UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Towards an architecture design for a future societal energy supply system

Nico Korthout

Master of Science Thesis
April 2017

Graduation committee:
prof.dr.ir. Mehmet Akşit (University of Twente)
dr. Maya Daneva (University of Twente)
dipl.ing. Jan-Peter Doomernik (Enexis)
drs. Gert Jan Timmerman (Info Support)
Laila van de Hoef, MSc (Info Support)

University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

The electricity supply system in the Netherlands has already been operational for more than a century. Its architecture has stayed mostly the same throughout several of the last decades. Meanwhile, the context in which the system operates has changed dramatically. Today, systems oftentimes are designed with user-centered approaches to satisfy the user's needs as much as possible. A societal system, like the electricity supply system, should in turn be designed with a society-centred approach to satisfy society's needs as much as possible. This master graduation thesis deals with this topic.

This thesis contributes the following: (1) an analysis of whether the current electricity supply system satisfies the needs of society as much as possible; (2) requirement specifications of a societal energy supply system; and (3) a feasibility analysis of a societal energy supply system from a high-level engineering perspective using the architecture synthesis process (Tekinerdoğan, 2000).

We conclude that an architecture for a future societal energy supply system can be synthesised, but additional research activities are needed to deal with the remaining unresolved technical problems.

“ *I can’t do this, Sam.*

I know. It’s all wrong. By rights we shouldn’t even be here. But we are. It’s like in the great stories, Mr. Frodo. The ones that really mattered. Full of darkness and danger, they were. And sometimes you didn’t want to know the end. Because how could the end be happy? How could the world go back to the way it was when so much bad had happened? But in the end, it’s only a passing thing, this shadow. Even darkness must pass. A new day will come. And when the sun shines it will shine out the clearer. Those were the stories that stayed with you. That meant something, even if you were too small to understand why. But I think, Mr. Frodo, I do understand. I know now. Folk in those stories had lots of chances of turning back, only they didn’t. They kept going. Because they were holding on to something.

What are we holding onto, Sam?

That there’s some good in this world, Mr. Frodo... and it’s worth fighting for.

”

Samwise Gamgee and Frodo Baggings, *The Lord of the Rings*

Preface

Dear reader,

About eleven months ago, I've started the work in front of you, in order to finish my master studies in computer science at the *University of Twente*. It is with great pride that I can finally present to you my master thesis.

The speech by *Samwise Gamgee*, is somewhat analogous to this work. At that moment in the story, Frodo and Sam had been struggling to reach their objective of destroying the ring for some time already and they still had a long road ahead of them. They felt lost because they encountered problem after problem. It shows that even though there were challenges, at least they had each other. This project has been with some challenges for me as well. But at least I had a lot of people around me to help me stay motivated and to keep on fighting. The road was long, but I didn't turn back and so it was, full of lessons, adventure and experience. Through it, I've bonded with my supervisors, found new colleagues and made new friends.

Another comparison can be made with the topic of the thesis. For, I've also encountered a shadow along the way. One as dark and dangerous as those in Mordor. Global warming is a very serious problem that needs to be dealt with, rather now than later. One day it will be one of the great stories. The ones that really mattered. I can only hope that my work, in some small way, may help the people of our planet to overcome this global issue. If we want to protect our planet for generations to come, then we have no other choice than to form a fellowship of the environment.

I sincerely hope that you will enjoy reading my thesis and that you, like me, think it's something worth fighting for.

Nico Korthout   Utrecht, 21 April 2017

Acknowledgement

This thesis is the culmination of eleven months of work that would not have been successful without the guidance and support of many wonderful people.

First and foremost, I wish to thank Jan-Peter Doomernik, as the initiator and radical force behind the main ideas presented in this thesis. You are not only a visionary, but also an innovator. One that questions authority and does not shy from changing the status quo. Without you this thesis would not exist. I hope it will help you succeed in making the world a better place.

Next, I want to thank Mehmet Akşit. I am very grateful for all your contributions, guidance and insights. Over the past few years, you've been such a great teacher to me. I look positively back on our wonderful discussions on geopolitical and philosophical topics. I wish you all the best in your future contributions to science.

I also want to thank Maya Daneva. Although you only joined the project somewhat later, you've been able to help me improve the work so much. I've always enjoyed your enthusiasm, friendliness and positivity in our meetings and calls. I hope you continue giving your excellent guidance to other students.

Technical supervision was provided by Gert Jan Timmerman. Thank you for your critical feedback. Laila van de Hoef supervised my process. Thank you for your guidance and all the lovely conversations.

Thank you to all of the graduate interns at Info Support for all the times they grovelled under the foosball table. Special thanks to Pim Verheij, Remco Westerhoud, Jur Braam, Clint Geense, Cas Diependaal, Thomas Peters, Julius Duijs and Nick Tönissen. Additional thanks to Mike van Dongen, Sjoerd Nijhof, the rest of the LifeApps team in room 4.6 and all the other Info Support employees.

I would like to thank Info Support for providing me a place to write my thesis, good supervision and new colleagues. I am happy to have received the opportunity to start my career at this innovative knowledge-driven organisation.

Thank you to my family. My parents who have always supported me. My brother, Michel, and my sister, Angela, and both of their families. Thank you all for the support. Finally, my dear girlfriend Linda. Thank you for always being there for me, for helping me push my boundaries and for keeping me motivated. I love you more than anything in this world.

Contents

List of figures

List of tables

Chapter 1

Introduction

The electricity supply system in the Netherlands has already been operational for more than a century (Hermsen, 2011). This system has matured during the late 19th and 20th century and has seen many changes to its design, but its architecture has stayed mostly the same throughout several of the last decades. Meanwhile, the context in which the system operates has changed dramatically.

Society is beginning to view the environment as an important political topic (TNS NIPO, 2016). More and more civilians want to do their part in solving the issues surrounding global warming, but investments for sustainable energy are costly and are often difficult for the average person. Companies, organizations and institutions have joined together to form the energy agreement in 2013 as a guideline to a sustainable society and economy (NRC, 2013); and the government has agreed to reduce CO2 emissions by signing the Paris agreement (NRC, 2016). In order for the Netherlands to meet the emission levels set in the Paris agreement, all its coal plants must be shut down (NOS, 2016a), while in contrast three newly build coal plants have recently begun their operation (Greenpeace, 2015). In addition, the political landscape is divided when it comes to solutions for the environmental issues and some political parties appear to actively work against potential solutions (Mommers, 2017). Something seems amiss with the electricity provisioning in the Netherlands.

Today, systems oftentimes are designed with user-centered approaches to satisfy the user's needs as much as possible. A societal system, like the electricity supply system, should in turn be designed with a society-centered approach to satisfy society's needs as much as possible.

We argue that political and commercial needs have had a greater impact on the electricity supply system than civil and environmental needs; and that a revision of the requirement specification should satisfy the needs of society as a whole as much as possible, because this system provides a societal function. This thesis provides the first work towards a future societal electricity supply system.

1.1 Research goal

The electricity supply system has been around for many years. It has matured (i.e. evolved) throughout those years. At the same time, the context in which it operates has changed as well. For example, many technical advancements have taken place, political perspectives of the people have changed and similarly the awareness of our environment has changed. In the same way, the electricity supply system must have changed together with that context, as described in section 1.2.2.

Today, systems oftentimes are designed with user-centered approaches to satisfy the user's needs as much as possible. A societal system, like the electricity supply system, should in turn be designed with a society-centered approach to satisfy society's needs as much as possible.

The main research goal is to perform a feasibility analysis for an electricity supply system, which is specifically designed to fit the needs of society as much as possible. This main research goal may be stated as a research question:

> *Is it possible to design an electricity supply system, that satisfies the needs of society as much as possible?*

This research question can be divided into several sub-questions:

1. *Does the current electricity supply system satisfy the needs of society as much as possible?*
2. *What are the requirements for a societal electricity supply system?*
3. *Can we solve the engineering problems involved with a societal electricity supply system?*

1.2 Research approach

A structured approach towards finding the answers to the research questions is needed, in order to do attain the research goal in a reliable manner. Figure 1.1 shows the research approach taken in this thesis.

The first research question is answered through a *Background research*. This background research consists of (1) a *Problem analysis*, in which problems of the current electricity supply system are exposed; and (2) a *Societal needs analysis*, in which an initial perspective towards answering the second research question is provided.

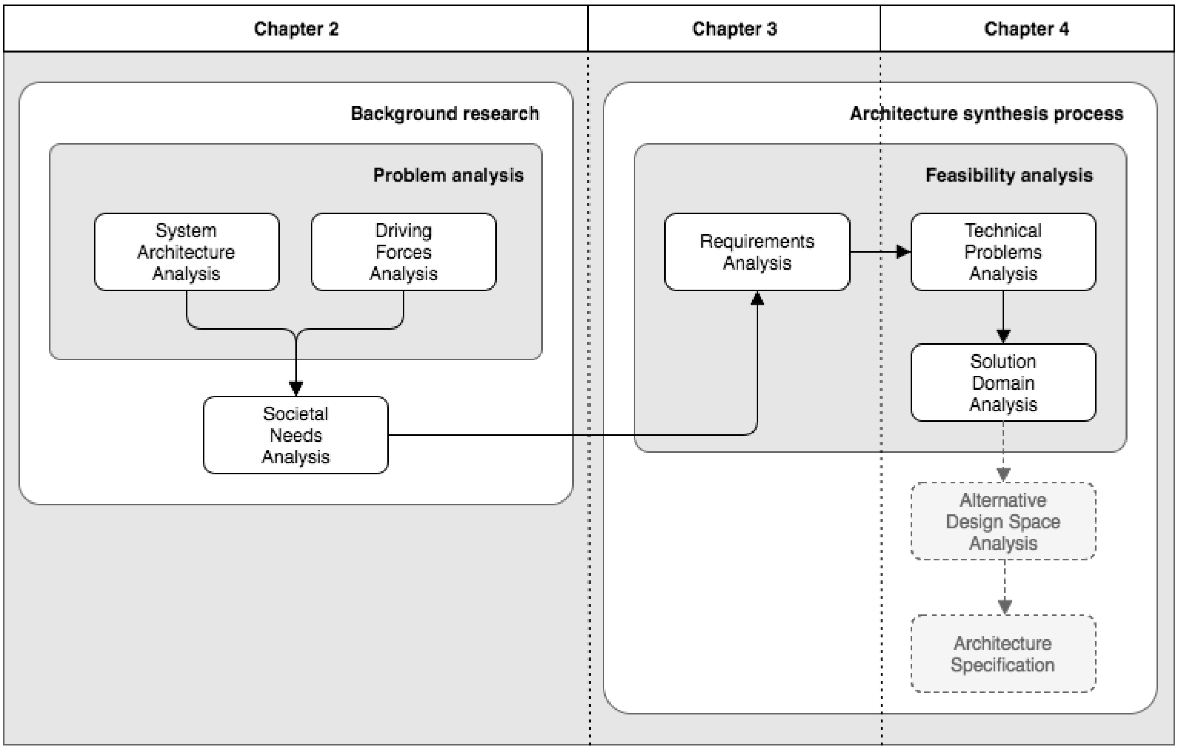

Figure 1.1: Research approach

The problem analysis is performed by (1) a *System architecture analysis*, resulting in an model of the current electricity supply system's architecture that can be used to reason about the system; as well as (2) a *Driving forces analysis* that provides insights into how the current system has changed over time into its present form.

The model of the current electricity supply system is arrived at from a commonality analysis (Coplien, Hoffman, & Weiss, 1998) of the available domain knowledge. A commonality analysis is generally applied as a first step in domain requirements engineering or in product-line engineering. However, this system already exists and domain knowledge about it is readily available. Therefore, we can apply a commonality analysis on this domain knowledge to derive the architecture of the current electricity supply system which can be used to reason about it.

Insights into the driving forces that drove the changes of the electricity supply system to its current state, are acquired through an historical analysis of the system. This is performed using published literature, domain knowledge and our own perspectives of political and commercial driving forces.

The societal needs analysis is performed using the ISO/IEC 25010:2011 standard model for in-use system qualities that can be used to describe system qualities from the user's perspective during operation. This analysis is performed from the 6 different societal view points or user groups, i.e. civilians depending on electricity, civilians providing electricity, businesses depending on electricity, businesses providing electricity, the state and the environment.

The second research question is answered through a *Requirements analysis*. This requirements analysis is the first step in the *Architecture synthesis process* (Tekinerdoğan, 2000). The requirements analysis approach is inspired by the work of Thayer, Bailin, and Dorfman (1997), but only addresses the main topics. The requirements have been arrived at from iterated discussions, meetings and document exchanges with the client. Term definitions provide a basis to form textual requirement specifications. A rationale discusses our reasoning.

The third research question is answered through a *Feasibility analysis* using the *Architecture synthesis process* (Tekinerdoğan, 2000). In this process, only the *Technical problem analysis* and the *Solution domain analysis* are applied for the purpose of a feasibility analysis. The *Alternative design space analysis* and the *Architecture specification* have been left-out of this thesis, but may be applied in the future to expand the research from a feasibility analysis to an architecture design research.

The architecture synthesis process is explained in more detail in the following subsections. First, the architecture synthesis process is further elaborated in section 1.2.1. Second, applying the architecture synthesis process to a feasibility study is further expanded upon in section 1.2.4. Third, the influence of a changing context on a system's requirement specification is described in section 1.2.2.

1.2.1 The architecture synthesis model

Engineering consists of many different branches. Each has its own ways to tackle the challenges specific to that field, but all have to solve problems of some form. In an attempt to improve the maturity of software engineering, Tekinerdoğan (2000) derived the concept of *Synthesis* from a broad analysis of philosophy and the traditional engineering disciplines (i.e. chemical engineering, civil engineering, electrical engineering and mechanical engineering) and applied this as a problem-solving perspective to software architecture design. A software architecture encompasses the overall structure of a software system. It provides semantic meaning to its components, making it easier to discuss and reason about the software system. This report proposes the application of Tekinerdoğan's architecture synthesis model (figure 1.2) as a guideline to design a novel architecture for the electricity supply system of the Netherlands. We apply it using the *Synthesis-based software architecture design approach*.

In his thesis, Tekinerdoğan describes the architecture synthesis model thoroughly, but to recapitulate, it is constructed from concepts (represented by rounded

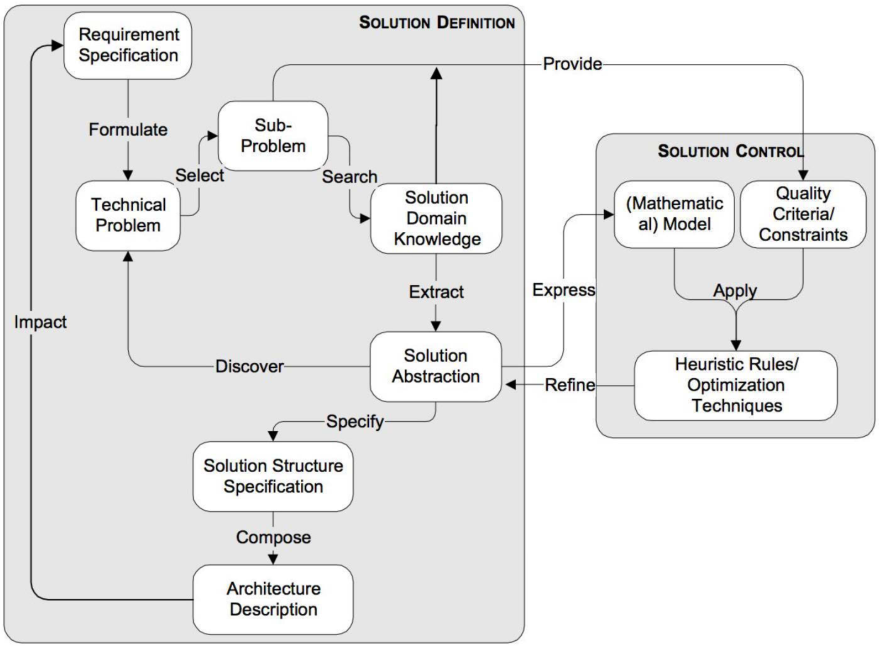

Figure 1.2: The architecture synthesis model (Tekinerdoğan, 2000)

rectangles) and functions (represented by arrows) and consists of two parts: a *Solution Definition* part, in which solution abstractions are identified and defined; and a *Solution Control* part, in which the solution abstractions can be quantified, measured, optimized and refined. In order to find a suitable architecture, a divide and conquer technique is applied to a requirement specification. Each requirement can be formulated as one or several technical problems. Solution abstractions to each of the technical problems can be extracted from the available solution domain knowledge. New technical problems may arise from a solutions abstraction, because it may not be trivial to apply the solution. Finally, a selection of compatible solutions can be synthesized into an architecture.

The *Synthesis-based software architecture design approach* is an implementation of the process of the architecture synthesis model. It consists of the following steps:

Requirements analysis. Understanding the client or stakeholder perspective of the software system.

Technical problem analysis. Mapping the requirements to technical problem, such that a problem solving process can be applied.

Solution domain analysis. Providing a solution domain model that can be utilized to extract the architecture design solution.

Alternative design space analysis. Depicting the set of possible design solutions.

Architecture specification. Extracting the architecture design.

1.2.2 A model to support changing requirements

Tekinerdoğan has based his architecture synthesis model (figure 1.2) on a controlled problem solving process, represented in the Controlled Problem Solving in Context model, or the CPC model for short (figure 1.3). The CPC model was, like the architecture synthesis model, also proposed by Tekinerdoğan and was based on studies of problem solving in the cognitive sciences.

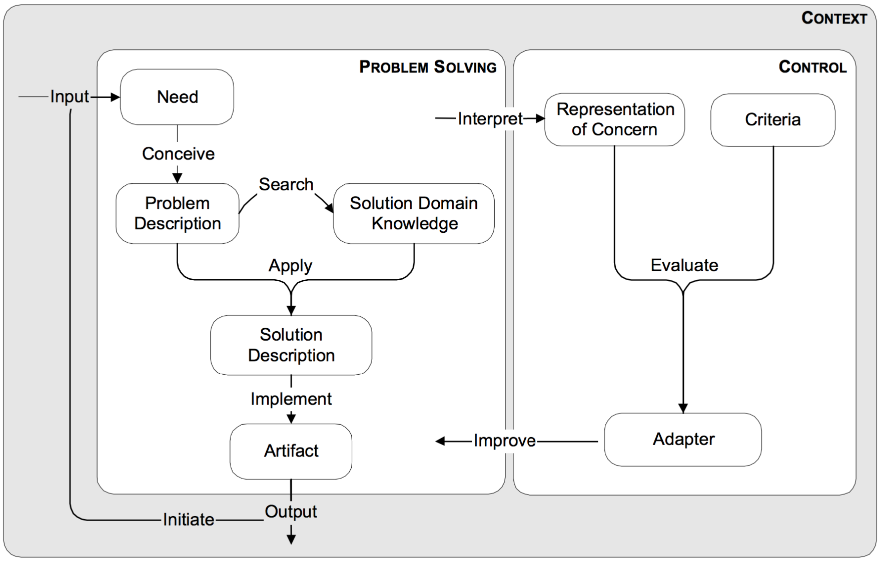

Figure 1.3: The controlled problem solving in context model (CPC model; Tekinerdoğan, 2000)

The CPC model places the problem solving process in a particular context. This context is "[...] expressed as the environment in which engineering takes place including a broad set of external constraints that influence the final solution and the approach to the solution. [...] the context may be very wide and include different aspects like the engineer's experience and profession, culture, history, and environment." (Tekinerdoğan, 2000, page 16).

The problem to be solved is conceived by a need, which in turn is created by the context. Changes to the context can thus create a different need and in turn a different problem description.

Tekinerdoğan's architecture synthesis model (figure 1.2) describes a way in which we might find an architecture that solves problems for a given set of requirements, but the CPC model (figure 1.3) shows that the needs (that conceive requirements) may change over time. In addition, it is widely accepted that system requirements change continuously over time (Jarke & Pohl, 1994). Therefore, we introduce the following model (figure 1.4) for the influence of a changing context on a system's requirement specification.

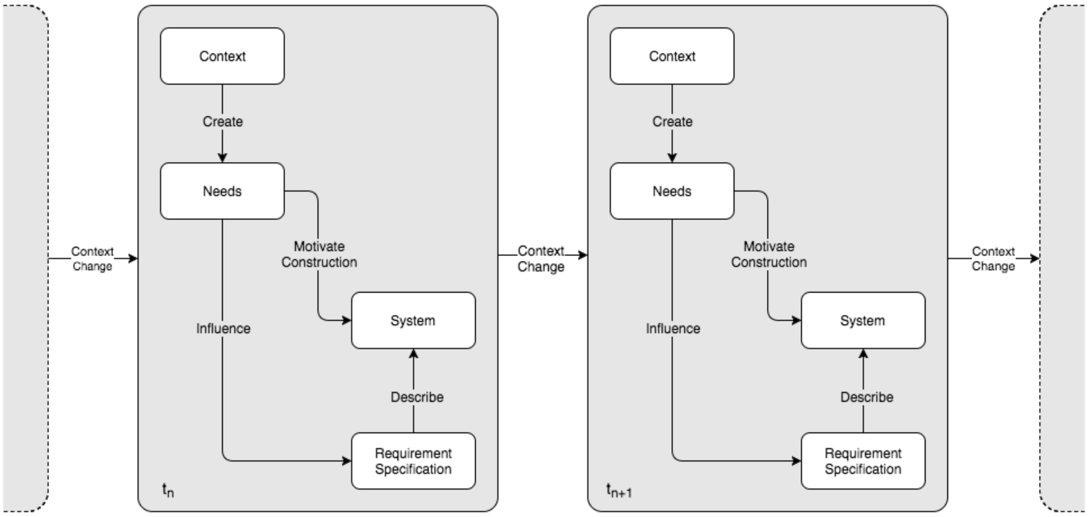

Figure 1.4: A model for requirements changes as a result of a changing context

In this model, we represent the state of the world (i.e. societal state, economic state, technological state, political state, public environmental awareness state, etc.) as the *Context*. This context creates particular *Needs*, which motivates the construction of a *System*. In addition, the needs influence the *Requirement Specification* of this system. In other words, when the context changes significantly, the needs change as well; different needs will motivate the construction of a system with different requirements.

1.2.3 The needs of a societal system

We argue that the requirement specification of the electricity supply system is influenced by needs from several different perspectives.

Society consists of political parties, commerce and civilians. In addition, the environment in which we live can be seen as part of our society as well. We can thus classify these needs as political, commercial, civil and environmental. For instance, the construction of a coal plant is an investment on which the owner wishes to see a return. The owner thus has a commercial perspective of the system and as such has specific needs for the system.

In addition, it means that the owner stands to benefit from reducing the influence of other parties on the requirement specification. Such parties will want to dictate what the system looks like by having more influence on the requirement specification than others. Civil and environmental needs may be less influential than political and commercial needs.

We argue that societal systems, like the electricity supply system, would better meet the needs of society when it would be designed to satisfy the needs of society as much as possible. We understand that this may clash with specific interests of commercial parties and governments, but we hope to find a solution that better suits the whole of society and thus we may have to change the status quo.

1.2.4 Applying the architecture synthesis model to a feasibility study

The application of the architecture synthesis model (figure 1.2) can also provide a way to check the feasibility of an engineering project. Using the techniques described by Tekinerdoğan (2000), an instantiation of his architecture synthesis model should be found for the project under investigation. An instantiation of the architecture synthesis model can be described as a tree structure of technical problems and solution abstractions in which each technical problem is solved by at least one solution abstraction. If an instantiation can be found, an architecture can be synthesized through this approach. If it can then also be developed within the project's constraints, the project is feasible.

But, if no instantiation can be found, the project is likely not feasible. This can manifest itself in: the inability of specifying requirements for the project; the inability to translate the requirements specification to technical problems; the inability of finding suitable solution abstractions in the solution domain knowledge for a technical problem; or the inability to compose the architecture from the available solutions due to constraints on the solution combinations. It is highly recommended to investigate further options when encountering any of these problems. Otherwise, the project does simply not seem to be feasible.

