Aalto University
School of Science
Degree Programme in ICT innovation

Relja Paunovic

Contracting Service For Industrial Internet

Re-inventing the Wheel

Master's Thesis
Espoo, June 18, 2011

DRAFT! — July 9, 2017 — DRAFT!

Supervisor:	Professor Petri Vuorimaa, Aalto University
Advisor:	Instructor?

Aalto University
School of Science
Degree Programme in ICT innovation

ABSTRACT OF
MASTER'S THESIS

Author:	Relja Paunovic		
Title:	Contracting Service For Industrial Internet Re-inventing the Wheel		
Date:	June 18, 2011	**Pages:**	vi + 23
Major:	Hypermedia	**Code:**	T-110
Supervisor:	Professor Petri Vuorimaa		
Advisor:	Instructor?		

A dissertation or thesis is a document submitted in support of candidature for a degree or professional qualification presenting the author's research and findings. In some countries/universities, the word thesis or a cognate is used as part of a bachelor's or master's course, while dissertation is normally applied to a doctorate, whilst, in others, the reverse is true.

!FIXME **Abstract text goes here (and this is an example how to use fixme).** FIXME! Fixme is a command that helps you identify parts of your thesis that still require some work. When compiled in the custom `mydraft` mode, text parts tagged with fixmes are shown in bold and with fixme tags around them. When compiled in normal mode, the fixme-tagged text is shown normally (without special formatting). The draft mode also causes the "Draft" text to appear on the front page, alongside with the document compilation date. The custom `mydraft` mode is selected by the `mydraft` option given for the package `aalto-thesis`, near the top of the `thesis-example.tex` file.

The thesis example file (`thesis-example.tex`), all the chapter content files (`1introduction.tex` and so on), and the Aalto style file (`aalto-thesis.sty`) are commented with explanations on how the Aalto thesis works. The files also contain some examples on how to customize various details of the thesis layout, and of course the example text works as an example in itself. Please read the comments and the example text; that should get you well on your way!

Keywords: ocean, sea, marine, ocean mammal, marine mammal, whales, cetaceans, dolphins, porpoises

Language: English

Acknowledgements

I wish to thank all students who use LaTeX for formatting their theses, because theses formatted with LaTeX are just so nice.

Thank you, and keep up the good work!

Espoo, June 18, 2011

Relja Paunovic

Abbreviations and Acronyms

IOT	Internet Of Things
IIOT	Industrial Internet of Things
ETSI	European Telecommunications Standards Institute
M2M	Machine To Machine
GSCM2MTF	Global Standards Collaboration Machine-to-Machine Task Force (GSCM2MTF)
3GPP	3rd Generation Partnership Project
ETSI	European Telecommunications Standards Institute
REST	Representational State Transfer
SCL	RESTful Service Capability Layer
OMA	Open Mobile Alliance
LWM2M	Light Weight Machine-to-Machine
CoAP	Constrained Application Protocol
DoS	Denial of Service

Contents

Chapter 1

Introduction

This is my master's thesis, and I am very proud of it. Of course, when I write my *real* master's thesis, I will not use the singular pronoun *I*, but rather try to avoid referring to myself and speak of the research *we* have conducted—I rarely work alone, after all. Yet, both *I* and *we* are correct, and it depends on the instructor and the supervisor (of course from you, too), which one they would prefer. Anyway, the tense should be active, and passive sentenses should be avoided (especially, writing sentences where the subject is presented with by preposition), so often you cannot avoid choosing between the pronouns. Life is strange, but there you have it.

By the way, the preferred order of writing your master's thesis is about the same as the outline of the thesis: you first discover your problem and write about that, then you find out what methods you should use and write about that. Then you do your implementation, and document that, and so on. However, the abstract and introduction are often easiest to write last. This is because these really cover the entire thesis, and there is no way you could know what to put in your abstract before you have actually done your implementation and evaluation. Rarely anyone write the thesis from the beginning to the end just one time, but the writing is more like process, where every piece of text is written at least twice. Be also prepared to delete your own text. In the first phase, you can hide it into comments that are started with % but during the writing, the many comments should be visible for your helpers, the instructor and supervisor.

The introduction in itself is rarely very long; two to five pages often suffice.

Problem statement

Undergraduate students studying technical subjects do not consider typography very interesting these days, and therefore the typographical quality of many theses is unacceptably low. We plan to rectify this situation somewhat by providing a decent-quality example thesis outline for students. We expect that the typographical quality of the master's theses will dramatically increase as the new thesis outline is taken into use.

