# Exploration of state-of-the-art technology, architectures and tools to create future-proof data spaces

Master Thesis (40 ECTS),
Software Engineering,
University of Southern Denmark


Authored by Nikolai Emil Damm*

Supervised by Jakob Hviid†

May 19, 2023

*~80 pages (2400 characters with spaces per page)*

*nidam16@student.sdu.dk
†jah@mmmi.sdu.dk

# Abstract

This thesis explores whether a data space can be implemented as a data mesh, focusing on challenges and potential benefits for collaboration among actors in the Danish energy sector. The study employs constructivism and the constructive research approach as its methodology, utilizing the grounded theory method to conduct fieldwork and create theories and hypotheses from the results. The research encompasses topics such as the climate, the Danish energy sector, data spaces, and data mesh, emphasizing a prototype of a data mesh's central component, a data product. The prototype demonstrates that the data mesh approach can be successfully applied to data spaces, enabling better separation of domains within sectors and enabling discoverability, observability, and governance. However, it lacks the features and the maturity to provide a production-ready and complete solution. Overall, this thesis contributes to the ongoing discussion on future-proof data spaces and provides insights into implementing a data space as a data mesh to achieve that goal.

***Keywords***— caching, climate change, climate status, constructive research, constructivism, csharp, data catalog, data governance, data infrastructure, data input/output, data mesh, data product, data space, docker, dotnet, energinet, energy sector, feature flags, feature pattern, feature toggles, flexible energy denmark, grafana, graphql, grounded theory, jaeger, kafka, linkedin datahub, openapi, opentelemetry, postgresql, prometheus, redis, sql server, sustainable development goals, swagger

# Acknowledgements

This thesis has been a long and challenging journey, and I would like to thank several people for their help and support.

First, I would like to thank my supervisor, Jakob Hviid, for his guidance and support on all aspects of the project. His knowledge and expertise have been invaluable, and I am grateful for his patience and willingness to help me learn.

Furthermore, I thank Jens Hjort Schwee, Bjørn Therkelsen, André Bryde Alnor, Peter Lyck Ingerslev, and Jakob Hviid for insightful interviews and discussions that have helped me understand the problem domain and requirements for the project. Your interest in the project and dedication to helping has been appreciated.

Finally, I thank my family, fiancée, and friends for their support and encouragement throughout the project—especially my fiancée for her indulgence when the project occupied all of my time.

# Reading Guide

This thesis assumes that the reader has a basic understanding of software engineering. Most chapters are written at an abstraction level suitable for a general academic audience. However, some chapters require a more in-depth understanding of software engineering principles and practices. These chapters are not intended to be read by the general academic audience and can be skipped if the reader finds them too technical. Section 1.4 provides more information on possible reading paths.

Furthermore, the thesis follows a few guidelines to improve the reading experience:

- The thesis is written in American English.
- All acronyms are mentioned in full length with their abbreviation in parentheses the first time they are used. Subsequent uses are abbreviated. The Glossary before the first chapter provides a complete list of acronyms.
- Footnotes are used for explaining basic terms and concepts, as well as parenthetical phrases, that would otherwise break the flow of the text.
- References use the **IEEE** citation style indicated by "[1, p. 11]" where "1" refers to the index of the reference, and "p. 11" refers to the page number in the referenced material. References to audio and video material are indicated by "[1, ts. 00:00] where "1" refers to the index of the reference and "ts. 00:00" refers to the timestamp in the referenced material (MM:SS). The references are listed at the end of the thesis.
- Assemblies are named according to the author's GitHub username, 'Devantler'. So when encountering these in the thesis and potentially wondering what it means to the assemblies, it is simply a username the author uses to identify that the assembly belongs to him.

# Contents

# List of Figures

# List of Tables



# List of Listings

# Glossary

**ANSI** American National Standards Institute. 103, 120, 128

**API** Application Programming Interface. 9, 12, 26, 27, 28, 31, 32, 35, 37, 39, 41, 43, 44, 45, 47, 49, 56, 58, 59, 61, 62, 87, 120, 124

**BI** Business Intelligence. 90, 92, 112, 124

**CA** Certificate Authority. 8

**CD** Continuous Delivery. 37, 76, 77

**CI** Continuous Integration. 37, 76, 77

**CIA** Confidentiality, Integrity and Availability. 103, 121, 130

**CNCF** Cloud Native Computing Foundation. 2, 20, 25, 27, 58, 59, 61, 62

**$CO_2$** Carbon Dioxide. 1

**$CO_2e$** Carbon Dioxide Equivalent. 1, 71, 72, 102, 117, 126

**CORS** Cross-Origin Resource Sharing. 49

**CPU** Central Processing Unit. vi, 52, 53

**CRA** Constructive Research Approach. vi, 3, 15, 16, 17, 18, 20, 61

**CRUD** Create, Read, Update and Delete. 28, 41, 44

**CSV** Comma-Separated Values. 50

**DA** Data Act. 6

**DAA** Decade Action Agenda. 72

**DAPS** Dynamic Attribute Provisioning Service. 9

**DCA** Danish Climate Act. 71

**DEA** Danish Energy Agency. 70, 72, 97, 121, 130

**DI** Dependency Injection. 32, 33

**DLL** Dynamic Link Library. 37, 38, 42

**DMA** Data Mesh Approach. 2, 3, 5, 10, 14, 15, 16, 20, 22, 23, 24, 25, 55, 56, 57, 61

**DMCEU** Danish Ministry of Climate, Energy and Utilities. 72

**DSI** Data Spaces Initiative. 2, 3, 5, 9, 13, 14, 16, 20, 22, 23, 25, 55, 56, 57, 61, 85, 96, 99, 100, 112, 114

**DSO** Distribution System Operator. 70, 93, 106, 109, 115, 119, 121, 130, 133

**DSSC** Data Spaces Support Centre. 9

**DTM** Dynamic Trust Monitoring. 8

**ELT** Extract, Load and Transform. 92, 111

**ePR** ePrivacy Directive. 6

**ETL** Extract, Transform and Load. 92, 111

**EU** European Union. 2, 5, 6, 7, 9, 86, 89, 91, 93, 97, 98, 108, 111, 114, 116, 119, 121, 123, 125, 130, 133

**FaaS** Function as a Service. 42

**FED** Flexible Energy Denmark. 1, 2

**GDP** Gross Domestic Product. 71

**GDPR** General Data Protection Regulation. 6

**GT** Grounded Theory. 3, 15, 17, 18, 19, 20, 21, 23, 55, 61

**HTML** Hypertext Markup Language. 48

**HTTP** Hypertext Transfer Protocol. 44, 52

**IaC** Infrastructure as Code. 94

**ID** Identifier. 52

**IDA** IT, Data and Analytics. 21, 89

**IDE** Integrated Development Environment. 30, 36

**IDS** International Data Spaces. 7, 8, 9, 14

**IDSA** International Data Spaces Association. vi, 7, 9, 14

**IEC** International Electrotechnical Commission. 93, 98, 103, 119, 120, 121, 128, 130, 133

**IEEE** The Institute of Electrical and Electronics Engineers. iii

**IO** Input/Output. 92, 118

**IoT** Internet of Things. 6, 23, 25, 28, 29, 97, 102, 117, 126, 132

**IPR** Intellectual Property Rights. 6

**ISO** International Organization for Standardization. 103, 120, 128

**IT** Information Technology. 6, 9, 54, 89, 103, 118

**JSON** JavaScript Object Notation. 34, 35, 36, 50

**LINQ** Language-Integrated Query. 46

**LoRaWAN** Long Range Wide Area Network. 102, 117, 126

**MIT** Massachusetts Institute of Technology. 30

**MVC** Model-View-Controller. 43, 44

**MVNO** Mobile Virtual Network Operator. 102, 103, 117, 118, 127

**MVP** Minimum Viable Product. 26

**NDA** Non-Disclosure Agreement. 17, 80, 82

**NGO** Non-Governmental Organization. 7

**ORM** Object-Relational Mapping. 51

**OT** Operational Technology. 89, 103, 118, 127, 132

**PaaS** Platform as a Service. 13

**PR** Pull Request. vii, 75, 77, 78

**RES** Renewables Share. 1

**RES-E** Renewables Share in Electricity. 1, 69

**REST** Representational State Transfer. 35, 41, 43, 44, 52, 58, 59

**SaaS** Software as a Service. 9

**SDG** Sustainable Development Goal. 5, 71, 72, 97, 108

**SDG13** Sustainable Development Goal 13. 1, 71

**SDG7** Sustainable Development Goal 7. vii, 1, 71, 73

**SDL** Schema Definition Language. 44

**SDU** University of Southern Denmark. 21, 23, 83

**SLO** Service-level Objective. 13, 113

**SOTA** State of the Art. 3, 4, 5, 13, 17, 55, 57, 61

**SQL** Structured Query Language. 46, 51, 54, 93, 124

**SRE** Site Reliability Engineering. 92, 112

**TSO** Transmission System Operator. 70, 93, 95, 96, 97, 106, 108, 109, 115, 119, 121, 123, 130, 133

**UI** User Interface. vi, 27, 43, 44, 45, 48, 54, 60

**UN** United Nations. 1, 71

**UP** Unified Process. 19, 79

**URL** Uniform Resource Locator. 35, 49

**URN** Uniform Resource Name. 49

**VSA** Vertical Slice Architecture. 32

**YAML** YAML Ain't Markup Language. 34, 50

# 1 | Introduction

Climate change is a huge threat now and in the future. It is paramount that humanity acts towards building a sustainable future by finding new and improved ways to reduce **Carbon Dioxide Equivalent ($CO_2e$)**[1] emissions. Doing so is essential to limit or counteract the effects of climate change.

In Denmark, an increasing amount of energy consumption is based on renewable energy, such as wind and solar power. Denmark's Climate Status and Outlook for 2022 states that Denmark's total **Renewables Share (RES)**[2] is between 42% to 51% and is estimated to reach 64% by 2030 [2, p. 16]. Meanwhile, the **Renewables Share in Electricity (RES-E)**[3] is between 65% and 93% and is estimated to achieve 109% by 2030 [2, p. 16].

Even though the **RES** and the **RES-E** are high in Denmark, it presents challenges. Renewable energy production is not constant, as the weather fluctuates. Sometimes there is neither wind nor sunlight to produce more renewable energy; other times, there is a surplus of renewable energy.

> "A 680 Mega Watt solar cell factory can deliver 90% one minute, three minutes later it might deliver 40%, and 30 seconds later it might deliver 100%. Clouds obstruct the sun's rays [3, ts. 04:23]." - Peter Lyck Ingerslev

The varying nature of weather only becomes more pronounced as the **RES** increases, and the energy sector becomes more reliant on renewable energy.

> "The more sustainable energy we produce in our energy system, the more fluctuating it becomes. The closer to 100% sustainable energy we get, the more stochastic the energy systems act [3, ts. 03:47]." - Peter Lyck Ingerslev

As such, a few scenarios can occur. One scenario is that the **RES** is higher than the energy demand, and the surplus of renewable energy must be sold to other countries, stored, or potentially wasted. Another scenario is that the **RES** is lower than the energy demand, and the energy sector must rely on non-renewable energy sources to meet consumer demand. These scenarios indicate how approaching 100% renewable energy will require a flexible energy system to manage the increasing stochastic behavior in the energy sector. More specifically, a need for balancing renewable energy consumption and production to maximize the use of renewable energy arises. The importance of achieving a better balance is further emphasized by **United Nations (UN) Sustainable Development Goal 7 (SDG7)** and **Sustainable Development Goal 13 (SDG13)**, which requires all member countries to take action toward achieving affordable and clean energy for all, as well as lowering **$CO_2e$** emissions [4].

⋆ ⋆ ⋆

In modern times we have the technology to collect and process enormous amounts of data from different actors and devices. Processing this data at scale can provide new insights into possible solutions for balancing energy consumption and production. For example, one solution in Denmark is a digitalization project called **Flexible Energy Denmark (FED)** [5]. It aims to increase the flexibility of the energy sector by finding new ways to use the surplus of renewable energy. For example, when consumption is low or the conditions for producing renewable energy are favorable.

The goal of the **FED** project is to, first and foremost, balance energy consumption with renewable energy production [5]. Ultimately this can lead to lowering **$CO_2e$** emissions because it can decrease the dependence on non-renewable energy sources [5]. It will result in cost savings because the need for new

[1] **$CO_2e$** is the metric tons of **Carbon Dioxide ($CO_2$)** emissions with the same global warming potential as one metric ton of another greenhouse gas [1].

[2] **RES** is the share of renewable energy consumption compared to the total energy consumption [2, p. 16].

[3] **RES-E** is the share of renewable energy consumption compared to the total electricity consumption [2, p. 16].

energy investments is reduced. Furthermore, it can decrease material waste, as the need for expanding the energy infrastructure is diminished [5].

An obstacle for the **FED** project and the energy sector, in general, is that data is scattered across many different systems, platforms, and organizations. There is an abundance of governance and standards for storing and sharing data between these systems. As **FED** is a centralized solution that collects and processes data in a data lake, it requires integrating with the many different systems, often legacy systems. Owning the responsibility for integrating with systems that lack governance and standards is a significant challenge, and it is not a scalable solution. The above is evident given the following statements from André Bryde Alnor, Department Manager in Digitalisation at Energinet, and Jens Schwee, Digital Business Developer in Digitalisation at Energinet:

> "No matter what initiatives we take, we still have a 40-year-ish old infrastructure that we must integrate with [6, ts. 12:08]." - André Bryde Alnor
>
> "We need some central standard, and that is an aspect of centralization [7, ts. 14:26]." - Jens Schwee

Exploring and categorizing the possibilities and pathways toward improving the energy sector's infrastructure is essential. An intelligent solution that avoids significant investments into new data infrastructure but supports being built on top of existing infrastructure is needed. It must be decentralized to move decision-making processes and responsibilities to the network's edge. It requires a standardized transaction-based communication model that integrates with central systems to provide governance. Energinet is already working towards these goals and is investigating possible solutions for an industry-wide initiative called the **Data Spaces Initiative (DSI)** [8]. This initiative aims to provide a decentralized solution for multiple sectors in the **European Union (EU)**. The **DSI** is a large and complex project. Exploring the applicability of novel data infrastructures, like the **Data Mesh Approach (DMA)**, is crucial.

## 1.1 Problem and Scope

This study explores the current collaboration challenges facing the energy sector. It focuses on the **DSI** in its current state and whether or not the **DMA** can be a viable solution to implement a data space. The following section will present limitations to the study, the problem statement, and the study's objectives.

### 1.1.1 Limitations

The energy sector is a large and complex domain with many actors and stakeholders. Likewise, the **DSI** is an evolving novel project that spans multiple industries in all of **EU**. As such, the scope of this study is limited as follows:

- The problem domain is limited to Denmark's energy sector and the **DSI**.
- Exploration of possible solutions for implementing the **DSI** is limited to the **DMA**.
- The development of a prototype is limited to the central unit of the data mesh, the data product.
- The development of the prototype will rely on .NET, **Cloud Native Computing Foundation (CNCF)** and open-source technologies.
- The prototype will be hosted in a self-hosted Docker environment for development, testing, and demonstration purposes.

### 1.1.2 The Problem Statement

As indicated, the study aims to explore the applicability of the **DMA** to implement a data space, and as such, the following problem statement is formulated:

> *How can a data space be designed as a data mesh to improve collaboration between actors in the Danish energy sector?*

As part of answering the problem statement, the following research questions must be answered, as they are prerequisites for answering the problem statement:

**RQ1** What is a data space?

**RQ2** What is a data mesh?

**RQ3** What are the challenges with collaboration in the Danish energy sector?

### 1.1.3 Objectives

Answering and understanding the problem requires completing a series of objectives. The objectives are as follows:

**OBJ1** Conduct interviews with experts at Energinet to understand challenges and ideas regarding the **DSI**, collaboration, and general information about the energy sector.

**OBJ2** Research and document the **DSI** and **DMA**, and synthesize theory with findings from fieldwork.

**OBJ3** Elicit requirements for a solution to demonstrate the challenges and possibilities of implementing a data space with a **DMA**.

**OBJ4** Develop a prototype and document the process, learnings, and challenges.

**OBJ5** Evaluate the results and document the findings.

## 1.2 Research Approach

The study will employ qualitative research methods, specifically the **Constructive Research Approach (CRA)** [9] and **Grounded Theory (GT)** [10]. The **CRA** will guide the overall process, and **GT** will be used to gather and analyze data from fieldwork. The data from fieldwork will result in a set of theories and hypotheses that can provide new insights into the challenges and needs of the energy sector in Denmark and clarify the current initiatives and potential solutions that can address these challenges and needs.

The study will examine the hypotheses to establish a prototype's non-functional and functional requirements. The prototype will be developed iteratively, potentially fostering learning and insights, fitting well with the **CRA**.