1.3 Thesis structure

This thesis has the following structure. It describes our motivations, the research goals and the research approach in chapter 1. In addition, this chapter describes how a problem-solving framework can be applied to a feasibility analysis. It analyses the current system, how it was shaped and society's future needs for it in chapter 2. It specifies requirements for a future societal energy supply system in chapter 3. It studies the feasibility of a novel architecture for a future societal energy supply system through the application of synthesis-based software architecture design in chapter 4. It discussions limitations of the work in chapter 5. It lists the open questions that may be answered by future research in chapter 6. Finally, it presents our conclusions in chapter 7.

Chapter 2

Analysis of the current electricity supply system

In this chapter, we discuss the current situation of the electricity supply system in the Netherlands; the driving forces that steered this system to its current form; and the future perspectives of societal needs that should drive the construction of the electricity supply system of the future as discussed in section 1.2

2.1 System architecture analysis

An analysis of the current system's architecture should provide insights into the choices made by the system's creators.

2.1.1 Commonality analysis of the current system

The Dutch law for production, transport and provisioning of electricity (Elektriciteitswet, 1998) provides restrictions on the national system by exacting rules on what is and what is not allowed for the Dutch electricity infrastructure and the institutes involved in its operation. It states that all the power lines that are destined for the transportation of electricity at a voltage level of 110 kV and up that operated as such and all trans-border power lines with alternating current together form the national high voltage network, which is operated and maintained by a single Transport Service Operator (TSO). Its goal is to manage the *transport of electricity* within the boundaries of the Netherlands. The *distribution of electricity* to the end users is performed by the regional high, mid and low voltage networks, which are operated and maintained by their respective Distribution Service Operators (DSO). In the Netherlands the DSOs are each provided with their own regions in which to operate.

When discussing electricity at different voltage levels, the following definition is used by the Dutch law: below and up to 1 kV is low voltage, higher than 1 kV is high voltage (Elektriciteitswet, 1998). In the literature on *electricity networks*, we also find other distinctions: $\leq$ 1 kV is low voltage; > 1 kV and < 35 kV is mid voltage; $\geq$ 35 kV and < 220 kV is high voltage; and $\geq$ 220 kV is extra high voltage (Netbeheer Nederland, 2014). From a commonality analysis perspective, electricity networks are the commonality and the different *voltage levels* are the variability.

These electricity networks are connected to each other and exchange electricity. In order to do so, the voltages need to be transformed to match the voltage level of the other network. This transformation is one of the functions of a *substation*. Other *substation functions* can be switching networks on and off (e.g. for maintenance); distribution to smaller networks or industrial consumers; or conversion between direct and alternating current. Substations can facilitate one or several of these functions to the system (Hoogspanningsnet, n.d.-c).

Electricity has to be provided to the networks, in order for consumers to use it. This task is performed by *generators*. In more technical terms, a generator creates and maintains the electric potential difference on the network. This electric potential difference allows a current to run through a closed circuit. Generators come in different types and sizes. Examples are rooftop solar panel systems producing in the kW range, solar panel and wind turbine parks producing in the MW range and coal plants or nuclear plants producing in the hundreds of MW to GW range.

Consumers can close a circuit by plugging in an electric device or turning on an already plugged-in electric device. Here an electric device can be anything from a low-powered light all the way to very high-powered industrial machinery. We can make a distinction between *domestic and industrial consumers*, because an industrial consumer may consume many orders more than a domestic user at any given moment. Generally, domestic users receive their connection via a separate *distribution hub* that acts as a small substation by distributing the electricity to its respective domestic users.

Electricity production and consumption has to be in balance within its networks. The frequency of the alternating current changes with each change in supply and demand (Mainsfrequency, n.d.). An out of bounds frequency may cause problems or even critical damage to electric devices and infrastructure. The TSO and DSO's measure and control the frequency of the alternating current in the network. Control is enacted through dynamically adjusting the input of energy to the generators, which affects frequency due to a simple energy balancing law (i.e. input energy = output energy + change in rotation speed, where the rotation speed defines the frequency of the current). Dealing with large changes in supply and demand is possible through import and export with *foreign networks*. Moreover, although electricity is difficult to store, *storage facilities* do exist and can also provide a balancing between supply and demand. An example of electricity storage is the Saurdal Hydroelectric Power Station, one of the largest hydroelectric power generators in Norway as well as an water-pumped electricity storage facility (Rosvold, 2012).

Electricity is produced and consumed in different places, we can say that electricity flows from a point in the network to another. This is called *load flow*. Typically this flow is *bidirectional* in the transmission network and *unidirectional* in the distribution network. This means that electricity can flow freely in the (extra) high voltage networks of the transmission network, supplying electricity to a region that has low supply and high demand. In addition, this means that the load flows from the substations via the distribution hubs to the domestic consumers, but not the other way. However, some domestic consumers provide electricity back to the network through *private generators*, like solar panels and wind turbines. Thus, some load flows back to the distribution hubs after which it can flow to other domestic consumers. Load flow from the distribution hub to the substation and back to the transmission network can cause problems for DSO's.

The Dutch *national electricity network*'s alternating current is synchronized with the electricity networks of several other countries. Together, these are called a *wide-area synchronous grid*. The Dutch electricity network is part of the Synchronous grid of Continental Europe (ENTSO-E, 2014).

2.1.2 A model of the current system architecture

From the commonality analysis we are able to derive a model of the current architecture of the Dutch electricity supply system.

In order to model an electricity supply system, we need to be able to show the flow of electric power, from producers to consumers. This is called *Load Flow* and can be either unidirectional or bidirectional. It flows from a source to a target and can thus be modelled as an association between the other modelled concepts. Figure 2.1 presents a simple inheritance model for the directionality of this load flow.

Figure 2.2 presents a model of the electricity supply system in its current form. In order to explain this model in more detail, it is split up into several smaller figures (see figures 2.3 to 2.7).

A *Wide Area Synchronous Grid* (figure 2.3) consists of at least one *National Electricity Provisioning Network*. National electricity provisioning networks provide electricity in their respective country. The Dutch electricity supply system is thus

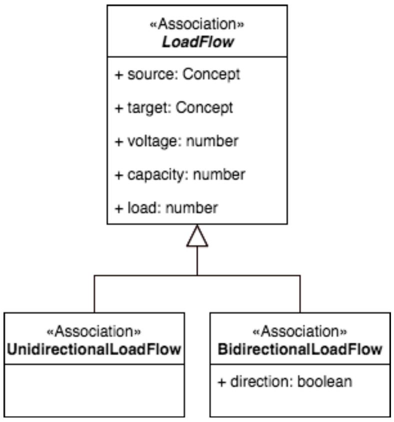

Figure 2.1: An association representing load flow

an instance of the national electricity provisioning network concept. It is part of the Synchronous grid of Continental Europe, which is an instance of the wide area synchronous grid concept. National electricity provisioning networks can exchange energy (i.e. from the perspective of the electricity supply system this energy is in the form of electricity) between each other. In reality, we find that the Dutch electricity provisioning network is connected to the Belgian and German electricity provisioning networks through alternating current and to the British and Norwegian electricity provisioning networks through direct current. In each of these connections we find a bidirectional load flow.

A national electricity network consists of two main network concepts: the *Transmission Network*, which is used to transmit electricity over longer distances (e.g. between cities or inter-provincially); and the *Distribution Networks*, which are used to distribute the electricity regionally. At a more technical level, these networks consist of smaller electricity networks operating at different voltage levels. The transmission network consists of *High Voltage Networks* and *Extra High Voltage Networks* and the distribution networks consists of *Low Voltage Networks*, *Mid Voltage Networks* and *High Voltage Networks*.

Figure 2.4 presents a refinement of the extra high voltage network concept. Its structure is the same as the high voltage network concept (figure 2.5) and the mid voltage network concept (figure 2.6), but uses the *Extra High Voltage* prefix to indicate that these concepts are able to deal with extra high voltages. In the extra high voltage network *Extra High Voltage Generators* provide energy to *Extra High Voltage Substations*. *Extra High Voltage Industrial Consumers* can use this energy. The energy can also be stored in (or taken out of) *Extra High Voltage Energy Storage* facilities. The extra high voltage substation can exchange energy with other substations.

Figure 2.5 presents a refinement of the high voltage network concept. Its

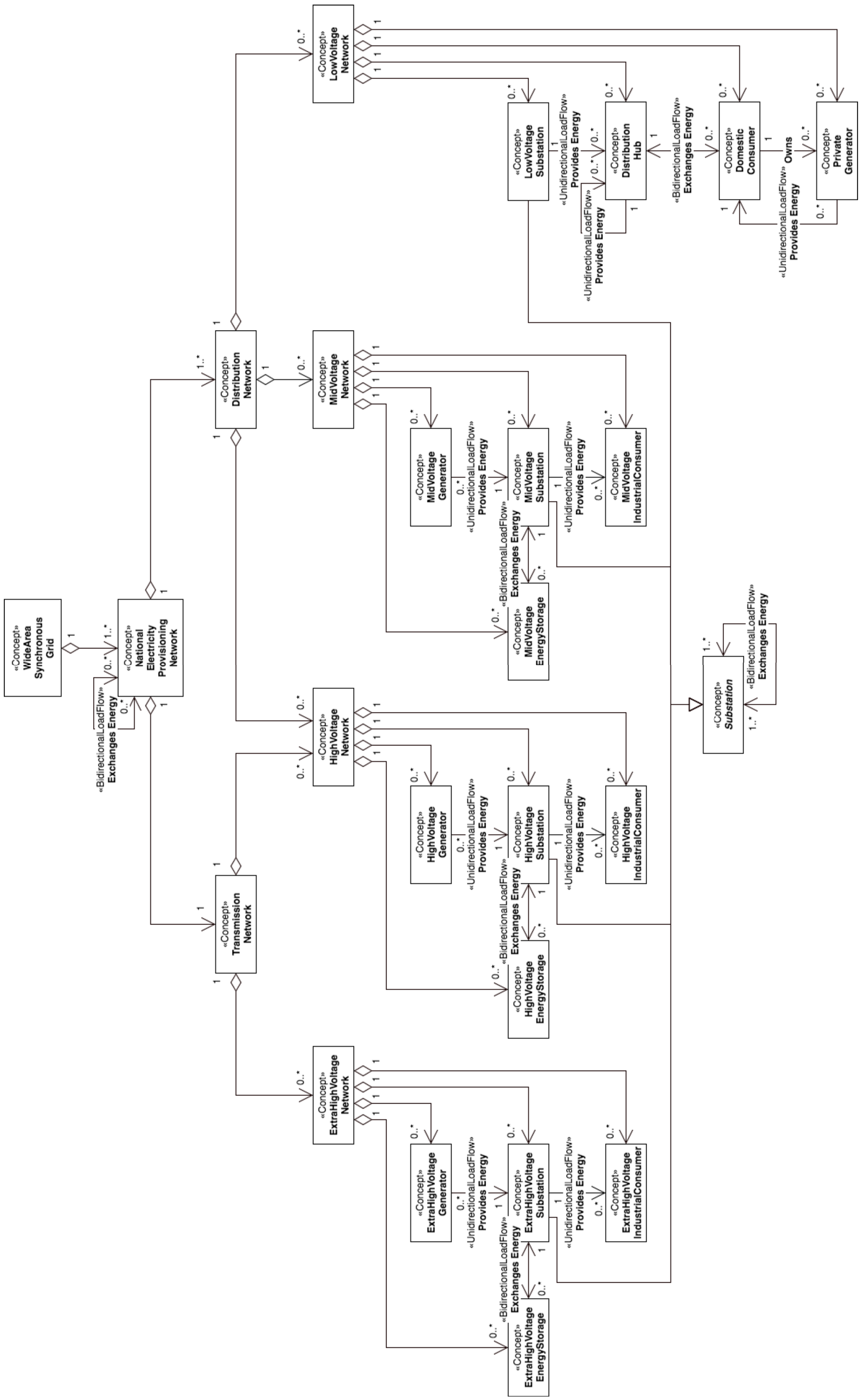

Figure 2.2: A model of the current electricity supply system

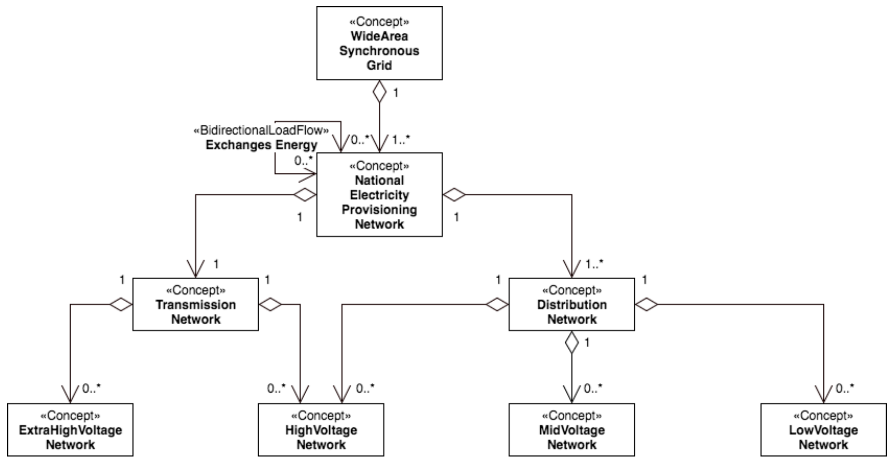

Figure 2.3: A model of the current electricity supply system - networks overview

structure is the same as the extra high voltage network concept (figure 2.4) and the mid voltage network concept (figure 2.6), but uses the *High Voltage* prefix to indicate that these concepts are able to deal with high voltages. In the high voltage network *High Voltage Generators* provide energy to *High Voltage Substations*. *High Voltage Industrial Consumers* can use this energy. The energy can also be stored in (or taken out of) *High Voltage Energy Storage* facilities. The high voltage substation can exchange energy with other substations.

Figure 2.6 presents a refinement of the mid voltage network concept. Its structure is the same as the extra high voltage network concept (figure 2.4) and the high voltage network concept (figure 2.5), but uses the *Mid Voltage* prefix to indicate that these concepts are able to deal with mid voltages. In the mid voltage network *Mid Voltage Generators* provide energy to *Mid Voltage Substations*.

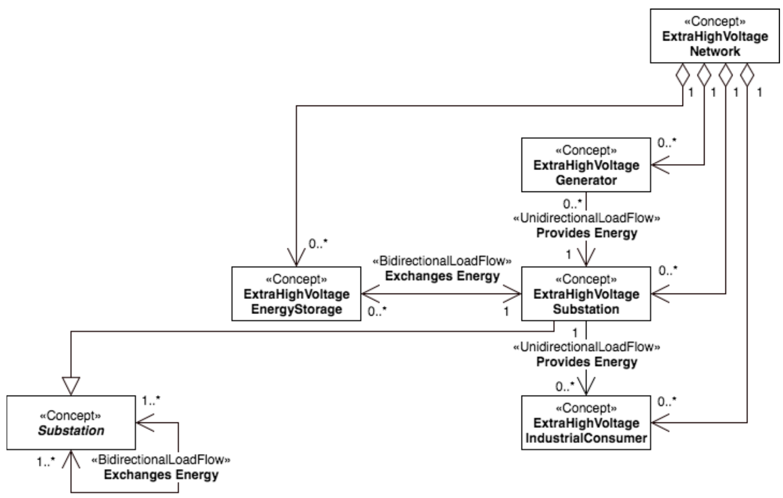

Figure 2.4: A model of the current electricity supply system - extra high voltage network refined

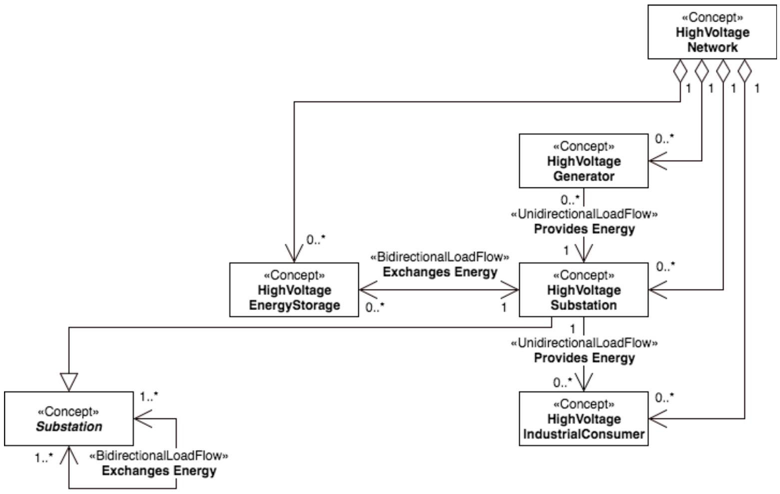

Figure 2.5: A model of the current electricity supply system - high voltage network refined

Mid Voltage Industrial Consumers can use this energy. The energy can also be stored in (or taken out of) *Mid Voltage Energy Storage* facilities. The mid voltage substation can exchange energy with other substations.

Electricity distribution works a bit different in the low voltage networks compared to the other electricity networks. Figure 2.7 presents a refinement of the low voltage network concept. Again the other electricity networks (figures 2.4 to 2.6) are connected via the substations, but the distribution to the *Domestic Customers* happens via *Distribution Hubs*. Distribution hubs can also distribute energy to the distribution hubs of subregions. Domestic consumers can produce their own energy through *Private Generators*. Excesses can be distributed to other domestic

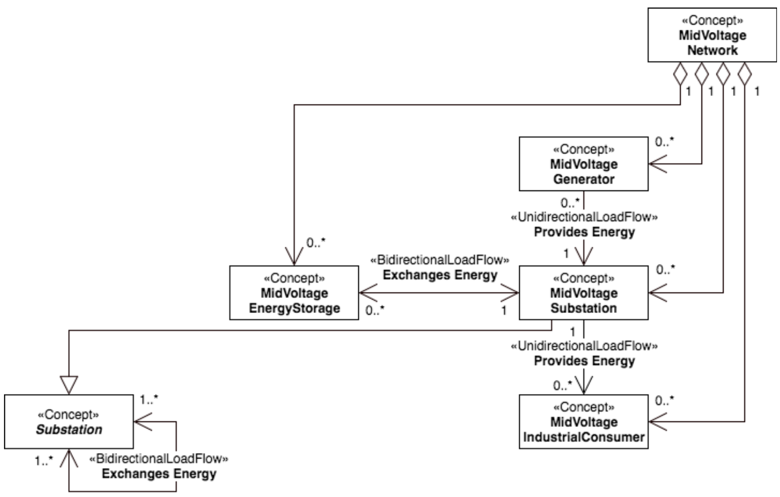

Figure 2.6: A model of the current electricity supply system - mid voltage network refined

consumers through the distribution hub, but distribution hubs are unable to provide energy back to substations.

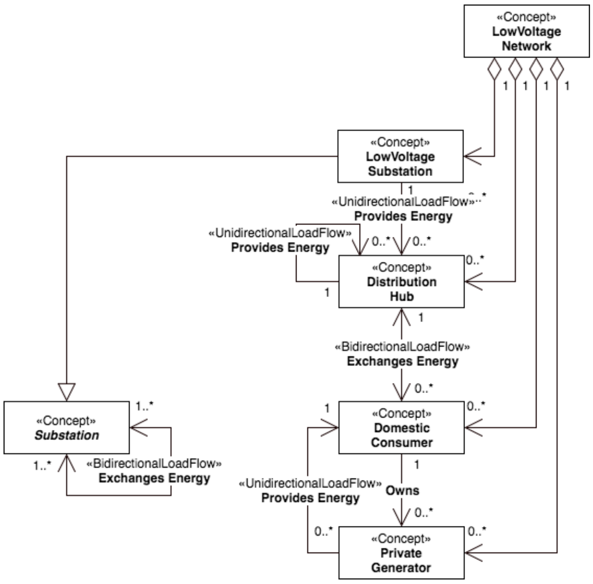

Figure 2.7: A model of the current electricity supply system - low voltage network refined

The model presented in figure 2.2 (and figures 2.3 to 2.7) exhibits a few clear characteristics. For instance, the structure is very hierarchical, due to the aggregate relations. This forms a model that is solid, but not flexible. Its parts are difficult to be reused in other parts of the system. One of the obvious technical reasons for this is that system parts that work on lower voltages can't simply work on higher voltages. Control systems in a substation for mid voltages would overheat and shutdown at high voltage levels, which may even result in critical damage and a power outage.

Moreover, energy can flow freely in the transmission network, but also for most part of the distribution network. Only in the low voltage network do we see a clear tree-structured top-down distribution. This may lead to problems when private generators provide too much energy back to their region. Control systems in distribution hubs are most of the time not yet able to deal with such load flows. Equipment may start to fail due to overcapacities. The transformers in private generators may shut down automatically when this happens as a safety measure. This problem has recently been occurring in Groningen (NOS, 2016b).

2.2 Driving forces analysis

The electricity supply system in the Netherlands was not designed and build in one day. Many different people and parties have played a role in its construction. We argue that political and commercial driving forces have helped shape the system to its current form.

There are many literary examples that discuss political and commercial driving forces on the shaping of the electricity supply system in the Netherlands. For instance, van den Noort (1984) presents thorough research into the motivations of the municipality of Rotterdam for its role in the electricity provisioning for Rotterdam and its surroundings; and van Empelen (n.d.) discusses political motivations for the completion of the national electricity supply system before and during the Second World War. In addition, we could simply look at the current political agenda and find plans and ideas for changes to the electricity supply system.

Although these examples show some involvement of political and commercial driving forces on the shaping of the electricity supply system, we discuss the driving forces involved with each of the significant changes to the system throughout history. In order to do so, we first describe a perspective of political and commercial driving forces.

2.2.1 A perspective of political and commercial driving forces

A state that can govern itself without any interference from outside is considered a sovereign state. A state's main objective is keeping this sovereignty intact. The Kingdom of the Netherlands (more commonly known as simply the Netherlands) as a rechtsstaat, protects its sovereignty using the rule of law. Likewise, the rule of law can be exercised within the state due to its sovereignty.

As Biersteker and Weber argue, "The modern state system is not based on some timeless principle of sovereignty, but on the production of a normative conception that links authority, territory, population (society, nation), and recognition in a unique way and in a particular place (the state)." (Biersteker & Weber, 1996, page 3), they define sovereignty as a composition of authority, territory, population and recognition.

Based on this, a state's internal sovereignty can be described as the state's authority over the population within its territory and a state's external sovereignty can be described as the recognition by other states of the state's internal sovereignty. A state's sovereignty can thus only be maintained through the protection of the internal and external sovereignty.

A state's internal authority of its population within its territory can be maintained either forcefully (e.g. through military or police force) or the authority can be accepted by the population. In the latter, the population must allow the sovereign its authority, either through divine right (i.e. divine right of kings) or through popular sovereignty. The population may demonstrate peacefully to show their dissatisfaction with the authority, but can also demonstrate violently or even overthrow its sovereign. Another option for them would be to find another state that does satisfy their needs and emigrate. Based on this, we can see that a state is able to adjust how it maintains its internal authority. Internal authority is either there or not, but its existence consists of several parameters that together provide the internal authority.

A state's external recognition of its authority within its territory can be maintained through military and economic power. Military may be required to protect the territory from other states. Economic power may be used as another means to achieve the same result.

Additionally, states are dependent on natural resources (i.e. biotic resources like fossil fuels and food sources, but also abiotic resources like air, fresh water, minerals and metals) for their population and economy. States may be lacking specific natural resources, necessitating their import and thus requiring a bigger economic power.