Helpful hints

Read the information from the university master's thesis pages [**?**] before starting the thesis. You should also go through the thesis grading instructions [**?**] together with your instructor and/or supervisor in the beginning of your work.

Structure of the Thesis

You should use transition in your text, meaning that you should help the reader follow the thesis outline. Here, you tell what will be in each chapter of your thesis.

Chapter 2

Background

!Fixme **connect with introduction, write later** Fixme!

Internet of Things

Internet of things (IOT) is a relatively new concept gaining momentum since the start of this century. Although, first examples of IOT date back as early as 1983 with an automated inventory system. IOT can be seen as an extension to Internet, with physical devices (such as sensors and actuators) communicating with each other and with humans creating an enormous network.

Nowadays, with the continuous decrease in cost of computational devices we are able to produce powerful devices with communication abilities for a very low price. With the increase in number of devices that needs to be connected (some estimates show up to 75 billion connected devices by year 2025[1]) issues regarding scalability, security, heterogeneity of devices, etc. have emerged.

These issues are holding back the progress of IOT because it forces companies to create their own proprietary systems to fit their needs, which is only an option for large companies with financial capabilities to do so. This leads to large number of systems designed for similar purposes but incompatible with each other (due to use of different protocols or architectures). Temporary solution for this problem is to introduce a middle-ware as proposed by [2] that acts like a bridge between two systems, but this has scalability issues and with the rise of number of connected devices will soon be unacceptable. In order to tackle these issues several standards have been proposed, most

[1]https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

notably smartM2M, oneM2M and most recently LightweightM2M (LWM2M) which will be explained in 2.1.2.1. None of these have yet became *de facto* standard.

In the remainder of this section, differences between Industrial Internet of Things (IIOT) and IOT will be introduced, along with a concrete example included in appendix A. Following, open issues of IOT and IIOT will be analyzed in 2.1.2.

Industrial Internet of Things

In IOT, a rough distinction is made between consumer and industrial IOT according to [2]. Consumer IOT applications are aimed to make everyday life easier by saving time and money, such as smart locks, smart homes and wearable hearth monitors. On the other hand, industrial IOT (such as production, automation and intelligent computation systems) focuses on how smart machines, data analytics and networked sensors can improve services in business-to-business domain [11]. As an example, predictive maintenance can generate savings up to 12% over scheduled repairs, leading to a 30% reduction in maintenance cost and a 70% cut in downtime from equipment breakdowns according to Accenture[2] !FIXME **Should I include that research in references? It is not a "standard" research paper, more like slide show** FIXME!.This usually implies extensive machine-to-machine (M2M) communication compared to consumer IOT, where in most cases real time guarantees are not required.

Generally, IIOT have stricter requirements regarding delay, security and general robustness compared to consumer IOT. This is because failures in these devices can have consequences on safety of people and environment. For example, in a factory setting, pressure sensor installed on an indoor crane can cause serious damage by failing to communicate about an obstruction.

Open Issues

!FIXME **You can mention role of midleware here, maybe?** FIXME! As previously mentioned, there are many issues surrounding IOT, most notably, lack of standards and security. These issues will be explained in the remainder of the section.

[2]https://www.accenture.com/us-en/insight-industrial-internet-of-things

Standardization

According to Global Standards Collaboration Machine-to-Machine Task Force (GSCM2MTF) there are more than 140 organizations involved in the M2M standardization process worldwide. This is a huge vertical fragmentation of IOT market, and it is a result of long history in Industrial use, starting from seventies with process control systems that continued to be used in todays process automation systems. As already mentioned, these systems are proprietary and are incompatible with each other.

In an attempt to resolve this fragmentation issue three notable initiatives stand out. SmartM2M is an initiative led by European Telecommunications Standards Institute (ETSI), it is based on RESTful Service Capability Layer (SCL) [1] which is available through open interfaces. Resource tree residing on SCL along with procedures for handling them is standardized following Representational State Transfer (REST) principles allowing technology agnostic way of accessing them. SmartM2M also defines security framework including authentication, M2M service bootstrap, key agreement and establishment, and M2M service connection procedures, based on a key hierarchy of the M2M node[6]. Unfortunately, it may have issues with scalability as pointed out by [7], thus, making it unsuitable for IOT and IIOT.