The prototype developed in this study will be evaluated using a comparison to the **State of the Art (SOTA)** and an acceptance test. The research and solution will be compared to **SOTA** in data spaces and meshes to determine its theoretical validity. This comparison will determine how the knowledge from research and the solution conform and deviate from the theoretical **SOTA** of the **DSI** and the **DMA**. By comparison, it will be possible to determine the potential impact and feasibility of using the **GT** method to explore and create knowledge from real-life problems and whether the **DMA** approach applies to the challenges facing the energy sector in Denmark.

The acceptance test will also determine whether the prototype meets the non-functional and functional requirements established from the hypotheses and theories derived from the **GT** method. This test will ensure that the prototype is a practical and viable solution for implementing a data space with a data mesh architecture in the energy sector in Denmark.

## 1.3 Thesis Outline

**Chapter 1 Introduction** Introduces the thesis's topic, scope, problem, objectives, research approach, and outline.

**Chapter 2 Background and State of the Art** Provides the necessary background and required context to understand the problem domain and why it is relevant. The background also represents the theoretical **SOTA** for the **DSI** and the **DMA**.

**Chapter 3 Methodology and Plan** Presents the methodology and process used for fieldwork and development. It culminates in a project plan that is followed throughout the project.

**Chapter 4 Research and Fieldwork** Explains and documents the research and fieldwork conducted throughout the project.

**Chapter 5 Requirements** Elicits and presents the requirements for the prototype based on the hypotheses from Appendix N.

**Chapter 6 Conceptual Design** Explains and documents the solution at a high level of abstraction.

**Chapter 7 Technical Design and Implementation** Explains and documents the prototype at a low level of abstraction.

**Chapter 8 Evaluation** Evaluate the theoretical validity by comparing the results from the research and fieldwork to the theoretical **SOTA** and evaluates the practical validity of the developed prototype through acceptance testing.

**Chapter 9 Conclusion** Concludes the project and discusses possible future work.

## 1.4 Reading Paths

Fig. 1.1 outlines the possible reading paths and for which audience they are intended. There are two main reading paths, one for technical readers and one for non-technical readers. Besides the main reading paths, a few chapters are optional, and the reader can skip them if they are not interested in all aspects of the project. Stippled lines indicate optional paths.

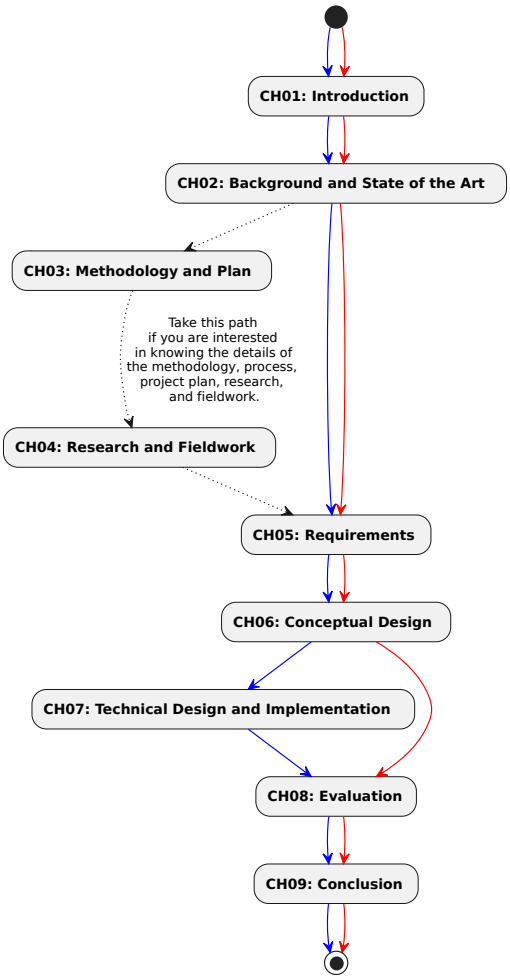


Figure 1.1: Recommended reading paths for technical (blue) and non-technical (red) readers.

# 2 | Background and State of the Art

This chapter aims to uncover the necessary background information for the problem domain and theoretical **SOTA** for the **DSI** and **DMA**. It covers what a data space and a data mesh are and the design principles underpinning them. Furthermore, the section about the **DSI** will cover an architectural reference model and outline the current landscape of the **DSI**. The section about the **DMA** will explain what exactly the data mesh and data product is. Lastly, the chapter will briefly overview the existing solutions in the data mesh space.

Besides understanding the **DSI** and **DMA**, a basic understanding of the Danish energy sector is recommended. The Danish energy sector overview is provided in Appendix A. Furthermore, not mentioning the climate crisis and the **Sustainable Development Goals (SDGs)** would be a missed opportunity to raise awareness of climate change. For this reason, Appendix B provides a brief overview of Denmark's current climate status and progress toward meeting sustainability goals.

## 2.1 The Data Spaces Initiative

The **EU** Commission has recently supported and invested heavily in an initiative dubbed data spaces. Data spaces are expected to be central to **EU**'s green transition, as the initiative tackles some of the underlying obstacles that challenge **EU**'s ability to reach climate goals [8, p. 24].

This section will provide an overview of data spaces and the current landscape of the **DSI**.

### 2.1.1 What is a Data Space?

According to [8, p. 5], data spaces are common data sharing and exchange standards in data ecosystems. It is a decentralized infrastructure that distinguishes it from traditional data infrastructures. Data spaces are designed to be distributed and decentralized, allowing data to be stored and processed in a distributed manner, whereas traditional data infrastructures are centralized and monolithic.

> "A data space is defined as a decentralized infrastructure for trustworthy data sharing and exchange in data ecosystems based on commonly agreed principles [11, p. 23]."

At a grand scale, data spaces can be seen as connected interoperable data ecosystems enabling sectors to collaborate while maintaining high trust, security, and privacy underpinned by common principles [8, pp. 8–9]. The envisioned solution is illustrated in fig. 2.1.

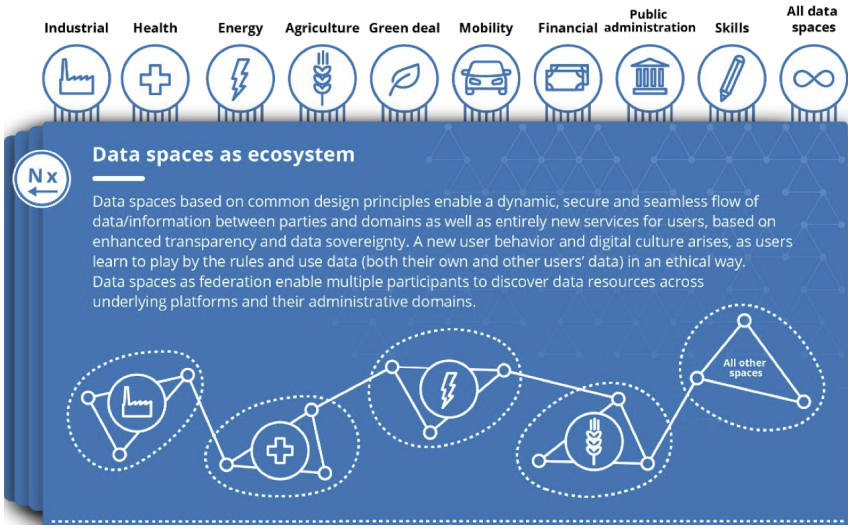


Figure 2.1: Data spaces as ecosystem [11, p. 16].

In [8, p. 5], the importance of knowing that data spaces are not a technology but rather a set of guidelines defining the ecosystem is emphasized. Setting and defining these guidelines is an ongoing process where organizations, universities, governments, and companies collaborate. Recently the **EU** Commission took on the responsibility of driving the development of data spaces forward through the European Data Strategy[4]. The current initiatives focus on establishing regulations to determine how data is shared and exchanged and investing in research and innovation projects, which can help drive the development of data spaces.

### The Design Principles

OpenDEI[5] [13] has had a significant role in defining the concept of data spaces and proposing the design principles that will guide its development [8, p. 32]. OpenDEI has proposed four design principles that touch upon the critical aspects of data spaces and define technical requirements to fulfill them.

**Design Principle 1 - Data sovereignty** Data sovereignty is the term used to describe a person's ability to access and manage the data generated by online services and **Internet of Things (IoT)** products. The **Data Act (DA)** [14], a legal rule put out by the **EU** Commission to defend the rights of data owners, governs data sovereignty [8, pp. 27–28]. In addition to deciding who gets access to their data, for how long, and for what reason, it also involves gaining access to standardized data formats. Furthermore, data owners should be free to readily sell or share their data with third parties and revoke consent. Data sovereignty builds on current laws and organizations like **General Data Protection Regulation (GDPR)** [15], **ePrivacy Directive (ePR)** [16], **Intellectual Property Rights (IPR)**, and MyData [17] to promote openness, dispersed digital trust, and the security of personal and corporate data [8, pp. 13–14].

**Design Principle 2 - Data level playing field** The principle of a level playing field tries to lessen the growing data monopolization by major digital firms. Due to their control of enormous amounts of data and the difficulty of switching providers, big **Information Technology (IT)** corporations have an edge. Switching between service providers and governing owned data should be made simpler by data spaces, encouraging competition based on data quality rather than quantity. A less constrained market would reduce entry barriers, establishing a level playing field. By allowing enterprises to share data, it is envisaged that a fair playing field for data will promote the growth of creative services and new business models. Furthermore, data spaces would be anticipated to improve transparency, making it more straightforward for users to profit from their data by selling it on marketplaces [8, p. 14].

**Design Principle 3 - Decentralized soft infrastructure** Decentralized soft infrastructure refers to the laws, regulations, contracts, and technological advancements that support and facilitate the operation of hard infrastructure[6]. In order to guarantee interoperability and cohesiveness between decentralized and fragmented data ecosystems, it involves common standards, technologies, and rules that must be agreed upon across data spaces and actors. Data sharing across sensors, networks, and platform technologies is essential for soft infrastructure. The soft infrastructure needs strong governance, which includes coordinated agreements and standards for data policies, security precautions, distributed digital trust, identity origin, and data information models. Governance also includes standard service level agreements, smart contracts, market conditions, business models, and collaboration criteria. The technological components of these agreements and standards must adhere to cybersecurity and data protection by design principles and **GDPR** and comply with all applicable laws and regulations [8, pp. 14–16].

**Design Principle 4 - Public-private governance** Public-private governance aims to establish a data market and economy that advances the interests of numerous participants. It refers to the balance between public and private management and control of data spaces. In order to create and execute data spaces, the principle highlights the need for democratic processes, including people, corporations,

[4]The European Data Strategy is a strategy set out by the **EU** Commission to make **EU** the leader in a data-driven society [12].

[5]OpenDEI [13] is a research and development project focusing on the digital transformation strategy of **EU**. It has received funding from the **EU** Commission.

[6]Hard infrastructure refers to the physical infrastructure, which does not entail software.

**Non-Governmental Organizations (NGOs)**, municipalities, states, and other actors. Regulation and design of data spaces should consider the interests of many stakeholders, including enterprises, and permit public and private organizations to use data for multiple purposes. Additionally, while considering protecting intellectual property rights, private data must be made available to public authorities under specific conditions [8, p. 16].

In summary, OpenDEI's proposed design principles set the guidelines and requirements for data spaces. The third principle is, in particular, necessary for establishing an architectural model for data spaces.

### The Reference Architectural Model

**International Data Spaces Association (IDSA)**[7] [18] has developed and promoted a full architectural reference model for data spaces. The model effectively presents the roles and components of a data space and aims to provide a data space as an international entity. According to information from an event held by Sitra in 2023 [19], several other actors are working on similar models or sub-parts in defining the soft infrastructure, for example, Gaia-X [20], MyData [17], Sitra Rulebook [21], FIWARE [22], and Data Governance Act [23].

This thesis focuses on the **International Data Spaces (IDS)** model proposed by **IDSA** due to its maturity and documentation. The currently available model is from 2019, but **IDSA** is putting efforts into a new model [24], which is currently under development. The current information on the new model is limited, and the overall architecture is not expected to change significantly. The model from 2019 is illustrated in fig. 2.2.

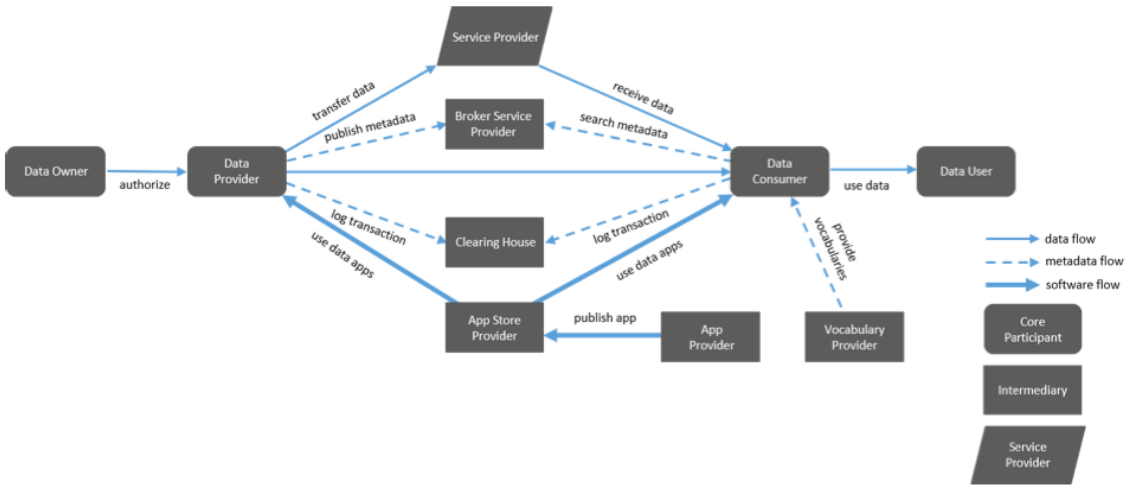


Figure 2.2: The Reference Architectural Model 3 by **IDSA** [25].

The model contains a set of roles and the relations between them. Understanding each role and its responsibilities is essential for understanding the architecture of data spaces. In total, there are ten roles divided into four categories.

**Core Participants** The roles involved in data exchange inside the **IDS** are called core participants. The data owner, data provider, data consumer, data user, and app provider are some of these responsibilities. Organizations may assume one or more roles depending on how much data they provide or consume [25, p. 21].

**Data Owner** An entity that produces and manages data is called a data owner. They are in charge of defining usage guidelines and contracts, granting access to the data, and specifying the payment model for third parties who utilize the data. The position of the data provider is typically assumed

[7]The **IDSA** is an association that aims to create an international data space that enables a standardized way to exchange data between different sectors in **EU** [18].

by a participant working as a data owner, but there may be instances where they are not the same entity. In these circumstances, the data owner's sole role is to authorize a data provider to make data accessible to data consumers, which should be made explicit in a contract that includes details of the data usage policy [25, pp. 21–22].

**Data Provider** Data is made available for interchange between a data owner and a data consumer by a data provider. Although this is frequently the case, the data owner and the data provider are not necessarily the same. A data provider is responsible for submitting metadata to a broker service provider or exchanging data with a data consumer, requiring the data provider to employ **IDS** compliant software components. Furthermore, the data provider can use apps to enrich or transform the data. To simplify billing or settle a dispute, the data provider may also log the specifics of a transaction at a clearing house. Lastly, the data provider can use a service provider to connect to the **IDS** if lacking the required technical infrastructure [25, p. 22].

**Data Consumer** The entity that gets data from a data provider is called a data consumer. Activities of the data consumer might be viewed as mirror entities because they are comparable to those of the data provider [25, p. 22].

**Data User** A data user is an entity with the authority to utilize a data owner's data per the usage policy. The person who uses data is typically the same person who consumes data from the data provider. There might be instances where different participants fill these roles [25, pp. 22–23].

**App Provider** Entities called app providers are in charge of creating data apps that can be accessed by the **IDS**. These applications must adhere to the **IDS** system design to be deployable. In order to increase trust in the apps handling sensitive information, data apps can also be certified by a certification body. The app publishers in the app store make these data apps available so that data consumers and providers can access and use them. When doing so, they should offer metadata that adheres to a metadata model, outlining each data app's semantics, functionality, interfaces, and other pertinent specifics [25, p. 23].

**Intermediaries** Trusted entities which offer particular services to participants in the data exchange process are referred to as intermediaries. These entities play a variety of responsibilities, including those of a broker service provider, clearing house, identity provider, app store, and vocabulary provider. Only trusted companies are permitted to fill these positions, and participants benefit from intermediaries because they build relationships of trust, provide metadata, and develop business models for their services [25, p. 23].

**Broker Service Provider** A broker service provider controls the metadata for the system's data sources. Although the broker service provider plays a crucial role, this position is not exclusive, and numerous providers may coexist simultaneously for various domains. A broker service provider receives metadata from data providers, stores it in an internal repository, and offers data consumers an interface to run queries. A broker service provider may expand the **IDS**-specified core metadata model to manage additional metadata items. One thing to note is that a broker service company could simultaneously act as a clearing house or identity provider [25, p. 23].