Economic power is directly related to a state's ability to produce valuable assets. Capitalism pushes commercial enterprises to compete, leading to the effective and efficient production of valuable assets. Governments, lacking this driving force, are not very successful in maintaining the production of increasingly diverse and valuable assets. To help sustain a state's economic power it is thus necessary for governments to support enterprises in their production of valuable assets through favorable rules and regulations. In addition, we see that through the desire for increasingly diverse and valuable assets, knowledge has become an important valuable asset in producing other valuable assets as well. This naturally stimulates individualization (valuable experts with knowledge); diversification (exploring other markets, business opportunities); distribution (increasing the customer base), globalization (growth), and suitable proactive dynamic organizations to support these. The importance of valuable assets and the lack in producing specific valuable assets increases the need for a larger economic power by the need for importing these assets like natural resources.

It so happens that these new economic powers (enterprises) crosscut sovereign states. Enterprises have the possibility to move to other states and take away their economic power from the state. Since sovereign states need economic power, they have to fulfil the needs of these enterprises in some way.

The driving forces of enterprises are based on capitalism, the objective to accumulate capital. Capital accumulation can happen through increasing the company's profit, growing the company and return on investments. For each of these concepts there are again many subvalues correlating, but for now these describe the main driving forces of commercial enterprises.

2.2.2 Driving forces analysis of system changes

The perspective given in section 2.2.1 provides a generally accepted description of political and commercial driving forces, we additionally provide examples of how these driving forces affected the electricity supply system in the Netherlands throughout history, providing evidence that indeed the current system is based on political and commercial needs instead of societal needs. An overview of the history of the electricity supply system in the Netherlands is provided in appendix A.

Electricity first became publicly available in the Netherlands on 19 April 1886 through the completion and startup of the 'N.V. Electrische verlichting Kinderdijk', providing DC (Direct Current) to 350 connections (i.e. light bulbs; appendix A, row 1). Before this, some industries had already invested in privately operated generators for factory machinery, but electricity and electric light, through the designs of Thomas Edison, provided a business opportunity for other businesses as well. Businesses that started in this field had to challenge the established steam-power industry (i.e. machinery operation industry) and oil lamp industry (i.e. lighting industry), leading to innovation through diversification and distribution in a competitive market, as well as individualization by gaining knowledge. Growth of these systems took over the lighting market for houses, cities (e.g. city streets) and industries, as well as machinery operation.

In the end of the 1890's and the early 1900's, many municipalities took over the electricity systems created by the first electricity companies, either by hostile takeover or by making regulations that favor government-owned companies over commercial companies (van Empelen, n.d.). These takeovers show a political interest in electricity provisioning. We can argue that, due to economic power being such an important factor in a state's struggle to maintain sovereignty, governments have to act like companies in order to effectively and efficiently produce valuable assets. Here this took the form of small government-owned companies. In addition, the first longer range connections are constructed in this time (appendix A, row 2). This was possible because of the invention (knowledge) of the transformer, but at the time this required the use of alternating current, while so far most connections used direct current (van den Noort, 1984). Investments to construct power lines with alternating current were not much higher than direct current, because systems could still be replaced with relative ease and without high financial repercussions.

Around 1910, the first high voltage lines are constructed, connecting cities to deal with an increase in the electricity demand of the population (appendix A, row 3). This can be seen as the state increasing its power over the system by growing the value of the system, but it also satisfies the population's increasing demand. In addition, this stimulated the economy by providing electricity access to more and more business. Similar things can be said about the rise of even higher voltage power lines in the 1920's (appendix A, row 4).

As the system grows towards a nationally connected network in the years up to the 1940's, more and more electricity state-owned companies arise (van Empelen, n.d.). Competing amongst each other they grow, and the system they support grows with them. We explained earlier that governments seem not to be very good at efficiently managing the production of valuable assets. A great example of how governments can use capitalism to its advantage.

Due to the threat of war in the late 1930's, a nationally connected network is proposed, but some municipalities did not want to hand over their generators, halting these plans. After the German nazi's invade the Netherlands on the 10th of May 1940, they also take over government rule and as such the driving forces on the electricity system changes. van Empelen (n.d.) describes thoroughly the political motivations of nazi Germany, from economic power to population satisfaction, for making changes to the Dutch electricity system. They completed the nationally connected network (appendix A, row 5) and made international connections to the German high voltage network (appendix A, row 6), which provided stability and efficiency improvements, due to size and the reduced need for overcapacity, respectively.

After the war, we see three big changes to the system: standardization of high voltage levels (appendix A, row 7), higher voltage levels than before (appendix A, row 8) and ring structures (appendix A, row 9). Such changes seem likely due to the destruction left by the war, but van Empelen shows that only small parts of the infrastructure were destroyed. Another explanation could be that as the economic power grew after the war, so did the infrastructure. The events in appendix A row 10 up to row 13 can be explained in a similar manner. Economic growth is seen to be correlating with electricity usage, and thus demands supportive changes to the infrastructure.

The construction of the NorNed and the BritNed cables (appendix A, row 14) can be explained economically. For instance, the NorNed cable provides cheap electricity from the Norwegian hydro generators during the day and at night the electricity produced in the Netherlands can be stored in the Norwegian water reservoirs. A similar economic construction is used for the BritNed cable.

Electricity has become a valuable asset that can be traded. In addition, these expansions provide more stability to each of the involved networks. Both the NorNed and BritNed cables are HVDC cables. Normally HVAC cables are used over long distances, but specific requirements (e.g. overseas connection) and technical issues (e.g. the skin effect) favored HVDC cables over HVAC cables (Hoogspanningsnet, n.d.-b).

Having discussed the driving forces involved with each of the significant changes to the system throughout history, we've now shown that political and commercial driving forces have played a significant role in the shaping of the system to its current form.

2.3 Societal needs analysis

An analysis of future perspectives should provide insights to what may be required of the future system. We've said that society's needs should be the main focus of the system's creators when making choices, but we have yet to explain why. This section first explains our reasoning for a society-centred system using future expectations through observed trends, and an analysis of society's needs for the system.

2.3.1 From political and commercial driving forces to societal needs

We cannot foresee the future, but we can make meaningful guesses and assumptions. In order to say something meaningful about the future, we may take a look at trends. For instance, in contemporary society the importance of the individual user is increasing. In politics, democracy has evolved to a more direct democratic influence of the civilians through referenda (Wet raadgevend referendum, 2015) and the Belastingdienst (i.e. the Dutch Tax and Customs Administration) uses the slogan "Leuker kunnen we het niet maken, wel makkelijker" (translates to: "We can't make it any more fun, but we can make it easier"), which shows the government's willingness to help the individual civilian and shows they care about how they are viewed by their population. This is of course highly related to how political parties gain power, because this happens through popular vote by the population, but this trend of more direct influence is interesting.

In business, we also find this trend. For instance, the age-old business mantra 'customer is king' holds now more than ever. Companies like Google, Apple and Facebook now provide users with free-to-use software and earn profits by collecting personal data about those users. Personal data on the individual has become the next valuable asset. A company's image (from a customer's perspective) has also become an even more important factor, with new companies like Tesla and SpaceX claiming to help society achieve new heights in technology and aiding in the battle against global warming, but at the same time rapidly accumulating capital. This also clearly shows the commercial influence on societal systems.

Another field where we find this trend of individual importance, is in technology. User-centered design is an increasingly popular subject for designing new systems and needs analysis tries to bridge the gap between 'what is' (i.e. the present) and 'what should be' (i.e. the future) for user-centred systems (Witkin & Altschuld, 1995). Technology can be seen as a driving force of this trend. Technological advances, like internet and smartphones, have provided control to civilians over many things in their life. This has happened throughout the third industrial revolution (e.g. electronics, IT and computers) and is happening even faster during the present-day fourth industrial revolution (e.g. cyber-physical systems).

These trends have one thing in common. Individuals (i.e. civilians or customers) and their needs are becoming more important. This is also the case for individuals and their needs in system development. In this report, we have focussed on electricity provisioning for the Netherlands, a system trying to meet the societal need for the supply of electricity. de Haan et al. describe such systems using the term societal systems, where "[...] societal systems are systems meant to meet societal needs, such as energy supply, transport, health care, food and water servicing [...]" (de Haan et al., 2014, page 122).

Societal systems are not the result of civilian needs alone. Society is build-up of civilians, businesses, the state and the environment in which each of these live, work and operate, respectively. They all have their own set of needs that motivate the construction of similar systems, but with different requirements. In order to benefit society the most, we propose that a future system should adhere to the needs of these different groups, together composing the societal needs for the system. The system should not be a user-centred system, but a society-centred system.

A society-centred system implies that the commercial driving forces we've discussed previously should not be allowed to influence system changes, as these may clash with reaching the societal needs. We assume that this does not hold for political influences, because we can argue that politics is meant to be a reflection of our society through our democracy.

2.3.2 Societal needs analysis of the Dutch electricity supply system

When we look at the needs of the societal groups described in the previous section for the electricity system, we can distinguish two forms: usage and supplying. A business may be supplying electricity by maintaining the infrastructure or by producing electricity, but another business may depend on electricity to produce their product or to provide their service. Civilians also depend on electricity, but nowadays more and more civilians have started producing their own electricity. Returning this electricity back to the grid has made these consumers into prosumers, consuming and producing electricity simultaneously. In this sense, we distinguish six different societal views for the electricity supply system: the state that helps to provide electricity, the environment, businesses constructing and operating the system; businesses depending on electricity; civilians producing electricity and civilians depending on electricity.

Their needs can be expressed in different ways. In order to describe needs, we apply the quality in use model presented in ISO/IEC 25010:2011. Quality in use encompasses a system's quality from the user's perspective during operation. In this case, there are different kinds of users, namely the societal views, each with their own needs for the system. We have chosen this model over the product quality model (of the same publication), because on top of the target software (i.e. system) of the product quality model, it also incorporates its users and the usage environment[1]. Furthermore, we chose this standard due to its broad application in the software engineering field. Moreover, we feel that this approach sufficiently suits this application.

ISO/IEC 25010's in use quality model uses quality categories and subcategories in which to define quality attributes. The quality categories are defined by the ISO as follows:

- effectiveness: "accuracy and completeness with which users achieve specified goals.";
- efficiency: "resources expended in relation to the accuracy and completeness with which users achieve goals.";
- satisfaction: "degree to which user needs are satisfied when a product or system is used in a specified context of use.";
  - usefulness: "[...] degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the

[1] See figure 5 of ISO/IEC 25022:2016

consequences of use.";

  - trust: "[...] the degree to which a user or other stakeholder has confidence that a product or system will behave as intended.";
  - pleasure: "[...] degree to which a user obtains pleasure from fulfilling their personal needs.";
  - comfort: "[...] degree to which the user is satisfied with physical comfort.".

- freedom from risk: "[...] degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment.";
  - financial risk mitigation: "degree to which a product or system mitigates the risk to [...] economic objectives related to financial status, efficient operation, commercial property, reputation or other resources that could be at risk.";
  - health and safety risk mitigation: "degree to which a product or system mitigates the risk to [...] health and safety objectives that could be at risk.";
  - environmental risk mitigation: "degree to which a product or system mitigates the risk to [...] environmental objectives that could be at risk.".

- context coverage: "[...] degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified.";
  - context completeness: "[...] degree to which a product or system can be used effectively, efficiently, free from risk and with satisfaction in all the specified contexts of use.";
  - flexibility: "[...] degree to which a product or system can be used with effectively, efficiently, free from risk and with satisfaction in contexts beyond those initially specified in the requirements.".

The needs of the state (table 2.1) are based on a state's desire to maintain its sovereignty. It can do this through economic power, military power and population satisfaction, as described earlier. Stable and plentiful electricity provisioning of the state is important to support the economic power, military power and the population satisfaction for obvious reasons. The effectiveness of an electricity supply system can thus be measured as the degree to which the state's sovereignty can be maintained through the electricity supply system. The combination of the levels of economic power, military power and population satisfaction may not drop below the point where the sovereignty of the state cannot be maintained. More concretely, this means that electricity needs to be continuously provided to all businesses (economic power), all national defense organizations (military power) and all civilians (population satisfaction), or in other words the availability and stability of electricity provisioning are important to the state. In addition, national defense may also mean that other states cannot take control over the national electricity supply system.

Efficiency is the degree to which resources are expended to reach a certain degree of effectiveness. Examples of resources at a state's disposal are: natural resources, valuable assets, capital, and manpower. For a state's needs of the electricity supply system, this means that the efficiency is the degree to which resources are expended to maintain the state's sovereignty through electricity provisioning. So, using less resources would mean a higher efficiency. An increased efficiency is beneficial to the state, because the state can expend the remaining resources for other important aspects of sovereignty (military, infrastructure, education, healthcare, etc.).

The satisfaction of the state can only be described from a political perspective. Political parties can bring emotion-based opinions to political questions due to the factor of population satisfaction. The usefulness, trust, pleasure and comfort all stem from the population's point of view, which we cover in the needs of civilians providing electricity (table 5) and the needs of civilians depending on electricity (table 6) perspectives. A high degree of the satisfaction in those perspectives will result in high satisfaction of the state.

Freedom from financial risk is dependent on the stability and availability of the electricity supply system, because failure to provide electricity to a businesses or to civilians may have a financial impact due to the inability to produce assets or the inability to consume already produced assets, respectively. Prolonged failure to provide electricity increases this financial impact. The state thus has the need for a high degree of stability and availability of electricity.

For businesses developing and operating the electricity supply system (table 2.2), the system is effective when it helps them to accumulate capital. This may mean increasing profit (e.g. by decreasing costs, increasing pricing or reaching more customers), but may also mean enterprise growth or returns on investments. Especially returns on investment play a big role for these businesses, as most of their investments are long term (30+ years).

The efficiency of this system can then be measured as the speed of capital accumulation (i.e. the time in which a particular number of capital can be accumulated), or the ease with which this capital can be accumulated.

Category	Sub-category	Need
Effectiveness		Degree to which the state's sovereignty must be maintained (i.e. degrees of economic power, military power and population satisfaction) through electricity provisioning
Efficiency		Degree to which resources are expended to maintain the state's sovereignty through electricity provisioning
Satisfaction	Usefulness	Depending on table 2.5 and 2.6
	Trust	Depending on table 2.5 and 2.6
	Pleasure	Depending on table 2.5 and 2.6
	Comfort	Depending on table 2.5 and 2.6
Freedom from risk	Financial risk mitigation	Need for high degree of stability and availability. Failure to supply electricity may impact economic power
	Health and safety risk mitigation	N.A.
	Environmental risk mitigation	N.A.
Context coverage	Context completeness	N.A.
	Flexibility	N.A.

Table 2.1: The needs of the state

A business' satisfaction of the system lies in their perception of the system's ability to accumulate capital. So its effectiveness and efficiency provide a basis for a business' satisfaction. Usefulness for a company is mostly based on capital accumulation. As customers are becoming more important to companies, as discussed previously, we see that business satisfaction due to a system's usefulness also starts to include customer satisfaction. Customer satisfaction is further discussed in tables 2.5 and 2.6. Moreover, businesses have to trust governments to provide regulations that do not work against an effective and efficient capital accumulation. Employees at these businesses may experience pleasure in their work through stable income and fulfilling a role in a societal system.

Governments have set-up rules and regulations regarding the construction and operation of the electricity supply system. Such regulations may place financial repercussions on construction or operation outside of predefined boundaries.

Financial risk mitigation thus comes in the form of following regulations closely. A similar construction holds for health and safety regulations and environmental regulations.

Category	**Sub-category**	**Need**
Effectiveness		Degree to which construction and operation of the system helps capital accumulation
Efficiency		Degree of speed to which electricity production helps accumulate capital
Satisfaction	Usefulness	System construction and operation provides opportunities for capital accumulation
	Trust	System construction and operation requires trust in the government to provide regulations that allows effective and efficient capital accumulation
	Pleasure	Capital accumulation provides stable income for employees; Employees may feel pride in fulfilling role in societal system
	Comfort	N.A.
Freedom from risk	Financial risk mitigation	System construction or operation outside of regulatory boundaries may lead to financial repercussions; Long term investments require a return
	Health and safety risk mitigation	System construction and operation happens under strict health and safety regulations
	Environmental risk mitigation	System construction and operation happens under strict environmental regulations
Context coverage	Context completeness	N.A.
	Flexibility	N.A.

Table 2.2: The needs of businesses developing and operating the system

Businesses depending on electricity (table 2.3) as a means of energy to produce their product or to provide their products have different needs for the electricity supply system than businesses constructing or operating this system. The effectiveness of the system for these companies is influenced by the degree to which their electricity demand can be met. This means availability and stability of the electricity provisioning. The degree to which this costs money to these businesses is their view of the efficiency of the system, where lower costs correspond to a higher efficiency.

In such a perspective the electricity supply system is useful to these companies, because the production of their products or the provisioning of their services allows them to accumulate capital. Higher costs for electricity lowers their profit and as such slows their capital accumulation. To be able to use this system they have to trust it. In this case, trust is only possible when the company can trust its government to safeguard its business needs through useful regulations on the system or through technical assurance of the correct workings of the system. Failure to receive electricity may lead to financial problems for the company. As a form of risk mitigation they may invest in separate technology allowing to them to be less depend of the system.

Civilians are depend on electricity (table 2.4) in most of their daily activities and at different levels of human needs. Looking at Maslow's hierarchy of needs (Maslow, 1943), civilians are at a physiological level reliant on food and water. The production of food requires energy to run the farms, factories and labs (e.g. food engineering), which happens mostly through electricity. The production of water (i.e. producing clean drinking water) also requires electricity to operate a city's water cleaning facility. The distribution of both require electricity as well. Civilians also need electricity for safety and security. Police, but also alarm systems and street lighting depend on electricity. Belongingness and love needs of civilians require electricity, nowadays, due to social interactions happening more and more over the internet and through electronic devices like smartphones and computers. A similar reasoning holds for esteem needs, where internet applications like LinkedIn, Facebook and Twitter are used to show off achievements and accomplishments. Perhaps self-actualization not yet depends on electricity, but people tend to seek help online to become a better employee, colleague, friend, lover or simply a better human being.

The degree to which this demand is met can be measured as the effectiveness of the electricity supply system towards civilians. So once again, availability and stability of electricity is important. The costs related to this effectiveness is the efficiency, where a lower cost equals a higher efficiency. As described above, electricity is useful for all civilians at all levels of their social needs, but in order to use it they will have to trust their government to uphold regulations to safeguard these needs.

Civilians like to be in control of their own lives, so it is important for them to be

Category	Sub-category	Need
Effectiveness		Degree to which electricity demands are met
Efficiency		Degree to which costs are made to meet electricity demands
Satisfaction	Usefulness	Electricity is necessary to perform tasks with which capital is accumulated
	Trust	Places trust in government to define and uphold regulations on businesses involved with providing electricity to safeguard business needs
	Pleasure	N.A.
	Comfort	N.A.
Freedom from risk	Financial risk mitigation	Failure to receive electricity may impact capital accumulation. Investments in third-party emergency electricity supply technologies may be necessary
	Health and safety risk mitigation	N.A.
	Environmental risk mitigation	N.A.
Context coverage	Context completeness	N.A.
	Flexibility	N.A.

Table 2.3: The needs of businesses depending on electricity

able to make choices. For instance, the options of how their electricity is produced (e.g. cheaply, environment-friendly, by themselves). At the same time, civilians don't like to think too much about electricity. Choosing their form of production is something they do once and don't want to be have redo every day, week or month. But, the lack of electricity has a very high unpleasant effect on civilians, as it clashes with their human needs.

Civilians trust their governments to safeguard their needs, but sometimes risks need to be mitigated. Civilians can take matters into their own hands by producing their own electricity, although some current techniques do not necessarily allow decoupling from the grid. Civilians should be allowed ways to put matters into their own hands, for sake of risk mitigation but also the feeling of satisfaction.

To reduce the electricity costs or as an extra source of income, some civilians

Category	**Sub-category**	**Need**
Effectiveness		Degree to which electricity demands are met
Efficiency		Degree to which costs are made to meet electricity demands
Satisfaction	Usefulness	Electricity is needed at any level of social needs of the individual civilian
	Trust	Civilians place their trust in the government to define and uphold regulations on businesses involved with providing electricity to safeguard civilian needs
	Pleasure	Civilians should be allowed to make some choices
	Comfort	See usefulness
Freedom from risk	Risk mitigation	Civilians should be allowed to put matters into their own hands, for instance by decoupling from the grid
	Financial risk mitigation	N.A.
	Health and safety risk mitigation	Failure to receive electricity may impact civilian health and safety
	Environmental risk mitigation	N.A.
Context coverage	Context completeness	N.A.
	Flexibility	N.A.

Table 2.4: The needs of civilians depending on electricity

have started producing electricity (table 2.5). Civilians producing their own electricity have similar needs of the electricity supply system as businesses producing electricity, because civilians profit from producing electricity in a similar way. Differences between this group and businesses can be found in the size and duration of investments. For instance, civilians have to buy solar panels and transformers to start producing electricity. Such an investment may be large for an individual, but is dwarfed when compared to the investments of businesses producing electricity. For the civilian producing electricity however, it is still necessary to mitigate financial, health and safety risks involved with the production of electricity. In addition, civilians will have to trust governments to provide regulations that allow to make a profit from producing electricity. Civilians may feel pride in producing electricity by fulfilling a role in a societal system.

Category	Sub-category	Need
Effectiveness		Degree to which producing electricity provides profit
Efficiency		Degree of speed to which electricity production provides profit
Satisfaction	Usefulness	Reduction of electricity costs; Extra source of income
	Trust	Civilians place trust in the government to provide regulations allowing effective and efficient electricity production
	Pleasure	Civilians may feel pride in fulfilling a role in a societal system
	Comfort	N.A.
Freedom from risk	Financial risk mitigation	Electricity price may form a risk on the initial investment of third-party hardware
	Health and safety risk mitigation	Health and safety regulations should protect the civilian when producing electricity using third-party hardware
	Environmental risk mitigation	N.A.
Context coverage	Context completeness	N.A.
	Flexibility	N.A.

Table 2.5: The needs of civilians producing electricity

The environment as a societal viewpoint is a lot different from the other societal views. For one, the environment does not benefit in any way from the system. There may however, be electronic devices helping the environment in some way, but these are either operated by civilians, businesses or nonprofit organizations (whose objectives reflect those of the civilian population), but those needs are already discussed. Instead, the effectiveness of the electricity supply system as measured from this viewpoint is the degree to which the environment is left unharmed while meeting the electricity demand. Its efficiency is then measured as the time spend or the speed with which it is being harmed, where less harming equals a higher efficiency. The system may not be useful to the environment, but it may be very unuseful when a threshold of harming the environment is reached where damage has become irreversible. This implies a trust relation between the environment and humans, where our civilization should try not to harm it.

Category	**Sub-category**	**Need**
Effectiveness		Degree to which the environment is left unharmed while meeting the electricity demand
Efficiency		Degree of resources spend with which the environment is left unharmed while meeting the electricity demand
Satisfaction	Usefulness	Harming the environment too much might lead to irreversible damage
	Trust	Trusts humans to not harm it
	Pleasure	N.A.
	Comfort	N.A.
Freedom from risk	Financial risk mitigation	N.A.
	Health and safety risk mitigation	N.A.
	Environmental risk mitigation	N.A.
Context coverage	Context completeness	N.A.
	Flexibility	N.A.

Table 2.6: The needs of the environment

The above has provided a perspective to the societal needs of an electricity supply system. To summarize, as a society we need:

- the state to maintain its sovereignty;
- the electricity demand of the businesses to be met;
- the electricity demand of civilians to be met;
- the cost of electricity provisioning to be minimal;
- the environment to be left unharmed.

2.4 Necessity of a novel electricity supply system

Our world is changing around us, but a fundamental system in our society on which everyone depends, has been left mostly the same for the last several decades. The electricity supply system of the Netherlands is due for a redesign to benefit everyone.

In this analysis we've looked at the individual needs of the groups that make-up society. The needs of the businesses trying to profit from the system may clash with reaching the societal needs. Commercial driving forces are currently shaping parts of the system.

In order to create a society-centred electricity supply system, these commercial driving forces should not be allowed to influence the shaping of the system. However, tasks performed by these businesses should still be performed. This will be a concern for the design of the future societal electricity supply system.