A follow up project was formed in order to resolve issues with SmartM2M, this time with a broader partnership. It is an international project started by seven telecom standards organizations: Association of Radio Industries and Businesses (ARIB) and Telecommunication Technology Committee (TTC), Japan; the Alliance for Telecommunications Industry Solutions (ATIS) and Telecommunications Industry Association (TIA), United States; the China Communications Standards Association (CCSA), China; the European Telecommunications Standards Institute (ETSI), Europe; and the Telecommunications Technology Association (TTA), Korea. This project is based on RESTfull design, same as SmartM2M, resource naming conventions same as SmartM2M and is grounded on horizontal service layer principle. Although, it relaxes the scalability constraints by using hierarchical organization of different actors in the system. With this improvement with respect to SmartM2M, it is a serious contester to became IOT standard, being already adopted by various companies according to [12]. Along with these improvements, it defines a security architecture in three layers: security functions, security environment abstraction and security environment.

Recently, a new standard has emerged from Open Mobile Alliance (OMA) targeting constrained devices named Light Weight Machine-to-Machine (LWM2M). It defines a fast deployable client-server specification while minimizing memory consumption and network overhead making it very appealing to IOT and

IIOT devices. It provides device management and security work-flow for IOT applications in a very light weight manner. Recent results from [13] show that memory footprint overheads on a client side protocol stack are no more than 6-9%. Detailed description of LWM2M will be given in 2.2 since this work is based on it.

Security and privacy

Devices in IOT generate, process and exchange vast amount of data that is safety-critical and/or private and they are subject to various attacks. Therefore, it is crucial to assure integrity of the devices code and data from malicious modifications [4]. In IIOT following two requirements are crucial for security according to [15]. Availability is most important requirement, because it can lead to loss in productivity and consequently loss of revenue, this is particularly affected by denial of service (DoS) attacks and preventing any system failure that may result in physical damage or harm to humans, particularly affected by sabotage.

There are many security architectures for embedded IOT devices, although, majority of them are too complex for low-end devices. Solutions for low-end devices usually rely on physical (hardware) isolation of security-critical code and data from other software on same device. Examples of such architectures are SMART [5], SPM [14], SANCTUS [10] and TrustLite [9] but they all have major flaws. SMART does not allow code changes after deployment, SPM and SANCTUS have hardware assisted task isolation but they are non-interruptible which violates real-time guarantees and TrustLite requires all software components to be loaded at boot time which reduces flexibility. TyTAN [3] is the only work that provides secure loading of tasks at runtime, secure inter-process communication, local and remote attestation and real-time guarantees.

Light Weight Machine to Machine (LWM2M)

OMA has approved first version (V1.0) of LWM2M standard in February 2017, since it is a very recent specification not many implementations exist, although, number has been steadily increasing since then. Most notable implementation by [13] focuses on client side architecture (residing on IOT devices).

LWM2M provides a light and secure communication interface with compact data model to enable device management and service for IOT (and IIOT) devices. It is a client-server architecture named OMA LWM2M Enabler. Enabler consists of LWM2M Server which is a central point that takes care of the devices, assigning security keys, registering device capabilities and more. Other part of Enabler is LWM2M Client which resides on a device and provides necessary information to the Server. Furthermore, it uses Constrained Application Protocol (COAP), instead of HTTP, which has a greatly reduced communication overhead making it ideal for constrained devices (or devices benefiting from efficient communication, as the case in IIOT) along with UDP/SMS transport bindings. Architecture of LWM2M is shown on Figure 2.1.

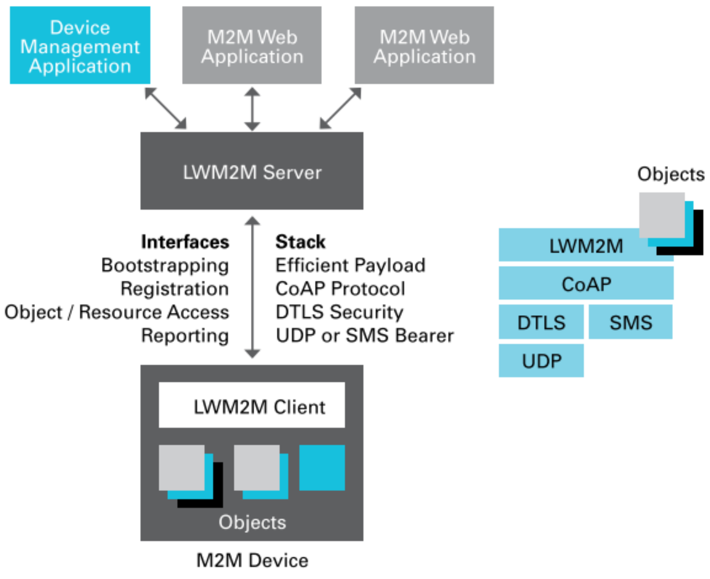


Figure 2.1: LWM2M Architecture

The LWM2M Enabler defines the application layer communication protocol between Server and Client. It is separated into four logical interfaces, namely, Bootstrap, Device Discovery and Registration, Device Management and Service Enablement, and Information Reporting. Diagram showing interfaces and corresponding messages is illustrated in 2.2.