**Clearing House** For all financial and data exchange transactions, the clearing house offers clearing and settlement services[8]. Although technically distinct from broker services, clearing activities might be performed by the same entity as broker service providers. Both involve serving as a reliable middleman between the data consumer and the data provider. The clearing house records every action taken during a data exchange, enabling the data provider and the data consumer to confirm the data transfer after it has occurred. It utilizes logging, providing reports on the logged transactions for billing, dispute settlement, and other uses [25, p. 23].

**Identity Provider** Identity provider offers services for managing identity to ensure the safe functioning of the **IDS** and prevent unwanted access to data. The identity provider's primary goal is to establish, maintain, manage, monitor, and validate the participants' identity information, which requires multiple services. First off, the **Certificate Authority (CA)** oversees the participants' digital certificates. Next, the **Dynamic Trust Monitoring (DTM)** continually checks the

[8] Clearing and settlement refer to the procedures required to settle financial transfers.

network's security and behavior while the **Dynamic Attribute Provisioning Service (DAPS)** maintains the participants' dynamic attributes [25, pp. 23–24].

**App Store Provider** An app store that manages the data apps is an app store provider. The app store provider is in charge of preserving information regarding the data apps made available by app providers, and they are required to provide **Application Programming Interfaces (APIs)** for publishing and retrieving data apps as well as relevant metadata [25, p. 24].

**Vocabulary Provider** The responsibility of the vocabulary provider is to manage and provide vocabulary that can be used to annotate and describe datasets. Ontologies, reference data models, metadata components, or domain-specific vocabularies can all be a part of the vocabulary [25, p. 24].

**Software / Service Provider** The software/service provider roles involve **IT** companies offering software and services based on a **Software as a Service (SaaS)** model [25, p. 24].

**Service Provider** On behalf of other organizations that lack the appropriate infrastructure, a service provider may host the technical infrastructure needed for data exchange. To enhance the quality of the shared data, the service provider may also provide supplementary data services such as data analysis, integration, cleansing, or semantic enrichment. Since it receives data from a data provider, delivers its service, and offers the processed data in the **IDS**, this function is also regarded as both a data provider and a data consumer. A service provider is not a data app; it cannot be implemented in a data provider or consumer's **IT** environment. This distinction is essential, as they share many similarities [25, p. 24].

**Software Provider** A software provider can provide software to implement the **IDS**'s necessary capabilities. How so is based on agreements between the users and software providers and is not covered by the **IDS** [25, p. 24].

**Governance Body** The bodies in charge of the **IDS** are called governance bodies. The governance bodies are the certification body, evaluation facilities, and the **IDSA**, and they collectively oversee the proper application and adherence of implemented governance systems and policies [25, p. 24].

### 2.1.2 The Landscape of Data Spaces in 2023

The information covered on the **DSI** is only a tiny fraction of the information available. However, covering all aspects of the **DSI** is a tremendous task, and it would entail covering many aspects of the **DSI** that are irrelevant to this thesis. However, in March 2023, the **Data Spaces Support Centre (DSSC)**[9] released a versioned report called "Starter Kit for Data Space Designers" [27]. The report documents most aspects of data spaces but does not delve deep into the design and technical aspects as OpenDEI or **IDSA** does. It provides a valuable overview of the landscape of the **DSI** and promises to keep it up-to-date [27, pp. 21–30]. The landscape is divided into six groups:

1. Data Spaces 101: Covers initiatives towards establishing a common understanding of the concept of data spaces.
2. Business: Value and Models: Covers initiatives towards uncovering the value of data spaces and the business models that can be used to monetize data spaces.
3. Legal Landscape and Governance Models: Covers initiatives towards establishing and documenting the legal framework for data spaces.
4. Functionality and Technology: Blueprints and Building Blocks: Covers initiatives towards defining and documenting the technical aspects of data spaces.
5. Use Cases: Covers initiatives for different use cases of data spaces.
6. Organizations and Associations: Cover the different organizations and associations involved in data spaces.

[9] The **DSSC** is an organization funded by the **EU** Commission that aims to support the development of data spaces by defining needs, requirements, and best practices [26].

Given the breadth and depth of the many organizations and initiatives encompassed by these six groups, a comprehensive analysis of all of them is beyond the scope of this thesis, which primarily focuses on the technical aspects of data spaces. Instead, the reader is encouraged to read the report and investigate the different projects and initiatives in the landscape for a deeper understanding of the data spaces ecosystem.

## 2.2 The Data Mesh Approach

The **DMA** [28, pp. 3–4] is a new paradigm for data management defined by Zhamak Dehghani. The **DMA** aims to solve the problems of traditional data management approaches, and it is a decentralized approach to managing analytical data in complex, large-scale environments within or across organizations. This section presents the concept of a data mesh and its central unit, the data product, and explains their relevance to the thesis.

### 2.2.1 What is a Data Mesh?

A data mesh is a decentralized data infrastructure that creates a mesh of interconnected data products. A data mesh enables teams and organizations to exchange data with each other in a standardized way, facilitating efficient data sharing and collaboration [28, p. 162] — an essential aspect for successfully implementing data spaces, which is a strong focus of this thesis.

According to [28, pp. 162–163], the data mesh consists of three planes, the mesh experience plane, the data product experience plane, the infrastructure utility plane, and the mesh itself. The logical architecture is depicted in fig. 2.3.

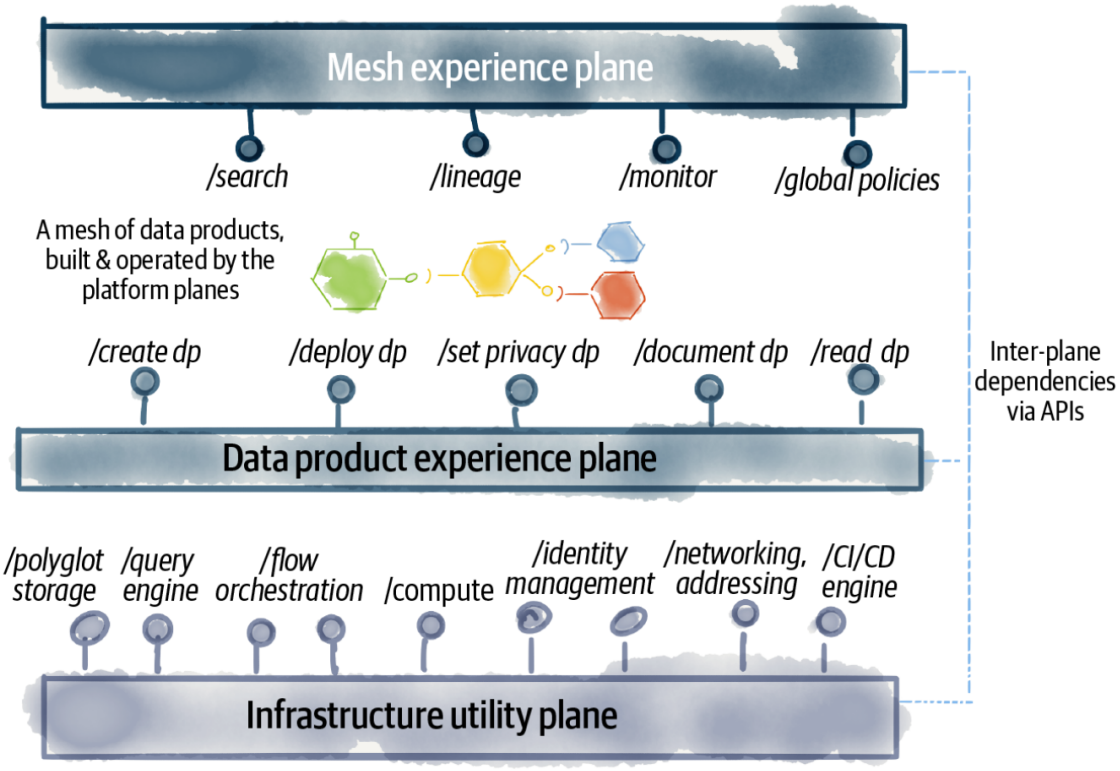


Figure 2.3: The data mesh and its supporting planes [28, p. 163].

The mesh experience plane [28, p. 162] provides observability and discoverability across the mesh. It should enable users to observe what is happening in the mesh and discover the data products they need.

The data product experience plane [28, p. 162] is the plane where data products' lifecycles are managed. It should allow the creation, deletion, and modification of data products and also enable reading data

from them. For example, to collect telemetry data and metadata for the mesh experience plane.

The infrastructure utility plane [28, p. 162] supports the data product experience plane to provide the required infrastructure for running the data products. It includes infrastructural components like storage and computing.

To fully understand the data mesh, it is necessary to understand the design principles that define it.

### The Design Principles of a Data Mesh

The data mesh is based on four principles that define its core concepts [28, pp. 6–9]. The principles are depicted in fig. 2.4 and presented below:

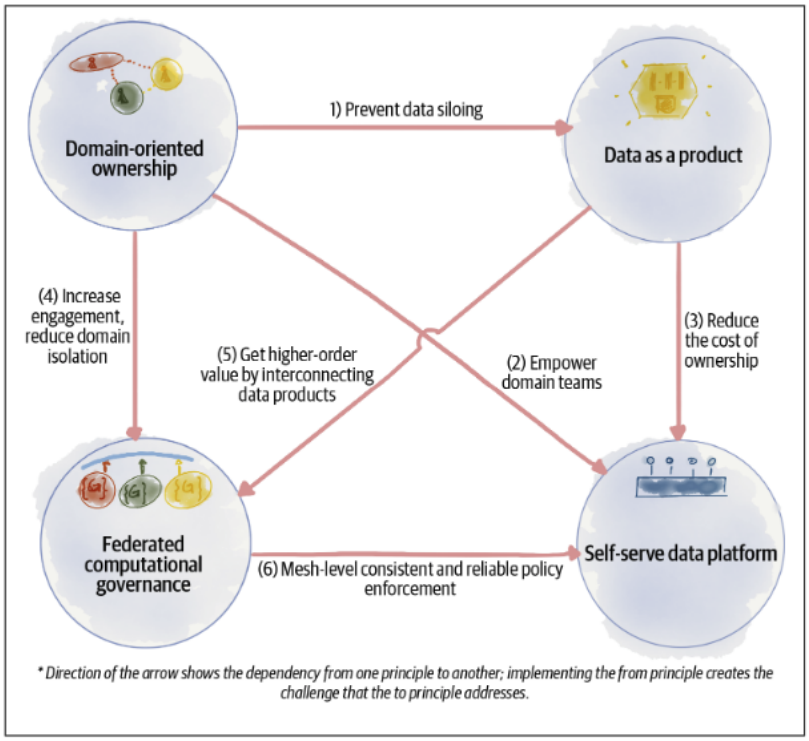


Figure 2.4: The four principles of data mesh and how they are related [28, p. 9].

**Principle of Domain Ownership** Domain ownership is vital for managing data in complex, large-scale environments within or between businesses. The strategy entails transferring data ownership control to the businesses closest to the data. Based on the business domain it represents, data is logically divided and separately maintained throughout its life cycle. Scaling out data sharing, optimizing for ongoing change, allowing agility, boosting data business integrity, and boosting the resilience of analytics and machine learning solutions are some of the reasons for domain ownership. Organizations can eliminate centralized bottlenecks, eliminate cross-team synchronizations, and bridge the gap between the real source of the data and its analytical use cases by introducing domain ownership [28, pp. 6–7].

**Principle of Data as a Product** The phrase "data as a product" refers to the usage of data as a valuable resource that is shared as a product with data users like scientists and data analysts. Data must meet usability requirements to be categorized as a data product, for example, discoverability, interoperability, and security. Each data product must be an isolated entity with its own life cycle making them autonomous. The fundamental components of a data product are code, data, and the declaration of infrastructure dependencies. The data as a product strategy is driven by many factors like breaking down data silos by altering how teams interact with data, developing a culture of data-driven innovation, boosting adaptability to change, and maximizing the value of data by sharing and utilizing it across organizational boundaries [28, pp. 7–8].

**Principle of Self-Serve Data Platform** A self-serve data platform is a set of services that makes it simple for cross-functional teams to share data. These platform services create a reliable mesh of interconnected data products and manage the whole life cycle of any individual data product. Additionally, they make it easier for data users to find, access, and use data products and for data providers to create, distribute, and maintain data products. Lastly, the services also aim to automate governance policies, which can ensure a high level of security and that the data products comply with standards [28, p. 8].

**Principle of Federated Computational Governance** Federated computational governance is an operating paradigm for data governance that relies on a decentralized framework for decision-making and accountability. This architecture seeks to compromise the global interoperability of the data mesh and the domains' independence and agility. Executing governance depends on platform services that codify and automate policies at a granular level for each data product. The goal of federated computational governance is to make it possible to incorporate cross-cutting governance requirements across a mesh of distributed data products and to mitigate the adverse effects of domain-oriented decentralization [28, pp. 8–9].

### 2.2.2 What is a Data Product?

In the context of data mesh, a data product is the smallest unit of architecture that can be deployed and managed independently. It is characterized by high functional cohesion, designed to perform a specific analytical transformation and securely share the result as domain-oriented data. A data product comprises three main types of structural components: code, data (metadata and configuration), and specifications of infrastructure dependencies [28, pp. 151–152]. The structural components are depicted in the left image in fig. 2.5.

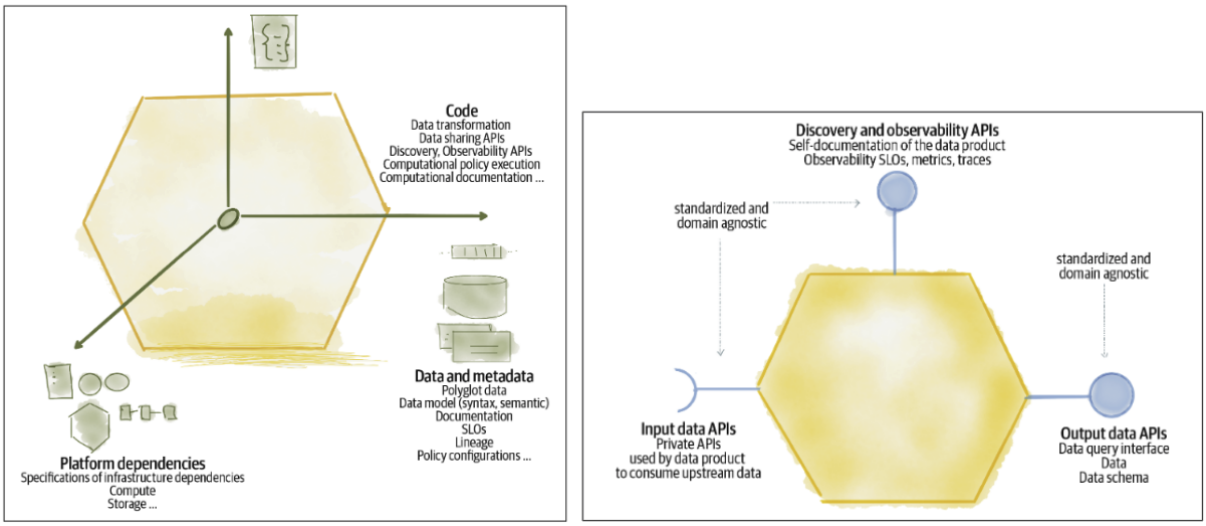


Figure 2.5: The structural components of a data product and its APIs [28, pp. 152, 156].

#### Code

The data lifecycle is fundamentally governed by code. The code comprises the implementation and automated tests that validate it and is contained as an internal implementation that resides inside a data product. Code could, for example, handle business logic, access control, and content sharing but also implements interfaces that adhere to stated contracts for output, input, discovery, and observability **APIs**. Code is required to alter data received from an upstream source or create data. It is worth noting that the upstream source is nearly always in charge of purifying the data; thus, data products deliver cleansed analytical data [28, pp. 153–157].

### Data

The life cycle of a data product depends heavily on the data and metadata. The data entails the analytical data and metadata for which the data product is responsible. Examples of metadata include documentation, syntax, semantic declarations, and **Service-level Objective (SLO)**[10]. One critical thing to mention is that, unlike traditional data architectures, a data product is expected to generate and expose its metadata. In this sense, metadata is an integral part of the data product, not a separate entity [28, p. 157].

### Infrastructure Dependencies

Infrastructure dependencies describe how a data product depends on a particular infrastructure. In order to support autonomous operation, the self-serve data platform is in charge of maintaining isolation for each data product while centrally managing infrastructure resources. Each data product defines and controls its expected infrastructure dependencies so the platform can provision and manage the relevant infrastructural components. One thing to note is that deploying or updating one data product must not interfere with the functionality of any other data products on the shared infrastructure [28, pp. 157–158].