2.5 Post-scarcity electricity supply system

Generally, systems with some form of monetisation on its service are build on the scarcity of that service. If a service, good or product is scarce, and people have a need for it, it can be capitalised (Weber, 1978).

Removing the influence of commercial driving forces from a system can be achieved by removing its monetary value, or in other words by removing the scarcity. An economy in which all the needs of society are provided abundantly is called a post-scarcity economy (Sadler, 2010).

A societal electricity supply system should work to help us achieve such a post-scarcity economy by removing the scarcity of the service it provides, i.e. electricity. We should aim to provide electricity abundantly in the design of the future societal electricity supply system.

Chapter 3

Specification of a societal energy supply system

So far, we've determined that a new energy supply system[1] must be designed to accommodate all of society. We've argued that this means that commercial driving forces must be blocked from having an influence on the system, because these driving forces may clash with the societal needs of the system.

Nevertheless, a societal energy supply system needs to do more than just keep-out these influences. For one, its main task should be to supply energy to its users. This chapter provides a short list of the most important requirements that should be satisfied by a societal energy supply system. Let us start with a short description of the main task of an energy supply system.

Society consists of several entities (e.g. citizens, factories, cities, etc.) that all rely on energy to power their devices (e.g. machinery, systems, facilities, consumer electronics, lighting, heating, etc.), in order to benefit from the services provided by those devices. This energy needs to be provided to them, because the energy is often not available at the desired location and not in the desired form. However, energy is already abundantly available on planet earth, e.g. through solar power, wind power, geothermal heat, and other renewable energy sources (De Vries, Van Vuuren, & Hoogwijk, 2007; Hoogwijk & Graus, 2008; Lu, McElroy, & Kiviluoma, 2009). In order to harness this available energy, it needs to be captured at places of abundance, transported to the desired places of usage and transformed to a usable form. This is the main task of an energy supply system.

[1] Note that previously, only the electricity supply system was discussed. This research project has had a change in the understanding of its objective. From this moment forward, we expand our notion of an electricity supply system to the more general notion of an energy supply system. An electricity supply system is of course an instance of an energy supply system, in which the form of the energy supplied is in an electrical form. The specification of this societal energy supply system can be applied as a societal electricity supply system as well.

3.1 Terms

In order to define specifications for the future energy system, several term definitions are provided. Emphasized terms refer to their respective specific definition.

Abundance. "An [...] oversufficient quantity or supply." (abundance, n.d., def. 1).

Driving force. "The main factor that causes something to happen." (driving force, n.d., def. 1)

Effectiveness. "Accuracy and completeness with which a specified goal is achieved." (ISO/IEC CD 25010.3)

Energy. "The capacity or power to do work, such as the capacity to move an object (of a given mass) by the application of force. Energy can exist in a variety of forms, such as electrical, mechanical, chemical, thermal, or nuclear, and can be transformed from one form to another. It is measured by the amount of work done, usually in joules or watts." (energy, n.d., def. 8)

Energy demand. The amount of *energy* needed at a specific moment.

Influence. "An effect of one person or thing on another." (influence, n.d., def. 1).

Minimize. "To reduce to the smallest possible amount or degree." (minimize, n.d., def. 1).

Operate. "To work, perform, or function, as a machine does." (operate, n.d., def. 1).

Optimal. "Most favourable or desirable; best." (optimum, n.d., def. 3).

Providing. "To make available; furnish; [or] to supply or equip." (providing, n.d., def. 1 & 2).

Society. "A highly structured system of human organization for large-scale community living that normally furnishes protection, continuity, security, and a national identity for its members." (society, n.d., def. 4).

Sustainability. "Environmental Science. the quality of not being harmful to the environment or depleting natural resources, and thereby supporting long-term ecological balance." (sustainability, n.d., def. 2).

The system. The system under design. The future energy system that *provides energy abundantly* in a *usable form* for *society*.

The total cost. "The price paid to acquire, produce, accomplish, or maintain anything." (cost, n.d., def. 1). Here meant as the total price paid to accomplish the ultimate goal of *the system*.

Transparency. "Any change in a computing system, such as a new feature or new component, is transparent if the system after change adheres to previous external interface as much as possible while changing its internal behaviour. The purpose is to shield from change all systems (or human users) on the other end of the interface. Confusingly, the term refers to overall invisibility of the component, it does not refer to visibility of component's internals (as in white box or open system)." (transparency, n.d., para. 1).

Usable form. Form of *energy* (e.g. electricity) in which it is generally used by *the user*.

User. "Any person, organisation, process, device, program, protocol, or system which uses a service provided by others." (user, n.d., def. 2). Here meant as a person using the service (i.e. providing energy) provided by *the system*.

3.2 Requirements

The following is our description of the societal energy supply system's requirements. It uses a top-down approach to requirements specification. We start with the goal statement of the system and derive the main requirements from the system's goal. The rational for each requirement is presented as well. We present the following requirement specification for a societal energy supply system.

3.2.1 Ultimate objective

Systems are designed to reach an ultimate objective or goal. For example, some systems inform their users on a specific or current situation. Other systems may help their users in travelling, scheduling, trading, storing, or one of the many other imaginable goals. Often such a goal does not imply the system can be disposed of as soon as the goal is reached. On the contrary, often the system must continue to provide its service in order for the goal to be continuously reached. Similarly, the system under design has an ultimate goal that continues indefinitely.

The system's ultimate objective is to *provide energy* in a *usable form abundantly* for *society*.

3.2.2 Main requirements

In order for the system to reach this ultimate goal, it can be divided into several main requirements that must be met. These main requirements will help the system reach its ultimate goal.

1. *Energy* must be *provided* to the *users* such that *society*'s *energy demand* is *effectively* met;

2. *The total cost* must be *minimized*;
3. *The system* must *operate sustainably*;
4. *The system* must be *optimally transparent* to its *users*;
5. *The system* may not be *influenced* by commercial *driving forces*.

3.2.3 Rationale

The main requirements help the system reach its ultimate goal. The following describes its rationale.

Energy must be provided to the users such that society's energy demand is effectively met.

This is the main task of the system. In order to reach the ultimate goal of energy abundance, the energy provided must be oversufficient for the energy demand. In other words, the energy demand must at the very least be met. However, abundance implies a form of overproduction. We consider effectively meeting the energy demand as sufficient, but the ultimate goal is to provide energy oversufficient. It does not have to reach this ultimate goal from the start, but it should aim to reach this goal at some point and maintain this goal.

The total cost must be minimized.

If the ultimate goal is to be reached, there is an implication of society adopting the system. Adoption of a system in our capitalist society will only happen when the system provides some necessary service and does this economically to society and its members. In this sense, the system's total costs should at least be smaller than the total costs of any competing energy supply system. Optimal minimization of the total costs will allow the system to be the most economical by definition, which in turn allows adoption of the system by society.

The system must operate sustainably.

If the ultimate goal is to be reached continuously and indefinitely, the system must operate sustainably. Sustainability is an important factor to operate endlessly, because an unsustainable operation (e.g. consuming finite resources faster than they are replenished; or producing waste faster than it can be disposed of) will eventually create a limit to that system's operation. Therefore, the system must operate sustainably.

The system must be optimally transparent to its users.

If the ultimate goal is to be reached, there is an implication of society adopting the system. Adoption of a system in any society will only happen when the system is easier to use than its competing systems. Usability in the case of energy can be considered to mean that the user does not want to have to think about using it or how to use it. It should mean that the user simply wants to use some device's functionality for its benefit, but does not specifically want to be bothered with the any concepts surrounding energy or how any of this works. Such aspects should be concealed from the user. Concealment in systems and interfaces is called transparency. In this sense, the system must be optimally transparent to its users.

The system may not be influenced by commercial driving forces.

As discussed in sections 2.3.1 and 2.4, a societal system should work to meet the needs of society, not just the needs of commercial businesses trying to make a profit. Such commercial needs may clash with societal needs. It is thus of vital importance that these commercial needs are not allowed to influence the system.

Chapter 4

Synthesis of a future energy supply system

The future energy supply system must be constructed to solve the energy supply problem. The requirements for this future energy supply system have been specified in chapter 3. In order to solve the energy supply problem several technical problems must be solved. These are described by the requirements of the future energy supply system. The energy supply problem is a composition of the following technical problems:

1. **Energy abundance problem.** Users desire energy. The main task of the system is to provide this energy. The users should thus always receive at least enough energy. In other words, starvation of the users must be avoided at all times. The goal is to eventually provide oversufficient energy;
2. **Cost minimization problem.** The users will only adopt the system if the costs of usage are lower than the costs of usage of other energy supply systems. The total costs must be minimized;
3. **Sustainable operation problem.** The system must operate to solve the energy supply problem continuously and indefinitely. This means that users must be able to use the system without interruption, but as the system operates in and interacts with the physical world as its environment, this sustainable operation also describes a form of ecological sustainability;
4. **System transparency problem.** Users should not have to be experts in all kinds of technologies in order to use the system. The system must be transparent to its users. Default behaviour of the system must be that energy is produced, consumed and transported in a transparent manner;
5. **Commercial influence problem.** Commercial influences must be kept out of the energy supply system to allow it to fully carry out its societal function.

Solving these technical problems of the energy system is difficult, but solutions may be found through the process of synthesis as described in section 1.2. Solution domains need to be identified for sub-problems such that solution abstractions may be found. The rest of this chapter describes technical problems, sub-problems, solution domains and solution abstractions from the perspective of synthesis. For readability, an overview of the synthesis (figure 4.1) is provided first, followed by more detailed descriptions of each solution domain in their respective sections. It is possible to write several books about each of the identified solution domains. We've tried to provide an overview of the problems and solutions of each solution domain regarding the future energy supply system described by the specifications of chapter 3.

For each solution domain, the respective section: (1) defines the terms used in the domain; (2) explains the invariants constraining such systems; (3) specifies the objectives of the domain; (4) translates the main requirements concerning the system under design to domain-specific requirements; (5) describes and discusses the engineering problems that must be solved to fulfil the domain-specific requirements; (6) describes and discusses potential solutions for the posed engineering problems; and (7) describes the open questions that are still to be resolved or researched.

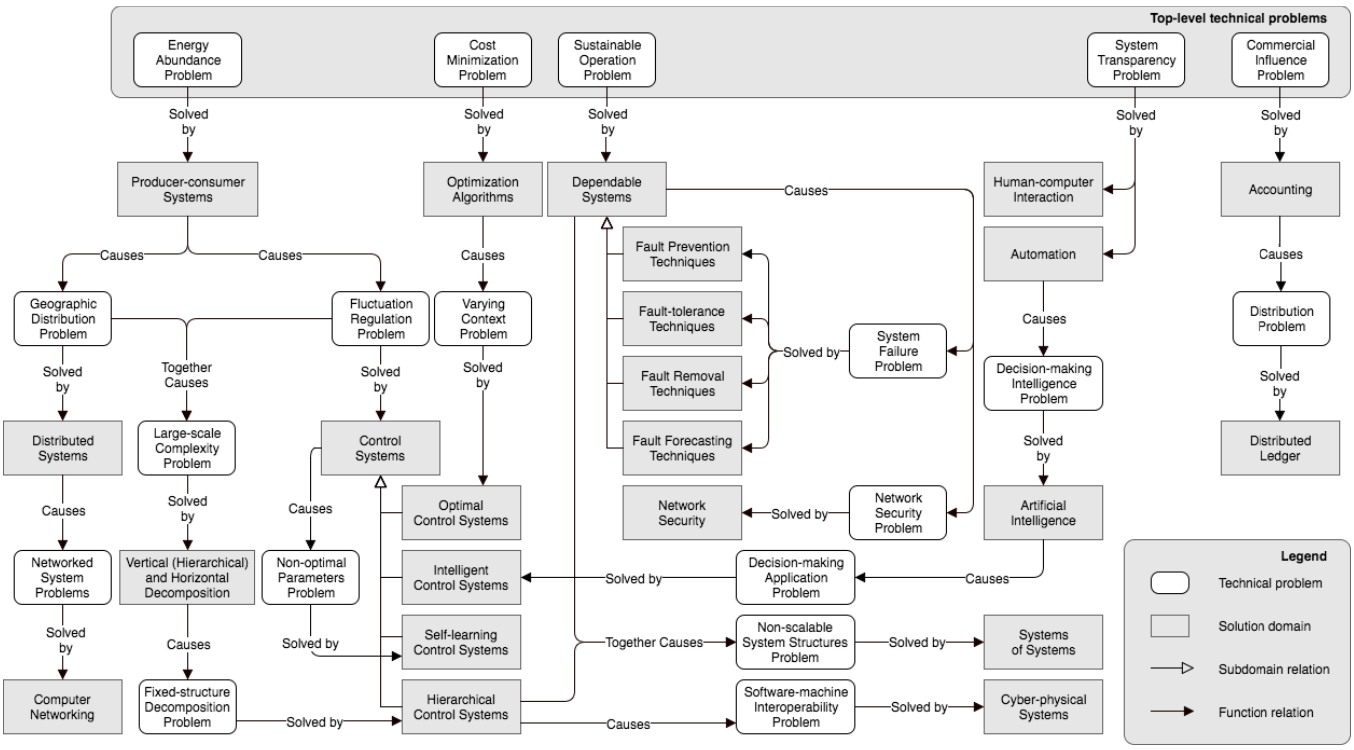

Figure 4.1: Synthesis diagram of a societal energy supply system

4.1 Synthesis overview

Figure 4.1 presents an overview of the synthesis that is performed. At the top of the diagram, the top-level technical problems are displayed as rounded rectangles. Each of these technical problems can be solved by solution abstractions of specific solution domains, which are displayed as grey rectangles. Other technical problems may arise in a solution domain.

The key solution to the energy abundance problem is located in the solution domain of producer-consumer systems. In this domain, producers and consumers are identified. Producers supply some good to consumers that in turn use that good. This is related to this technical problem because enough energy must be provided to users (i.e. consumers). In order to provide energy, it needs to be retrieved from somewhere outside of the system, which we can see as the act of production[1].

Within this domain, problems concerning the fluctuations in tempo or speed of the production and consumption are discussed as the fluctuation regulation problem. It is necessary because users must at least receive sufficient energy. Several approaches may be applied to solve this problem, but information about the state of the production-consumption fluctuation is needed in order to make meaningful decisions. Since the producers and consumers are geographically distributed, information needs to be gathered from several locations to gain these meaningful insights. This poses the problem of geographic distribution, which can be solved using knowledge from the distributed computing solution domain. This in turn, poses the problems of networked systems, which can be solved using knowledge from the computer networking solution domain.

When the producers are connected to consumers via a network with buffering capabilities it allows new perspectives on the fluctuation regulation problem. In such a network, the producers can supply energy to the network. The network can then in turn provide this energy to the consumers that desire it. Buffers may be connected to this network to help deal with fluctuations in supply and demand. In this model, the network is thus composed of producers, consumers and buffers. The network can be represented as an unrestricted graph where the producers, consumers and buffers can be hierarchically nested. This means that no assumptions about the size of the network have to be made, because the producers and consumers can be connected in many different ways. This hierarchical nesting also occurs naturally in the geographic distribution of the system. For example, in the form of the country, provinces, cities, districts, areas,

[1]Note that energy cannot be produced, nor consumed due to the law of conservation of energy, but it can be brought into a system or taken out of a system.

streets, houses and devices.

Within this network, users must receive at least sufficient energy. In average case, producers must thus produce at least sufficient energy. We assume there is a network with buffering capability, so on average the network should have at least sufficient production, at least sufficient buffering capacity and at least sufficient transportation capacity. In other words, users may not exceed the system's capacity in their consumption. Eventually, overproduction is assumed advantageous as it transitions the system from sufficient energy to an abundance of energy. Section 4.2 discusses the problems and solutions of producer-consumer systems. The problems and solutions of distributed systems are discussed in section 4.4. The problems and solutions of networked systems are discussed in section 4.5.

The first level of synthesis is to consider a producer-consumer system that is modelled as a graph network consisting of producers, consumers and buffers; a set of constraints and a set of usage patterns to solve the problem. For example, regulation may be applied to production and consumption; buffers may be applied and buffering size may be regulated; producers and buffers may be scaled, etc. Regulation of parameters in a system implies the application of the control systems solution domain. In the control systems domain, systems are not designed to be perfect, but instead system parameters are directed (i.e. controlled) towards some goal or objective.

There are many kinds of control systems and control systems can be applied at any level. However, since the highly complex nested graph structure is naturally hierarchically organized, control systems may be needed at each of these levels. In this sense, producers, consumers and buffers can be grouped together to allow reasoning and control at different levels of abstraction (e.g. grouped together producers, consumers and buffers in a district or in a city can be considered different levels of control). Similarly, control must also be enacted on individual producers, consumers and buffers. Because of this, control structures may be applied at the very fine-grained level (i.e. electric devices, for example controlling smart washing machines and the charging of electric car batteries),but also at the very abstract or high level (e.g. controlling the flow of energy within large areas).

Depending on the need, these different levels of control require hierarchical control systems, meaning that sub-level control systems must provide the needs of higher level control systems. The system is highly dynamic and not all parameters may be known initially, which can be solved using intelligent self-learning control systems (e.g. using neural network control, fuzzy control, Bayesian control, genetic control, etc.). Additionally, at the lower levels of abstraction, cyber-physical systems may be needed to deal with the software-machine interoperability, because there will be machinery involved. Lastly, the second technical problem discusses the minimization of costs. Minimization problems can be solved by optimization algorithm solutions. However, the context of the system may change over time, so control may be needed here as well. Optimal control systems solutions are needed to deal with this problem. Section 4.3 discusses the problems and the solutions of control systems and its specific forms in more detail.

Producer-consumer systems, control systems, distributed systems and networking already solve the first and second technical problems. However, the system must also solve the sustainable operation problem. This means that the system must operate both indefinitely continuous and ecologically sustainable, at the same time. Solutions to the indefinite continuous operation can be found in the dependable systems solution domain. We assume that ecological sustainability can be solved through policies and regulations (e.g. by not allowing fossil fuel energy production as producers in the system). It may be difficult to detect connected energy producing devices as polluting or not. Such a classification could be provided through standardizations of equipment information. This would allow the system to only connect zero-emission energy producing devices. Only allowing approved components to the system is discussed in the accounting solution domain (section 4.8).

Dependable systems is composed of the quality attributes availability, reliability, safety, confidentiality, integrity and maintainability. Problems of the dependable systems for the energy supply system are security and dealing with system failure. Security problems mainly arise from the use of a network. In computer networks it is often impossible to avoid attacks and difficult to mitigate threats. However, solutions from the domain of dependable systems may be applied to solve these issues. In addition, solutions from the network security solution domain may be required. Section 4.6 discusses the problems and solutions of the dependable systems solution domain.

Dependable systems together with hierarchical control systems bring-up another technical problem. The system structure may not be scalable, but have to be due to the geographic distribution of its users. The solution domain systems-of-systems addresses this problem by allowing systems to be composed of smaller component systems. These component systems can operate independently from the rest of the system, but can also connect to a larger system in which emergent behaviour arises. The problems and solutions of systems-of-systems are discussed in section 4.7.

Next, the system transparency problem must be dealt with. System transparency must be applied, because users should not have to be experts in many of the things the system does. This requires both automation for the parts the system can do and solutions from the human-computer interaction solution domain for the parts where the users still have to interact with the system. Automation is discussed in this thesis as the computer science and software technology perspective that has been taken towards this system.

However, when decisions have to be made, the problem of automation becomes more complex. This can be solved using solutions from the artificial intelligence solution domain. Generally, the decisions that have to be made intelligently by the system, are located in the control systems solution domain. The application of artificial intelligence is thus discussed from the perspective of intelligent control in section 4.3.

The system described so far is able to provide energy to users. It can deal with the problems relating to production and consumption fluctuations and with the geographic distribution of the users. It can provide this service with minimal costs and can do so sustainably, meaning both ecologically sustainable as well as indefinitely continuous. Furthermore, the system is automated to allow the users to receive the service without expert knowledge of the system's workings as well as provides those services through well designed interfaces with high usability and a good user experience. However, in this system commercial driving forces may still influence the system.

Accounting is necessary to register the components of the system. This allows the system to decide when to expand and which new components to use for this. The system can use this bookkeeping of components to rejects commercial components from joining the operation. The distributed nature of the system brings new challenges to accounting, like replication and consensus. Section 4.8 discusses the problems and solutions of the accounting solution domain.

4.2 Producer-consumer systems

This section discusses producer-consumer systems. We assume producer-consumer systems to contain components that can be classified as producers or consumers.

4.2.1 Terms

In order to discuss producer-consumer systems, several domain-specific term definitions are provided.

Buffer. A buffer represents any kind of device, machine, person, system, thing or piece of code that temporarily keeps or stores some goods, either physical or virtual, in the period of time between its creation and usage. For example, a part of the computer temporarily storing data during its transfer between two computers that process it at different speeds can be considered a buffer of data; and a battery can be considered a buffer of electrical energy.

Consumer. A consumer represents any kind of device, machine, person, system, thing or piece of code that absorbs, destroys, devours, exhausts, expends, spends or otherwise uses up some goods, either physical or virtual. For example, a person eating a bread can considered a consumer of bread; and a smartphone device using up electrical energy to operate can be considered a consumer of electrical energy[2].

Demand. The property demand represents the total number of goods, either physical or virtual, desired to be used up by all devices, machines, persons, systems, things or pieces of code within a producer-consumer system at a specific moment.

Overproduction. The property overproduction represents that more goods, either physical or virtual, are being created than there are goods being used up at a specific moment.

Producer. A producer represents any kind of device, machine, person, system, thing or piece of code that creates, makes, provides or otherwise brings into existence some goods, either physical or virtual, with the intent that it be used. For example, a baker can be considered a producer of bread; the sun as a producer of heat and electromagnetic radiation can be considered a producer of solar energy; and a solar panel can be considered a producer of electrical energy[3].

Supply. The property supply represents the total number of goods, either physical or virtual, provided by all devices, machines, persons, systems, things or pieces of code available to be used up in a producer-consumer system at a specific moment. This property's value increases by production and decreases by consumption.

4.2.2 Invariants

Producer-consumer systems operate under specific constraints. These constraints can be considered invariants of such systems. In order to discuss producer-consumer systems, the significant invariants concerning them are provided.

[2]Note that energy cannot be consumed due to the law of conservation of energy, but electrical energy can be transformed into other forms of energy (e.g. mechanical energy) and thus be consumed in the sense that the electrical energy no longer exists after being used up.

[3]Note that energy cannot be produced due to the law of conservation of energy, but forms of energy can be transformed into other forms of energy which we consider production of energy in the sense that the resulting form of energy can now be used up our the sense of the term consumption.

Availability. Consumers can only use up goods when there is an available supply (i.e. only when $supply > 0$).

Buffer limit. The amount of goods stored in a buffer at any moment is finite.

Demand fluctuation. Over time the value of property demand will vary, according to the current desire for the goods of the consumers. For example, in the case of the energy system, throughout the day individual consumers may desire different amounts of energy.

Production fluctuation. Over time the speed at which the producers provide energy may vary. For example, in the case of the energy system, throughout the day different amounts of energy may be provided by individual solar panels as the power of the solar rays changes throughout the day.

Production limit. The amount of goods provided by a producer at any moment is finite.

Starvation. When demand cannot be met by supply, the consumers will not be able to receive goods when trying to get goods.

Synchronization in space. Handover of a good can only be done at the same place.

Synchronization in time. Handover of a good can only be done at the same time.

4.2.3 Objectives

Producer-consumer systems operate to reach particular objectives. These objectives may help to fulfil the system's main requirements.

Effectiveness. The effectiveness of a producer-consumer system can be measured as the degree to which it is able to provide goods to the consumers according to their desire. A fully effective producer-consumer system thus makes sure that the demand is met by the supply (i.e. $supply \geq demand$).