1. Bootstrap: allows LWM2M Bootstrap server to manage keying, access control and to configure device for communication with the Server
2. Device Discovery and Registration: allows Server to discover devices and register their capabilities, e.g. which objects and how to access them does a device have.

3. Device Management and Service Enablement: allows LWM2M Server to manage devices and provide M2M service by sending operations to devices and getting corresponding responses from them
4. Information Reporting: This is a core interface in LWM2M which allows reporting of resource information. It can be triggered periodically, on events or on request

Communication model of LWM2M is based on CoAP methods similar to HTTP verbs GET, POST, PUT, DELETE to manipulate *resources* on devices. Compared to HTTP, CoAP starts with only four bytes of overhead in binary encoded message and is easily translatable to HTTP. Unlike HTTP, CoAP messages are exchanged asynchronously between CoAP end-points over a datagram-oriented transport, in this case UDP.

In LWM2M Enabler, each individually addressable piece of information is called a Resource and groups of Resources are logically organized into Objects. For example, predefined object *Location* contains all resources needed for locating the device. Full list of existing objects can be found at OMA registry [3].

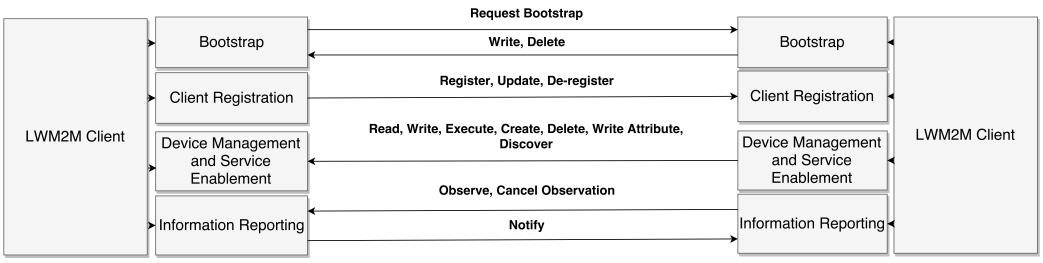

LWM2M Client
Bootstrap
Client Registration
Device Management and Service Enablement
Information Reporting
Request Bootstrap
Write, Delete
Register, Update, De-register
Read, Write, Execute, Create, Delete, Write Attribute, Discover
Observe, Cancel Observation
Notify
Bootstrap
Client Registration
Device Management and Service Enablement
Information Reporting
LWM2M Client

Figure 2.2: LWM2M Interfaces

Policy Based Communications for 5G Mobile with Customer Edge Switching

Particularly interesting work by [8] proposes a policy based communication with built in security, while also addressing classical weaknesses of Internet, namely, address spoofing, unwanted traffic and DoS attacks. It is based on a principle that before establishing communication between two hosts (or

[3]http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html

networks) they need to negotiate interests, and only if they are matching communication is established. These interests are described with a policy.

This work proposes to replace Network Address Translator (NAT) from the edge of the network with their own Customer Edge Switch (CES) node. This node will act exactly like NAT if the sender, who is behind a CES node, is interacting with receiver using legacy ip. On the other hand, if the situation is reversed CES node will act as a *realm gateway*. Only if both actors are behind CES node it will act as a cooperative firewall negotiating interests of sender and receiver. Because of this, CES can be applied one network at a time making it suitable for IIOT purposes where vertical fragmentation is a big issue.

Furthermore, CES allows efficient communication by dropping unwanted traffic at the edge and in that way reducing amount of traffic that passes trough network. Also, CES makes use of Domain Name Servers (DNS) to find receivers faster using Fully Qualified Domain Names (FQDN) and MS-ISDN numbers.

In essence, policy database holds all policies and can be accessed through API. Before communication is established, policies of both actors are checked and if they match communication is allowed. Example is shown on 2.3.