## 2.3 Existing Solutions

In the context of this research, it is necessary to explore the current landscape of data spaces and data mesh implementations to understand the **SOTA** and identify any existing gaps or opportunities. Data spaces are still in their infancy, and implementations are expected to emerge in the coming years as research and development converge toward a standard solution. To learn more about the current state of **DSI**, the author refers the reader to study the current landscape of data spaces as presented in sec. 2.1.2. However, the data mesh is a more mature concept, and some solutions have emerged.

The author conducted numerous searches for solutions in the data space and data mesh space using Google in English from 2022-09-01 to 2023-02-14—the searches aimed to identify existing solutions and understand their relevance to the research. One notable article by K2View [30] presents different data meshes and their proprietary solution in the data mesh space. The author used the article as a guideline for the existing solutions in the data mesh space. In tbl. 2.1, the different solutions are presented.

Table 2.1: Competing products in the data mesh space [30].

| Name | Architecture | License |
|---|---|---|
| K2View [31] | Data Mesh | Proprietary |
| Talend [32] | Data Fabric | Proprietary |
| Starburst [33] | Data Fabric | Proprietary |
| Informatica [34] | Data Fabric | Proprietary |
| Denodo [35] | Data Fabric | Proprietary |

The table shows five competing vendors in the data mesh space, all providing proprietary solutions. Further research might reveal more solutions, but they will most likely be at an early stage of development, hence not being mature enough to be considered a threat to the existing solutions. It is worth noting that K2View is the only solution built from the ground up as a data mesh solution that includes the concept of a data product. The other solutions are data fabrics[11] [36] adapted to fit the data mesh architecture through different means. Data fabrics generally focus on unifying different data sources for seamless access, integration, and processing, while data meshes emphasize decentralized data management and domain-oriented data sharing. Understanding these differences is crucial when considering the implications of adopting various solutions.

[10] A service-level objective is an agreement on measurable characteristics—for example, availability, throughput, frequency, response time, and quality [29].

[11] A data fabric is an architecture that unifies different data sources, enabling seamless access, integration, and processing, often through a **Platform as a Service (PaaS)** [36].

Overall, the solutions available are hard to investigate, as accessing them is impossible without paying or arranging a demo session with the organization(s). As all the solutions are proprietary, working toward a flexible, open-source solution is still valuable, as it can contribute theoretically and practically. Establishing how data spaces and meshes can be implemented freely with existing technology is essential to ensure that organizations and governments have a fair chance to comply with the standards that will underpin the industry in the coming years. Furthermore, most of these solutions are data fabrics, and they do not compete directly with the data mesh vision, as they do not include the concept of data products. However, they compete with the self-serve data platform, which overlaps with many of the capabilities of data fabrics and, by extension, the possible solutions presented in tbl. 2.1.

## 2.4 Summary

This chapter presented the **DSI** and **DMA**, both crucial to understanding ambitions for modernizing data infrastructures. The **DSI** is a large project, and this chapter only presents a small fraction of it, namely the design principles from OpenDEI and the architectural reference model (**IDS**) for a soft infrastructure by **IDSA**. The design principles explain how the **DSI** should be designed at a high level to achieve data sovereignty, a level playing field, a decentralized infrastructure, and a good balance between public and private governance. The presentation of **IDS** demonstrated how implementing **DSI** requires specific components to achieve these design principles.

The **DMA** is presented as a set of principles set forward by Zhamak Dehghani that details a data infrastructure that is decentralized and domain-oriented [28, p. 160]. The **DMA** is a response to the challenges of traditional data architectures, which are centralized and monolithic. **DMA** consist of a data mesh comprising many individual data products. Each data product enables different organizations and teams to share and enrich data in a decentralized manner, promoting data sovereignty. It segregates large domains into smaller, more manageable units that can be developed, deployed, and managed independently. The data mesh vision requires a new generation of data platforms that can manage the entire life cycle of individual data products and provide a reliable mesh of interconnected data products. This new generation of data platforms combines a data catalog, a provisioning system, and a data management system under one roof.

The landscape of data spaces and data meshes is limited, and only recently has research and development in the areas started to converge toward actual implementations. The author could not find any implementations of data spaces, but five proprietary vendors were found for data meshes, where most were adaptations of their existing data fabric products. The only vendor with a fully developed data mesh solution is K2View.

The information provided on the **DSI** and the **DMA** is highly relevant to the thesis, as it forms the theoretical foundation for evaluating the findings from fieldwork presented in ch. 4, and their theoretical validity. It enables assessing whether the research aligns with the **DSI** and **DMA** concepts. The actual evaluation of how the **DMA** applies to **DSI** and the resulting theoretical and practical validity is presented in ch. 8.

# 3 | Methodology and Plan

The methodology used in this thesis project is guided by constructivism, specifically the constructive research methodology, which aims to develop innovative solutions to real-world problems. During the early stages of the project, an inductive approach known as **GT** is utilized to generate knowledge through conducting and analyzing interviews. This approach helps identify practical challenges and understand the problem domain. The outcome is a set of theories and hypotheses (Appendix N) used to define the requirements (ch. 5) and design a prototype.

In the later stages of the project, a deductive approach is employed. The deductive approach involves comparing and contrasting the findings from **GT** with established theories and concepts to identify consistencies or deviations, specifically, the theories on data spaces and the **DMA** presented in sections 2.1 and 2.2.

Combining the inductive and deductive approaches allows the author to develop and evaluate a prototype based on practical problems identified in the early stages while theoretically ensuring the solution's feasibility. This combination contributes to a comprehensive understanding of the research problem and helps bridge the gap between theory and practice.

Activities unrelated to research and fieldwork are carried out through an agile, iterative process, which guides the development of the prototype, project documentation, and thesis writing. The process relies on agile principles and practices from the Kanban method and good developer practices by following the GitHub flow. To learn more about this process, see Appendix C and Appendix D.

The following sections describe this thesis project's research methodology, process, and practices in more detail.

## 3.1 The Constructive Research Approach

Constructivism is a theoretical framework in which knowledge is constructed through an individual's active participation in constructing meaning from their experiences. The potential depends on the researcher's prior knowledge, beliefs, and cultural background [37].

The **CRA** (depicted in fig. 3.1) is closely related to constructivism. The approach aims to produce innovative constructions for real-world problems while also contributing to the theory of the discipline in which it is applied. The core features of the approach require that it focuses on real-world problems with practical relevance, produces an innovative construction meant to solve the problem, and includes an attempt to implement the developed construction and test its practical applicability. The approach involves close involvement and cooperation between the researcher and practitioners in a team-like manner, with explicit links to prior theoretical knowledge and particular attention paid to reflecting empirical findings to theory [9, pp. 83–85].

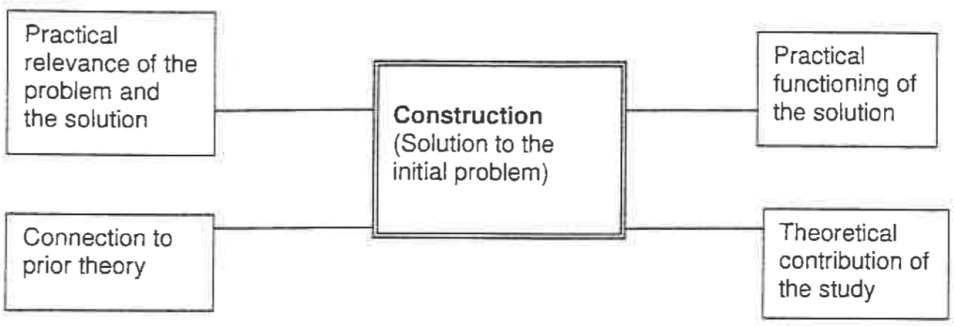


Figure 3.1: A model of the central elements in the **CRA** [9, p. 85].

The **CRA** is vital to the project. A considerable emphasis has been put into following the approach, allowing the author to construct knowledge from fieldwork (ch. 4). Doing so has ensured that the project has practical relevance and that a solution remains feasible. The approach is critical in the project's early stages, where the author had no prior knowledge of the problem domain and had to rely on the **CRA** to guide the project. It is a difficult task, requiring discipline, to not research the problem domain before conducting fieldwork. Doing so risks creating bias and limits the author's ability to construct knowledge from practice. As such, the approach requires high trust, but exploring the problem domain through pragmatism has proven valuable.

### 3.1.1 The Process

The process of constructive research is a seven-step process that enables new and novel constructions to be developed and tested from a deep understanding of the problem area [9, pp. 85–91]. Each step is described in the following list:

1. "Find a practically relevant problem with the potential for theoretical contribution [9, p. 86]."
2. "Examine the potential for long-term research cooperation with the target organization(s) [9, p. 86]."
3. Obtain a deep practical and theoretical understanding of the topic area [9, pp. 86–87].
4. Define an innovative solution idea and develop a problem-solving construction with a potential theoretical contribution [9, p. 87].
5. Implement and test the solution [9, pp. 87–88].
6. Evaluate the applicability of the solution [9, pp. 88–89].
7. "Identify and analyze the theoretical contribution [9, pp. 89–91]."

The project plan described in sec. 3.3 is inspired by the constructive research process, and time is carefully allocated to ensure each step is covered.

### 3.1.2 The Need

The need for constructive research is based on the relevance of the research topic, both practical and theoretical, and the importance of two-way communication between the researcher and the target organization(s). It allows active and intensive cooperation, bringing prior knowledge into the research process and advocating truth by pragmatism - the truth is what works in practice [9, pp. 91–92].

Constructive research is valuable for the project as it researches a problem domain with the potential for practical and theoretical contributions. Energinet requires finding innovative solutions to decentralize data infrastructure. Both the **DSI** and **DMA** are novel theories that present guidelines to accomplish this but with much wiggle room for exploration and experimentation. Furthermore, collaborating with Energinet, the target organization, is a solid incentive to conduct constructive research. Working with an organization allows gaining practical research experience and a deeper understanding of the problem domain. The project is also practical-oriented, as a prototype is developed to explore a solution to a real-world problem; this matches well with the approach.

Due to the scope and time constraints of this particular master's thesis, deep and long-term cooperation with Energinet is not feasible. Instead, the author will rely on the methodology to interview Energinet experts and involve them in the design process to ensure the prototype stays relevant to the problem domain.

### 3.1.3 Benefits

Conducting constructive research offers new possibilities for gaining access to exciting research sites, and it also enables gaining practical experience in the field of research. From an organization's point of view, it creates the incentive to start cooperation with researchers, where their problems can potentially be analyzed and solved by the critical thinking applied by an academic researcher. As such, it provides organizations access to new knowledge and expertise. The approach narrows the gap between practice and research, allowing the interchange of knowledge between researchers and practitioners. This close collaboration fosters an incentive to develop or hone knowledge [9, pp. 96–97].

The benefits of conducting constructive research are felt throughout the project. The author has gained practical knowledge of the problem domain and an understanding of the Danish energy sector's challenges. Gaining access to Energinet has also been beneficial as it has allowed the author to experience the organization's culture and work environment, talk to employees, and observe the challenges they face in their day-to-day work. Hearing people's interest in the subject has emphasized the importance of the problem and the need for solutions.

Moreover, creating knowledge through practice has allowed the author to experience and experiment with the prototype. It is valuable in the later stages of the project, where the author experienced many of the challenges the author faced during the project were relatable to the challenges the **SOTA** aims to solve, further strengthening the author's trust in the approach. Experiencing that the **CRA** leads to developing a relevant prototype, the author feels confident that the approach is practical.

### 3.1.4 Risks

Constructive research is difficult and time-consuming and can require a high level of competence and dedication from the researcher and the target organization(s). The research findings can be sensitive, which makes them challenging to publish and puts the researcher's interests at odds with those of the target organization(s). As such, creating an explicit contract that considers publication concerns early on is critical. Furthermore, the research may become irrelevant without constant communication with the target organization(s). Another risk is that it can be challenging to avoid bias, as creating an innovative solution can compromise the stance of neutrality or skepticism required of academic research. From the perspective of the target organization, a reluctance to cooperate can arise from a lack of trust in the researcher's ability to solve the problem or the fear of losing control of trade secrets [9, pp. 96–97].

The risks have been clear from the start of the project. Committing to the process is challenging as it requires the author to trust that knowledge will emerge from practice. Initially, the author was not convinced that interviews would guide the project toward novel solutions; however, this doubt waned as the project progressed. It is also time-consuming to keep ongoing relations with the organization, but the value of it far outweighs the cost.

As the master's thesis is not published, and no **Non-Disclosure Agreement (NDA)** has been signed, the publication risks are irrelevant.

### 3.1.5 Summary

In conclusion, the **CRA** is essential to the thesis project, as it helps bridge the gap between theory and practice, providing practical relevance and theoretical contributions. The approach enables the author to develop a feasible solution based on fieldwork and insights from the target organization, Energinet. The benefits of this approach outweigh the risks and challenges faced during the project. By following the **CRA**, the author can ensure that the research outcomes are relevant, innovative, and applicable in the real world, contributing to the field of study.

## 3.2 Grounded Theory

Grounded theory is a qualitative research method defined by Glaser and Strauss in 1967 [10]. The method allows new insights and patterns to emerge without imposing preconceived notions or theories. It is a good fit for the exploratory nature of the project, and it fits well with constructive research.

The method involves four activities that are conducted iteratively:

**1 Theoretical sampling** is a method to select participants for data collection. The participants are selected based on the theory developed during prior iterations, which refer to the earlier data collection, analysis, and theory development cycles in the **GT** process. The selection of participants is not based on random sampling but on the researcher's judgment, allowing for a more targeted approach to data collection. As such, the data collection methods are expected to be qualitative, such as semi-structured and structured interviews [10, pp. 45–47].

**2 Open Coding** is a method to organize and analyze data (fig. 3.2). The data collected during the research process are systematically broken down and analyzed by assigning codes to data segments representing a concept or idea. The codes are used to identify patterns in the data that can be used to develop theories and hypotheses [38, ts. 00:24–00:44].

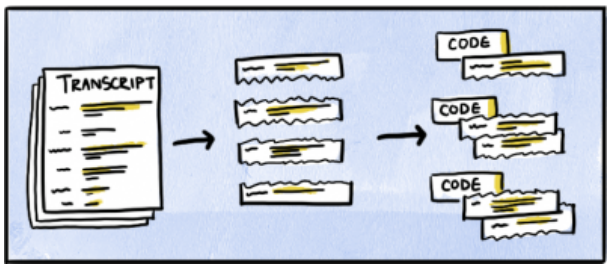


Figure 3.2: The process of open coding [38, ts. 00:34].

**3 Axial coding** is a method to link codes and categories together to develop a more comprehensive theory (fig. 3.3). The process involves identifying relationships between codes and categories and organizing them into a framework. This framework is used to develop a theory that explains the relationships between the concepts identified during the research process [38, ts. 00:44–01:00].

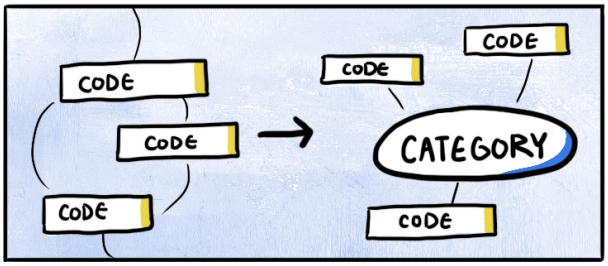


Figure 3.3: The process of axial coding [38, ts. 00:53].

**4 Selective coding** is a method to refine and focus the theory developed during the research process (fig. 3.4). The process involves selecting a core category, the central concept or idea that explains the relationships between the other categories identified during the research process. The core category is then used to guide the selection and analysis of data to refine the theory further [38, ts. 01:00–02:04].

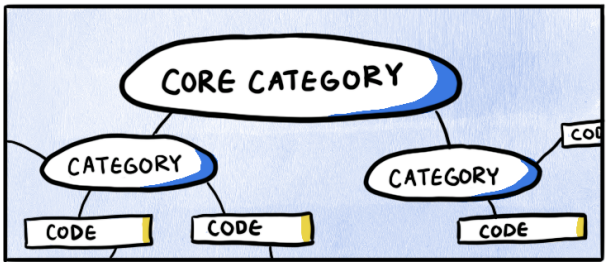


Figure 3.4: The process of selective coding [38, ts. 01:08].

In conclusion, **GT** is a valuable method for conducting qualitative research that contributed to the success of the **CRA** used in this project. The method allowed the author to structure collected data and make emerging patterns and insights visible, which formed the basis for the prototype's requirements and design. However, the method is very time-consuming, and the work required to complete the coding processes increased exponentially with the data collected. As such, the number of interviews is carefully selected to ensure the time consumption remains manageable. Time-related issues could have been mitigated by using better tools to support the coding process, such as the automatic transcription of interviews into

LaTeX-compatible tables, making the process less manual and tedious. Despite these challenges, the **GT** proved essential in developing the prototype and the overall project.

## 3.3 The Project Plan

The methodology, process, and practices culminate into a project plan. Planning and managing a project is vital to ensure a project is completed on time and that all aspects of a project are carefully considered when allocating time to the project's different stages.