In the case of the system under design, we can measure the effectiveness as the degree to which it is able to provide energy to the energy consumers according to their energy demand. The system's energy demand (E^{Demand}) must be measured as well as the energy actually supplied (E^{Supply}). We expect the system to be fully effective, which means that the system must make sure that the energy demand is met by the energy supply (i.e. $E^{Supply} \geq E^{Demand}$).

Efficiency. The efficiency of a producer-consumer system can be measured as the degree to which resources are expended in relation to its effectiveness.

In the case of the system under design, we can measure the efficiency as the degree to which resources are expended in relation to meeting the energy demand. Efficiency can be divided into two categories: (1) energy efficiency, measured as the energy produced divided by the total energy consumed; and (2) cost efficiency, measured as the total cost in relation to meeting the energy demand. In a producer-consumer system that produces its energy from renewable energy sources, cost efficiency does not take into account the energy efficiency. This implies that the energy efficiency may be relatively low for this system, as long as it is fully effective and cost efficient.

4.2.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using this domain, they are translated to domain-specific requirements.

1. The system must keep the energy supply greater than or equal to energy demand;
2. The system must minimize costs related to meeting the energy demand.

4.2.5 Engineering problems

In order to fulfil the domain-specific requirements of the system as a producer-consumer system, several engineering problems must be solved.

Fluctuation. Energy demand must be met by energy supply. However, challenges arise because energy demand and energy production may both fluctuate. The consumer will fluctuate in its desire for goods (i.e. the demand will fluctuate) and the producer may fluctuate in its production of goods. The system must be designed to deal with these fluctuations to meet the demand.

Synchronization. The energy producer and the energy consumer need to be able to 'hand over' energy. Therefore, they need to be in the same place or be in someway connected to each other; and have to make the transfer at the same time. The same holds when an energy buffer is added in between. This may be a problem, because the producer and the consumer may have a different tempo.

Complexity. The system will consist of many producers and many consumers. This introduces a problem of complexity.

4.2.6 Solutions

The following solutions are proposed for the engineering problems of producer-consumer systems.

Fluctuations. In a producer-consumer system where the production and demand may fluctuate, several general approaches may be taken: (1) production and consumption may be controlled to balance the fluctuation, keeping both at the same level; (2) buffering may be applied to smooth the fluctuations; and (3) horizontal scaling may be applied to provide a form of overproduction.

Balanced fluctuations is an approach taken by the current electricity provisioning system. This may also be called direct production-consumption (figure 4.2). Consumer C has a demand of energy. Producer P provides this energy to consumer C. This implies that C has to consume as fast as P is producing. However, both P and C have their own processes for production and consumption, which are fluctuating. Balancing these fluctuations can in this situation be achieved in several ways: (a) production may be regulated; and (b) consumption may be regulated.

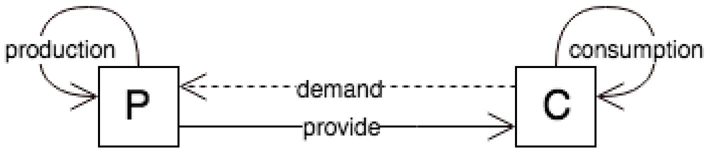

Figure 4.2: Direct production-consumption

Regulation of production can be applied by reducing the production of P according to the demand of C. In the current electricity provisioning system this also saves costs, because less fossil fuels have to used, which is optimally efficient. However, in a system that only uses renewable energy sources, this does not affect the cost efficiency. On the contrary, by definition this balancing will never allow energy to be provided abundantly, because it will only provide enough and never *extremely plentiful* or *oversufficient*.

Regulation of consumption can be applied by reducing the consumption of C according to the production of P. However, this comes at the cost of satisfaction of C, because the demand is not met immediately. This approach is applied by several smart consumer devices in the current electricity provisioning system. For example, smart washing machines may delay their operation until after peak times.

Instead of balancing the fluctuations, buffering may be applied to smooth the fluctuations (figure 4.3). Buffer B is added in between producer P and consumer C, and has a finite storage capacity. Both processes can stay the same, but energy is temporarily stored in a buffer after production and before it is consumed. This can help deal with the fluctuations of production and demand. However, B may be empty during demand of C causing starvation in C; or B may overflow during production of P, because buffers have a finite capacity. A solution must be found for both if this approach is to be taken. Buffer overflows may be solved by reducing production of P or providing some kind of disposal device, hole or sink to waste the excess. Furthermore, the starvation problem is actually the same as meeting the demand, so this should be solved using one of the solutions provided in this section.

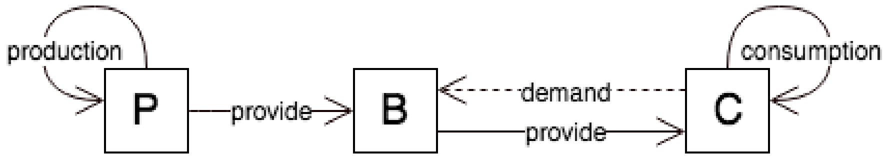

Figure 4.3: Buffered production-consumption

Another way to apply buffering to smooth the fluctuations is by adding a buffer, but not placing it in between the producer and the consumer (figure 4.4). In this situation, producer P may still provide all the demanded energy to consumer C, but any overproduced energy can be provided to the buffer simultaneously. Similarly, when C is demanding more than P produces, B may provide energy to C. Again, B may be empty during demand of C causing starvation in C; or B may overflow during production of P, because buffers have a finite capacity. A solution must be found for both if this approach is to be taken.

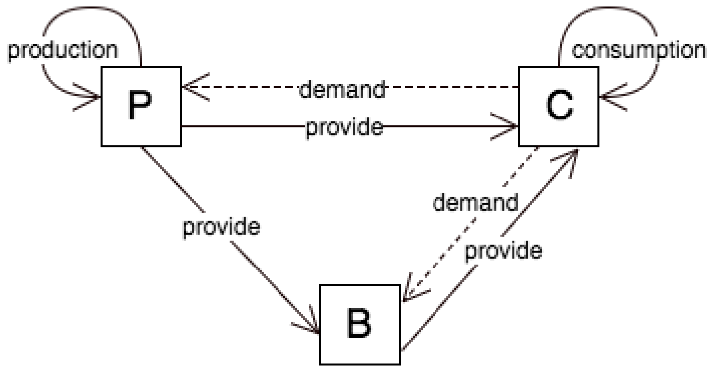

Figure 4.4: Separate buffering for production-consumption

Overproduction is another way to meet demand during fluctuations. As long as production fluctuates well above the demand, the problem has been already been solved. Production cannot be increased, because it is reliant on an environment function (e.g. solar and wind power). However, horizontal scaling (i.e. simply adding more producers) may be applied to achieve the same (figure 4.5). A similar approach by horizontally scaling the buffers may be used to increase the buffer size of the system. Horizontal scaling cannot be applied in real-time, nor in near real-time, especially in the case of energy producers and energy buffers the act of adding a device is one that takes time. A new producer or buffer has to be acquired and installed.

Complexity. Several solutions exists to deal with the complexity: (1) by connecting each energy consumer to only one energy producer (and if needed to only one energy buffer); or (2) by decomposition.

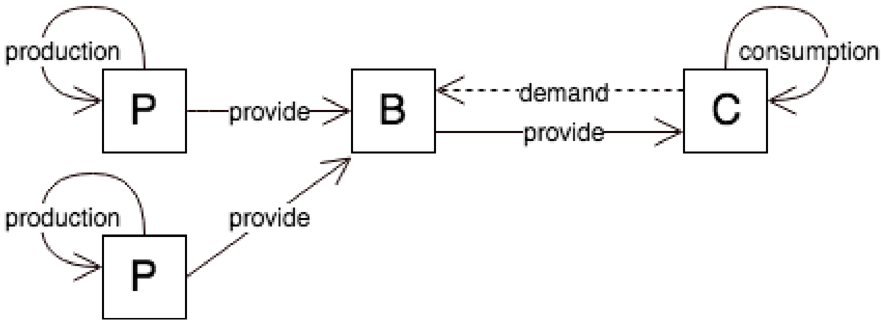

Figure 4.5: Overproduction through horizontal scaling of producers

Connecting each energy consumer to only one energy producer (and if needed to only one energy buffer) will help reduce the complexity. However, it will also reduce the reliability of the system, because the producer (or the buffer) may stop functioning. In systems with many components, failure is likely to occur, and should be designed for.

In any other situation, several energy consumers will be connected to several energy producers (and, if needed, to several energy buffers) like a network, resulting in an increased complexity for the producer-consumer problem (i.e. meeting the energy demand). This may be dealt with, by decomposing the system into smaller subsystems for which the producer-consumer problem is less difficult to solve. The subsystems may then of course also be decomposed into even smaller subsystems. And so forth. Luckily, this decomposition occurs naturally in the energy system, due to natural decomposition of the geographic location of the producers and consumers. For example, we may use the decomposition of: country, provinces, cities, districts, areas, streets, houses and devices. However, connecting these geographically distributed parts like a network may bring its own challenges.

It may be best to decompose the system into smaller subsystems, that can determine for themselves which of the fluctuation strategies to apply in order for the system as a whole to provide energy abundantly. In such a system, each subsystem can manage its own supply and demand while informing its parent system that it has too less or abundant energy. This may work by each subsystem providing a range for control to its parent system, after which the parent system can evaluate the best option, which means there needs to be a form of control of the subsystems.

Synchronization. The transfer of energy from the producer to the consumer has to happen at the same place and at the same time. The production and consumption processes may have different speeds, so this may be problematic. This problem is called synchronization in time. The problem seems partially tackled by the solutions for the fluctuation problem, but it is not yet solved. In each of the solutions proposed in that section, there is still a moment of ‘handover’ or transfer of energy between two entities (i.e. producer, consumer and buffer). Timing this correctly is difficult, because a shared clock may be missing. In a centralized systems, the solution is trivial. In a decentralized system, one of several solutions may be applied: MARS (Kopetz et al., 1989); CASD protocols (Cristian, 1989); Reference broadcast infrastructure synchronization (Cena, Scanzio, Valenzano, & Zunino, 2015); or GPS may be used to synchronize time. However, there are many more options to choose from as the topic has been researched for some time already. (Lamport, 1978)

In computing, producer-consumer systems are characterized by the application of the bounded-buffer problem. First posed by Dijkstra (1972), the bounded-buffer problem arises due to producers of data and consumers of data, sharing a finite data buffer, where each of the producer(s) and the consumer(s) are separate processes that sleep (i.e. wait) and wakeup (i.e. continue) accordingly, to make sure the producers do not put data in an already full buffer and the consumers do not take data from an already empty buffer. In other words, these processes need a form of synchronization. Systems with synchronization have to watch out for concurrency problems of multiple processes, like a deadlock. Solutions may be provided using inter-process communication like semaphores (Dijkstra, 1968), mutexes (Courtois, Heymans, & Parnas, 1971) and monitors (Hansen, 1973; Hoare, 1974).

4.2.7 Open questions

After providing solutions to the engineering problems of producer-consumer systems, several questions are left unanswered.

Geographic distribution. The natural geographic distribution of the producers and consumers in the system poses the problem of synchronization in space. The domain of distributed computing may be applied to deal with this problem (section 4.4).

Fluctuation regulation. In order to regulate the fluctuations using the solutions mentioned above, some form of control is needed. This can be achieved by designing an imperfect system that corrects its errors to reach some improved (or goal) state. This can be done using the domain of control systems (section 4.3).

Large-scale complexity. The large-scale of the system increases the difficulty of solving the fluctuations problem. Decomposing the system vertically hierarchically as well as decomposing the system horizontally peer-to-peer can help decrease this complexity. This may be solved through hierarchical control systems, which is described in section 4.3.

4.3 Control systems

Systems provide some kind of functionality to their users. This functionality is described in a requirement specification. The mapping from requirements to system design is often difficult. Sometimes it is possible to design systems to be perfect from the start, i.e. they are able to provide some function in the optimal way from the beginning and are able to maintain that ability indefinitely. In other cases, this may be highly difficult, highly costly or not possible at all. Control systems are used when it is easier to correct errors in an imperfect system than to design a perfect system. For example, a boiler correcting its water temperature can be considered a control system.

"Control systems have been successfully applied in almost all areas of engineering and extensive literature exists in this field." (Akşit, 2004, page 87). There are two general kinds of control systems: open loop control systems and closed loop control systems. Open loop control systems control the system on an external condition independent of the system's condition. For example, a boiler correcting its water temperature using a timer to determine when to apply heat can be considered an open loop control system. Closed loop control systems (also called feedback control systems) control the system using the system's condition and some desired condition. For example, a boiler correcting its water temperature using a measurement of its water temperature and a desired water temperature value to determine when to apply heat can be considered a closed loop control system.

This chapter discusses closed loop control systems from the perspective of the energy system as a software system.

4.3.1 Terms

In order to discuss control systems, several domain-specific term definitions are provided.

Actuator. An actuator represents a component, device, machine, person, system, thing or piece of code that can influence its environment.

Comparator. A comparator represents a component, device, machine, person, system, thing or piece of code that can compare two models describing conditions of the system in order to evaluate whether there is a difference between the two. Here meant as a comparator for the model of the controlled system and its reference model.

Compensator. A compensator represents a component, device, machine, person, system, thing or piece of code that can choose an appropriate correction for a difference between two models of conditions. Here meant as a compensator for a difference between the model of the controlled system and its reference model.

Controlled system. A controlled system represents the system that is being influenced.

Feedback loop. A feedback loop represents a connection between the sensor and the model of the controlled system that allows the sensory data to be interpreted and applied to the model of the controlled system. A feedback loop keeps the model of the controlled system up-to-date with the input from the sensor.

Model of the controlled system. A model of the controlled system represents the set of properties of the system that reflect its actual state and can be used to reason about its perfect or imperfect condition.

Property of the controlled system. A property of the controlled system represents some characteristic, quality or trait of the system that is being controlled. A value can represent the current state of the controlled system with respect to the particular property.

Reference model. A reference model represents the set of properties of the system that reflect its perfect condition and can be used in comparison to another condition.

Sensor. A sensor represents a component, device, machine, person, system, thing or piece of code that informs a system of its environment. Here meant as a sensor that informs the control system of its controlled system.

4.3.2 Invariants

Control systems operate under specific constraints. These constraints can be considered invariants of such systems. In order to discuss control systems, the significant invariants concerning them are provided.

Controllability. A system can only be a control system if it has the controllability property, i.e. “A system is controllable if it is possible to cause its state vector to move from any initial value, to any value, in a finite time.” (Akşit, 2004, page 87)

Observability. A system can only be a control system if it has the observability property, i.e. “A system is observable if it is possible to reconstruct the state-vector completely from measurements made.” (Akşit, 2004, page 87)

Observer effect. When a property of the controlled system is measured (i.e. observed), the observation of the controlled system impacts the systems. This may or may not have influence on the measured property itself and thus impact the observability of the system.

Stability. A system is considered stable if the properties of the controlled system can be controlled within the specified amount of time.

4.3.3 Objectives

Control systems operate to achieve one particular objective. This objective may help to fulfil the system's main requirements.

Imperfection. As described in the introduction of this chapter, a control system can be applied to correct errors in an imperfect system, in order to avoid having to design a perfect system.

The domain of producer-consumer systems (section 4.2) has described that demand and supply will fluctuate for energy supply systems. Correcting the errors (fluctuations) of such an imperfect system is the objective of a control system. The properties of the controlled system that are not controllable can be considered: demand and number of consumers. The property of the controlled system that must be controlled can be considered: supply. In order to control the supply, the following properties of the controlled system can be controlled: production rate, consumption rate, buffering, number of producers and number of buffers.

Correcting errors in the energy supply system may be necessary in different ways at each of the different subsystem decomposition levels (i.e. country, provinces, cities, districts, areas, streets, houses and devices), as described earlier. For this, we described that each subsystem's supply and demand may be managed by a parent system. This is supposed to decrease the complexity of the fluctuation regulation problem.

4.3.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using this domain, they are translated to domain-specific requirements.

1. The system must effectively control the production rate, in order to keep the energy supply greater than or equal to the energy demand;

2. The system must effectively control the consumption rate, in order to keep the energy supply greater than or equal to the energy demand;

3. The system must effectively control buffering, in order to keep the energy supply greater than or equal to the energy demand;

4. The system must effectively control the number of producers, in order to keep the energy supply greater than or equal to the energy demand;

5. The system must effectively control the number of buffers, in order to keep the energy supply greater than or equal to the energy demand.

4.3.5 Engineering problems

In order to fulfil the domain-specific requirements of the system as a control system, several engineering problems must be solved.

Problems of defining. Akşit (2004) describes multiple problems of defining control systems.

- difficulty of obtaining the actual parameters of controlled systems;
- semantic gap between the sensed data and the model of the controlled system;
- difficulty of defining the model of the controlled system;
- difficulty of defining an ideal reference model;
- difficulty of defining an automated comparator;
- difficulty of defining an automated compensator;
- difficulty of applying the desired control actions;
- difficulty of defining an optimal control architecture;
- difficulty of determining an adaptation and/or optimization strategy;
- lack of means to compute the quality factors of control.

All of these non-trivial problems have to be dealt with in some way.

4.3.6 Solutions

The following solutions are proposed for the engineering problems of control systems.

Feedback controller. Generally, a feedback control system consist of the *Controlled system* plus a controller, where the controller consists of the following components: *Sensor*, *Feedback loop*, *Model of the controlled system*, *Reference model*, *Comparator*, *Compensator* and *Actuator*; as shown in figure 4.6. The controlled system's properties are observed by a sensor. The feedback loop updates the model of the controlled system according to input of the sensor. The comparator calculates the *Error*, i.e. the delta or difference between the reference model and the model of the controlled system. The compensator uses the error to choose the appropriate correction that can be performed by the actuator, thus having an effect on the properties of the controlled system.

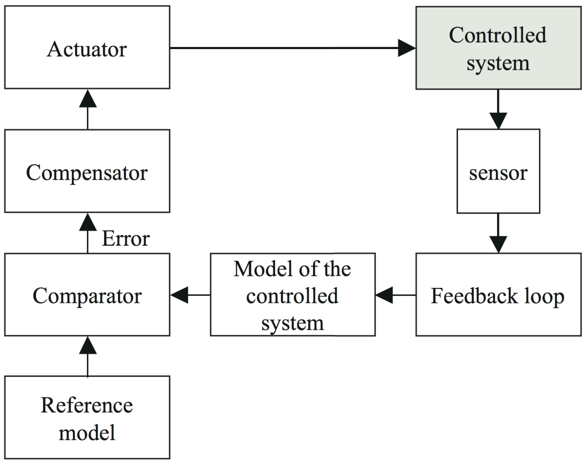

Figure 4.6: A feedback controller (Akşit, 2004)

Architectures. figure 4.6 shows a simple structure of a single feedback control loop. However, several architectures exist for control systems. A simple controller can control a single property of the controlled system, but we may also imagine controllers that control multiple properties of a controlled system. Similarly, multiple controllers may concurrently control properties of a single controlled system. This would then happen via multiple feedback control loops. Here, the controlled properties may each be controlled by a single controller or multiple controllers may control a single property. Control problems are often tackled using a combination of multiple controllers.

Whenever these controllers are acting as autonomous agents and are distributed throughout the system, the control system can be considered a *decentralised control system* or *distributed control system*. These agents may cooperate to achieve their objectives. Controllers may operate autonomously, but it is also possible that the behaviour of controllers is controlled. For instance, the reference model with which the comparator determines the error may be controlled. An example of this is the thermostat in a living room that controls the room temperature, where the desired temperature (i.e. reference model) can be changed. However, other parts of the controller may also be controlled. Controllers may be used for this control of other controllers. Higher-order controllers have control over lower-order controllers in a *hierarchical control system*. Controllers may sometimes need to adapt to a controlled system where the parameters may vary or are unknown during design. Control systems with such adaptiveness are considered *adaptive control systems*.

Artificial intelligence can be applied to control systems via intelligent agents as *intelligent control* (Linkens & Nyongesa, 1996). Several approaches of artificial intelligence like Bayesian probability, fuzzy logic, genetic algorithms, machine learning and neural networks. Possible control strategies using these approaches are: Bayesian control (Hinderer, Rieder, & Stieglitz, 2016), fuzzy control (Etik, Allahverdi, Sert, & Saritas, 2009; Liu, Wang, Golnaraghi, & Kubica, 2010), genetic control (Nassif, Kajl, & Sabourin, 2005), expert system control (Etik et al., 2009) and neural network control (Ge, Hang, Lee, & Zhang, 2013; Zhang, Ge, & Hang, 2000).

Furthermore, there exist optimal control architectures, in which a strategy to minimize a mathematically predefined and measured performance index is implemented (Akşit, 2004). This can be applied when an optimal property value is known or can be calculated.

4.3.7 Open questions

After providing solutions to the engineering problems of control systems, several questions are left unanswered.

Application. The solutions presented above only give an introductory overview of possible solutions that are available in the field of control systems. Further research to the application of these solutions to the energy supply system is required. The problems of defining control systems must be dealt with during the application of these solutions. The solutions to these problems of the energy supply system are non-trivial and require further research attention.

Decision theory. Shortly highlighted in the solutions is the domain of intelligent control systems in which artificial intelligence is applied to control systems. The field of artificial intelligence is very large and lots of research is currently happening in this field.

Physical. The solutions provided in the control systems section only discuss the ability to change properties of the system. In reality, this will mean to have an effect on the physical world. This report only focusses on the software part of the system, but the interaction between the cyber (i.e. software) and the physical world must be researched as well. The field of cyber-physical systems has been the subject of active study for many years, but has recently had a boost in popularity, due to the *Internet of Things* (Whitmore, Agarwal, & Da Xu, 2015). The application of cyber-physical systems to energy supply systems has been documented well

(Khaitan, McCalley, & Liu, 2015).

4.4 Distributed systems

This section discusses distributed systems. Distributed systems is a widely accepted grouping term for systems consisting of a geographically distributed collection of autonomous components that work together to form a system with emergent properties. In the literature, we may find several descriptions of such systems. In their brief introduction to distributed systems, van Steen and Tanenbaum (2016, page 968) state: “A distributed system is a collection of autonomous elements that appear to its users as a single coherent system.”. Such systems can generally be described as having the following three characteristics:

Component concurrency. The components are computers operating independently of each other, but are doing so at the same time.

Independent component failures. Each of the components may fail at any time for reasons that may be unknown (e.g. hardware failure, connection failure, etc.). This brings uncertainty which must be designed for. The system must be able to deal with these failures.

Lack of a global clock. The computing of each of the components is asynchronous to the computing of other components. This implies that a single computer is not classified as a distributed system.

4.4.1 Terms

In order to discuss distributed systems, several domain-specific term definitions are provided.

Availability. “For a distributed system to be continuously available, every request received by a non-failing node in the system must result in a response.” (Gilbert & Lynch, 2002, page 3)

Consistency. “There must exist a total order on all operations such that each operation looks as if it were completed at a single instant. This is equivalent to requiring requests of the distributed shared memory to act as if they were executing on a single node, responding to operations one at a time.” (Gilbert & Lynch, 2002, page 3)

Network. “[...] a system of interconnected computer systems, terminals, and other equipment allowing information to be exchanged.” (network, n.d., def. 6)

Partition tolerance. “[...] the network will be allowed to lose arbitrarily many messages sent from one node to another. When a network is partitioned, all messages sent from nodes in one component of the partition to nodes in another component are lost." (Gilbert & Lynch, 2002, page 3)

4.4.2 Invariants

Distributed systems operate under specific constraints. These constraints can be considered invariants of such systems. In order to discuss distributed systems, the significant invariants concerning them are provided.

CAP Theorem. The impossibility for a distributed system to provide all 3 of the guarantees consistency, availability and partition tolerance, at the same time. Only 2 out of 3 guarantees can be provided at the same time. (Gilbert & Lynch, 2002).

4.4.3 Objectives

A system may be distributed for several reasons and distributed systems may have goals of their own. The following are considered goals of distributed systems (Tanenbaum & van Steen, 2007). These objectives may help to fulfil the system's main requirements.

Making resources accessible. It may be useful to apply distributed systems techniques to make remote resources more easily accessible to the users.