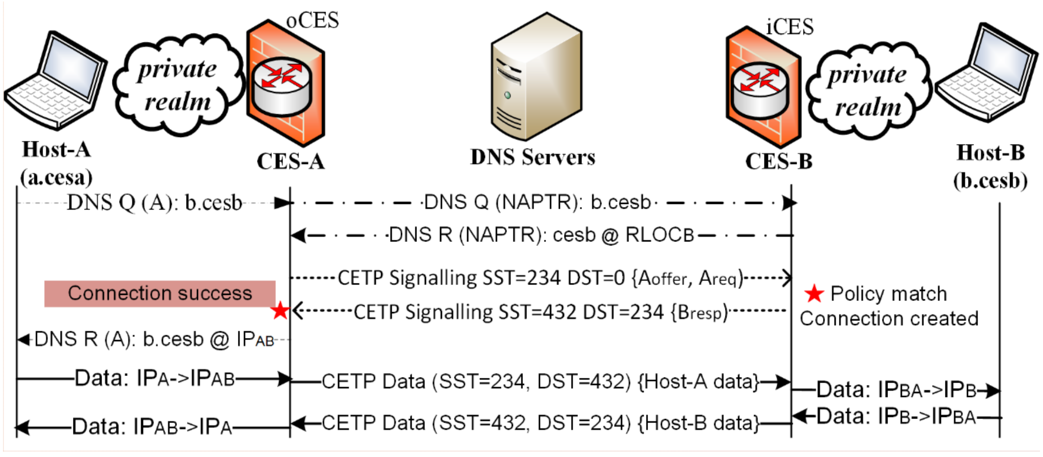


Figure 2.3: CES communication example

Chapter 3

Environment

A problem instance is rarely totally independent of its environment. Most often you need to describe the environment you work in, what limits there are and so on. This is a good place to do that. First we tell you about the LaTeX working environments and then is an example from an thesis written some years ago.

LaTeX working environments

To create LaTeX documents you need two things: a LaTeX environment for compiling your documents and a text editor for writing them.

Environment

Fortunately LaTeX can nowadays be found for any (modern) computer environment, be it Linux, Windows, or Macintosh. For Linuxes (and other Unix clones) and Macs, I'd recommend *TeX Live* [**?**], which is the current default LaTeX distribution for many Linux flavors such as Fedora, Debian, Ubuntu, and Gentoo. TeX Live is the replacement for the older *teTeX*, which is no longer developed.

TeX Live works also for Windows machines (at least according to their web site); however, I have used *MiKTeX* [**?**] and can recommend it for Windows. MiKTeX has a nice package manager and automatically fetches missing packages for you.

Editor

You can write LaTeX documents with any text editor you like, but having syntax coloring options and such really helps a lot. My personal favourite for editing LaTeX is the *TeXlipse* [?] plugin for the Eclipse IDE [?]. Eclipse is an open-source integrated development environment (IDE) initially created for writing Java code, but it currently has support for editing languages such as C, C++, JavaScript, XML, HTML, and many more. The TeXlipse plugin allows you to edit and compile LaTeX documents directly in Eclipse, and compilation errors and warnings are shown in the Eclipse *Problems* dialog so that you can locate and fix the issues easily. The plugin also supports reference traversal so that you can locate the source line where a label or a citation is defined.

Eclipse is an entire development environment, so it may feel a bit heavy-weight for editing a document. If you are looking for a more light-weight option, check out TeXworks. TeXworks is a LaTeX editor that is packaged with the newer MiKTeX distributions, and it can be acquired from `http://www.tug.org/texworks/`.

And if you are attached to your *emacs* or *vim* editor, you can of course edit your LaTeX documents with them. Emacs at least has syntax coloring and you can compile your document with a key binding, so this may be a good option if you prefer working with the standard Linux text editors.

Graphics

When you use `pdflatex` to render your thesis, you can include PDF images directly, as shown by Figure 3.1 below.

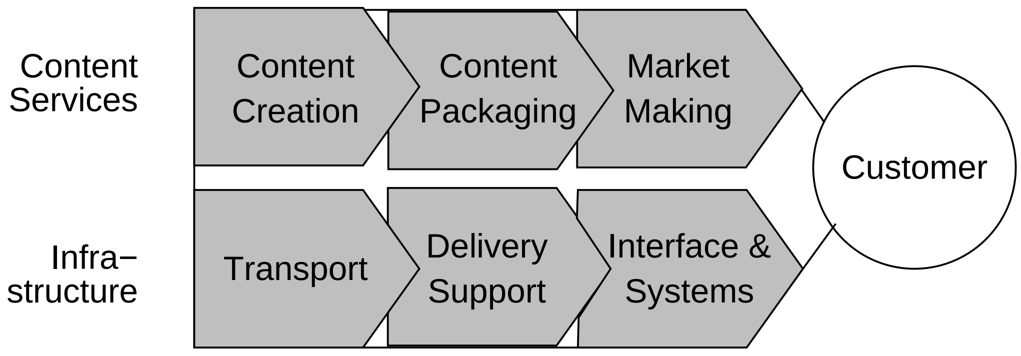


Figure 3.1: The INDICA two-layered value chain model.

