere architecturen. De principles mogen binnen een Enterprise omgeving niet als compleet worden beschouwd, mede wegens het ontbreken van autorisatie en authenticatie principles en basis beveiligingsprincipes.

Denk ook aan principes zoals het samenwerkingsverband tussen sensoren en actoren om gevaarlijke situaties uit te sluiten -ook bij onverwachte situaties-. (Bijv. Een grote roldeur die

naar beneden komt terwijl er iemand onder staat) Zowel mechanisch als software gestuurd moet er rekening gehouden worden met voldoende diepgang in de uitzonderingssituaties.

Het KISS principe krijgt voorzichtig ook toenadering bij een dergelijke structuur om steun te bieden aan de architectuur. Wat vaak complex kan zijn is de orchestratie van het geheel (wat moet wanneer gebeuren). AWS gebruikt Shadow techniek om enkele zaken “eenvoudig” te benaderen.

XVIII. Interview framework validation respondent H

I: We zijn aan het opnemen, oké, super dank je wel voor het willen helpen en het willen bijdragen aan mijn framework met hulp van een interview. Dan gaan we direct door eigenlijk naar de eerste vraag en dat is kijkend naar het framework wat zou je dan eigenlijk kunnen zeggen over de functionaliteiten hiervan.

R: Kijkend, naar het totale plaatje is het helemaal compleet uhm, ik begin wel meestal beneden bij de sensoren want die spreken mij het meeste aan. Verder uhm, wat ik wel mis is voorbeelden hiervan dan mensen beter begrijpen van oké wat is het nou, waar kan ik dat in terugzien in het dagelijks leven. Verder wel echt een hele goede basic, laat ik het zo zeggen. Ik als IT'er zou het gewoon begrijpen ook als ik niet veel kennis heb van data science. Dat is wel heel erg fijn, maar ik kijk ook naar mensen die niet technisch zijn en in een IT-bedrijf werken zoals business managers of zulke dingen. Hoe zou je het hun dan uitleggen, dan zou het misschien wat meer Jip & Janneke taal. Maak het dan op die manier iets duidelijker.

I: Dus jij zou zeggen dat je de business kant toch wel meer meenemen met voorbeelden.

R: Ja want IoT wordt een hot-topic op dit moment, iedereen wil er iets mee doen maar raken ze de kluts weer kwijt, dus maak die mensen ook duidelijk waarom IoT wel of niet handig kan zijn.

I: Oké top, dan gaan we door naar vraag twee en dat is eigenlijk wat zou je kunnen zeggen over de compleetheid van het framework, bijv. zitten alle aspecten van IoT in het framework

R: Ja dat sowieso, wat heel belangrijk is, is dat bij sensoren wil zien wat voor sensoren en hoe kan ik daarmee omgaan en wat ik bij front-end dan denk is hoe kunnen we dan visualiseren en kunnen we dit ook op andere manieren doen. Daar komen vragen naar boven bij mij zoals bijvoorbeeld een led-kubus of in de vorm van ja, hoe zeg ik dat, bijvoorbeeld ook VR. Dus ook andere manieren van data visualisatie laten zien.

I: Dan gaan we door naar vraag drie en dat is wat zou je kunnen zeggen over de consistentie van het framework dus of alle aspecten goed geplaatst zijn en op de juiste manier terugkomen.

R: Ja wat ik heel erg belangrijk vind is dat bijvoorbeeld de sensoren en de schakelaars heel flexibel moeten blijven. Iedereen zou alles met zijn of haar opties moeten kunnen aansluiten. Dit is heel duidelijk erbij gezet in het framework. Wat ik ook erg belangrijk vind is dat de visualisatie van het framework en dat dat ook flexibel blijft. Iedereen wil namelijk op zijn of haar eigen manier iets zien. Ik denk dat je dat ook hier hebt toegepast. Het zou wel meer getoond mogen worden. Maar tegelijkertijd moet je het ook weer simpel houden. Je wilt het natuurlijk zo makkelijk mogelijk toelichten maar je wilt ook laten zien dat alles mogelijk is. Probeer daar gewoon een balans in te zoeken inderdaad. Daarbij is het belangrijk dat je ook worden gebruikt bij bijv. Zigbee & LoRa gebruik daar ook Wifi of BLT omdat mensen daar eerder herkenning bij hebben dan de moeilijkere termen. Het hoeft niet hoe zeg je dat, het hoeft vast te zijn. Dat is IoT weer niet.