The constructive research process, **GT**, and the **Unified Process (UP)** [39] inspire this project's plan. **UP** has been used as a framework for defining the project's different stages and as a general guideline toward a more agile approach. **UP**'s different lifecycles, workflows, and how they are related are described in more detail in Appendix E.

The project plan is presented in fig. 3.5 as a gannt chart representing the project from start to finish. Stages are horizontal lines separating the project's objectives. The objectives are represented as horizontal bars, and the length of the bars represents the time allocated to the objective. Lastly, deliverables are represented as diamonds placed at the end of the objective bar that they are associated.

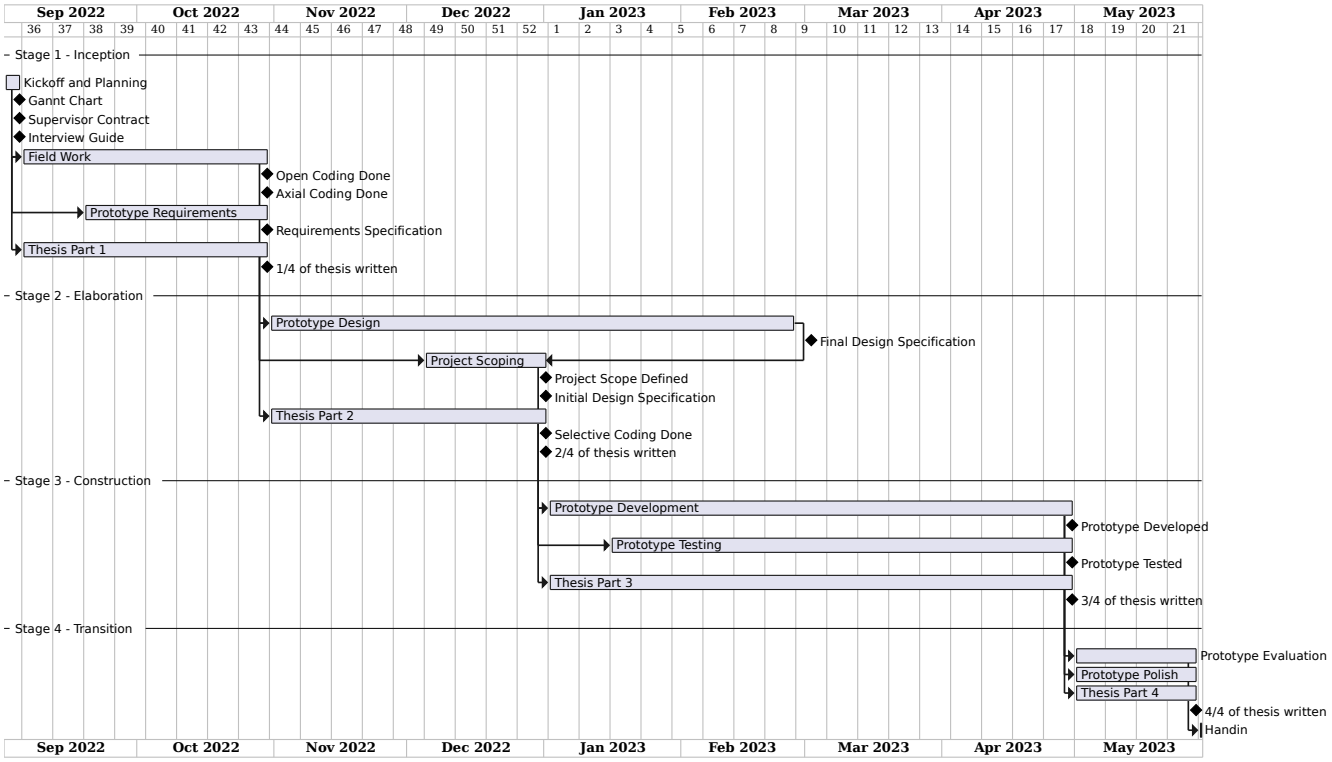


Figure 3.5: Gantt chart for the thesis's project plan.

# 4 | Research and Fieldwork

This chapter will describe the research and fieldwork conducted to gather information about the thesis's problem domain. The **GT** method, described in sec. 3.2, is utilized. First, the author's prior knowledge and practice are identified, followed by initializing the **GT** process. The **GT** process is then documented for each interview, including the open, axial, and selective coding processes.

## 4.1 Prior Knowledge and Practice

Before conducting interviews and initiating the **GT** process, it is essential to establish an aim and initial theory to explore for the theoretical sampling activity. However, prior knowledge cannot be too extensive, as it can lead to bias.

The author explored a few topics before deciding on the final topic and direction of the thesis. First, a few books on how to write a thesis and conduct qualitative research were studied.

- Den Gode Opgave by Peter Stray Jørgensen [40].
- Specielt om Specialer by Peter Stray Jørgensen [41].
- Qualitative Research Methods by Natasha Mack et al. [42].

Preparing for the thesis project by obtaining knowledge on how to write a thesis and how to use qualitative research methods, the author is well prepared for what a thesis project requires. Furthermore, being aware of different qualitative methods, such as open-ended, semi-structured, and structured interviews, make it straightforward to determine which interview type best suits a specific interviewee and topic.

Early in the project, the supervisor referred the author to study specific literature to spark an interest in a direction and to gain a better understanding of constructivism as a methodology:

- Data Mesh by Zhamak Dehghani [28].
- The Danish National Energy Data Lake by Ben Hamadou et al. [43].
- Constructive Research and Info-Computational Knowledge Generation by Dodig Crnkovic [37].
- The Constructive Research Approach by Kari Lukka [9].
- Energinet's website [44].

Being aware that the energy sector relies on centralized systems, like data lakes, and that new and exciting approaches, like the **DMA**, provide much value through being decentralized and distributed, the author is interested in further exploring the **DMA** [28]. Also, having read more about the **CRA**, the author is convinced that it is a good fit for the thesis project, where the author plans to explore the **DMA** by constructing a prototype.

Exploring Energinet's website is informative, allowing the author to understand Energinet's domain and work. The author identified a few exciting projects and initiatives, like energy islands, DataHub, and a global engagement toward the green transition [44].

Based on the knowledge gained from referred material, a discussion was established with the supervisor on what technologies could potentially be used to implement data spaces. During the discussion, the supervisor informed the author that the **DMA** overlaps with some ideas and concepts of the **DSI**. The author was immediately sparked by exploring whether the **DMA** applies to the **DSI** and how it can be implemented.

Besides the theoretical background presented above, the author possesses a technical background with expertise in .NET development and a particular interest in **CNCF** technologies. Because of the specific interests, the author wanted to explore how the **DMA** can be implemented using .NET, **CNCF** technologies, and open-source technologies while using a self-hosted Docker environment for development, testing, and demonstration.

## 4.2 Initializing Grounded Theory

Each interview follows the **GT** method, as explained in sec. 3.2, and utilizes a general interview guide, as presented in Appendix G. The general interview guide serves as a reminder of essential steps and questions to ask to facilitate the interview process. All interviews are conducted in Danish and are recorded or filmed. Each conducted interview is transcribed and translated into American text by the author. Each interview also employs a specific guide containing data about the interviewee and the interview itself, including questions to ask during the conversation. Open and axial coding is performed for each interview to categorize data into codes and core categories. The codes are used as input for selective coding to generate theories and hypotheses for each core category. Only the codes from open coding directly related to the thesis topic are included as input to the axial coding process. The rest are filtered out; the interviews touch upon many topics, some of which do not provide any value to the thesis.

The subsequent sections summarize each interview and detail the open, axial, and selective coding processes.

## 4.3 Interview 1 with Jens Hjort Schwee - Grounded Theoery Cycle 1

Interview 1 was conducted with Jens Hjort Schwee on the 8th of September, 2022. Jens Hjort Schwee is a Senior Digital Business Developer in digitalization at Energinet, primarily focusing on data spaces. He has a Ph.D. in Software Engineering at **University of Southern Denmark (SDU)** with expertise in privacy, data sharing, risk identification, and governance [7, ts. 00:20].

The first interview focuses on gathering general knowledge about collaboration and data spaces and comparing decentralization and centralization in the energy sector. The interview is semi-structured, as the author prepared a few questions beforehand to guide the conversation. These questions can be found in Appendix H. The supervisor observed the first interview to ensure that the interview was conducted according to **GT** principles and the chosen interview type. The author and supervisor agreed upon this to ensure the author was well-prepared for future interviews.

The open coding process resulted in a table of 89 codes grouped into ten categories. The results are visible in tbl. H.1.

After the open coding process is completed, the axial coding process is initiated. The ten categories from the open coding process are refined, and codes are filtered into nine new core categories indicated by an asterisk "*" in the list below.

- Business Ecosystem* - 4 codes are added.
- Centralization and Decentralization* - 16 codes are added.
- Collaboration* - 10 codes are added.
- Data Mesh* - 3 codes are added.
- Data Spaces* - 26 codes are added.
- Governance* - 9 codes are added.
- Metadata* - 9 codes are added.
- Roles and Actors* - 1 code is added.
- Software Qualities* - 2 codes are added.

## 4.4 Interview 2 with Bjørn Therkelsen - Grounded Theory Cycle 2

The second interview was conducted with Bjørn Therkelsen on the 20th and 22nd of September, 2022. Bjørn Therkelsen is a Senior Enterprise Information Architect in the department of **IT, Data and Analytics (IDA)** at Energinet [45, ts. 00:16].

The second interview aims at some of the more technical aspects of data management and gaining more perspectives on data spaces and the technologies that can be implemented. The interview is semi-structured, with a few questions prepared beforehand to guide the conversation. These questions can be found in Appendix I. The interview was split into two sessions, as there was not enough time to cover the topics of interest in one session.

The open coding process resulted in 129 codes, where 115 of those are from the first session, and the last 14 are from the second session. The codes are grouped into 42 categories. The results are visible in tbl. I.1 and tbl. I.2.

The axial coding of interview 2 resulted in the 129 codes being split between 11 core categories. Of these categories, six are new.

- Collaboration - 11 codes are added.
- Data Management* - 17 codes are added.
- Data Mesh - 23 codes are added.
- Data Spaces - 6 codes are added.
- Domain Modelling* - 15 codes are added.
- Governance - 4 codes are added.
- Infrastructure: Digital* - 2 codes are added.
- Legislation and Regulation* - 2 codes are added.
- Open Source vs. Proprietary* - 4 codes are added.
- Roles and Actors - 1 code is added.
- Users* - 2 codes are added.

## 4.5 Interview 3 with André Bryde Alnor - Grounded Theory Cycle 3

The third interview was conducted with André Bryde Alnor on the 3rd of October, 2022. André Bryde Alnor is the Department Manager in Digitalization at Energinet [6, ts. 00:27].

The third interview aims to obtain more knowledge on the business aspects of the energy sector, the **DSI** and the **DMA**. For example, what governs the decisions made in the sector and how legislation and regulation affect the development of new solutions. The interview is semi-structured, with a few questions prepared beforehand to guide the conversation. These questions can be found in Appendix J.

The open coding process resulted in 69 codes grouped into 22 categories. The results are visible in tbl. J.1.

The axial coding of the third interview resulted in the 69 codes being split between 14 core categories. Of these categories, two are new.

- Business Ecosystem - 7 codes are added.
- Centralisation and Decentralisation - 8 codes are added.
- Collaboration - 5 codes are added.
- Data Management - 2 codes are added.
- Data Mesh - 2 codes are added.
- Data Spaces - 10 codes are added.
- Domain Modeling - 2 codes are added.
- Flexibility and Grid Balance* - 2 codes are added.
- Infrastructure: Digital - 1 code is added.
- Infrastructure: Physical* - 2 codes are added.
- Legislation and Regulation - 7 codes are added.
- Metadata - 4 codes are added.
- Roles and Actors - 8 codes are added.
- Software Qualities - 1 code is added.

## 4.6 Interview 4 with Peter Lyck Ingerslev - Grounded Theory Cycle 4

The fourth interview was conducted with Peter Lyck Ingerslev on the 3rd of October, 2022. Peter Lyck Ingerslev is Chief Principal Architect for Energinet's Digitalization and Innovation Initiative [3, ts. 00:58].

The fourth interview was decided to be an unstructured interview, as the interviewee is a very experienced architect with much knowledge about the digital and physical infrastructure of the energy sector. Before the interview, a context is given to the interviewee, and the rest of the interview is conducted as a conversation, with as few interruptions as possible. The context given to the interviewee focused on **DSI**, **DMA**, data storage, data discovery, availability, and performance which can be found in Appendix K.

The open coding process resulted in 67 codes grouped into seven categories. The results are visible in tbl. K.1.

The axial coding of the fourth interview resulted in the 67 codes being split between seven core categories, and none of these categories are new.

- Business Ecosystem - 16 codes are added
- Collaboration - 1 code is added
- Flexibility and Grid Balance - 21 codes are added
- Infrastructure: Digital - 6 codes are added
- Infrastructure: Physical - 7 codes are added
- Legislation and Regulation - 2 codes are added
- Software Qualities - 6 codes are added

## 4.7 Interview 5 with Jakob Hviid - Grounded Theory Cycle 5

The fifth interview was conducted with Jakob Hviid on the 24th of October, 2022. Jakob Hviid is a Senior Architect at Energinet with a Ph.D. in Software Engineering at **SDU**, focusing on **IoT** and energy flexibility.

The fifth interview aims to obtain more in-depth knowledge of the **DSI** and **DMA** and how they relate to storage, data, metadata, data discovery, management, and provisioning. The interview is conducted as a semi-structured interview, with a few questions prepared in advance. The questions can be found in Appendix L.

The open coding process resulted in 41 codes grouped into 17 categories. The results are visible in tbl. L.1.

The axial coding of the fifth interview resulted in the 41 codes being split between seven core categories, and none of these categories are new. The results are summarized in the list below:

- Business Ecosystem - 1 code is added
- Data Management - 10 codes are added
- Data Mesh - 2 codes are added
- Data Spaces - 22 codes are added
- Domain Modelling - 1 code is added
- Infrastructure: Digital - 2 codes are added
- Metadata - 1 code is added

## 4.8 Selective Coding of Interview Data - Grounded Theory Conclusion

The **GT** process resulted in a total of 17 core categories, with a total of 338 codes distributed between them. The categories are summarized in the list below but can also be seen in full in Appendix M:

1. Business Ecosystem - 28 codes
2. Centralization and Decentralization - 24 codes
3. Collaboration - 31 codes
4. Data Management - 34 codes
5. Data Mesh - 28 codes
6. Data Spaces - 64 codes
7. Domain Modelling - 20 codes
8. Flexibility and Grid Balance - 25 codes
9. Governance - 13 codes
10. Infrastructure: Digital - 12 codes
11. Infrastructure: Physical - 9 codes
12. Legislation and Regulation - 11 codes
13. Metadata - 14 codes
14. Open Source vs. Proprietary - 4 codes
15. Roles and Actors - 10 codes
16. Software Qualities - 9 codes
17. Users - 2 codes

The selective coding process involves analyzing the core categories to identify relationships between codes and subsequently develop theories and hypotheses. Relationships between codes within each category were identified by applying several criteria, such as thematic similarity, causal connections, frequency of co-occurrence, and conflicting or complementary perspectives. These criteria aimed to ensure a comprehensive and rigorous analysis of the code relationships, and doing so enabled the discovery of insights that might not have been evident if examining codes individually.

The analysis seeks to create a unified theory for each core category, which aids in maintaining a manageable number of theories and minimizes the risk of producing vague theories. Identified theories will be labeled with a "T," followed by a number signifying the theory number, a title, and an explanation.

Hypotheses derived from the theories will be presented as testable statements and marked with an "H" followed by a number indicating the hypothesis number. To keep the identified hypotheses manageable, the author restricted himself to keeping the number of identified hypotheses at or below four for each

theory. While testing all hypotheses falls beyond the scope of this thesis, their presentation offers an intriguing foundation for future research.

Theories will be listed before the corresponding hypotheses, which will reference the theory from which they were derived.

The hypotheses and theories set the foundation for eliciting non-functional and functional requirements for the prototype implementation of **DMA** in ch. 5. They also serve as the basis for evaluating the theoretical and practical validity of the research and the developed prototyped, as presented in ch. 8. Furthermore, the results articulate the theoretical and practical contributions of the thesis, presented in ch. 9.

Given the extensive nature of the results, the author believes including them in the Appendix is most suitable. Please refer to Appendix N to explore the resulting theories and hypotheses. Readers are encouraged to navigate back and forth between the theories and hypotheses relevant to the requirements in the next chapter. Links have been created to ease the experience.

# 5 | Requirements

This chapter outlines the requirements for a decentralized data space that utilizes a data mesh approach considering the Danish energy sector's needs. The requirements are elicited from hypotheses in Appendix N but also draw inspiration from the theory of the **DSI** (sec. 2.1) and **DMA** (sec. 2.2).