You can also include JPEG or PNG files, as shown by Figure 3.2.

You can create PDF files out of practically anything. In Windows, you can download PrimoPDF or CutePDF (or some such) and install a printing

Figure 3.2: Eeyore, or Ihaa, a very sad donkey.

driver so that you can print directly to PDF files from any application. There are also tools that allow you to upload documents in common file formats and convert them to the PDF format. If you have PS or EPS files, you can use the tools `ps2pdf` or `epspdf` to convert your PS and EPS files to PDF.

Furthermore, most newer editor programs allow you to save directly to the PDF format. For vector editing, you could try Inkscape, which is a new open source WYSIWYG vector editor that allows you to save directly to PDF. For graphs, either export/print your graphs from OpenOffice Calc/Microsoft Excel to PDF format, and then add them; or use `gnuplot`, which can create PDF files directly (at least the new versions can). The terminal type is *pdf*, so the first line of your plot file should be something like `set term pdf ...`.

To get the most professional-looking graphics, you can encode them using the TikZ package (TikZ is a frontend for the PGF graphics formatting system). You can create practically any kind of technical images with TikZ, but it has a rather steep learning curve. Locate the manual (`pgfmanual.pdf`) from your LaTeX distribution and check it out. An example of TikZ-generated graphics is shown in Figure 3.3.

Another example of graphics created with TikZ is shown in Figure 3.4. These show how graphs can be drawn and labeled. You can consult the example images and the PGF manual for more examples of what kinds figures you can draw with TikZ.

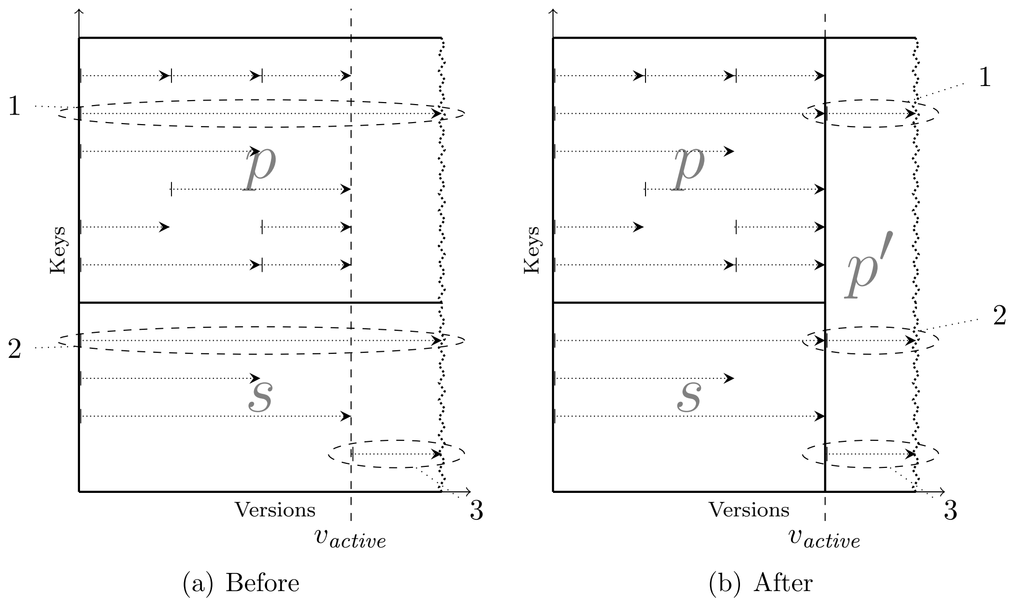
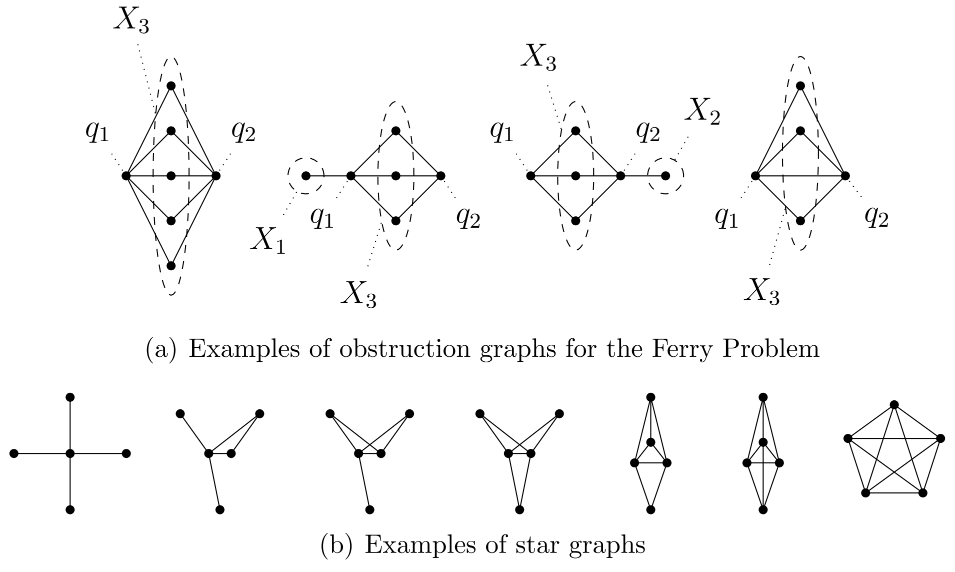