I: Dan gaan we naar de vraag, naar vraag vier en dat is over de bruikbaarheid van het framework dus als consultant zie je het jezelf gebruiken en zo ja hoe.

R: Ja ik zie gewoon letterlijk als ik de architectuur plaat dan zie ik letterlijk gewoon een soort keuzemenu en dat is voor mij dus ideaal. Van oké ik wil deze sensoren gebruiken, deze protocollen gebruiken voor draadloze communicatie. Wat betreft data storage dat is niet mijn vakgebied en dat zou je voor mij mogen bepalen en daarna natuurlijk de visualisatie kan ik op deze manieren toepassen. Dan denk ik echt van links naar rechts dat zou echt ideaal zijn om dat te kunnen proberen. Dus ja, ook

voor mensen die IoT of zich erin willen specificeren zouden dit ook kunnen gebruiken om steeds beter te weten wat voor protocollen er zijn en wat er mogelijk is en dat vind ik ook leuk hieraan.

I: Gaan we door naar de volgende vraag over de architecturale structuur van het framework of alles duidelijk geschreven is en ook goed gelaagd is.

R: Wat ik vind is, ik kijk naar beneden wat ik start bij sensoren, want wat heb ik allemaal nodig en dan kijk ook naar microcontroller en microprocessor dat vind ik dus heel erg op elkaar lijken als hardware engineer want je programmeert ze allebei met C of C++, dus laat maar zeggen die hard programmeren, dus ik zat echt zo van mmm, is wel waar maar misschien is een beetje wel inderdaad het beste zoals de bestaande en de andere zet je zelf in elkaar in te bouwen. Dus misschien moet je dat verschil duidelijker laten zien. Denk ik dat het allemaal duidelijk is, ik had hier verder geen opmerkingen aan.

I: Dan gaan we door naar vraag 7, en dat is eigenlijk of je wat kan zeggen over de rol van communicatietechnieken, die slaat eigenlijk al weer een beetje op het antwoord van de vorige vraag.

R: Inderdaad dus geef voorbeelden en combineer het met dagelijks taken, dus het voorbeeld dat Wifi en BLT meer wordt gebruikt dan Zigbee. En vooral als je dat aan niet technische mensen wilt uitleggen. Dus als je dit aan een bedrijf laat zien zou het top zijn maar daarbuiten niet.

I: oké dan gaan we naar vraag acht en die gaat over geleverde services en de rol daarvan. Dus dat gaat over de verwerking van alle data en hoe dat uiteindelijk gepresenteerd wordt aan de client.

R: Ja, daar heb ik ook al eerder iets over genoemd, ik vind dus ook dat dat meer getoond kan worden. Ik bedoel IoT is best wel een groot onderwerp en het kan in verschillende wegen getoond worden dus laat dat ook zien. Het in VR het kan met een applicatie of front-end ik vind dat gewoon heel erg leuk om te zien. Zoals producten leuker maken voor mensen en mensen daarbij ondersteunen. Want mijn persoonlijke doel is hoe kan ik mensen dan helpen met techniek en dan ook echt alle mensen dus daar denk ik vaak aan als ik iets ontwerp of probeer toe te lichten.

I: Dan gaan we naar vraag negen en dat is de rol van front-end communicatie en dat is eigenlijk hetzelfde antwoord zoals je eerder zei, focus niet enkel op dashboard maar ook met andere opties zoals licht of VR. Dat je op meerdere manieren je data kan overbrengen.

R: Ja precies en hetzelfde met API dat je het daar ook lekker makkelijk houdt en ook de opties geeft voor andere toepassing zoals graphQL.

I: oké dan gaan we naar vraag 10 en dat is de rol die de middleware vervult en wat zou je daarover kunnen zeggen?

R: Ja mijn ervaring met middleware en data science is niet zo denderend maar wat ik hoor van mijn collega's van infra dat ze het soms fijn vinden dat de keuzes in Cloud oplossingen al zijn gemaakt en welke tools ze mee zouden willen werken zoals Azure. Dat was wel twee jaar geleden maar ik hoop dat die wensen nog steeds zo zijn en snel die keuzes willen maken. Voor de rest dat je analyse, automate en communication doet is voor mij duidelijk hoe je dat wilt doen.

I: Gaan we naar de een na laatste vraag en de laatste vraag over het framework en dat is over je nog algemene feedback hebt over het framework, zie je het jezelf gebruiken bijvoorbeeld, is het goed genoeg als IoT framework.