Distribution transparency. It is a goal of distributed systems to hide the fact that its processes and resources are physically distributed across multiple computers. A distributed system that seems like it is a single system can be called transparent.

Openness. The syntax and semantics of services that are offered by a distributed system must be described by standard rules. Interfaces and their behaviour must be specific precisely by protocols. Flexibility through composability in the form of easily configuring the system out of different components and being able to replace these is also mentioned under this objective by Tanenbaum and van Steen.

Scalability. There are at least three dimensions on which a system's scalability can be measured: size, i.e. scalable in the number of users or resources; geographically, i.e. scalable in the distance between users and resources; and administratively, i.e. scalable in the number of independent administrative organizations (Neuman, 1994). Such problems may be solved with distributed systems techniques like replication, sharding, etc. The geographic distribution that is a fundamental part of the energy supply system finds the applicability of distributed systems in this objective.

Reliability. On top of the other goals of distributed systems, distributed systems may also be applied to achieve reliability through data replication. When one or several computers with data on it crash, still other computers may have a copy of the data, effectively providing a form of fault tolerance.

4.4.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using this domain, they are translated to domain-specific requirements.

1. The system must be geographically scalable over a wide-area (i.e. to cover territory of the Netherlands);
2. The system must be scalable in size to provide energy to all users in the Netherlands;
3. The system's distribution must be transparent;
4. The system must be able to interoperate with many different heterogeneous energy producing devices.

4.4.5 Engineering problems

In order to fulfil the domain-specific requirements of the system as a distributed system, several engineering problems must be solved.

General pitfalls. Although not really an engineering problem. Deutsch (n.d.) formulated general mistakes in the form of false assumptions (i.e. fallacies) that developers may make when creating distributed systems. These fallacies of distributed computing are formulated as:

1. The network is reliable.
2. The network is secure.
3. The network is homogeneous.
4. The topology does not change.
5. Latency is zero.
6. Bandwidth is infinite.
7. Transport cost is zero.
8. There is one administrator.

Scaling techniques. Scaling seems easy enough. Simply replicate the data by replication of databases, servers, files or caches. However, problems may arise from scaling. Having multiple copies may lead to inconsistencies between these copies when changing data. This may be solved by updating all the copies in a transaction, but transactions require synchronization through a global clock which is not available in distributed systems. In addition, in the failure prone environment of distributed systems, it is difficult to maintain the ACID property of transactions.

Transparency. Distributed systems have to hide that everything is distributed. This allows the system to be experienced as a single system by the user. However, this means many different parts of the distribution must be hidden.

Openness. In order to interoperate well with many different devices. The system must conform to well-defined interfaces and support different policies specified by those devices.

4.4.6 Solutions

The following solutions are proposed for the engineering problems of distributed systems.

Fallacies of distributed computing. Arguably the best solution to these mistakes, is to avoid making the system distributed. However, that is not possible in the case of the energy supply system due to the geographical distribution of its components. The only other solutions is to keep these fallacies in mind in order to design and implement the system carefully without making these mistakes. Rotem-Gal-Oz (2006) provides a good starting point to avoiding these mistakes.

Synchronizing replications. The CAP theorem shows that it may be very difficult to have strong consistency, i.e. synchronous replication. However, instead of strong consistency, a system may also aim for weak consistency (i.e. best effort) or eventual consistency (i.e. asynchronous replication, that eventually will result in consistent data). As long as we tolerate inconsistencies temporarily, eventual consistency may provide a good solution. Many different technologies have been developed that deal with eventual consistency.

For storage, we may look at technologies like *HDFS* (Shvachko, Kuang, Radia, & Chansler, 2010), *MongoDB* (MongoDB Inc., 2009) or *Apache Cassandra* (Apache Software Foundation, 2008). For computing, we may look at technologies like *Apache Hadoop* (Apache Software Foundation, 2011a), *Apache Spark* (Apache Software Foundation, 2014) or *Apache Storm* (Backtype, 2011). For messaging, the technology *Apache Kafka* (Apache Software Foundation, 2011b) may provide a solution. For consensus, the technology *Apache Zookeeper* (Apache Software Foundation, 2010) might be used or one might even look at the application of *Blockchain*, the technology underpinning the cryptocurrency *Bitcoin* (BTC) (Nakamoto, 2008). Blockchain is a technology for consensus in a probabilistic state machine (Saito & Yamada, 2016). Of course, there may be many more technologies each with there own benefits and limitations.

Availability patterns. In order to achieve high availability, we may look at availability patterns. Generally, there are two patterns used to achieve high availability: fail-over; and replication.

Distribution transparency. Transparency of the distribution is a very broad problem. Several different parts of the distribution have to be hidden. Raymond (2013, page 11) describes the different transparencies:

- "access transparency — hides the differences in data representation and procedure calling mechanism to enable interworking between heterogeneous computer systems";
- "location transparency — masks the use of physical addresses, including the distinction between local versus remote";
- "relocation transparency — hides the relocation of an object and its interfaces from other objects and interfaces bound to it";
- "migration transparency — masks the relocation of an object from that object and the objects with which it interacts";
- "persistence transparency — masks the deactivation and reactivation of an object";
- "failure transparency — masks the failure and possible recovery of objects, to enhance fault tolerance";
- "replication transparency — maintains consistency of a group of replica objects with a common interface";
- "transaction transparency — hides the coordination required to satisfy the transactional properties of operations".

Support different policies. It may be best to only provide different policies through mechanisms. For illustration, we can look at secure communication over SSL/TLS (Dierks, 2008). In this protocol, the encryption algorithm(s) used to encrypt and decrypt the messages are decided upon by the communicating entities themselves. A special 'hand-shake' part of the communication, that sets-up the secure connection, allows to communicate which encryption algorithm is going to be used. This way of allowing different policies through a mechanism thus allows a wide-range of interoperability between devices, where some devices may not know the currently most used algorithms. This example can be applied to the energy supply system interface and protocol design.

4.4.7 Open questions

After providing solutions to the engineering problems of producer-consumer systems, several questions are left unanswered.

Networked system. A system can only be distributed when its components are connected as a network. This means that networking capability is needed. The domain of computer networking may solve this and its sub-problems (section 4.5).

Transaction failure. Although the technologies mentioned for replication synchronization may solve some of the problems surrounding transactions, transactions failures also have to be dealt with. System failures in a more general sense are discussed in the domain of dependable systems using fault prevention, fault-tolerance, fault removal and fault forecasting techniques (section 4.6).

4.5 Computer networking

This section discusses computer networking. A computer network is a group of computers that are connected like a graph structure in which the connected computers can exchange information.

4.5.1 Terms

In order to discuss networking, several domain-specific term definitions are provided.

Connection. A connection represents a bond or link between two devices, machines, persons, systems, things or pieces of code, within the network that can be used to transfer messages between each other. A direct connection explicitly represents a direct bond or direct link between the two devices, machines, persons, systems, things or pieces of code.

Destination. A destination represents a property of a message referencing the device, machine, person, system, thing or piece of code that it wants to go to.

Node. A node represents a device, machine, person, system, thing or piece of code that is connected in the network to other devices, machines, persons, systems, things or pieces of code.

Packet. A packet represents the smallest unit of a message transferred from one device, machine, person, system, thing or piece of code to another.

Protocol. A protocol represents a set of rules that state how communication is conducted.

Source. A source represents a property of a message referencing the device, machine, person, system, thing or piece of code that it originates from.

Topology. A topology represents the structure in which the devices, machines, persons, systems, things or pieces of code are bonded or linked within the network. For example, in a bus topology all the nodes are connected to a shared link; and in a ring topology each node is connected to two other nodes such that the structure has the shape of a ring.

4.5.2 Invariants

Networks operate under specific constraints. These constraints can be considered invariants of such systems. In order to discuss networks, the significant invariants concerning them are provided.

End-to-end delay. It takes time for a packet to be transferred from a source to a destination.

Pairing. A direct connection can only exist between two nodes, except in a bus topology.

4.5.3 Objectives

The objective of a networked system depends on the reason why networking is applied. Networks can be applied when a system:

- is naturally geographically distributed;
- has a natural or artificial decomposition separating expertise (or specialism) into different nodes;
- requires cooperation or interaction between its components;
- requires reliability through the application of redundancy;
- requires security, through the application of concern separation into different nodes allowing security breaches of one node/concern while maintaining security in the other nodes/concerns;
- requires performance, through the application of horizontal scaling;
- requires a way to deal with conflicting information, through the application of competitive computations fighting for consensus.

Some of these objectives are sometimes covered better by the distributed systems domain, but they are mentioned here for completeness. The objectives above may help to fulfil the system's main requirements.

The system is naturally geographically distributed, because the producers and consumers are naturally distributed throughout the country. This was already covered in the distributed systems section (section 4.4).

The system has a natural decomposition separating expertise into different nodes, because from the energy domain we already understand the separated concepts of energy producers, energy transporters and energy consumers. In addition, when the usable form of energy is set to electrical energy, then the field of electrical engineering naturally decomposes those concepts into many more concepts through the application of different pieces of equipment for the production, transport and consumption of electrical energy.

The system requires interaction between its components for the provisioning of energy, but also for the further communication necessary for the computation operations of the system.

Reliability, security, performance and dealing with conflicting information may all be topics of interest for the system, although they have yet to arise from the main requirements and the applied domain of producer-consumer systems.

4.5.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using networking, they are translated to domain-specific requirements.

1. system nodes must be effectively able to connect to the network;
2. system nodes must be effectively able to disconnect from the network;
3. system nodes must be effectively able to interact through the network.
4. the total cost of building and maintaining the network must be minimized;
5. the network must operate sustainably;
6. the network must be optimally transparent to its users.

4.5.5 Engineering problems

In order to fulfil the domain-specific requirements of the system using networking, several engineering problems must be solved.

Costs. A system requires a lot of hardware and software to allow the functioning of a network. All nodes of the network have to run software that allows the communication using the same protocols. Development costs of the necessary hardware and software may be very high.

Complexity. A network's complexity is related to its size, topology and heterogeneity. It can be the number of nodes and different paths that exist between each pair of nodes in the network. In which case the choice of topology may help reduce the complexity of the network. Complexity can also stem from the heterogeneity of connections, nodes and protocols, in which case compatibility (both procedural and semantic) can become an important factor.

Procedural interoperability. Interaction and cooperation both require procedural as well as semantic interoperability. Procedural interoperability describes the ability of communication using a common syntax (i.e. grammar and vocabulary), as well as having the same procedures in place (e.g. for beginning and ending a conversation). Making sure that nodes in the system are procedurally interoperable may be difficult.

Synchronization in time. Nodes need to be take time to go through the steps of their protocol in the right moments of time. They will have to wait for one another. In other words, the nodes have to be synchronized in time. This can be described to be covered by technical interoperability (i.e. the ability of understanding signals).

Semantic interoperability. Interaction and cooperation both require procedural as well as semantic interoperability. Semantic interoperability describes the ability of understanding whatever is communicated between two entities. Making sure that nodes in the system are semantically interoperable may be difficult.

4.5.6 Solutions

The following solutions are proposed for the engineering problems of networks.

The internet. The costs involved with constructing (or buying) the necessary networking hardware, designing and implementing the necessary protocols and maintaining such a computer network are incredibly high and will increase with the size of the network. The easiest solution to this is to simply make use of the infrastructure that is already available: *the internet*.

The internet is already widely used by many services. For example, the *World Wide Web*, *electronic mail*, *VoIP*, *streaming media*, *multi-player online gaming* and many more, including lots of business and banking applications. The internet already provides all the general computer networking needs a system might have. The hardware and software are already in place. Scalability is provided through a layered system of protocols called the *Internet protocol suite*. In addition, the network is highly robust (Cohen, Erez, ben Avraham, & Havlin, 2000). However, it is also vulnerable to intentional attacks (Cohen, Erez, ben Avraham, & Havlin, 2001). On top of that, the internet comes with its own problems. For example, security is an issue for the internet and special care must be taken to safeguard the proper working of the system and the privacy of its users.

A custom network. Instead of using the already available internet infrastructure, we may also create our own custom network. One of the advantages of this approach is that it can be completely isolated from the internet. This may be helpful in safeguarding the system from malicious hackers. However, the biggest disadvantage is most likely the costs involved with implementing such a network. Our advice is thus to focus on using the internet and dealing with the security issues as engineering problems. For completeness, we also describe some solutions that tackle the issues of creating a custom network.

First of all, a custom network must be able to have its nodes communicate with each other. Technical interoperability provides the basis for procedural and semantic interoperability through an understanding of sending data signals. This also requires synchronization in time. Several timing solutions exist that may help with synchronization in time. For instance, a node can be event-driven (i.e. the node starts operating after an event has happened) or clock-driven (i.e. the node starts operating at a defined moment in time). Producer-consumer systems already discusses several clock-driven solutions (section 4.2.6).

Procedural and semantic interoperability have been mentioned before as engineering problems of computer networking. Solutions for procedural interoperability are predefined protocols (e.g. the *OSI model* or the *TCP/IP model*). In addition, it is necessary here to support different policies (as described in section 4.4.6) which may also require versioning of these policies. Furthermore, a common information exchange reference model (again with versioning) is necessary to help deal with the semantic interoperability.

Dealing with the complexity and time delays of a network is related to its topology, its size and its heterogeneity. Of these, only the topology can really be chosen by the architect. Many different topologies exist, each with their own benefits and drawbacks.

Lastly, in an isolated network, security must still be protected. Hackers may find ways to connect to the network and disrupt the functional correctness of the system or gather private user and usage information.

4.5.7 Open questions

After providing solutions to the engineering problems of networking, several questions are left unanswered.

Security. In networking one of the major topics of research and interest is the topic of network security. Data travelling over networks can be intercepted, read and altered if the network has not been designed against attacks and contains vulnerabilities. Security and network security in particular should be further researched in order for these solutions to be applied. This topic is discussed further in the dependable systems domain (section 4.6).

Topology In the case of a custom network, a choice of topology has not been made, yet. Which topology or which combination of topologies would work best for this system is left open for now. This allows the designers to choose what is best for their own specified networked system. Perhaps the system could even decide to grow its system in such a way that helps use the best topology for its current state. The distributed systems domain (section 4.4) should be mindful of the possibility of a changing topology due to the fallacies of distributed systems.

4.6 Dependable systems

Systems that value the quality attributes availability, reliability, safety, confidentiality, integrity and maintainability are considered dependable systems (Laprie, 1995). Systems that are dependable may continue indefinite operation, even during the occurrence of errors. Dependable systems should provide their services at any time, indefinitely, safely, correctly and should be easy to repair.

4.6.1 Terms

In order to discuss dependable systems, several domain-specific term definitions are provided.

Attributes. Attributes represents qualities of the system that determine its dependability. Such qualities can be assessed using qualitative and quantitative measures. These qualities are: availability; reliability; safety; confidentiality; integrity; and maintainability. Other qualities have been attributed to dependable systems as well, but are combinations of the ones mentioned here. However, secondary attributes may be needed to describe dependability fully. For example, accountability, authenticity and non-repudiability are examples of secondary attributes of security, which in turn is a combination of availability, confidentiality and integrity. (Laprie, 1995)

Dependability. Dependability represents "[...] the ability to deliver services that can justifiably be trusted." (Kumar, Khan, & Khan, 2015, page 314)

Environment. Environment represents "[...] other entities, i.e. other systems, including hardware, software and the physical world with its natural phenomena" (Avizienis, Laprie, Randell, & Landwehr, 2004, page 12), with which the system interacts.

Means. Means represent the approaches that can be utilised to attain dependability. These can be classified as: fault prevention; fault tolerance; fault removal; and fault forecasting. (Laprie, 1995)

Service. Service represents the behaviour of the system as perceived by the users. (Avizienis et al., 2004)

Threats. Threats represents the things that may cause the system to decrease its dependability. There are three main threats: *faults*; *errors*; and *failures*. Faults are defects of the systems, which may or may not lead to a failure. Errors are invalid system states, which may or may not lead to a failure. Failures are the difference in actual behaviour and the system's specification. (Laprie, 1995)

4.6.2 Invariants

Dependable systems operate under specific constraints. However, the domain of dependable systems is so broad that we were not able to specify these as invariants. Our invariants would not constrain the whole domain, but only subsets of the domain. Instead, these constraints are now described in the respective topics of dependable systems.

4.6.3 Objectives

Dependable systems operate to reach particular objectives. These objectives may help to fulfil the system's main requirements. We use the attributes of dependable systems as qualities that can be attained, where attaining these attributes is the goal of dependable systems. The definitions are cited from Avizienis, Laprie, Randell, et al. (2001, page 6).

Availability. "Readiness for correct service."

Reliability. "Continuity of correct service."

Safety. "Absence of catastrophic consequences on the user(s) and the environment."

Confidentiality. "Absence of unauthorized disclosure of information."

Integrity. "Absence of improper system state alterations."

Maintainability. "Ability to undergo repairs and modifications."

4.6.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using dependable systems, they are translated to domain-specific requirements.

1. The system must be effectively available, i.e. services must be readily available for usage, as much as possible.

2. The system must be effectively reliable, i.e. services must be continuously provided, as much as possible.

3. The system must be effectively safe, i.e. users and the environment should be protected from system-related catastrophic consequences, as much as possible.

4. The system must be effectively confidential, i.e. privacy-sensitive user information must be protected from unauthorized disclosure, as much as possible.

5. The system must be effectively integer, i.e. the system must block users from performing unauthorized system state alterations, as much as possible.

6. The system must be effectively and efficiently maintainable, i.e. the system must be designed such that the system can easily undergo repairs and modifications.

4.6.5 Engineering problems

All of the objectives of dependable systems are related to faults in some way. Dealing with these faults requires a combination of the before mentioned means. These techniques are defined by Avizienis et al. (2001, page 7) as:

- **Fault prevention.** “How to prevent the occurrence or introduction of faults.”

- **Fault tolerance.** “How to deliver correct service in the presence of faults.”

- **Fault removal.** “How to reduce the number or severity of faults.”

- **Fault forecasting.** “How to estimate the present number, the future incidence, and the likely consequences of faults.”

4.6.6 Solutions

The following solutions are proposed for the engineering problems of dependable systems.

Fault prevention techniques. The first form of dealing with faults is by simply preventing them. This means that the design, and eventually the implemented system, should be free of faults as much as possible. This can be achieved through the application of general engineering techniques as well as through the application of more specific ones. (Avizienis et al., 2004)

General engineering by itself is already concerned with the design and development of functional systems. Many general engineering techniques, like process methodologies, requirement specification techniques, architecture design, modelling techniques, implementation tools, etc. help us design and development systems that function the way they are supposed to. Generally, we can assume that these techniques help produce better systems than than when not applying any techniques, and as such help to prevent faults in the design and implementation of the system.

When concerned specifically with the fault prevention techniques for the design and implementation of software, we can still find many possible solutions that may be applied. Please note, that software engineering is still a young profession and not all of the engineering techniques have been proven to work scientifically. However, we know a lot about the applications of structured programming (Dahl, Dijkstra, & Hoare, 1972), typeful programming (Cardelli, 1991), information hiding and abstraction (Kiczales, 1991), modularisation (Parnas, 1972) and new paradigms like aspect-oriented programming (Kiczales et al., 1997), as well as how these relate to prevention of faults. Naturally, there may be many more approaches. Section 4.7.6 discusses collaborative software engineering tools, platforms and standards that may help prevent faults in decentralised engineering environments.

Other approaches mentioned by Avizienis et al. (2001, page 7) are: “[...] rigorous design rules for hardware. Shielding, radiation hardening, etc., intend to prevent operational physical faults, while training, rigorous procedures for maintenance, ‘foolproof’ packages, intend to prevent interaction faults. Firewalls and similar defenses intend to prevent malicious faults.”.

Fault tolerance techniques. Fault tolerance is a combination of *fault acceptance* and *failure avoidance*. It accepts that faults are present and that errors will occur, but it will avoid failures that consequently happen from these errors. This is generally achieved through *error detection* and *error recovery*. A complete overview of the fault tolerance techniques is provided in figure 4.7.

Error detection requires the detection of the error and signalling to the system of said error. Two forms of error detection techniques exist: *concurrent error detection*; and *pre-emptive error detection*. Concurrent error detection is performed during system operation, while pre-emptive error detection temporarily suspends system operation to check for latent errors.

Errors maybe left undetected, which may or may not lead to a failure. This is called a *latent error*, or in more general terms a *false positive*. It may also happen that an error is detected, while in reality it could not lead to a failure (i.e. it was not a fault after all). This is called a *false negative*.

Error recovery is the act of changing the system state from containing errors to a system state without those errors. Additionally, error recovery may help such that a fault cannot introduce new errors again. Recovery is made-up of two parts: *error handling* and *fault handling*.

Error handling is the part that changes the system state to remove the detected errors. There are three approaches to this:

- *Rollback* can be used to change the system state back to a saved state before the error was detected. Generally, this saved state is a checkpoint where there were no detected errors.
- *Compensation* uses redundancy to remove the error.
- *Rollforward* makes the new state the state without detected errors.

Fault handling is the second part of error recovery. It is the part that makes sure that a fault that caused the detected error, cannot introduce new errors again. Fault handling takes the following four steps:

1. *Fault diagnosis* is the localisation of the fault that caused the detected error.
2. *Fault isolation* is the exclusion of the component that contains the fault from the system.
3. *System reconfiguration* puts a new component in place of the excluded component.
4. *System reinitialisation* actually changes the system to the new configuration.

Fault removal techniques. Fault removal is the act of removing faults (i.e. defects) from the system. This can be done during development and during operation of the system.

There are three steps to fault removal during development of the system: *verification*, in which particular system properties (i.e. verification conditions) are checked; *diagnoses*, in which the fault is identified and which only happens

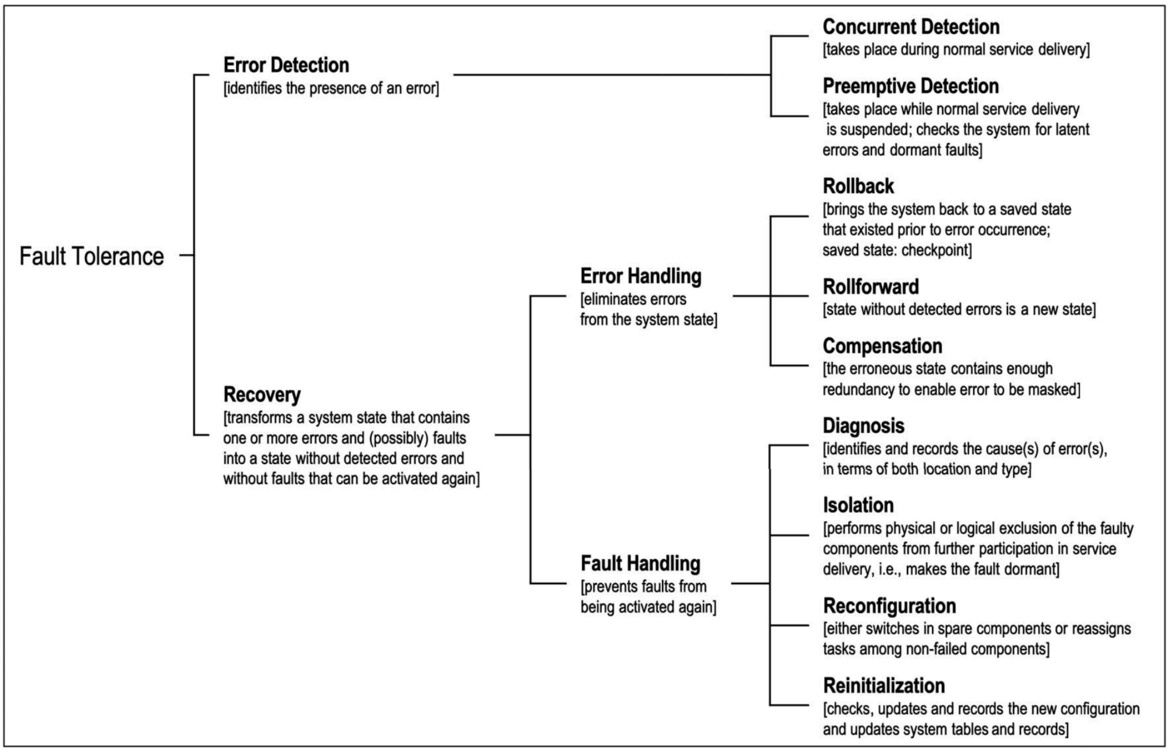

Figure 4.7: Fault tolerance techniques (Avizienis et al., 2004)

when the verification conditions are not met; and *correction*, in which the fault is removed. These three steps must be repeated after the correction to make sure that the removal of the fault did not cause other faults.