Figure 3.3: Example of a multiversion database page merge. This figure has been taken from the PhD thesis of Haapasalo [?].



X_3
q_1
q_2
X_1
X_2
(a) Examples of obstruction graphs for the Ferry Problem
(b) Examples of star graphs

Figure 3.4: Examples of graphs draw with TikZ. These figures have been taken from a course report for the graph theory course [?].

Chapter 4

Methods

You have now stated your problem, and you are ready to do something about it! *How* are you going to do that? What methods do you use? You also need to review existing literature to justify your choices, meaning that why you have chosen the method to be applied in your work.

If you have not yet done any (real) metholodogical courses (but chosen introduction courses of different areas that are listed in the methodological courses list), now is the time to do so or at least check through material of suitable methodological courses. Good methodologial courses that consentrates especially to methods are presented in Table 4.1. Remember to explain the content of the tables (as with figures). In the table, the last column gives the research area where the methods are often used. Here we used table to give an example of tables. Abbreviations and Acronyms is also a long table. The difference is that longtables can continue to next page.

Code	**Name**	**Methods**	**Area**
T-110.6130	Systems Engineering for Data Communications Software	Computer simulations, mathematical modeling, experimental research, data analysis, and network service business research methods, (agile method)	T-110
Mat-2.3170 Simulation (here is an example of multicolumn for tables)		Details of how to build simulations	T-110
S-38.3184	Network Traffic Measurements and Analysis	How to measure and analyse network traffic	T-110

Table 4.1: Research methodology courses

Chapter 5

Implementation

You have now explained how you are going to tackle your problem. Go do that now! Come back when the problem is solved!

Now, how did you solve the problem? Explain how you implemented your solution, be it a software component, a custom-made FPGA, a fried jelly bean, or whatever. Describe the problems you encountered with your implementation work.

Chapter 6

Evaluation

You have done your work, but that's[1] not enough.

You also need to evaluate how well your implementation works. The nature of the evaluation depends on your problem, your method, and your implementation that are all described in the thesis before this chapter. If you have created a program for exact-text matching, then you measure how long it takes for your implementation to search for different patterns, and compare it against the implementation that was used before. If you have designed a process for managing software projects, you perhaps interview people working with a waterfall-style management process, have them adapt your management process, and interview them again after they have worked with your process for some time. See what's changed.

The important thing is that you can evaluate your success somehow. Remember that you do not have to succeed in making something spectacular; a total implementation failure may still give grounds for a very good master's thesis—if you can analyze what went wrong and what should have been done.

[1]By the way, do *not* use shorthands like this in your text! It is not professional! Always write out all the words: “that is”.

Chapter 7

Discussion

At this point, you will have some insightful thoughts on your implementation and you may have ideas on what could be done in the future. This chapter is a good place to discuss your thesis as a whole and to show your professor that you have really understood some non-trivial aspects of the methods you used...

Chapter 8

Conclusions

Time to wrap it up! Write down the most important findings from your work. Like the introduction, this chapter is not very long. Two to four pages might be a good limit.

Bibliography

[1] Alaya, M. B., Banouar, Y., Monteil, T., Chassot, C., and Drira, K. OM2M : Extensible ETSI-compliant M2M service platform with self-configuration capability. *Procedia - Procedia Computer Science 32* (2014), 1079–1086.

[2] Bandyopadhyay, D., and Sen, J. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications 58*, 1 (2011), 49–69.

[3] Brasser, F., Mahjoub, B. E., Sadeghi, A.-r., Wachsmann, C., and Koeberl, P. TyTAN: Tiny Trust Anchor for Tiny Devices.