R: Ja wat ik al eerder zei, ik begin wel beneden bij de sensoren, dus in mijn geval is dat, dat ik graag wil weten wat ik wil meten en wat zou ik nou eigenlijk willen maken. Daarom denk ik ook dat je de plaat misschien anders moet doen, dat je eerst begint bij de sensoren en daarna je weg naar

beneden internet toe werkt. Dus dat lijkt mij logischer eigenlijk. Voor de rest is het echt begrijpelijk en toen ik het voor het eerst zag. Alleen linksboven de internet benaming twijfelde ik nog even over of dat daar visualisatie in kan vallen.

I: Dan gaan we naar de laatste vraag en dat is eigenlijk over de principles het lijst hier.

R: Ja uhm, de principles zijn heel kort en duidelijk uitgelegd als ik er doorheen lees en als ik informatie zoek kan ik het ook zo heel gemakkelijk doornemen. Wat ik wel mis is een paar voorbeelden zoals waar moet ik aan denken als het bijvoorbeeld low-cost is.

I: Oke top, dit waren alle vragen dus hierbij stop ik de opname en hartelijk bedankt voor al je tijd en input.

XVIII. Interview framework validation final iteration respondent F

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
Het transparante en het simpele overzicht van een drie laag overzicht is goed uitlegbaar aan de beginner en kenner (Vinke, 2020).
2. Wat zou u kunnen zeggen over de compleetheid van het framework?
Voor het doel van partijen in een beginstadia bij elkaar brengen is dit framework zeer compleet (Vinke, 2020).
3. Wat zou u kunnen zeggen over de consistentie van het framework?
Door het gebruik van literatuur, empirische ervaringen en interviews vanuit het veld vind ik de consistentie niet het belangrijkste aspect, maar juist de toepasbaarheid. De consistentie binnen het framework wordt wel zoveel als kan meegenomen (Vinke, 2020).
4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
De toepasbaarheid van het framework zit zoveel mogelijk tegen de waarheid aan, maar helpt concreet bij het begrijpen van alle partijen welke betrokken moeten worden bij een IoT project (Vinke, 2020).
5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?
Als high-over solution architectuur plaat is dit referentiemodel erg geschikt. Het framework faciliteert enorm om de afhankelijkheden vooraf in kaart te brengen voor een IoT implementatie (Vinke, 2020).

XVIII. Interview framework validation final iteration respondent K

I: Welkom bij het interview voor de validatie van het framework, we gaan straks 5 variabelen die iets zeggen over hoe sterk framework eigenlijk is en hoe goed het framework eigenlijk is opgebouwd na al die iteraties. Dan gaan we nu naar vraag een en dat is wat zou je kunnen zeggen over de functionaliteiten die het framework biedt.

R: Eigenlijk om het heel algemeen te beschrijven het laat eigenlijk de belangrijkste kenmerken van een flow-based oplossing binnen IoT. Hierbij vind ik de communicatie tussen de verschillende lagen van het grootste belang vindt en dat wordt goed afgevangen. Daarbij zie ik dat je het publish/subscribe model, ik denk dat dit heel direct is en de gebruiker van het framework kan helpen met het vormgeven van de oplossing. Zeker omdat je met publish/subscribe eigenlijk op die manier heel snel en simpel kan abonneren op berichten en daardoor snel kan koppelen aan onderwerpen

I: Oke top dan gaan we door naar vraag twee en die gaat over de compleetheid van het framework, en eigenlijk moet dat zeggen of alle aspecten van IoT keurig zijn weergegeven in het framework en dus of het gehele plaatje wordt laten zien.

R: Wat ik vind over de compleetheid is dat het heel goed over een high-level perspectief zien hoe een IoT flow-based solution moet functioneren. Omdat het juist zo high-level is het eigenlijk vrij gemakkelijk te zien dat het framework compleet is. Je kunt in een opzicht zien dat de verschillende lagen van een IoT oplossing terugkomen. De belangrijkste componenten komen in deze drie lagen terug. Het framework is ook compleet in de details die het weergeeft in de bouwblokken om zo te helpen bij de implementatie.

I: Oké top dan gaan we door naar vraag drie en dat is, wat zou je kunnen zeggen over de consistentie van het framework. Dat is dan meer gebaseerd op van wat is het detail niveau en is die overall gelijk en wordt dezelfde terminologie overall toegepast en ook kwa oplossing dus niet ergens een oplossing geven met specifieke tool en ergens anders in het framework weer generaliseren. Wat zou je er dan over kunnen zeggen?