## 5.1 Non-Functional Requirements

The non-functional requirements define a system's characteristics and constraints. The following non-functional requirements are derived from the hypotheses in Appendix N. Formulating the requirements as whole sentences for each expressed need in the hypotheses will quickly become unmanageable, so the non-functional requirements are expressed based on the system's quality attributes that the different hypotheses allude to. The qualities clarify the critical system characteristics for the Danish energy sector while allowing the author freedom in implementation choices.

1. Adaptability — H4, H23
2. Discoverability — H48, H49, H50
3. Flexibility — H5, H6, H7, H47, H52, H53
4. Governance — H22, H32, H33, H34, H35
5. Interoperability — H9, H21, H58, H60
6. Modularity and Extendibility — H42
7. Observability — H17
8. Performance — H13, H24, H36
9. Robustness and Resilience — H62, H63
10. Scalability — H20, H30, H31
11. Security — H38, H43
12. Usability — H61, H64, H65, H66

By addressing these non-functional requirements, a prototype will be better equipped to tackle the challenges associated with the problem statement, namely implementing a decentralized data space using a data mesh approach and improving collaboration among actors in the Danish energy sector.

## 5.2 Functional Requirements

Functional requirements define the specific features and functionalities a system must possess to fulfill its purpose. The elicited functional requirements are linked to the non-functional requirements they address.

1. **Schema Management:** The system should provide tools for managing data schemas, including creating, updating, and sharing schemas — NFR1
2. **Dynamic Domain Models:** The system should support and facilitate different domain models — NFR1, NFR3
3. **Data Discovery and Search:** The system should offer data discovery and search functionality, enabling users to find and access information about the data and its context — NFR2
4. **Data Transformation:** The system should provide the tools and capabilities for data transformation, normalization, and enrichment — NFR3, NFR6
5. **Data Cleansing:** The system should provide tools for deleting, anonymizing, and obfuscating data to ensure compliance with data protection regulations — NFR4
6. **Data Ingestion:** The system should be able to ingest data from various sources, including existing systems, **IoT** devices, and external data providers — NFR5
7. **Data Egestion:** The system should be able to egest data to various destinations, including existing systems, **IoT** devices, and external data consumers — NFR5
8. **Integrate with CNCF tools and technologies:** The system should integrate with **CNCF** tools and technologies to ensure high digital maturity and robustness — NFR5, NFR9

9. **Modern APIs:** The system should provide modern **APIs** to facilitate seamless data exchange, interoperability, and collaboration — NFR5
10. **Modularity and Extensibility:** The system should be designed modularly, allowing for easy addition or removal of components, updates, and extensions — NFR6
11. **Telemetry:** The system should provide capabilities to monitor key performance metrics, such as system load, response times, and error rates, and track critical system behaviors, including data ingestion, processing, and egestion, to ensure optimal performance and reliability — NFR7
12. **Caching:** The system should be able to cache data to improve performance and reduce resource utilization — NFR8
13. **Real-time Data Processing and Querying:** The system should process, analyze, and query data in real-time, leveraging suitable technologies and formats to facilitate rapid decision-making, efficient data exchange, and responsive user experiences — NFR8
14. **Validation** — The system should provide tools for validating data to ensure data quality and compliance with data schemas — NFR9
15. **Microservice Architecture:** The system should be designed as a collection of microservices, allowing for the scalability of individual components and a less coupled architecture — NFR10
16. **Data Storage:** The system must store data in a secure and scalable manner, supporting different data formats and storage solutions — NFR10, NFR11
17. **Authentication and Authorization:** The system should implement fine-grained access control mechanisms to ensure data confidentiality and compliance with regulations — NFR11
18. **Dashboard:** The system should provide a dashboard for interacting with the system and managing data — NFR12
19. **Data Visualization and Analytics:** The system should provide data visualization and analytics tools, catering to the distinct needs of business users and data scientists — NFR12

These functional requirements outline the features and capabilities the prototype should possess to address the challenges of implementing a decentralized data space using a data mesh approach in the Danish energy sector. A production-ready system will likely require additional features and functionalities, but these requirements should provide a solid foundation for a **Minimum Viable Product (MVP)**.

# 6 | Conceptual Design

This chapter provides a high-level overview of the proposed solution, focusing on its core concepts and capabilities while considering the functional requirements presented in ch. 5. The solution is inspired by the concept of data spaces in sec. 2.1.1 and the notion of data mesh in sec. 2.2.1.

The proposed solution is a flexible, distributable data product accommodating many use cases in the Danish energy sector, specifically for Energinet. It leverages **CNCF** technologies to create a data space supported by a data mesh, where data products connect teams, organizations, and backend systems. Figure 6.1 illustrates the solution's conceptual design.

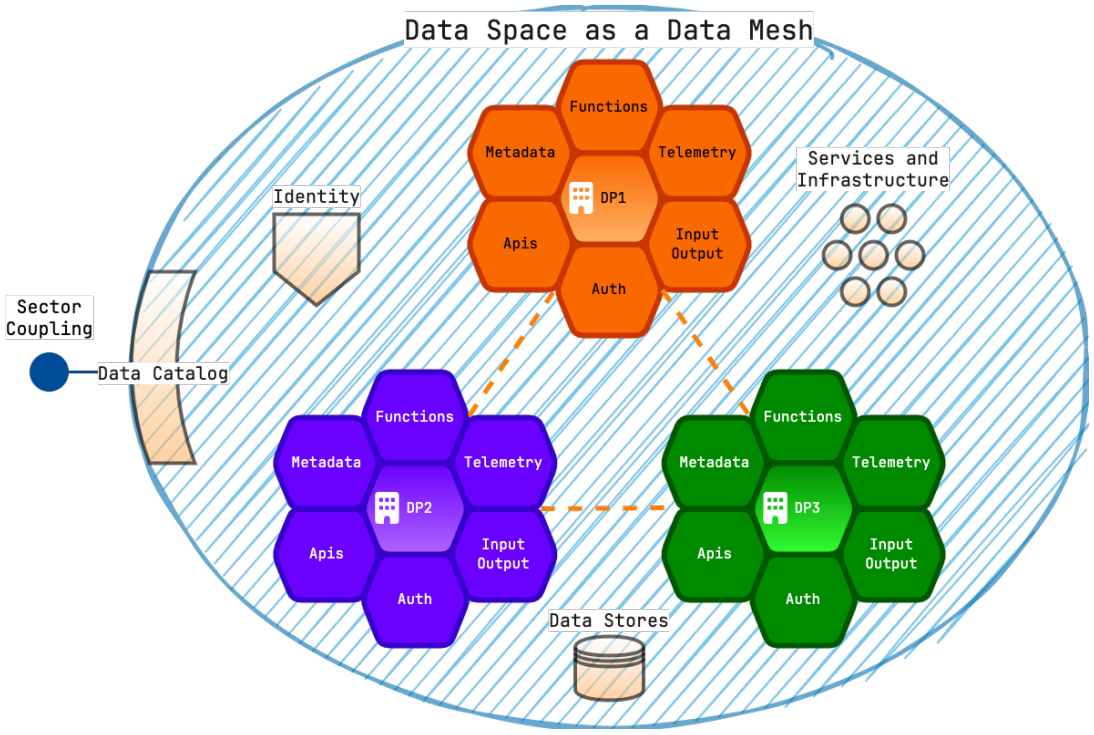


Figure 6.1: The conceptual design of the solution.

The figure shows that data products are deployed within a data space representing some organization or sector's data infrastructure. This infrastructure encompasses a shared data catalog, data stores, identity management, and services and infrastructure components depending on the data products' infrastructure requirements. The data products aim to facilitate better separation of concerns within the sector and provide a uniform way to access and manage data. Each team can concentrate on its domain area and integrate with the necessary services and infrastructure to enrich and expose its data.

Users can interact with the data space by searching for data products in the shared data catalog. The catalog allows users to discover data products and learn about their capabilities. With appropriate permissions, users can interact with the data products programmatically or through a **User Interface (UI)**.

Each data product provides the capabilities required to support its use cases. For example, a data product may provide **APIs** for querying, modifying, and subscribing to data and functions for transforming and enriching data. Ideally, the data product allows access to all its capabilities without requiring users to interact with external systems. This usability feat can, for example, be obtained by proxying requests and embedding external **UIs**.

In an ideal scenario, the solution would include a platform to manage the life cycle of data products, a

function store for teams to find and install functions in their data products, and other enhancements. However, these aspects warrant different solutions and, in many cases, do not yet exist in the form described in theory. If there were to be a platform and a function store, it would be separate entities that would not significantly alter the conceptual design.

## 6.1 The Capabilities of the Solution

The data product is designed from the ground up to be a dynamic entity that can be configured to fit the needs of most teams and organizations. For this purpose, the product can be described with a focus on six critical components:

1. **APIs**
2. Auth
3. Functions
4. Input/Output
5. Metadata
6. Telemetry

These components are the core of the data product and the solution's building blocks. **APIs** are the interfaces that allow for interaction with the data product, the **APIs** must support the common use cases like **Create, Read, Update and Delete (CRUD)** operations, querying, subscriptions, streaming, and more while facilitating seamless data exchange, interoperability, and collaboration.

It is essential that a data product is secure and that it can be configured to only be accessible by users with the proper permissions. It is also essential that the data product can be configured only to allow certain operations to be performed by specific users. For this, authentication and authorization are needed, which should be implemented with fine-grained access control mechanisms to ensure data confidentiality and compliance with regulations.

Functions provide ways to implement business logic on the data product. The functions can implement data transformations, enrichment, and more. Functions must be easy to implement such that the data product can be extended to fit the owner's needs and use cases beyond the data product's core capabilities.

No data product is complete without being able to input and output data. A new data product should be able to source data from existing systems, **IoT** devices, and external data providers, optionally transform it and expose it in a standardized way. Likewise, the data product should be able to output data to support custom data pipelines, long-term storage, and egest data to various destinations, including existing systems, **IoT** devices, and external data consumers, similarly to the input capabilities.

Metadata tells us about the data, capabilities, and context of the data product. The metadata is essential to make the data product discoverable and usable. The system should offer robust data discovery and search functionality, enabling users to explore data products easily. The metadata should provide information about the data product's capabilities, dependencies, data sources, and documentation. Essentially, the more metadata, the better.

Telemetry tells us what is happening and how the data product is used. It is commonly separated into tracing, logging, and metrics. It is essential to have telemetry to monitor the data product, how users use it, and how it performs. Telemetry is also essential to be able to debug the data product and to be able to improve it. It can do so by providing capabilities to monitor key performance metrics, such as system load, response times, and error rates. In many ways, telemetry is the cornerstone of a data product's observability.

Considering these six components, the data product can solve various use cases in the Danish energy sector. However, creating a data product like this is not trivial. Having one product adapt to different data domains and providing standardized functionality for each component requires specific implementations and designs that, to some degree, can depend on the domain model itself. As such, a solution must be able to rely on technologies that allow for dynamicity and flexibility.

## 6.2 Supporting Capabilities

Besides the six main components, a data product requires supporting capabilities to enable the core components to function optimally. There can be many supporting capabilities, but the most important ones for this solution are:

1. Data storage
2. Schema management
3. Dynamicity and flexibility

Data storage ensures that data can be stored and retrieved for each data product. The data storage should be flexible and allow for different storage solutions depending on the use case and preferences of the data product owner. For example, a data product may require a relational database, a document database, a time series database, or even a data lake. Considering where to host the data storage is also essential, as some data products may run on closed networks with restricted internet access, for example, on an **IoT** device. In these cases, data storage should be able to be hosted and run on the same network as the data product or on the device itself.

Schema management should provide tools for managing schemas. Data product owners must be able to create, update, and share schemas. Schemas are essential to data products, as they define the domain-specific data structure a data product acts upon. Sharing these schemas enables interoperability, as schemas can be used to build domain models and enable a shared standard for interpreting data. Moreover, schemas should be flexible regarding schema formats like Avro and JSON schemas or even formats introspected from data sources.

Dynamicity and flexibility are also essential to support the core components. As already implied, the system should be able to adapt to different use cases and data domains. However, this dynamicity and flexibility should also apply to the different implementations of the core components — the system will not thrive well in the Danish energy sector if flexibility demands are not met. Due to the many different data domains and use cases, the system should be able to adapt to the needs of the data product owner, and the data product owner should be able to configure the system to fit their needs.

# 7 | Technical Design and Implementation

This chapter describes the technical design and implementation of the prototype. It first introduces the GitHub repositories containing the code for the prototype and then presents the assemblies that make up the prototype. It then describes the prototype's architecture, including its logical structure and feature pattern used extensively. After this, it describes the configuration and code generation systems in depth.

At this point, the prototype's overall architecture and technical design have been presented, and the chapter will delve into the features and capabilities the prototype provides. The design and implementation will be described for each feature. Lastly, a brief overview of the infrastructural dependencies of the prototype is presented.

Moreover, because it can be challenging to describe technical details to a non-technical audience, the reader is assumed to be comfortable with technical jargon, object-oriented programming, and best practices in .NET development. If not, the reader might want to skip the entirety of the chapter or the technical parts of it.

## 7.1 GitHub Repositories

The prototype's code is open-source under the **Massachusetts Institute of Technology (MIT)** license. The author has tried to split the code between a few repositories to improve the reusability of the code and keep each repository in a manageable size. As such, the prototype is split between three public repositories.

1. The Data Product's repository - Contains the data product and assemblies for its core functionality.
2. The .NET Commons repository - Contains shared libraries that have been used to help implement numerous features.
3. The Homelab repository - Contains the docker-compose files for all infrastructure and services the author is currently running in his homelab. The files include the infrastructure and services required for running the data products.

These repositories, including a few others, have been consolidated in a mono-repository[12] using git submodules. Mono-repositories is a practice recommended by The DevOps Handbook, as it can be beneficial to quickly navigate growing codebases and thus enable teams to work more efficiently [46]. The mono-repository is not public, so it will not be linked here. This decision allowed the author to keep the projects in separate repositories while still being able to consolidate them in one workspace for the thesis project, which makes use of all of them. The approach enabled the author to seamlessly work on all projects at once without having to clone multiple repositories and switch between them frequently—all from within the same **Integrated Development Environment (IDE)**.

## 7.2 The Different Assemblies

The prototype is implemented as a WebApp using .NET 7.0, Roslyn source generators [47] to generate code and a few supporting libraries to help with everyday tasks encountered when developing .NET applications. For example, code generation, string manipulation, and shared functionality. Below is a list of the assemblies that make up the prototype and a short description of their purpose:

- `Devantler.DataProduct` - A .NET 7.0 WebApp, and the main assembly of the prototype. It contains the application's main entry point and logic for all its features.

[12] A mono-repository is a repository that contains multiple projects. It is often used to enable teams to organize better and manage dependent codebases, documentation, and other related assets [46].

- `Devantler.DataProduct.Core.Configuration` - A .NET Standard 2.1 library containing the data product's configuration system. It is used by both the `Devantler.DataProduct` and `Devantler.DataProduct.Generator` assemblies to build the configuration from the user-provided configuration.
- `Devantler.DataProduct.Generator` - A .NET Standard 2.1 Roslyn Source Generator library that generates code for the data product based on the configuration provided by the user.
- `Devantler.DataProduct.CodeGen.*` - A set of .NET Standard 2.1 libraries that contain a modern **API** for generating code with the help of Scriban [48], a popular templating engine for .NET. The `Devantler.DataProduct.Generator` library uses the `Devantler.DataProduct.CodeGen.*` libraries to generate code at compile time.
- `Devantler.DataProduct.StringHelpers` - A .NET Standard 2.1 library with various utility and extension methods for working with strings, for example, converting a string between different casings.
- `Devantler.SchemaRegistry.Client` - A .NET Standard 2.1 library that contains an abstract client for connecting to external Schema Registries. The `Devantler.DataProduct` and the `Devantler.DataProduct.Generator` library uses the `Devantler.SchemaRegistry.Client` to register and retrieve schemas from the local or Kafka schema registry.

## 7.3 The Architecture

The overall architecture of the data product is shown in fig. 7.1. The architecture revolves around a set of features and their interactions with the data pipeline and external infrastructure.

