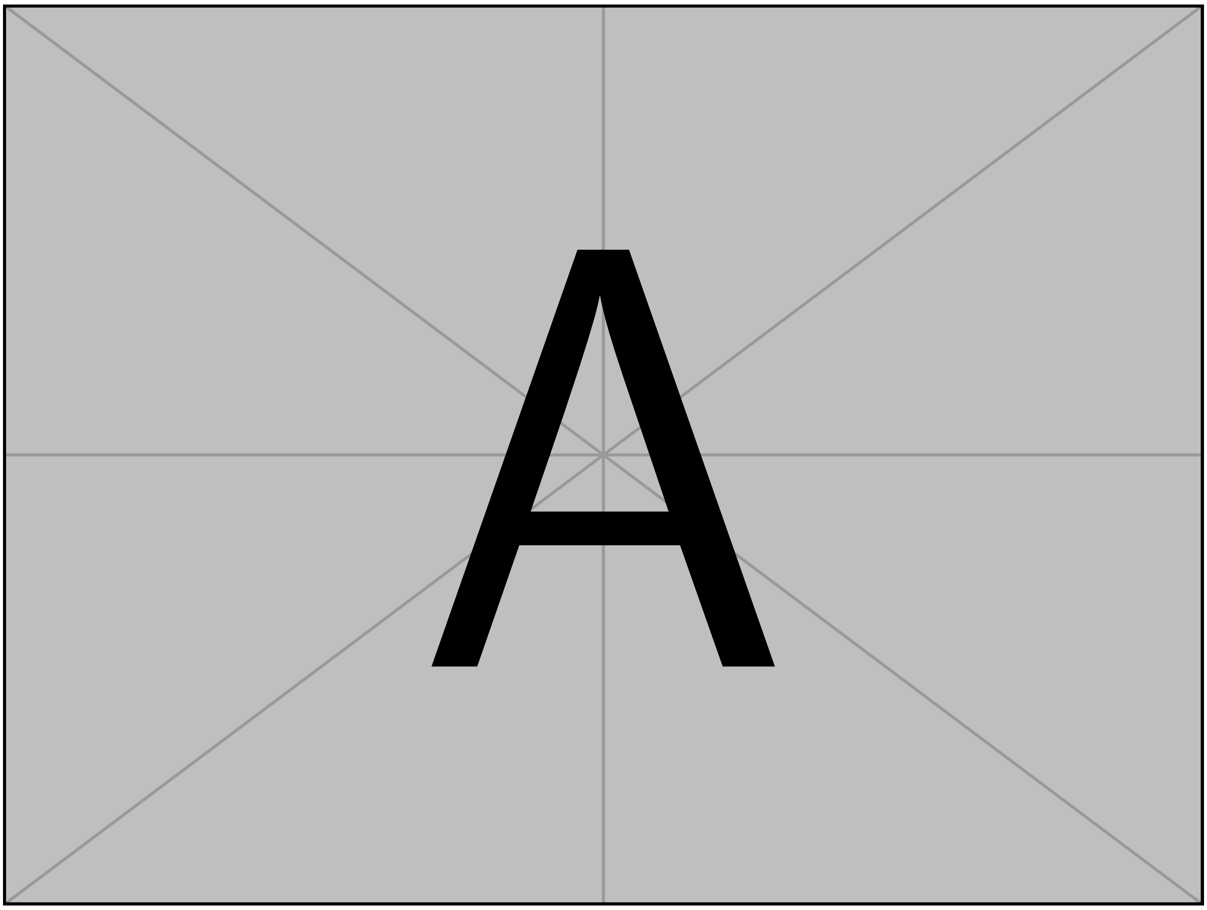
Embedded IoT for Eclipse Arrowhead



Albin Martinsson

Dept. of Computer Science and Electrical Engineering
Luleå University of Technology
Luleå, Sweden

Supervisor:

Jan van Deventer

To my dad Bengt-Göran Martinsson a special thanks for proof reading is required...

Abstract

The abstract is a mini thesis on its own. It should contain the briefest of motivation and problem description, what has been done, and summarize the results. The purpose is to give the reader a quick view of the content, and encourage the reader to read the rest of the thesis. It's also helpful to put the reader in the right frame of mind for the rest of the thesis. This section is typically anything between one paragraph for short research papers (of 8 pages) to a page for a full thesis

Contents

Acknowledgments

Here it is supposed to say something really smart about my thesis I think. (Maybe leave this part out)

Chapter 1

Introduction

1.1 Background

The number of devices connected to the Internet has risen from 7 billions in 2018 to an estimated 35 billions in 2021 according to Security Today and that number is only going to increase according to a survey done by security today. [1]

1.2 Motivation

With the numbers of devices connected to the Internet, IOT-devices from now on, rising to an estimated 38.6 billion devices world wide by 2025 the need to enable communication between those devices have never been bigger.

1.3 Problem definition

This project aims to investigate the possibilities, benefits and limitations of using Eclipse Arrowhead framework on embedded devices in contrast to commercially available solutions such as Amazons Amazon Web Services, AWS from now on, and Microsofts Azure.