R: Eh, als je er op deze manier naar kijkt dus puur vanuit hoe alles in detail is weergegeven dan zie ik dat het niveau overall wel gelijk. Het is eigenlijk net zoals ik eerder zei dat het goed vanuit een high-level overview gemaakt en zo heb je het ook toegepast. Je zadelt de gebruiker van het framework niet met specifieke keuzes op in welke oplossing ze moeten gebruiken zoals Microsoft Azure als clouddienst.

I: Dan gaan we naar vraag vier en dat is stel jij als consultant wordt gevraagd een IoT oplossing te bouwen binnen een bedrijf hoe bruikbaar vind je vanuit deze optiek het framework. Omvat het framework alle aspecten om je eerste blauwdruk te kunnen creëren?

R: Ja en dan kom ik weer beetje terug op wat ik net gezegd heb. Opnieuw door de high-level approach is het heel generiek toepasbaar en kan daardoor ook voor veel specifieke doelen gebruikt worden. Het kan dan ook in meerdere stadia van een projecten gebruikt worden mochten er al aspecten van het framework aanwezig zijn in het bedrijf. Daarnaast is het gebruik van het framework gemakkelijk gemaakt doordat je zowel top-down als bottom-up het framework kan benaderen.

I: Dan gaan we alweer door naar de laatste vraag en dat is wat zou je kunnen zeggen over de structuur van het framework dus de architecturale structuur dus of alle componenten goed bij elkaar passen en de overloop logisch is.

R: Als ik naar jouw model kijk is het eerste wat bij mij opkomt een soort piramide waarbij de componenten elkaar logisch en in hiërarchische volgorde elkaar opvolgen waarbij boven de Things zitten en in de middelste en onderste lagen heb je het internet. Dus ik zou dan zeker zeggen dat de componenten in het framework elkaar logisch opgevlogen

I: Oké dat waren dan de vijf vragen over het framework, bedankt voor je antwoorden en je tijd en moeite. Dan stap ik hierbij de opname.

De vragen die gesteld gaan worden nemen elk van deze middelen door en verwachten een passend antwoord waarom het gepresenteerde framework wel of niet voldoet aan deze variabelen.

Vragen:

1. Wat zou u kunnen zeggen over de functionaliteiten van het framework?
 - a. Kijkend naar internet/things (plaatje is compleet). Detailniveau misschien hoger voor WSN-voorbeelden voor things/objects. Eventueel extra voorbeelden of keuzes toevoegen. Iets meer kenbaarheid geven aan privacy, security and trust binnen de building blocks. Hoe zal de gebruiker ermee omgaan en kan iedereen het gebruiken. Het zo jip en janneke duidelijk maken. Volgende stap zou zijn simpeler en begrijpbaarder maken. Zo kunnen meer mensen worden bereikt. Het doel duidelijk. Wat doe je met de data en welk doel wil je bereiken.
 - b. Niet veel developers die kennis hebben over IoT of daarmee oplossingen ontwikkelen.
 - c. Framework helpt met het begrijpen van het concept van IoT en hoe het te gebruiken. Dat is erg belangrijk bij een framework ook voor het front-end.
2. Wat zou u kunnen zeggen over de compleetheid van het framework
 - a. Hoe ga je om met de sensoren en schakelaars.
 - b. Bij de internet kant dat het goed gevisualiseerd wordt. Denk ook aan aspecten die het ook nog interessanter maakt bijv. led of licht.
 - c. De basic staan er
 - d. Het belang van mensen helpen met techniek
3. Wat zou u kunnen zeggen over de consistentie van het framework?
 - a. Niet afbakenen van specifieke protocollen.
 - b. Misschien enkel voor uitleggen of informeren als je voorbeelden wilt geven.
 - c. Uitdaging zo makkelijk mogelijk maken zonder specifiek keuzes vast te zetten.
 - d. Geef makkelijke voorbeelden bijv. WiFi in plaats van Zigbee of LoRa
 - e. Haal voorbeelden aan voor startende oplossingen
4. Wat zou u kunnen zeggen over de bruikbaarheid van het framework?
 - a. Framework is goed te visualiseren en klinkt ideaal als de nodige flexibiliteit er is.
 - b. Challenge is het creëren van een flexibel framework wat toch kan helpen bij het maken van de verschillende keuzes.
5. Wat zou u kunnen zeggen over de architecturale structuur van het framework?
 - a. De rol van Internet zou duidelijker kunnen (output). Niet meteen logisch er moet eerst worden gekeken naar things & middleware. De rest wel duidelijk.
6. Wat zou u kunnen zeggen over de rol van devices in het framework?
 - a. Micro/Micro processor onderscheid groter maken. Omdat ze inherent hetzelfde zijn. Misschien veroorzaakt het meer verwarring. Misschien voorbeeld geven aan eigen PCB/ontwerp. (buy/build)
7. Wat zou u kunnen zeggen over de rol van de communicatietechnieken in het framework?
 - a. Niets toe voegen eventueel Wi-Fi. (Begin of start met makkelijkere voorbeelden). BLT low-energy. Voeg herkenning toe aan het framework. Zodat het doel duidelijker wordt