[4] Control, E., and Security, S. Detecting Industrial Control Malware Using Automated PLC Code Analytics.

[5] Defrawy, K. E., Perito, D., and Tsudik, G. SMART : Secure and Minimal Architecture for (Establishing a Dynamic) Root of Trust.

[6] ETSI. Machine-to-machine communications (m2m); functional architecture, 2013.

[7] Grieco, L. A., Alaya, M. B., Monteil, T., Drira, K., and Bari, P. Architecting Information Centric ETSI-M2M systems. 211–214.

[8] Kantola, R., and Beijar, N. Policy Based Communications for 5G Mobile with Customer Edge Switching.

[9] Koeberl, P., Schulz, S., and Sadeghi, A.-r. TrustLite : A Security Architecture for Tiny Embedded Devices.

[10] Noorman, J., Agten, P., Daniels, W., and Strackx, R. Sancus : Low-cost trustworthy extensible networked devices with a zero-software Trusted Computing Base.

[11] Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., and Ladid, L. Internet of Things in the 5G Era: Enablers, Architecture, and Business Models. 510–527.

[12] Park, H., Kim, H., Joo, H., and Song, J. Recent advancements in the Internet-of-Things related standards : A oneM2M perspective. *ICT Express 2*, 3 (2016), 126–129.

[13] Rao, S., Chendanda, D., Deshpande, C., and Lakkundi, V. Implementing LWM2M in Constrained IoT Devices. 52–57.

[14] Raoul Strackx, Frank Piessens, B. P. Efficient isolation of trusted subsystems in embedded systems.

[15] Sadeghi, A., Wachsmann, C., and Waidner, M. Security and privacy challenges in industrial internet of things. *Proceedings of the 52nd* (2015).

Appendix A

KoneCranes Use Case

KoneCranes have donated a smart Crane to Aalto Industrial Campus. This Crane has many sensors attached to make it automated and smart as possible. These sensors bring many features, namely **load weighting** (which can also be used to detect whether the crane is stuck somewhere), **remote monitoring** of the position of the crane, **signaling** when some error or warning has occurred, **live video feed** from camera attached above the hook (at the moment, not used for automation but could be supported in the future, for example, image processing to detect obstructions) and more.

There are multiple actors in this ecosystem and all of them require some of the data that Crane generates, although not all information should be available to them, only the bare minimum that is required by their business. These actors are KoneCranes, maintenance service (part of the KoneCranes) and users of the crane. Currently, KoneCranes has access to all the data Crane generates, but in order to increase security, this should not be the case.

Maintenance

Maintenance service makes use of usage data (alarms, usage parameters) to monitor and predict maintenance needs of the cranes. They also use the data together with the customer, for example, to review their maintenance spend of the assets, study patterns to reveal relationships between variables, etc.

For example, part of the crane that needs to be changed most often are the breaks. Breaks have limited number of uses and this number is approximately known (there is a regulation in place that requires brakes to be changed regularly). Information like, how many times has the Crane been moved from idle state, should be available to maintenance service so they know when do they need to change the breaks (or some other part of the Crane). Crane smart sensors can also detect some irregularities and issue a warning (or error), which is useful information for the maintenance companies.

Information like this should be disclosed to the maintenance service, while restricting the access to, for example, position of the crane in a certain point in time, or any kind of data that can be used to infer the processes inside the factory.

Users of the Crane

Inside a factory there are different actors with different privileges. Worker that operates Crane does not need to know the history of the movements of the Crane, or condition of the brakes. Thus, he should only have access to information he needs, e.g. from where to where the load should be moved. Manager of those workers on the other hand should be able to monitor all the Cranes in the factory.

In the future, machine to machine communication will be utilised much more, for example, in a setting with two Cranes operating (automated, not by human) in the same room they should be able to signal to each other with their planned path in order to avoid collisions and to optimize the process. In this way, need for a centralized control is eliminated (or at least minimized).

KoneCranes

KoneCranes also wants to access some of the data that Crane generates, in order to make product improvements, react to possible reliability problems and get better specification for new product generations, adjust warranties, etc.. They analyze all types of data that Crane generates: manufacturing data (component lists, manufacturing dates, etc.), usage (starts, lifting hours, loads, etc.) and sensor data (vibrations, temperature, etc.), and maintenance data (maintenance task history, ordered materials, etc.).

At the moment KoneCranes has access to all the data Crane generates and their customers are aware of that. For privacy reasons, and in environment with multiple manufacturers, data access should be restricted to only what is needed for a specific role.