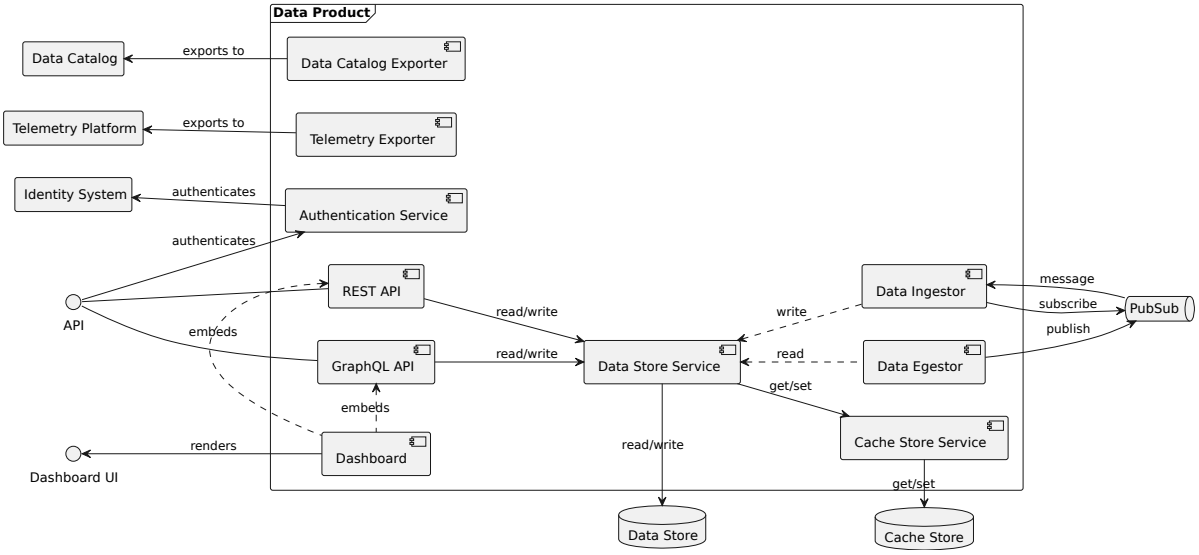


Figure 7.1: The architecture of the data product prototype.

Most features are autonomous, meaning they do not need to interact with the data pipeline to provide their functionality. As such, it is primarily the data pipeline that needs explaining.

Data is either queried, mutated, ingested, or egested from the **APIs**, the data ingestors, and egestors. The data pipeline handles all events related to data input and output. As such, **APIs** and data ingestors and egestors interact with the data store service to query or mutate data. The data pipeline also includes supporting activities like mapping and validating data to ensure that the data is valid and in the correct format. Depending on whether the data is cached, the pipeline can operate efficiently by reading and writing to a data store and a cache store.

Furthermore, the architecture demonstrates how the dashboard can embed internal and external services to provide a single point of entry for users and, thus, a more seamless user experience.

### 7.3.1 The Logical Structure

The data product consists of multiple features (sec. 7.7) implemented by various types, such as classes and interfaces. The types are grouped into folders based on the feature they implement. This decision was made as the project is implemented following a feature pattern (sec. 7.3.2). Thus it enables high cohesion between the code's logical structure and the concepts it implements. The logical structure also makes locating the code responsible for a specific feature more accessible.

An overview of the folder structure is shown in fig. 7.2., which also shows an example of the code that implements the dashboard. Each feature could be implemented as a separate assembly, as the logical structure and feature pattern mixture ensures that the code is loosely coupled. However, as these features are not intended for reuse, doing so is unnecessary.

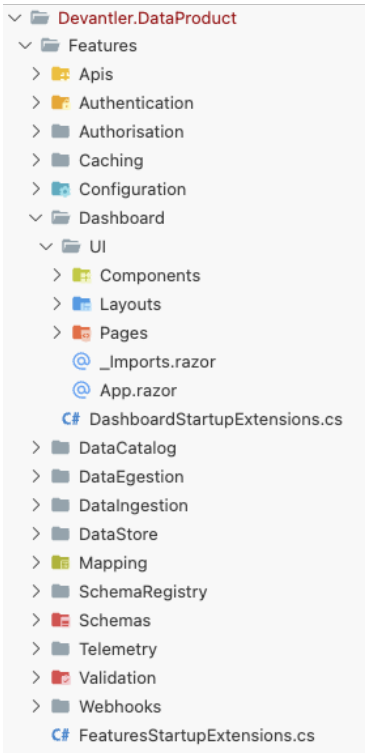


Figure 7.2: The logical structure of code in the prototype.

### 7.3.2 The Feature Pattern

Jimmy Bogard's **Vertical Slice Architecture (VSA)** inspires the feature pattern employed. The **VSA** sets out to solve a common problem with traditional monolithic software architectures where code is organized based on types and layers. **VSA** proposes that code should be organized based on vertical slices instead, where a vertical slice is a collection of code concerned about the same thing. Organizing code based on its purpose means code from multiple layers can be in the same module, minimizing the coupling between different slices and maximizing cohesion within a slice [49]. A demonstration of the **VSA** is shown in fig. 7.3.

The **VSA** is a good fit for the data product because it is a monolithic application consisting of a set of modular features where the **VSA** principles apply. In this context, a slice is considered a feature and is organized as presented in fig. 7.2, hence calling it the feature pattern.

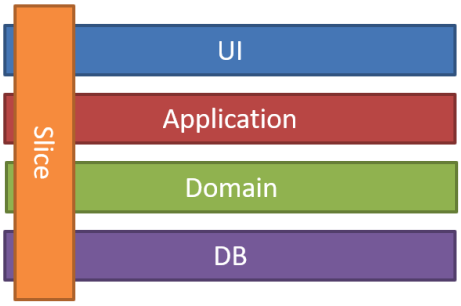


Figure 7.3: The vertical slice architecture [49].

Ensuring that the logical structure and the architecture align with this modular approach is vital for the data products' readability, maintainability, and extensibility. Furthermore, it is imperative to enable the different features to be extracted and migrated to microservices in the future. Doing so will not require extensive changes to the code, as a typical microservice will follow the same principles as the **VSA**. For example, a microservice will typically have business and data access logic in the same module, as both would be needed for a microservice to provide analytical data for a specific business use case.

Quite a few functionalities have been implemented in the prototype to support the feature pattern. `Microsoft.FeatureManagement` [50] is used to enhance the capabilities of feature flags, allowing much more granular control over the data product's features—for example, the option to disable specific endpoints or enable features for a percentage of users. The library does not have an **API** for managing **Dependency Injection (DI)** registrations, so the data product relies on conditional logic. In the prototype, managing **DI** registrations cover most use cases for feature flags, but designing the system around more advanced use cases can become important for the data product's future. For example,

having granular control over features can be helpful for A/B testing[13] or Canary Releases[14].

The feature pattern is implemented using a **DI** flow, where the entry point of the application is responsible for calling the `AddFeatures()` extension method, which then propagates the **DI** registrations to the different features. The design enables a simple **DI** registration flow, where each service is responsible for its service registrations, thus fostering high cohesion between the code and its logical structure. An example of the feature registration is shown in fig. 7.4.

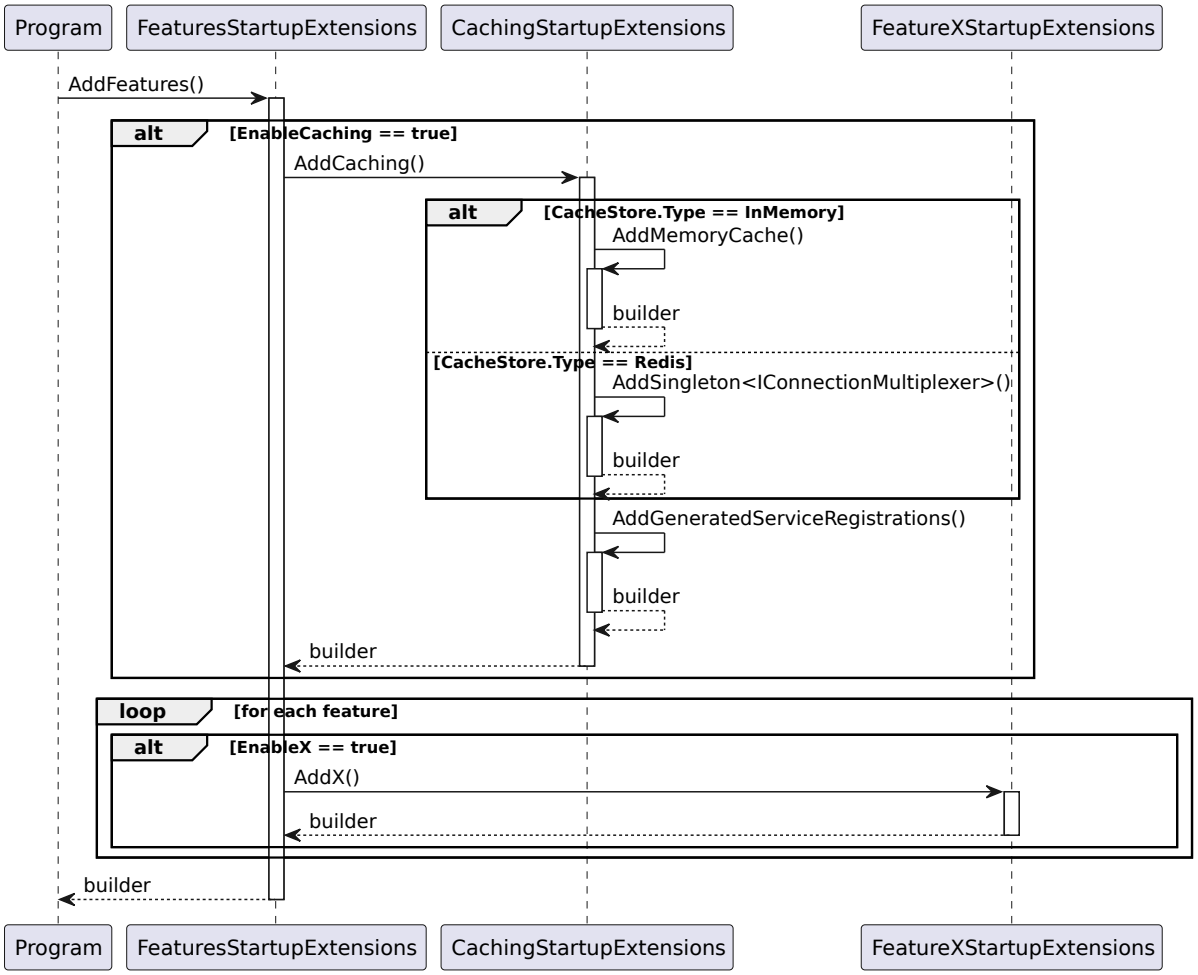


Figure 7.4: An example of the data product's feature registration flow.

In the example, the `AddCaching()` method is responsible for registering the correct cache store and its dependent services and configuring them. The cache store implementations require dynamicity, as they depend on the domain model. The `AddGeneratedServiceRegistrations()` register these implementation-specific services by delegating the registration to its partial mirror method on a generated class.

## 7.4 The Configuration System

The configuration system is a core functionality of the data product, as it is designed to be highly configurable and adaptable to a wide range of environments. This flexibility allows the data product to

[13] A/B testing is a technique that can be used to compare two versions of a feature to determine which version is favored by users [51].

[14] Canary releases is a technique that can test new features in production by gradually rolling out the feature to a subset of users [51].

cater to the specific needs and requirements of different teams and use cases. The configuration system is extensible and easily integrates new features and capabilities.

Users can provide configuration values as a **JavaScript Object Notation (JSON)** or **YAML Ain't Markup Language (YAML)** file, environment variables, or command line arguments. The system prioritizes the configuration sources in the following order:

1. Command line arguments
2. Environment variables
3. **YAML** files
4. **JSON** files

The prioritization ensures that if a higher-priority configuration source provides a configuration value, it will override the value provided by a lower-priority configuration source. For example, if a value is provided as an environment variable and in a **YAML** file, the value provided by the environment variable will be used. This prioritization of configuration values is a common practice and should be followed to ensure the configuration system is predictable and easy to use.

.NET supports command line arguments, environment variables, and **JSON** files out of the box. However, it does not support **YAML** files. As such, the prototype uses the `YamlDotNet` library [52] to read and parse **YAML** files and the `NetEscapades.Configuration.Yaml` library [53] to map the configuration values from the **YAML** file to .NET's configuration system.

An example of a simple **YAML** configuration file for one configured data product can be seen in lst. 7.1. The configuration can become more extensive, but the complexity is limited to the added cognitive load that comes with a more extensive set of configuration values. The different configuration sections are similar and, thus, easy to understand.

**Listing 7.1** An example of a YAML configuration file for a simple data product.

```yaml
Name: Contoso University Data Product Local
Description: A data product to query and manage students and courses...

License:
  Name: MIT
  Url: https://opensource.org/licenses/MIT

Owner:
  Name: Test User
  Email: test@email.com

FeatureFlags:
  EnableDataIngestion: true

DataIngestors:
  - Type: Local
    FilePath: assets/data/contoso-university-multiple.json
```

### 7.4.1 The Capabilities of the Configuration System

The configuration system supports a wide range of features and capabilities. However, it only implements the use cases the author experienced necessary to build, test, and demonstrate the prototype. The capabilities of the configuration system are:

1. **Multiple configuration sources** - The configuration system supports multiple configuration sources, which allows the user to provide configuration values from command line arguments, environment variables, and **YAML** or **JSON** files.
2. **Polymorphic objects** - The configuration system supports polymorphic objects, which allows the user to configure different implementations of the same feature.
3. **IntelliSense and validation** - The configuration system supports IntelliSense[15] and validation

[15] Intellisense is a general term for a tool's capability to provide code completion and validation [54].

through a **JSON** schema, which allows the user to get feedback on what configuration values are valid and what they do.

Besides these capabilities, the configuration system can configure the following features and metadata:

- **Name and description** - Configures the name and description of the data product.
- **Release** - Configures the current version release of the data product. When deploying the data product with containers, this value is automatically set from the image semver tag.
- **License** - Configures the data product's license.
- **Owner** - Configures the owner of the data product.
- **Feature Flags** - Configures which features to enable or disable.
- **APIs** - Configures the **APIs** exposed by the data product.
  1. **Representational State Transfer (REST) API** - Configures the **REST API**, its endpoints, and its features.
  2. **GraphQL API** - Configures the GraphQL **API**, and its features.
- **Schema Registry** - Configures the schema registry used by the data product.
- **Cache Store** - Configures the cache store used by the data product.
- **Data Store** - Configures the data store the data product uses.
- **Data Ingestors** - Configures the data ingestors the data product uses.
- **Data Catalog** - Configures the data catalog used by the data product.
- **Dashboard** - Configures the dashboard, its embedded services, and links.

Most of these capabilities have a configuration section, with the possibility of configuring different aspects of the setting. For example, the data product's schema registry has settings for configuring the schema registry type, and if it is a Kafka schema registry, the schema registry **Uniform Resource Locator (URL)** can also be set.

### 7.4.2 Polymorphic Objects

Polymorphic objects are a crucial feature for some use cases of the data product. However, the .NET configuration system has limited support for polymorphic objects, as it requires mapping objects in the configuration to concrete models in the code. The limitation can make it challenging for users to understand and configure data products. An example of this limitation and the resulting inflexibility can be seen in lst. 7.2.

**Listing 7.2** An example of a YAML configuration with or without polymorphic objects.

```yaml
DataStore: # Polymorphic configuration. The same section can be used for different data stores.
  Type: SQL # Determines the data store type.
  Provider: MySQL # Determines the data store type provider (if applicable)
  ConnectionString: "Server=localhost;Database=DataProduct;Trusted_Connection=True;"
  TypeAndProviderSpecificProperty: 1

MySQLDataStore: # Static configuration. Each Data Store requires a new section.
  ConnectionString: "Server=localhost;Database=DataProduct;Trusted_Connection=True;"
  MySQLSpecificProperty: 1
```

The prototype addresses this issue by manually resolving the configuration at startup, which enables mapping the configuration to polymorphic objects while maintaining a small maintenance cost, requiring adding a few lines of code for each new configuration section. This approach relies on interfaces and overrides and is implemented through extension methods in the `ConfigurationExtensions` class. The resulting flexibility is essential for accommodating different configurations and requirements.

The implementation revolves around a concrete type based on the configuration key and the `Type` property, instructing the manual resolution to use the matching object to deserialize the configuration section. With the approach, configurations sharing the same key can have different object types if they share the same base type and interface.

Although the configuration aims to utilize the polymorphic capability implemented, consolidating the configuration as much as possible is preferred whenever it is possible to use the same configuration across different implementations of a feature. It helps to avoid a configuration hell.

### 7.4.3 Configuration Schema and User Experience

A **JSON** schema [55] is provided to enhance the user experience of configuring the data product. This schema provides IntelliSense [54] and validation features for **IDEs** that support **JSON** schemas and is also used to generate documentation for the configuration file. Default values are provided for the configuration, allowing the data product to function even if the user does not provide any configuration. However, a user-defined configuration is required for most use cases. In lst. 7.3, a part of the configuration schema is shown.

**Listing 7.3** A part of the JSON schema responsible for documenting the Name, Description and Release property.

```
{
    "title": "Data Product Config",
    "description": "Specification for a data product config",
    "type": "object",
    "properties": {
        "Name": {
            "description": "The name of the data product",
            "type": "string"
        },
        "Description": {
            "description": "A description of the data product.",
            "type": "string"
        },
        "Release": {
            "description": "The current release of the data product. Uses semantic versioning (vX.Y.Z).",
            "type": "string",
            "pattern": "^v[0-9]+\\.[0-9]+(\\.[0-9]+)?$"
        },
    },
    "required": [
        "Name",
        "Release"
    ],
}
```

The main challenge with using a **JSON** schema is maintaining consistency between the actual configuration and its documentation. Maintaining consistency may require additional effort to ensure the schema accurately reflects the configuration.