8. Wat vindt u van de rol van de geleverde services?
 - a. Voorbeeld van de LED'jes om ook informatie weer te geven. Niet enkel de focus op dashboard. Iets meer mogelijkheden laten zien, basic is goed. Goede start om met IoT te starten en mee kunnen omgaan.
9. Wat vindt u van de rol van de front-end communicatie?
 - a. Niet enkel de focus op dashboard
10. Wat zou u kunnen zeggen van de rol die de middleware vervult?
 - a. Wat voor cloud/storage niet flexibel maken. Niet overal de keuzes in maken, misschien vaste keuze nemen.
11. Overall feedback over het framework

Hele goede basic, aangetrokken door sensoren en things aspect.
Logischer als het framework begint met Things en eindigt met Internet. Begin bij het aspect van meten. De benadering vanuit things is logischer.
12. Wat zou u kunnen zeggen over de beschreven principles?
 - a. Voorbeelden van sensoren voor temperatuur bijv.
 - b. Voorbeelden van communicatie technieken
 - c. Hiervoor kunnen ook situaties worden benoemd, voorbeelden om te visualiseren.
Idem dito voor data en visualisatie.
 - d. Duidelijk en kort dus dat is goed.

XVIII. Interview framework validation final iteration respondent L

Hallo Karsten,

Een algemene opmerking, aan de linker kant zie ik privacy, trust en security staan. Die zie ik alleen niet echt terugkomen in je plaatjes. Misschien kun je daar nog een accent opleggen.

De vragen:

1. De scheiding van de 3 lagen is goed om te maken. Wat ik interessant vind is dat je gaat van data naar informatie door middel van de middleware laag.
2. Het framework omvat vele soorten mogelijkheden IoT devices, communicatie standaarden en outputmogelijkheden. Ik zou graag nog zien dat er aangegeven wordt dat het framework bruikbaar is tot een bepaalde omvang van je systeem.
3. De consistentie is goed, het is ingezoomd op alle componenten die nodig zijn voor een dergelijk systeem, zonder te veel in de materie te gaan. Alles is op het niveau van 1 device/datastream gemaakt, wat het geheel goed begrijpbaar maakt.
4. Als een IT'er kan ik goed begrijpen welke componenten er in de basis nodig zijn om een IoT platform op te zetten. Je geeft al een aantal hints over de technologieën die soms gebruikt worden, waardoor je ook een beeld kan krijgen hoe het in de praktijk er ongeveer uit zal zien bij een praktische vertaling. Wat ik wel mis is soms de kanttekening dat er veel IoT apparaten bestaan met verschillende standaarden, en je misschien afhankelijk van je implementatie daar een keuze in moet maken/support moet voorzien voor meerdere technologieën naast elkaar.
5. Op een architecturniveau is dit een duidelijk houvast voor het daadwerkelijk opzetten van een systeem. Hier zal nog een technische vertaling gemaakt moeten worden maar dat kan zeker op basis van dit framework.

Met vriendelijke groet,

Bas de Jong

XVIII. Interview framework validation final iteration respondent E

Hi Karsten, Here's my quotes feedback:

- Functionalities
‘Clear distinction between different partitions & tasks of architecture. I enjoy seeing the data-based structure that chronologically flows from one functionality to another.’
- Completeness
‘The partitioning of the architecture into components makes for a clear and complete story, from initial sensor readings all the way to end-user dialogue.’
- Consistency
‘Through the usage of a consistent layout and flow annotation, the framework covers different parts of the solution in equal measure.’
- Usability
‘The framework is utilitarian in the sense that every part has its function and utility, and no space is wasted. It’s something you can pick up and start working with right away.’
- Architectural structure
‘Aligning framework with the sequences and flow of data makes for a functional and easy to read architectural structure. ’

Met vriendelijke groet / Kind regards,