The way to measure the difference between either using a central broker, the MQTT protocol used by AWS and Azure or using peer to peer, HTTP protocol used by the Arrowhead framework to handle the communication between devices in terms of latency, energy consumption and security.

1.4 Equality and ethics

The ability to own and control your data are becoming more and more rare these days with giant corporations establishing their own cloud services. You as a consumer always takes a risk when pushing sensitive data the a cloud owned by someone else, the right to own your data should not have to be infringed upon.

1.5 Sustainability

The use of small embeded devices instead of monolithic machines used by the industry today provides a much needed decrease in energy consumption.

1.6 Delimitations

This thesis will not cover a solution to the numerous secruity risks and issues associated with IoT-devices. This thesis will also only cover the three core systems of the Eclipse Arrowhead framework, which are the service registry, authorization and orchestrator.

1.7 Thesis structure

In chapter 2 related work is presented, a literature review of IoT, Industry 4.0, security and the Eclipse Arrowhead framework is conducted.

In chapter 3 theory is covered, describing what scientific methods where used in this thesis.

Chapter 4 covers implementation, describing how the different systems used in this thesis uses are designed from a software engineering perspective.

In chapter 5 an evalution of the experimet conducted will be performed.

Chapter 7 presents the conclusion of the work done in this thesis. The chapter also describes how to further investigate the questions raised in this thesis.

In chapter 8 there is a list of references used in this thesis.

Chapter 2

Related work

2.1 Related work

2.1.1 Internet of things

Rob van Kranenburg defines IoT as:

> a dynamic global network infrastructure with self-configuring capabilites based on standard and interoperable communcation protocols where physical and virutal 'Things' have identites, physical attributes, and virtual parsonalites and use intelligent interfaces, and are seamlessly integrated into the information netork.

2.1.2 Industry 4.0

Lasi [2] argues that the term industry 4.0 was coined beforehand as a planned fourth industruial revolution. The use of internet of things devices, IoT-devices from now on, and cyber physical systems, CPS from now on is what defines the fourth industrial revolution Vadiya means [3]. See figure x for a short historic overview of previous industrial revolutions.

According to Vadiya [3] industry 4.0 promotes the connection of sensors and devices both to the internet and to other sensors or devices.

Hozdić [4] states that a sensor is a device capable of providing an appropriate output in response to a measured value. One key feature of intelligent sensor is that in order to increase the level of information proeccesing it processes the information at a logical level Hozdić [4] argues. A Intelligent sensor is capable of executing actions based on the measured value in contrast to regular sensors, making them easier to set up and use means Hozdić [4].

Hozdić defines cyber physical system, CPS, as a new generation of system that integrate both physical and computer abilities. [4] A cyber physical system consists of two

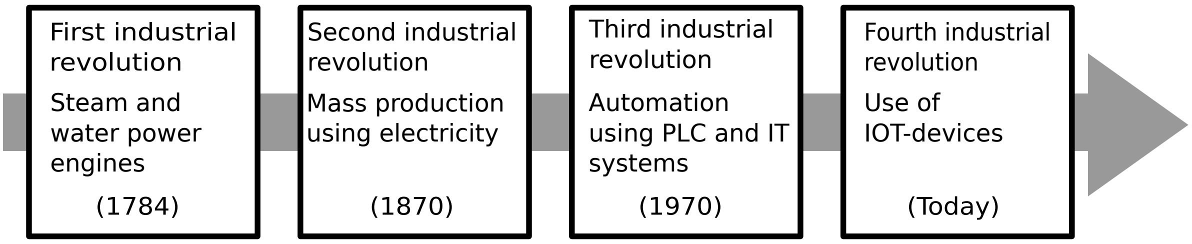

First industrial revolution
Steam and water power engines
(1784)
Second industrial revolution
Mass production using electricity
(1870)
Third industrial revolution
Automation using PLC and IT systems
(1970)
Fourth industrial revolution
Use of IOT-devices
(Today)

Figure 2.1: Historic overview of previous industrial revolutions

parts, one cybernetic and one physical. The cybernetic part of the system can be viewed as a summation of logic and sensor units while the physical part of the system can be viewed as a summation of the accuator units Hozdić adds. [4] Xu et. al. states that cyber physical systems is a key part of Industry 4.0. In contrast to the simple embedded systems of today will be exceeded due to advances in CPS that enables enhanced capability, scalability, adaptibility, resilency, safety, usability and secruity. [5] Hozdić argues that it is the CPS abilities to share and recieve information from intelligent sensors that connect to digital networks is what enables and form an internet of things. [4]

2.1.3 Arrowhead framework

A local cloud is defined as a self contained network with at least the thrre mandatory systems deployed, more on those in a later paragrapg. Delsing et.al. also argues that except the three mandatory core systems running a local cloud also needs at least one application system deployed [6].

Two terms have to be introduced in order to further understand what the Eclipse Arrowhead framework aims to accomplish, services and systems. Delsing et. al. defines a system as what is providing or consuming a service. [6] Furthermore a service is defined as what is used to convey information between a provider and a consumer Delsing et. al. argues [6]

The Eclipse Arrowhead framework, Arrowhead from now on, consists of three mandatory core systems according to Delsing et. al. [6] To fully operate a local cloud as defined in the previous section it must according to Delsing [6] contain:

- Service registry system.
- Authorization system.
- Orchestration system.