Several approaches can be taken to verification: static analysis, theorem proving, model checking, symbolic execution and testing.

Fault forecasting techniques. Fault forecasting can be used to predict likely faults, so they can be prevented. This is achieved through a system behaviour evaluation considering fault occurrence.

Fault forecasting is either performed through *qualitative evaluation* by identifying, classifying, ranking the component failures and environmental conditions under which the system would fail; or through *quantitative evaluation* by measuring and evaluating the extent to which the dependability attributes are satisfied, in terms of probabilities.

Possible methods for qualitative evaluations are *failure mode and effect analysis*. Possible methods for quantitative evaluations are *Markov chains* and *stochastic Petri nets*. In addition, *reliability block diagrams* and *fault-trees* can be used for both forms of evaluations.

4.6.7 Open questions

After providing solutions to the engineering problems of dependable systems, several questions are left unanswered.

Precision. False positives and false negatives may be reduced by a more precise fault model (used for fault detection). However, such models do not scale well. Reducing the model's precision may lead to false negatives (i.e. latent errors). Similarly, the model may also contain mistakes. Fault models with mistakes in them may lead to false positives and false negatives. Increasing the model's precision may lead to more mistakes in the fault model.

Measurements. The field of measurement theory may be required to deal with the detection of faults.

Security. The domain of dependable systems touches on the topic of security as a secondary attribute consisting of a combination of availability, confidentiality and integrity. This means that the solutions described in this section are applicable to security, where faults are security vulnerabilities, errors are security breaches and failures are consequences of security breaches. However, security has very specific problems and solutions that require further detailing, especially in the case of a networked system.

4.7 Systems-of-systems

Traditionally, in systems engineering, a system provides its functionality as a single entity, machine or product. The field of distributed systems has greatly changed our perspective of what constitutes as a system. Today, we understand that large-scale geographically distributed systems can also operate to provide a function like a single system. Some contemporary systems are composed of multiple integrated systems, each operating to provide their own function, but also operating together providing new emergent behaviour. These systems are studied in the relatively new domain of systems-of-systems (SoSs) and their design and implementation is actively researched and applied in the domain of system-of-systems engineering (SoSE).

4.7.1 Terms

In order to discuss system-of-systems, several domain-specific term definitions are provided.

Component system. A component system represents a part of the system that exist as a system in its own right. The system-of-systems is composed of component systems.

Emergent behaviour. Emergent behaviour represents a form of behaviour exhibited by a system due the collaboration of its component systems, where the component systems do not exhibit this behaviour by themselves.

Independence. Independence represents the ability of a component system to fully functionally operate without the existence of other component systems or the system-of-systems. In other words, the component system can operate independently from the system-of-systems.

4.7.2 Invariants

Systems-of-systems is a widely accepted grouping term for systems consisting of parts that are systems in their own right. However, the term system-of-systems has been the topic of discussion in the literature. According to Maier (1996, page 268), a system can be called a system-of-systems, or rather a *collaborative system*, when both:

1. "Its components fulfilled valid purposes in their own right and continued to operate to fulfill those purposes if disassembled from the overall system, and;"

2. "The components systems are managed (at least in part) for their own purposes rather than the purposes of the whole."

According to Tekinerdogan (2016, page 21), "The concept of system and system-of-systems is generally applicable to different categories of systems including:"

- "*Technological Systems*: include man-made engineered artifacts or constructs; including physical hardware, software and information.";
- "*Social Systems*: include elements, either abstract human types or social constructs, or concrete individuals or social groups.";
- "*Natural Systems*: include elements, objects or concepts which exist outside of any practical human control.".

4.7.3 Objectives

System-of-systems engineering can be applied to achieve a particular objective. This objective may help to fulfil the system's main requirements.

"Systems-of-systems should be distinguished from large but monolithic systems by the independence of their components, their evolutionary nature, emergent behaviors, and a geographic extent that limits the interaction of their components to information exchange." (Maier, 1996, Summary, page 1)

From this, we may assume that systems-of-systems have the following objectives:

Component independence. Component systems have the ability to provide their own function independently from the larger system. They do not need the larger system to operate. However, the larger system is dependent on the component systems to provide its function.

Emergent behaviour. Each component system provides its own function. Through collaboration of the component systems, together all of the component systems provide an emergent behaviour that can be experienced as the service of a coherent system.

Evolvability. Systems-of-systems maintain a high level of evolvability. According to Rowe, Leaney, and Lowe (1994, page 4), evolvability is "[...] simply a system's ability to accept change.". It is composed of *generality*, *adaptability*, *scalability* and *extensibility*.

4.7.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using system-of-systems, they are translated to domain-specific requirements.

1. The system must be coherently composed of smaller independent component systems.

2. The component systems must be able to collaborate with heterogeneous consuming devices.

4.7.5 Engineering problems

In order to fulfil the domain-specific requirements of the system using systems-of-systems, several engineering problems must be solved.

Collaboration. In order for the component systems to be able to collaborate, they have to be able to communicate with each other. Similar to the communication problems of computer networking (section 4.5.5), these component systems must have procedural interoperability (i.e. the ability of communication using a common syntax) and semantic interoperability (i.e. the ability of understanding whatever is communicated between two entities). Additionally, it may be necessary for the component systems to have pragmatic interoperability (i.e. the ability to know what procedures and methods are available on each system). For systems-of-systems it is often described that most of the engineering effort should be directed at the interfaces, for these allow the communication and thus the collaboration of the component systems.

Decentralised engineering. Engineering of a system-of-systems is different from the engineering of a traditional single system. A system-of-systems' components are likely to be designed, developed, operated, maintained and evolved in a decentralised manner. Special techniques are required for each of these decentralised engineering tasks.

Operational independence of components. Component systems must be able to operate independently. They must aim to be self-reliant. In addition, this requires an engineering effort to design component systems to be able to deal with failure of other component systems.

Managerial independence of components. Component systems not only *can* operate independently, but actually *do* operate independently. This distinction emphasizes the difference between the ability of operating independently and actually operating autonomously [4].

4.7.6 Solutions

The following solutions are proposed for the engineering problems of systems-of-systems.

Collaborative software engineering. The open source communities have been developing software in a decentralised manner for many years. Their efforts have helped improve how software can be developed as a collaboration between people without organisational assumptions. Git, a decentralized version control system, has provided a way for decentralised software development. New collaboration platforms, e.g. GitHub, Bitbucket and GitLab, and special so-called branching workflows, provide further communication tools and standards necessary for large-scale decentralised software development. Other platforms, e.g. Trello and JIRA, provide the communication tools necessary for decentralised project management. This is just a small drop in the vast ocean of collaborative software engineering tools, platforms and standards. There are tools for automated code analysis, testing and reporting (i.e. continuous integration) and tools for automated deployment of releasable code (i.e. continuous deployment). It must be clear that many solutions exist to deal with the problems of decentralised engineering.

Interfaces, protocols and standards. Design principles for systems-of-systems are focussed in interfaces and policies (i.e. protocols and standards).

[4]autonomous is a synonym for independence, but is here meant as automated independence

Maier (1996, page 283) writes: “A collaborative system is defined by its interfaces. The interfaces, whether thought of as the actual physical interconnections or as higher level service abstractions, are the primary points at which the designer can exert control. [...] The collaborative system designer will not have coercive control over the systems configuration and evolution. This makes choosing the points at which to influence the design more important. In communication-centric systems, this means that design leverage will frequently be found in relatively abstract components (like data standards and network protocols).”. It is clear from his description that the solutions for collaboration are solved by careful attention to the design of interfaces, protocols and standards.

Stable intermediate forms. According to Rechtin (1991, page 91), “Complex systems will develop and evolve within an overall architecture much more rapidly if there are stable intermediate forms than if there are not.”. This means that a complex systems, like a system-of-systems, should be designed in such a way that it allows component systems to be fully operational without the existence of the complete system-of-systems. This allows that the creation of the system-of-systems is done in small steps, each time adding a new component system that can operate on its own and provide its own function. Then slowly, as more component systems are connected, the system-of-systems can start to fulfil its purpose. This can be solved by careful attention to the existence and stability of component systems (Maier, 1996). This also deals with the evolvability of a system-of-systems.

4.7.7 Open questions

After providing solutions to the engineering problems of systems-of-systems, a question is left unanswered.

Application. The solutions presented above only give an introductory overview of how the system must be engineered. Further research to the application of these solutions to the energy supply system is required. Interfaces, protocols and standards must be designed for the communication and collaboration purposes on which the system will be constructed.

Socio-technical perspective. This thesis already provides a societal perspective towards the needs of the system. However, Tekinerdogan (2016, page 28) discusses the importance of further research of a socio-technical perspective: “In SoS [systems-of-systems], people are not just users of the system but will be an active part who design, develop, use, test and maintain the system. Hence, it will be hard to understand the aspects of an SoS without full consideration of the human behavior in the system. When designing current systems, the focus tends to be on technology, and the functional and quality concerns of the technical system. A socio-technical perspective that takes into account both the technical and the human aspects is largely missing. Hereby, not only the behavior of individuals but also the collective behavior of groups of users and developers will need to be analyzed to get an insight in how the SoS is used, viewed, accepted, and maintained."

4.8 Accounting

The fifth main requirement (section 3.2.2) of the system is keeping-out commercial driving forces. Businesses and governments should not be allowed to meddle with the system for their financial gains, because that is not in the benefit of the public from the point of view that society needs energy.

This may be achieved via laws, policies, rules and regulations. However, the system has no control over these. Instead, governments have control over legislation. Therefore, such policies may not be able to fully keep-out commercial driving forces, after all. It is therefore necessary that we take on this problem from an engineering perspective.

We will use the domain of accounting to workout the details regarding this problem.

4.8.1 Terms

In order to discuss accounting, several domain-specific term definitions are provided.

Accounting. Keeping track of information. Also known as bookkeeping.

Ledger. An account book in which transactions are recorded.

4.8.2 Invariants

We were unable to define invariants for accounting.

4.8.3 Objectives

Accounting is done to achieve a particular objective. This objective may help to fulfil the system's main requirements.

Bookkeeping. Accounting is performed to keep track of transactions and to be able to mutually agree on what was recorded.

For the energy supply system, accounting is needed to keep track of which components are part of the system. This arises partially from the horizontal scaling of producers and buffers in section 4.2, but also from the distributed nature of the system. It must be able to scale together with the number of connected users, producers, buffers and with the geographic size of the system.

When components can simply be added to the system by anyone, then businesses and governments are allowed to make the decisions about which components to add, when to add them and where to add them, instead of letting the needs of the system, which reflect the needs of society, determine these. Therefore, the system must be the one to make these decisions. We've reflected that in section 4.2 in the part about horizontally scaling producers and buffers, as well.

In order for the system to be able to make these decisions, the system must make sure that other decisions are simply not allowed. In this case, that results in not allowing any components to be added to the system that have not been approved by the system. This requires accounting.

4.8.4 Domain-specific requirements

In order to fulfil the main requirements (section 3.2.2) using accounting, they are translated to domain-specific requirements.

1. The system must account new components as part of the system.

2. Only the system may account new components as part of the system.

4.8.5 Engineering problems

In order to fulfil the domain-specific requirements of the system using accounting, several engineering problems must be solved.

Accounting. New components need to be accounted. What information needs to be kept track of and how can the system keep track of this information?

Distribution. The system is distributed geographically as described in section 4.4. This brings the problems of distributed systems to accounting. Most importantly, how should the system synchronise the accounted data over all of the components?

Authentication. Due to the distributed nature of the system, it is unclear whether a component can be trusted in its claim to being part of the system. How can the system make sure that a component can be trusted in its claim to being part of the system?

Authorisation. Commercial driving forces should be kept out. In order to do that, only the system is allowed to add new components. This will be accounted, but how can the system make sure that only components of the system can account new components?

4.8.6 Solutions

The following solutions are proposed for the engineering problems of accounting.

Ledger. The system must keep track of all of the components that are a part of the system. In order to distinguish between components, each component must get a unique identifier. How it receives this identifier is not yet important, but that it has one is important.

The identifier of each component that is part of the system can be stored as a list of identifiers (i.e. a list of identifiers of components of the system). In accounting, such a list is called a ledger. Ledgers are used to keep track of accounted information. A ledger keeps track of transactions. The addition of a component to the system can be considered a transaction.

If a new component connects to the system it must announce its presence in order to take part. This announcement must contain the components identifier, such that the system can verify that is part of the system. For this verification it simply needs to check whether the identifier is accounted for in the ledger. If so, the component is apparently part of the system and can join operation. If not, the system can simply ignore the component.

Distributed ledger. Due to the distributed nature of the system, there is no single component of the system that all other components can trust to keep track of the ledger with the information about the components of the system. If such a centralised ledger would be used, then the control over the ledger falls into the hands of the person, people, group, organisation, business or government that controls the computer on which this ledger is stored. Therefore, it must be clear that this information must be distributed to the components throughout the system.

Each component may manage its own copy of the ledger. A component that announces its presence can easily be verified using this copy of the ledger. Changes to ledger can be made by the component that owns that copy of the ledger. In other words, the component can update its own ledger copy. To update the ledger, the component needs to be sure that the system has added a new component to it. This can happen in two ways. Either, the component itself has made the decision that a new component must be added to the system or some other component has made this decision.

When a component makes a decision to add a new component, it needs to update its own ledger and inform the other components of the update. When it informs the other components, it needs to send along its own identifier to show that it is part of the system already. Other components can then verify this, before adding the new component to their own ledger. In addition, components must be able to ask other components for a current version of the ledger.

In order to reach eventual consistency, a component that decides to add a new component must inform all other components. However a single component will likely not be connected to all other components directly. Thus, the message must be propagated to all directly known components and from there this must be repeated until all of the components are informed. (Tschorsch & Scheuermann, 2015).

Public key cryptography. So far, each copy of the distributed ledger is managed by a single component. Components may state that they are part of the system, by sending along their identifier. However, every component may grab a copy of the ledger and simply use one of the identifiers from it to fake being part of the system. Somehow, a component must be able to verify the identity stated by the other component.

To do this, public key encryption (i.e. asymmetric cryptography) may be applied to form a digital signature. A digital signature provides non-repudiation (i.e. identification of the sender) and integrity (i.e. the message has not been altered in transit). This is achieved by a public and private key pair. Public-private key pairs are a mathematical scheme with a relation between the keys that can be used for encryption purposes.

A public-private key pair can be generated using a *key generation algorithm*. The public key is the identifier discussed previously. The *private key* is only known by the sender and is used together with the *message* to form a *signature* for the message. Forming this signature is performed using a *signing algorithm*. The *public key* can be used together with the message and the signature to verify the authenticity of the message. Many different digital signature algorithms are available for this purpose. For example, RSA-based signature schemes such as RSA-PSS and the Elliptic Curve Digital Signature Algorithm (ECDSA). (Johnson, Menezes, & Vanstone, 2001; Jonsson & Kaliski, 2003)

Blockchain. The distributed ledger is updated individually and concurrently for each of the components. Although eventual consistency can be guaranteed, consensus over the current state of the ledger may still pose difficulties. Ledger verification may be necessary for new components to trust the copy of the ledger they received from another component.

In order to verify a ledger for correctness, event ordering can be applied. Updating the ledger with a new component adds a new ledger state that references the previous ledger state. For any ledger state, the currently known components are defined. In addition, the component that added that particular component to the ledger must also be noted. Now it is possible for any component to verify whether a particular ledger state is valid. Valid, in this case, means that each component can only be added to a ledger state by a component that exists in the previous ledger state. Using this a component can thus be sure that only components of the system have added other components to this ledger.

Using ledger states in this way, effectively turns the distributed system into a distributed finite state machine. Such a system is generally called a *blockchain* (Tschorsch & Scheuermann, 2015).

Due to the distributed nature of blockchains, multiple blocks may be created before all the nodes have received notice. This leads to chain branching, because both created blocks will refer to the same previously accepted block. When a chain branches, there exist multiple states in the distributed finite state machine, at the same time. For a node, it thus unclear whether its version of the state is also the accepted state of the system as a whole. If we allow one branch as the accepted state of the system, we can continue to reason about the authorisation of components. If branching occurs, a choice must be made. Only one of the branches should be allowed to continue.

This can be achieved by simply making a choice of which branch to accept as the current state. One approach could be to choose the longest chain or the first to arrive when both chains are of equal length. Choosing the longest chain will allow the chain with the most information to be the accepted state. Another approach would be to always use the branch with the latest time stamp. However, a problem with all of these approaches it that it disregards the state information in the other branches.

Another approach is to try to merge the branches. Let's imagine a node that receives two different blocks of two branched chains. These blocks were both created by components that were added to the chain many blocks ago. Even though, these blocks were created around the same time, no problem arises from simply using both. Their information (i.e. the newly added components) can simply both be accepted as the truth. When that node decides to create a new block, this block may refer to both previous blocks. However, this requires additional work in authorising components, validating blocks and creating blocks.

Lastly, a hard-coded root block is necessary to make sure that the system always keeps control of its own growth. If a completely new chain would be accepted, there is no way to make sure that all of the components were added by the system itself.

4.8.7 Open questions

After providing solutions to the engineering problems of accounting, a question is left unanswered.

New technologies. Distributed ledgers and blockchains are still new technologies. It is often unclear whether a blockchain is necessary or that a more general distributed ledger already provides enough functionality. In addition, they are the topic of new research, which must be considered in the application of these technologies.

Chapter 5

Discussions

The synthesis process has shown there is a multitude of solution techniques that may be synthesised into a system architecture that satisfies the specified requirements of the future societal energy supply system. However, it is important to discuss potential shortcomings in our approach and its outcomes.

This chapter discusses and evaluates the assumptions of the requirements specification from the perspective of the system's ultimate objective in section 5.1 and discusses: (1) the applicability of synthesis to a feasibility analysis; (2) the feasibility of the applying the solution techniques; and (3) the possibility of emergent problems that may arise from putting together the selected solution techniques in section 5.2. As our research is multi-disciplinary in nature, the discussion draws on some established theories in a variety of disciplines.

5.1 Evaluation of the objective

We've considered: The system's ultimate goal is to provide energy in a usable form abundantly for society. However, whether this ultimate goal is the optimum for providing energy to society's members is debatable. Let us breakdown the goal, in order to discuss it.

It's difficult to debate the first parts of the goal. Providing energy in a usable form is at the basics of energy supplying. Not providing energy would not make for a functional energy supply system. Furthermore, providing energy in a non-usable form does make for a functional energy supply system, but it is not effective when the users are unable to effectively and efficiently use that energy. So it is clear that energy must be provided in a usable form.

The second part of the goal is debatable. Providing anything abundantly implies a form of overproduction. When cost minimization is important and production uses up resources, overproduction is never the optimum. Nevertheless, in the case of this system, we've chosen for production using renewable resources only. Overproduction in such a system does not affect the cost. Cost in such a system is only based on the other factors, like operational costs, maintenance costs, etc. Considering cost minimization, overproduction should not be a problem.

However, abundance and overproduction may involve other problems. Consumption of energy does not dispose of the used energy. The first law of thermodynamics tells us that energy is always conserved. In other words, it cannot be consumed in the sense that it is disposed of. The second law of thermodynamics tells us that entropy always increases in an isolated system. This can be read as: the energy may be converted (i.e. used) to drive some device or perform mechanical work, but it will never be optimally efficient. Generally, some of that energy will be transformed into a waste product. Typically this is in the form of heat. (Cengel & Boles, 2002)

If overproduced energy is wasted, for example as heat, then that process cannot be reversed unless external work is performed. An abundance of energy may, by definition of overproduction, involve an increase in heat in our planet's atmosphere. We may need to build systems to actively get rid of this heat, for instance by disposing it into outer-space, to fulfil the sustainability requirement of the system. Further research to the effects of overproduction on the global temperature is needed. In addition, from the results of such a study other research activities may be necessary to design such heat disposal systems.

Furthermore, it is uncertain what effects an abundance of energy may have on society. It is likely that the transport of people and goods will both become cheaper. It seems clear from consumer theory that consumer behaviour increases with the decrease of costs. Many consumer goods require energy in their production, so a decrease in costs and an increase in consumer behaviour may result from an abundance of energy. This may in turn have consequences to an increase of industry and advancements in technology. Population may increase as healthcare improves through technological advancements. Overpopulation may become a problem.

In addition, other natural resources may become scarcer. For example, as population increases, the demand for land increases as well, while the supply for land is mostly invariable. Energy may no longer be a reason to fight wars, but wars may be continued (or started) over other such scarce resources. Research has already provided some insights into some of these impacts (Meadows, Meadows, Randers, & Behrens III, 1972; Meyer & Nørgaard, 2010; Van Vuuren & Faber, 2009).

Further research to the effects of an abundance of energy on our society is necessary before such an energy supply system should be applied without human control. Moreover, a reduction in overall energy consumption may also be necessary to further limit environmental impact.

5.2 Evaluation of synthesis applicability

Synthesis is a great framework for engineers to solve problems, because it allows us to deal with concerns by splitting them up both horizontally as technical problems and vertically as abstractions of solution techniques. There is a sense of abstraction in this approach, because each solution technique may solve a technical problem, but can from another perspective also be the root of new technical problems.

Richard Feynman, sometimes referred to as *The Great Explainer*, was a renowned American theoretical physicist and Nobel Prize winner. He was able to explain highly complex ideas in easily understandable ways. His methods for explaining include, among others, a sense of abstraction with a focus on the level of detail given a particular context. He could explain something to a person at the level of detail the other person was comfortable with. This means that in order to explain something, often is has to be simplified or abstracted to mean something to the other person. Of course there's more to explaining than this, but it seems a good metaphor for the concept of abstraction that is applied in the synthesis process. A solution to a problem can be simple and solve the problem, yet it may pose new questions to the engineer.

When synthesis is applied to a problem, it is often a quite idealistic way of looking at the problem. A simple solution may already be enough to solve the problem. For example, in order to end all wars, the solution is simple: just stop fighting. However, that solution may not be as simple to apply, because there are reasons that people are fighting.

Previously, we've argued that a feasibility analysis can be performed using the synthesis process by simply checking whether a path can be found connecting problems and solutions. However, from the war example it must be clear that synthesis does not have to provide a clear answer to the feasibility of the construction of a system by just solving high level problems. The selected solution techniques may solve the technical problems at that level of abstraction, but may pose new problems that may not be trivial to solve. A proper synthesis process should thus work all the way down to trivial solution techniques that solve the technical problems. The same holds for a perfect feasibility analysis.

However, a feasibility analysis does not have to solve the complete problem. It should indicate whether it is clear or not that a problem can be solved. This implies an accuracy or a probability. For the purpose of this thesis, we've performed a synthesis process that solves the main problems of a societal energy supply system, to the extend where we can understand the major pitfalls and challenges that still need to be overcome in order to actually construct it. This provides a feasibility analysis to the extend where we can be quite certain that it must be possible to synthesise a solution architecture.

This also means that a lot of those challenges are not trivial to solve and some may in fact be considered among the NP-hard problems as described in Van Leeuwen (1991). Further research activities are required to fully understand the feasibility of implementing each of the solution techniques.

Furthermore, when adequate solution techniques are synthesised, emergent cases may show up. In other words, other problems may appear when solutions are put together. There may be unforeseen emergent problems in the provided solution domain analysis. This means we cannot claim that the provided solution would resolve the energy problem completely. There may always be additional problems that show up due to a combination of technology.