The service registry system is responsibly for enabling discovery and registring services Delsing et. al. [6] states. According to the Eclipse Arrowhead projects own github page [7] the service registry system provides the database which stores the offered services in the local cloud. The github page also states the three main objectives of the service registry system are:

- To allow application system to register available services to the database.
- Remove or update available services from the database.
- Allow application system to use the lookup functionality of the registry.

The Authorization system is contains two databases for keeping track on which system can consume services from which other system, depending on whether or not the Application system are in the same cloud or not according the projects github page [7] The github documentation also states that if the authorization happens within the same cloud it is called intra-cloud authorization and if it happens accross two local clouds it is called inter-cloud authorization. [7]

The Orchestration system is responsible for pairing and finding service providers and consumer Delsing et. al. declares. [6] Delsing et. al. continues to state that the orchestrator also stores the orchestration requirments and the resulting orchestration rules. [6] The projects documentation argues that the main objective of the orcherator system is to find an appropriate provider for the requesting consumer system. [7] The documentation also states that there are two types of orchestration, store orchestration and dynamic orchestration. Store orchestration uses the database orchestration store to find predefined orchestration rules. Dynamic orchestration on the other hand searches the entire local cloud, or even other clouds, in order to find mathing provider. [7]

2.1.4 Security

2.1.5 Communication

Chapter 3

Theory

3.1 Theory

Primarily this section should be about scientific methods and theories you need to evaluate/compare/invent to solve your problems from 1.3. In some cases it may be ok to describe different technologies, but the purpose is to describe something and then draw a conclusion from that. Example, if you decide to discuss different databases, it may be for the purpose of selecting the best type for your implementation later on (based on for example data representation, scalability, speed, etc.). Optimally the problems in 1.3 are not solved by anyone else yet, in which case this section needs to describe how to solve them (new algorithms, mathematical approaches, etc.).

This section can have a lot of subsections (3.1, 3.2, 3.3, etc).

Chapter 4

Implementation

4.1 Implementation

This is not a step-by-step instruction or diary to your work. Instead, you should describe your technical approach and solution, describe architecture, components, etc. Think software engineering... Perhaps use a few useful uml-diagrams or illustrate the system architecture. Keep in mind that the purpose of the implementation section is to describe your implementation to solve the problems from 1.3.

4.1.1 System architecture

4.1.2 System component

Chapter 5

Evaluation

5.1 Evaluation

Describe the test setup to verify that your problems from 1.3 have been solved. This can be done in different ways depending on focus of your problems. Some problems may purely objective, such as "improve the performance of X compared to Y". These are easy to evaluate since you simply need to compare the performance, and perhaps compare against a few more technologies that you have listed in Section 2 (related work). In other cases the problems may be very subjective, such as "Create a mobile app that can be used while driving, and which shows the most fuel efficient time to change gear". This problem will require a user-study in which several persons drive without the application, you calculate the fuel consumption, then they drive with the application and then you calculate the fuel consumption again. Then you collect the objective measurements (fuel consumption comparisons) and the subjective opinions from the users about whether the application was unobtrusive, usable, etc. (typically via a questionnaire)

Chapter 6

Discussion

6.1 Discussion

Discuss how you solved your problems from 1.3, and what the results were (from section 5). Describe alternative solutions, what you could have done differently, problems you encountered, how your results compare to other peoples' results, etc. Go through each problem individually, and then in the end add general remarks and discussion points which are outside the problems themselves but that you think may be valuable to share with the reader. This section can have several subsections.

Chapter 7

Conclusions and future work

7.1 Conclusions and future work

This section describes the outcome of your work and summarizes your efforts. It also outlines things that are left to do to reach a full solution, or to integrate your solution with something else.

Chapter 8

References

8.1 References

This section should be easy if you always write down your references the moment you read them / use them. There are, however, several acceptable ways of writing references. There are plenty of instructions online to help you get these correct. The typical format is: [1] Authors separated by comma, "Title in cursive and within citation marks", Place of publication, (for articles you also add issue, number, and pages as well), date.

instead of [x] you can also use [First Author (, et. al), year]

References

[1] The iot rundown for 2020: Stats, risks, and solutions – security today.

[2] Heiner Lasi, Peter Fettke, Hans Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business and Information Systems Engineering*, 6:239–242, 8 2014.

[3] Saurabh Vaidya, Prashant Ambad, and Santosh Bhosle. Industry 4.0 - a glimpse. volume 20, pages 233–238. Elsevier B.V., 1 2018.

[4] Elvis Hozdić. Smart factory for industry 4.0: A review, 2015.

[5] Li Da Xu, Eric L. Xu, and Ling Li. Industry 4.0: State of the art and future trends. *International Journal of Production Research*, 56:2941–2962, 2018.

[6] Delsing Jerker. *IoT Automation: Arrowhead Framework - 1st Edition - Jerker Delsing* -. 2017.

[7] eclipse-arrowhead/core-java-spring.