Chapter 6

Future work

As discussed, the following research activities still require further attention.

1. **Justification of the requirement specification.** This thesis only provides a first work towards the requirements of a societal energy supply system. Scientists, domain experts and visionaries may help to justify these requirements or to otherwise improve them. However, it must be clear that potential commercial interests of contributors should be identified and their motivations questioned.

2. **Justification of the solution domain analyses.** Well-defined experts in the solution domain areas can provide their opinions on the provided solution techniques and rationales to justify their validity.

3. **More detailed solution domain analyses.** Our discussion on the applicability of synthesis (section 5.2) suggests that it is necessary to do more detailed analyses. Additional solution domain analyses may be performed both horizontally (i.e. other solution domains may be analysed) as well as vertically (i.e. the provided solution domain analyses may be analysed in more detail). Only then, we could achieve an optimal solution.

4. **Alternative design space analysis.** The performed feasibility analysis can be perceived as very abstract, so it may look like there are no alternative solutions, but actually there are many alternative solutions possible. In this thesis, we have sometimes provided several different solution techniques for a single technical problem. A thorough alternative design space analysis should be performed to clearly depict the possible solution alternatives.

5. **Architecture specification.** This thesis has only performed a feasibility analysis. It may be expanded by actually synthesising an architecture specification, after an alternative design space analysis has been performed. This requires additional work in all of the research activities above.

Chapter 7

Conclusions

In this final chapter, the research question posed at the start of this thesis is answered. This is followed by a short discussion of contributions provided by this work.

This thesis set out to perform a feasibility analysis for an electricity supply system, which is specifically designed to fit the needs of society as much as possible. This is reflected by the research question:

> *Is it possible to design an electricity supply system, that satisfies the needs of society as much as possible?*

In order to answer this question, we have split it up into several sub-questions. Let us answer these one by one. The first research sub-question is:

> *Does the current electricity supply system satisfy the needs of society as much as possible?*

Our analysis of the current electricity supply system has shown that political and commercial driving forces have played a significant role in the shaping of the system to its current form. Especially, these commercial driving forces seem to clash with the needs of our society for this system. Our analysis of society's needs of the electricity supply system has made it clear that our current system does not satisfy the needs of society as much as possible. The second research sub-question is:

> *What are the requirements for a societal electricity supply system?*

Our analysis of society's needs of the electricity supply system has provided us with our first insights into what the requirements of such a system should be. A change in the understanding of the project's objective has expanded the notion of an electricity supply system to an energy supply system. The requirements have been arrived at from iterated discussions, meetings and document exchanges with the client. Our rationale explains our reasoning of these requirements. To summarise, the requirements of a societal energy supply system as defined by section 3.2.2 are:

1. *Energy* must be *provided* to the *users* such that *society*'s *energy demand* is *effectively* met;
2. *The total cost* must be *minimized*;
3. *The system* must *operate sustainably*;
4. *The system* must be *optimally transparent* to its *users*;
5. *The system* may not be *influenced* by commercial *driving forces*.

From our solution domain analysis in chapter 4, additional domain-specific requirements are added to these main requirements. The third research sub-question is:

> *Can we solve the engineering problems involved with a societal electricity supply system?*

Similar to the second research sub-question, the notion of an electricity supply system has changed to an energy supply system. Using the architecture synthesis process a feasibility analysis on the requirements defined in section 3.2.2 has been performed. In this process, only the technical problem analysis and the solution domain analysis have been applied for the purpose of this feasibility analysis.

The solution domain analysis has been able to find solution abstractions to each of the technical problems posed in the technical problem analysis. Solution abstractions often posed new technical problems. We've been able to describe solution abstractions to the technical problems at a high-level perspective. Technical problems that have arisen from particular solution abstractions are not always trivial to solve, but we've discussed current and future research activities focussing on these problem domains. It is, at this moment, not yet possible to claim that the system is feasible due to these non-trivial problems and potential unforeseen emergent cases from the solution synthesis.

However, a feasibility analysis does not have to solve the complete problem. It should indicate whether it is clear or not that a problem can be solved. For the purpose of this thesis, we've performed a feasibility analysis to the extent where we can understand the major pitfalls and challenges that still need to be overcome in order to actually construct it. From this analysis, we are quite certain that it is possible to synthesise a solution architecture.

References

abundance. (n.d.). *Dictionary.com Unabridged.* Retrieved January 10, 2017, from http://www.dictionary.com/browse/abundance

Akşit, M. (2004). The 7 c's for creating living software: A research perspective for quality-oriented software engineering. *Turkish Journal of Electrical Engineering & Computer Sciences*, *12*(2), 61–96.

Apache Software Foundation. (2008). *Apache cassandra.* Computer software. Retrieved March 16, 2017, from https://cassandra.apache.org/

Apache Software Foundation. (2010). *Apache zookeeper.* Computer software. Retrieved March 16, 2017, from https://zookeeper.apache.org/

Apache Software Foundation. (2011a). *Apache hadoop.* Computer software. Retrieved March 16, 2017, from https://hadoop.apache.org/

Apache Software Foundation. (2011b). *Apache kafka.* Computer software. Retrieved March 16, 2017, from https://kafka.apache.org/

Apache Software Foundation. (2014). *Apache spark.* Computer software. Retrieved March 16, 2017, from https://spark.apache.org/

Avizienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, *1*(1), 11–33.

Avizienis, A., Laprie, J.-C., Randell, B., et al. (2001). *Fundamental concepts of dependability.* University of Newcastle upon Tyne, Computing Science.

Backtype. (2011). *Apache hadoop.* Computer software. Retrieved March 16, 2017, from https://storm.apache.org/

Biersteker, T. J., & Weber, C. (1996). The social construction of state sovereignty. *Cambridge Studies in International Relations*, *46*(1), 1–21.

Cardelli, L. (1991). *Typeful programming.* Springer-Verlag.

Cena, G., Scanzio, S., Valenzano, A., & Zunino, C. (2015). Implementation and evaluation of the reference broadcast infrastructure synchronization protocol. *IEEE Transactions on Industrial Informatics*, *11*(3), 801–811.

Cengel, Y. A., & Boles, M. A. (2002). Thermodynamics: an engineering approach. *Sea*, *1000*, 8862.

Cohen, R., Erez, K., ben Avraham, D., & Havlin, S. (2000, November). Resilience of the internet to random breakdowns. *Phys. Rev. Lett.*, *85*, 4626–4628. doi: 10.1103/PhysRevLett.85.4626

Cohen, R., Erez, K., ben Avraham, D., & Havlin, S. (2001, April). Breakdown of the internet under intentional attack. *Phys. Rev. Lett.*, *86*, 3682–3685. doi: 10.1103/PhysRevLett.86.3682

Coplien, J., Hoffman, D., & Weiss, D. (1998). Commonality and variability in software engineering. *IEEE software*, *15*(6), 37–45.

cost. (n.d.). *Dictionary.com Unabridged.* Retrieved February 15, 2017, from `http://www.dictionary.com/browse/cost`

Courtois, P.-J., Heymans, F., & Parnas, D. L. (1971). Concurrent control with “readers” and “writers”. *Communications of the ACM*, *14*(10), 667–668.

Cristian, F. (1989). Probabilistic clock synchronization. *Distributed computing*, *3*(3), 146–158.

Dahl, O.-J., Dijkstra, E. W., & Hoare, C. A. R. (1972). *Structured programming.* Academic Press Ltd.

de Haan, F. J., Ferguson, B. C., Adamowicz, R. C., Johnstone, P., Brown, R. R., & Wong, T. H. (2014). The needs of society: A new understanding of transitions, sustainability and liveability. *Technological Forecasting and Social Change*, *85*, 121–132.

Deutsch, P. (n.d.). *The eight fallacies of distributed computing.* Website. Retrieved March 13, 2017, from `https://blogs.oracle.com/jag/resource/Fallacies.html`

De Vries, B. J., Van Vuuren, D. P., & Hoogwijk, M. M. (2007). Renewable energy sources: Their global potential for the first-half of the 21st century at a global level: An integrated approach. *Energy policy*, *35*(4), 2590–2610.

Dierks, T. (2008). The transport layer security (tls) protocol version 1.2.

Dijkstra, E. W. (1968). Cooperating sequential processes. In *The origin of concurrent programming* (pp. 65–138). Springer.

Dijkstra, E. W. (1972). Information streams sharing a finite buffer. *Information Processing Letters*, *1*(5), 179–180.

driving force. (n.d.). *Collins English Dictionary – Complete and Unabridged, 12th Edition 2014. (1991, 1994, 1998, 2000, 2003, 2006, 2007, 2009, 2011, 2014).* Retrieved February 15, 2017, from `http://www.thefreedictionary.com/driving+force`

Elektriciteitswet. (1998, July 2). Law. Retrieved November 2, 2016, from `http://wetten.overheid.nl/BWBR0009755/2016-07-01`

energy. (n.d.). *The American Heritage® Science Dictionary.* Retrieved January 10, 2017, from `http://www.dictionary.com/browse/energy`

ENTSO-E. (2014, September). *Entso-e at a glance* (Tech. Rep.). European

Network of Transmission System Operators for Electricity. Retrieved November 2, 2016, from https://www.entsoe.eu/Documents/Publications/ENTSO-E%20general%20publications/140925_ENTSO-E_at_a_Glance.pdf

Etik, N., Allahverdi, N., Sert, I. U., & Saritas, I. (2009). Fuzzy expert system design for operating room air-condition control systems. *Expert Systems with Applications*, *36*(6), 9753–9758.

Ge, S. S., Hang, C. C., Lee, T. H., & Zhang, T. (2013). *Stable adaptive neural network control* (Vol. 13). Springer Science & Business Media.

Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News*, *33*(2), 51–59.

Greenpeace. (2015, December 3). Factsheet kolenexit in nederland. *Greenpeace Nederland*. Retrieved November 2, 2016, from http://www.greenpeace.nl/Global/nederland/2015/klimaat%20en%20energie/Factsheet%20kolenexit%20in%20Nederland.pdf

Hansen, P. B. (1973). *Operating system principles*. Prentice-Hall, Inc.

Hermsen, R. (2011, April 11). 130 jaar elektriciteit in nederland (kinderdijk 1886 - 2016). *Stichting Willem Smit Historie Nijmegen*. Retrieved November 2, 2016, from http://www.willemsmithistorie.nl/historische-nieuwsflits/65-willemsmithistorie/uitgelicht/495-1e-elektrische-centrale-kinderdijk-125-jaar-19-04-1886-19-04-2011

Hinderer, K., Rieder, U., & Stieglitz, M. (2016). Examples of bayesian control models. In *Dynamic optimization* (pp. 411–434). Springer.

Hoare, C. A. R. (1974). Monitors: An operating system structuring concept. In *The origin of concurrent programming* (pp. 272–294). Springer.

Hoogspanningsnet. (n.d.-a). *Historische netkaarten.* Website. Retrieved November 2, 2016, from http://www.hoogspanningsnet.com/netkaarten/historische-netkaarten

Hoogspanningsnet. (n.d.-b). *Hvdc-zeekabels.* Website. Retrieved November 2, 2016, from http://www.hoogspanningsnet.com/over-hoogspanningsmasten/luchtlijnen-grondkabels/hvdc-zeekabels/

Hoogspanningsnet. (n.d.-c). *Soorten hoogspanningsstations.* Website. Retrieved November 2, 2016, from http://www.hoogspanningsnet.com/techniek/hoogspanningsstations/

Hoogwijk, M., & Graus, W. (2008). Global potential of renewable energy sources: a literature assessment. *Background report prepared by order of REN21. Ecofys, PECSNL072975*.

influence. (n.d.). *Collins English Dictionary - Complete & Unabridged 10th Edition.* Retrieved February 15, 2017, from `http://www.dictionary.com/browse/influence`

ISO/IEC. (2009). *Iso/iec cd 25010.3. systems and software engineering. systems and software quality requirements and evaluation (square). system and software quality models* (Tech. Rep.).

ISO/IEC. (2011). *Iso/iec 25010:2011. systems and software engineering. systems and software quality requirements and evaluation (square). system and software quality models* (Tech. Rep.).

ISO/IEC. (2016). *Iso/iec 25022:2016. systems and software engineering. systems and software quality requirements and evaluation (square). measurement of quality in use* (Tech. Rep.).

Jarke, M., & Pohl, K. (1994). Requirements engineering in 2001:(virtually) managing a changing reality. *Software Engineering Journal*, *9*(6), 257–266.

Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, *1*(1), 36–63.

Jonsson, J., & Kaliski, B. (2003). Public-key cryptography standards (pkcs)# 1: Rsa cryptography specifications version 2.1.

Khaitan, S. K., McCalley, J. D., & Liu, C. C. (2015). *Cyber physical systems approach to smart electric power grid*. Springer.

Kiczales, G. (1991). Towards a new model of abstraction in software engineering. In *Object orientation in operating systems, 1991. proceedings., 1991 international workshop on* (pp. 127–128).

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., & Irwin, J. (1997). Aspect-oriented programming. In *European conference on object-oriented programming* (pp. 220–242).

Kopetz, H., Damm, A., Koza, C., Mulazzani, M., Schwabl, W., Senft, C., & Zainlinger, R. (1989). Distributed fault-tolerant real-time systems: The mars approach. *IEEE Micro*, *9*(1), 25–40.

Kumar, R., Khan, S. A., & Khan, R. A. (2015). Revisiting software security: durability perspective. *International Journal of Hybrid Information Technology (SERSC)*, *8*(2), 311–322.

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, *21*(7), 558–565.

Laprie, J.-C. (1995). Dependable computing: Concepts, limits, challenges. In *Special issue of the 25th international symposium on fault-tolerant computing* (pp. 42–54).

Linkens, D. A., & Nyongesa, H. O. (1996). Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications. *IEE Proceedings-Control Theory and Applications*, *143*(4), 367–386.

Liu, J., Wang, W., Golnaraghi, F., & Kubica, E. (2010). A novel fuzzy framework for nonlinear system control. *Fuzzy Sets and Systems*, *161*(21), 2746–2759.

Lu, X., McElroy, M. B., & Kiviluoma, J. (2009). Global potential for wind-generated electricity. *Proceedings of the National Academy of Sciences*, *106*(27), 10933–10938.

Maier, M. W. (1996). Architecting principles for systems-of-systems. In *Incose international symposium* (Vol. 6, pp. 565–573).

Mainsfrequency. (n.d.). *Measurement of the mains frequency.* Website. Retrieved November 2, 2016, from http://www.mainsfrequency.com/

Maslow, A. H. (1943). A theory of human motivation. *Psychological review*, *50*(4), 370.

Meadows, D. H., Meadows, D. H., Randers, J., & Behrens III, W. W. (1972). *The limits to growth: a report to the club of rome (1972)*. Universe Books, New York.

Meyer, N. I., & Nørgaard, J. (2010). Policy means for sustainable energy scenarios. *Collection of Extended Abstracts*, 133.

minimize. (n.d.). *Dictionary.com Unabridged.* Retrieved January 10, 2017, from http://www.dictionary.com/browse/minimize

Mommers, J. (2017, March 13). *Dit zijn de partijen die klimaatverandering serieus tegengaan.* Website. De Correspondent. Retrieved March 14, 2017, from https://decorrespondent.nl/6353/dit-zijn-de-partijen-die-klimaatverandering-serieus-tegengaan/700157777-a3614274

MongoDB Inc. (2009). *Mongodb.* Computer software. Retrieved March 16, 2017, from https://www.mongodb.com/

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved November 2, 2016, from https://bitcoin.org/bitcoin.pdf

Nassif, N., Kajl, S., & Sabourin, R. (2005). Optimization of hvac control system strategy using two-objective genetic algorithm. *HVAC&R Research*, *11*(3), 459–486.

Netbeheer Nederland. (2014, April 29). *Betrouwbaarheid van elektriciteitsnetten in nederland - resultaten 2013* (Tech. Rep.). Netbeheer Nederland, Tech. Rep. RM-ME-13L10440006. Retrieved November 2, 2016, from https://www.kivi.nl/uploads/media/57e7ed5724248/169_0_nl_betrouwbaarheid_van_elektriciteitsnetten_in_nederland__resultaten_2013_definitief.pdf

network. (n.d.). *Collins english dictionary - complete & unabridged 10th edition.* Retrieved March 16, 2017, from http://www.dictionary.com/browse/network

Neuman, B. C. (1994). Scale in distributed systems. *Readings in Distributed*

Computing Systems., 1–28.

NOS. (2016a, September 2). *Sluiting nieuwe kolencentrales onvermijdelijk.* Website. Retrieved March 14, 2017, from http://nos.nl/artikel/2129512-sluiting-nieuwe-kolencentrales-onvermijdelijk.html

NOS. (2016b, July 18). *Te veel zonnepanelen in groningen.* Website. Retrieved November 2, 2016, from http://nos.nl/artikel/2118256-te-veel-zonnepanelen-in-groningen.html

NRC. (2013, August 27). *Energieakkoord bereikt: in 2020 14 procent duurzame energie.* Website. Retrieved March 14, 2017, from https://www.nrc.nl/nieuws/2013/08/27/energieakkoord-bereikt-in-2020-16-procent-duurzame-energie-a1432096

NRC. (2016, April 22). *171 landen ondertekenen klimaatakkoord parijs.* Website. Retrieved March 14, 2017, from https://www.nrc.nl/nieuws/2016/04/22/168-landen-ondertekenen-klimaatakkoord-parijs-a1407731

operate. (n.d.). *Dictionary.com Unabridged.* Retrieved January 10, 2017, from http://www.dictionary.com/browse/operate

optimum. (n.d.). *Dictionary.com Unabridged.* Retrieved January 11, 2017, from http://www.dictionary.com/browse/optimum

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, *15*(12), 1053–1058.

providing. (n.d.). *Dictionary.com Unabridged.* Retrieved January 10, 2017, from http://www.dictionary.com/browse/providing

Raymond, K. (2013). Reference model of open distributed processing (rm-odp): Introduction. *Open Distributed Processing: Experiences with distributed environments*, 1.

Rechtin, E. (1991). *Systems architecting: Creating and building complex systems*. Prentice Hall.

Rosvold, K. A. (2012, November 8). *Saurdal kraftverk.* Website. Retrieved November 2, 2016, from https://snl.no/Saurdal_kraftverk

Rotem-Gal-Oz, A. (2006). Fallacies of distributed computing explained. Website. Retrieved March 16, 2017, from https://www.cse.unsw.edu.au/~cs9243/16s1/papers/fallacies.pdf

Rowe, D., Leaney, J., & Lowe, D. (1994). Defining systems evolvability-a taxonomy of change. *Change*, *94*, 541–545.

Sadler, P. (2010). *Sustainable growth in a post-scarcity world: consumption, demand, and the poverty penalty*. Gower Publishing, Ltd.

Saito, K., & Yamada, H. (2016). What's so different about blockchain?—blockchain is a probabilistic state machine. In *Distributed computing systems workshops (icdcsw), 2016 ieee 36th international conference on* (pp. 168–175).

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The hadoop distributed file system. In *Mass storage systems and technologies (msst), 2010 ieee 26th symposium on* (pp. 1–10).

society. (n.d.). *Dictionary.com Unabridged.* Retrieved January 10, 2017, from `http://www.dictionary.com/browse/society`

sustainability. (n.d.). *Dictionary.com Unabridged.* Retrieved February 15, 2017, from `http://www.dictionary.com/browse/sustainability`

Tanenbaum, A. S., & van Steen, M. (2007). *Distributed systems: Principles and paradigms, 2nd edition.* Pearson.

Tekinerdogan, B. (2016). *Engineering connected intelligence: a socio-technical perspective* (Tech. Rep.). Wageningen University & Research. (Inaugural lecture)

Tekinerdoğan, B. (2000). Synthesis based software architecture design. *University of Twente*. (Ph.D. Thesis)

Thayer, R. H., Bailin, S. C., & Dorfman, M. (1997). *Software requirements engineerings*. IEEE Computer Society Press.

TNS NIPO. (2016, September 20). *Zorg voorlopig belangrijkste verkiezingsthema.* Website. Retrieved March 14, 2017, from `http://www.nipo.nl/nieuws/persberichten/zorg-voorlopig-belangrijkste-thema-verkiezingen/`

transparency. (n.d.). *Wikipedia.* Retrieved February 15, 2017, from `https://en.wikipedia.org/wiki/Transparency_(human-computer_interaction)`

Tschorsch, F., & Scheuermann, B. (2015). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, *18*(3), 2084–2123.

user. (n.d.). *The Free On-line Dictionary of Computing.* Retrieved February 15, 2017, from `http://www.dictionary.com/browse/user`

van den Noort, J. W. P. P. (1984). Het begin van het rotterdams gemeentelijk elektriciteitsbedrijf (1895). *Jaarboek voor de geschiedenis van bedrijf en techniek*, *1*, 288–310. Retrieved November 2, 2016, from `http://jvdn.nl/Downloads/1984/BeginGEB.pdf`

van Empelen, L. (n.d.). De grootsche gedachte van het koppelen der centrales: De aanleg van het hoogspannings-koppelnet in zuid-holland in de periode 1930-1945. Retrieved November 2, 2016, from `http://www.pv-eon-benelux.nl/Historie/1%20Transport/Transport%20documenten/EZH%20boek.pdf`

Van Leeuwen, J. (1991). *Handbook of theoretical computer science (vol. a): algorithms and complexity*. Mit Press.

van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing*, *98*(10), 967–1009.

Van Vuuren, D., & Faber, A. (2009). *Growing within limits. a report to the global assembly 2009 of the club of rome* (Tech. Rep.). Netherlands Environmental Assessment Agency PBL.

Weber, M. (1978). *Economy and society: An outline of interpretive sociology*. Univ of California Press.

Wet raadgevend referendum. (2015, July 1). Law. Retrieved November 2, 2016, from http://wetten.overheid.nl/BWBR0036443/2015-07-01

Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The internet of things—a survey of topics and trends. *Information Systems Frontiers*, *17*(2), 261–274.

Witkin, B. R., & Altschuld, J. W. (1995). *Planning and conducting needs assessments: A practical guide*. Sage.

Zhang, T., Ge, S. S., & Hang, C. C. (2000). Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. *Automatica*, *36*(12), 1835–1846.

Appendix A

History of the Dutch electricity supply system

#	Timeframe	Change	Source
1	1886	First public system (regionally centralized)	Hermsen, 2011
2	1890 and later	Connecting cities and villages over different mid voltage lines at around 10 kV, providing electricity to every household	Hoogspanningsnet, n.d.-a
3	Around 1910	First high voltage lines at 50 kV, interconnecting cities to support an increased demand	Hoogspanningsnet, n.d.-a
4	1920	First 100+ kV lines to support an increased demand	Hoogspanningsnet, n.d.-a
5	Around 1940	First fully interconnected system at national level	Hoogspanningsnet, n.d.-a
6	1940's	First connections to foreign electricity systems (i.e. Germany, Belgium)	Hoogspanningsnet, n.d.-a
7	1950's	Standardisation to 50, 100, 110, 150 kV lines	Hoogspanningsnet, n.d.-a
8	1950's	First 220 kV lines to support an increased demand	Hoogspanningsnet, n.d.-a
9	1950's	First ring structures in the network to increase fault-tolerance	Hoogspanningsnet, n.d.-a
10	1960's	Interconnected high voltage system at 110 and 150 kV	Hoogspanningsnet, n.d.-a
11	1970's	Replacement of interconnected system by extra high voltage lines at 380 kV	Hoogspanningsnet, n.d.-a
12	1980's	Ring structures in the extra high voltage network at 380 kV	Hoogspanningsnet, n.d.-a
13	1990's	Reduction of old 50, 110 and 150 kV lines	Hoogspanningsnet, n.d.-a
14	2000's	First HVDC lines (i.e. NorNed, BritNed)	Hoogspanningsnet, n.d.-a

Table A.1: History of the Dutch electricity supply system