## 7.5 The Code Generation System

The data product is designed to be highly configurable and dynamic, and code generation is crucial in achieving this dynamicity. Code generation allows the data product to adapt to different configurations and schemas, offering greater flexibility and customization. The code generation system is responsible for generating concrete types and methods for the dynamic parts of the data product, and it is implemented with Roslyn source generators [47].

This section describes the code generation system by first discussing the motivation for using code generation over other options. Then the inner workings of the code generation system are discussed, including its reliance on source generators, its limitations, and how it uses the Scriban templating engine [48] to generate high-quality code. Lastly, all the different code generators are presented.

### 7.5.1 Reflection vs. Code Generation vs. Dynamic Data Modelling

Choosing the appropriate method for making the data product dynamic was challenging. There are three primary options:

1. Reflection.
2. Code generation.
3. Use of a dynamic data model, e.g., with dictionaries.

Reflection is a powerful tool for inspecting and modifying code at runtime. It enables the highest level of dynamicity, allowing types and methods to be created at runtime, but it comes with trade-offs. Reflection can be a performance bottleneck and negatively impact the code's readability and maintainability. Additionally, many third-party libraries do not support reflection well, which can complicate development. Although a reflection-based approach would theoretically provide the highest level of dynamicity, it could introduce issues and limitations that make maintaining and developing the data product complex [56].

On the other hand, code generation addresses the dynamicity problem by generating code at or before compile time. Generating code at compile time allows for better performance and readability. However, code generation cannot be used to generate code at runtime, necessitating recompilation of the codebase whenever new code must be generated. This requirement creates limitations, such as being unable to update a data product's schema at runtime, resulting in the data structure potentially becoming incompatible with the generated schema and breaking the data product's capabilities.

The third option involves building a dynamic data model with, e.g., a dictionary, to allow the data product to manage various models by mapping their values to a dictionary before processing them. This approach requires writing boilerplate code to handle different data models and storing data in non-optimal data structures, which can affect performance. However, it also enhances flexibility and unifies the storage model, allowing for data storage in different data stores without implementing a specific and optimized data model for each data store. This option was ultimately ruled out due to its impracticality and misalignment with data modeling best practices, such as when modeling relational data.

Between the two viable options, code generation emerged as the best choice for the prototype, as it enables a more maintainable and readable codebase and a faster development process. The limitation of dynamicity being bound to compile time is not necessarily a negative factor. It will not cause issues if a team or organization agrees to avoid updating database schemas manually and instead relies on **Continuous Integration (CI) / Continuous Delivery (CD)** pipelines to make breaking changes in a controlled manner. However, a hybrid approach might be worth considering if the data product is to be developed further. This approach would involve code generating most of the code at compile time, with reflection being used to generate domain models at runtime. Such an approach would allow the data product to react to changes in the database schema at runtime while maintaining a readable and maintainable codebase. Ultimately, enabling the database schema to be reflected would make the data product more flexible and robust. Nonetheless, this thesis does not further investigate the feasibility of this approach.

### 7.5.2 Generating Code with Source Generators

As mentioned, the code generation system uses Roslyn source generators [47], specifically incremental source generators [57]. Source generators are a relatively new feature introduced with .NET 5.0, and their **API** was further improved with the release of incremental source generators in .NET 6.0.

Source generators enable the compiler to analyze and generate code at compile time. They are more commonly used in the industry for creating analyzers that can assess code and provide feedback to developers. In addition to analyzing code, source generators can also generate code at compile time, as the prototype demonstrates [47].

Incremental source generators build upon the capabilities of source generators by allowing the compiler to cache results and only re-run a generator if its generation output has changed. This feature is highly beneficial as it enables the code generation system to be fast and efficient, generating code only when necessary [57].

### 7.5.3 Limitations of Source Generators

Source generators, being a relatively new feature, have some limitations. Firstly, they do not support dependencies well and necessitate verbose configuration in the `.csproj` file. Another limitation is that source generators cannot access the file system directly due to security and reproducibility concerns.

The `.csproj` file must be extended with the appropriate configuration to load the necessary **Dynamic Link Librarys (DLLs)** to add assembly dependencies from other libraries to the source generator. The configuration requires adding a custom target, `GetDependencyTargetPaths`, which executes before

the generator and loads the **DLLs** into the source generator's context. The prototype's `.csproj` file configuration can be seen in lst. 7.4.

**Listing 7.4** A snippet of how to add third-party libraries to Source Generators.

```
<Project Sdk="Microsoft.NET.Sdk">
    <ItemGroup>
        <PackageReference Include="Chr.Avro.Json" Version="9.5.0" GeneratePathProperty="true"
↪  PrivateAssets="all" />
    </ItemGroup>
    <Target Name="GetDependencyTargetPaths">
        <ItemGroup>
            <TargetPathWithTargetPlatformMoniker Include="$(PKGChr_Avro_Json)\lib\netstandard2.0\*.dll"
↪  IncludeRuntimeDependency="false" />
        </ItemGroup>
    </Target>
</Project>
```

To enable a source generator to read static files, they must be embedded in the assembly using the `AdditionalFiles` property in the `.csproj` file. The inability to access the file system presents a challenge when generating code with a template engine, as template engines often require template files representing the code structure to be generated. Since the source generator cannot read template files from disk without loading all the template files into the assembly, a workaround is required. Loading the template files into the assembly was ruled out as a viable option due to difficulties related to static files being in different libraries and source generators not handling that well. Instead, the prototype implemented a custom template loader to load templates from string properties. The template loader, `CSharpTemplateLoader`, is demonstrated in lst. 7.5 and enables the source generator to generate code entirely from memory.

**Listing 7.5** The CSharpTemplateLoader implementation.

```
public class CSharpTemplateLoader : InMemoryTemplateLoaderBase
{
    public override string Load(TemplateContext context, SourceSpan callerSpan, string templatePath)
    {
        var type = Type.GetType(templatePath);
        var property = type.GetProperty("Template");
        return (string)property.GetValue(type, null);
    }
}
```

Together these two workarounds enable the code generation system to generate code from templates while using third-party libraries to support the generation process.

### 7.5.4 Implementing Generators

Once suitable workarounds for the limitations of source generators were identified, implementing generators became a straightforward process. Each generator inherits from the base class `GeneratorBase`, responsible for loading additional files and building the configuration. After inheriting from the base class, a generator must provide the necessary code to generate the types or methods for which it is responsible. A simple example of a code generator is the `SchemaGenerator`, as shown in lst. 7.6.

**Listing 7.6** A trimmed version of the SchemaGenerator implementation.

```
[Generator]
public class SchemaGenerator : GeneratorBase
{
    public override Dictionary<string, string> Generate(Compilation compilation
    ↪  ImmutableArray<AdditionalFile> additionalFiles, DataProductOptions options)
    {
        var codeCompilation = new CSharpCompilation();
        foreach (var schema in rootSchema.Flatten().FindAll(s => s is RecordSchema).Cast<RecordSchema>())
        {
            //Omitted @class construction for brevity
            _ = codeCompilation.AddType(@class);
        }
        var generator = new CSharpCodeGenerator();
        return generator.Generate(codeCompilation);
    }
}
```

The **API** used for code generation is in the custom library, `Devantler.DataProduct.CodeGen.CSharp`, that generates C# code by constructing a `CSharpCompilation`, adding types to it, and then passing it to the `CSharpCodeGenerator` that generates the code and returns it as a dictionary of file names and code content.

A custom code generator was chosen over alternatives like `System.CodeDom` [58] or `SyntaxFactory` [59] because it allowed for a more flexible and user-friendly **API**. Both options were considered, but `System.CodeDom` generated outdated and inflexible code, while the `SyntaxFactory` **API** was complex and poorly documented, making it challenging to use. The latter might have been a viable option, but the added flexibility of owning a custom code generator made generating the desired code output more accessible.

### 7.5.5 Using the Scriban Template Engine

The code generation process involves more than just using the `CSharpCodeGenerator` **API** to generate code. Behind the scenes, a templating engine, Scriban [48], defines and generates code from highly flexible templates.

Each model that can be fed to the `CSharpCompilation` has a corresponding Scriban template that defines the code for its model. One example is the `CSharpClass` model's template, as shown in lst. 7.7. The templates are recursively evaluated when the `CSharpCodeGenerator` generates the code.

In the template, various concepts of Scriban are presented. The `{{}}` syntax presents a Scriban expression, which can be a variable, a function call, or a control flow statement.

- The `{{~}}` syntax is used to trim whitespace before an expression.
- The `{{-}}` syntax is used to trim whitespace after an expression.
- The `{{ include }}` syntax is used to include another template in the current template.
- The `{{ for }}` syntax is used to iterate over a collection.
- The `{{ if }}` syntax includes a part of the template conditionally.
- The `{{ end }}` syntax ends a control flow statement.

As demonstrated in lst. 7.7, the templates can become quite complex. One aspect that helps to avoid this complexity is the ability to include other templates. This feature has been used extensively to make the templates more readable and prevent code duplication. For example, the `CSharpClass` template includes the `CSharpField`, `CSharpConstructor`, `CSharpProperty`, and `CSharpMethod` templates, which are used to generate the code for the fields, constructors, properties, and methods of the class, respectively.

This modular approach allows for better organization and maintainability of the templates. Breaking down the templates into smaller, more focused pieces makes understanding and modifying them easier. Moreover, this strategy encourages reusability, as these smaller templates can be included in multiple places as needed, reducing the overall amount of duplicated code.

In conclusion, using the Scriban template engine in combination with the custom `CSharpCodeGenerator` **API** provides a powerful and flexible code generation system. Developers can easily define complex code

structures using templates while the generator efficiently produces the desired output. This approach significantly reduces the manual coding needed and helps maintain consistency across the generated code. Overall, this code generation method is invaluable, increasing productivity and ensuring high quality in the generated code.

**Listing 7.7** The CSharpClass template.

```
{{~ if !(base_class?.namespace | string.empty)  ~}}
using {{ base_class.namespace }};
{{~ end ~}}
{{- for using in imports ~}}
{{ include 'using' using }}
{{~ end ~}}
{{~ if !(namespace | string.empty) ~}}
namespace {{ namespace }};
{{~ end ~}}
{{~ if doc_block ~}}
{{ include 'doc_block' doc_block }}
{{- end ~}}
{{ visibility != "Private" ? (visibility | string.downcase) + " " : "" }}{{ is_static ? "static " : ""
↪  }}{{ is_abstract ? "abstract " : "" }}{{ is_partial ? "partial " : "" }}class {{ name }}{{ if
↪  base_class || (implementations | array.size > 0) }} : {{ end }}{{ base_class ? base_class.name : ""
↪  }}{{ if implementations | array.size > 0 }}{{ base_class ? ", " : "" }}{{ for implementation in
↪  implementations }}{{ implementation.name }}{{ if !for.last }}, {{ end }}{{~ end ~}}{{~ end }}
{
    {{~ for field in fields ~}}
    {{ include 'field' field }}
    {{~ end ~}}
    {{~ for constructor in constructors ~}}
    {{ include 'constructor' constructor }}
    {{~ end ~}}
    {{~ for implementation in implementations ~}}
    {{~ for property in implementation.properties ~}}
    {{ include 'property' property }}
    {{~ end ~}}
    {{~ end ~}}
    {{~ for property in properties ~}}
    {{ include 'property' property }}
    {{~ end ~}}
    {{~ for implementation in implementations ~}}
    {{~ for method in implementation.methods ~}}
    {{ include 'method' method }}
    {{~ end ~}}
    {{~ end ~}}
    {{~ for method in methods ~}}
    {{ include 'method' method }}
    {{~ end ~}}
}
```

### 7.5.6 The Code Generators

Twelve code generators have been implemented in the prototype to generate various types and functions required by features. These generators play a vital role in reducing development time and effort. Some of these generators generate partial code, taking advantage of .NET's ability to mark a class as partial. Partial types allow certain functionality to be implemented in another file offering the following benefits:

- Generates the least amount of code possible by not requiring generating new types and methods to extend the functionality of non-generated code.
- Facilitates code extensibility by allowing developers to add custom functionality to the generated code without affecting non-generated code.

The implemented code generators are as follows:

- `AutoMapperProfileGenerator` - Generates an AutoMapper profile for mapping between request and response models and entities.
- `CachingStartupExtensionsGenerator` - Generates a startup extension for registering the configured caching service.

- `DataIngestionStartupExtensionsGenerator` - Generates a startup extension for registering the configured data ingestion service.
- `DataStoreServiceGenerator` - Generates generic data store service implementations to provide data access for each entity in the data product.
- `DataStoreStartupExtensionsGenerator` - Generates a startup extension for registering the configured data store service.
- `DbContextGenerator` - Generates an Entity Framework Core `DbContext` for the configured relational database, with the necessary `DbSet` properties for each entity in the data product.
- `EntitiesGenerator` - Generates the entity classes for each entity in the data product.
- `GraphQLQueryGenerator` - Generates the GraphQL `Query` class with the necessary methods to query the different entities in the data product.
- `RepositoryGenerator` - Generates the implementations of the generic repository to provide the necessary data access operations for each entity in the data product.
- `RestBulkControllerGenerator` - Generates the **REST API** controller for **CRUD** bulk operations on the entities in the data product.
- `RestControllerGenerator` - Generates the **REST API** controller for single **CRUD** operations on the entities in the data product.
- `SchemaGenerator` - Generates the schema classes for each entity in the data product.

These twelve code generators efficiently automate the creation of various components of a data product, ranging from data access and caching to **REST** and GraphQL **API** support. By employing these generators, developers can significantly reduce manual coding effort and deliver high-quality data products quickly.

## 7.6 The Compilation Process

This section will explain the purpose of code generation for the data product. To better understand the role of code generation, a sequence diagram of the initialization of a data product is presented in fig. 7.5, demonstrating its importance in enabling the data product to handle various configurations and schemas.

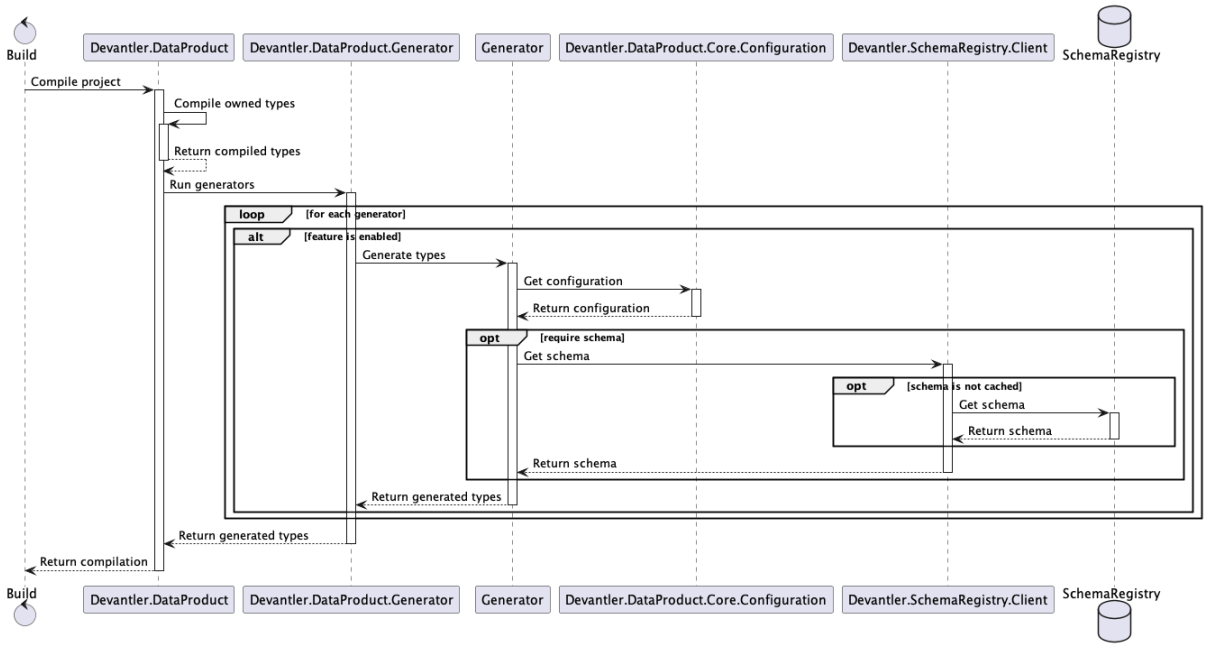


Figure 7.5: The sequence of events that occur when a data product is initialized.

When the data product is built, the Roslyn source generators will be activated to regenerate implementations for its various dynamic features. The generation relies on the configuration to determine the exact

code to generate, so changes to the configuration may require rebuilding the project. The code generation might require knowing the schema of the data product, which can require a direct connection to the schema registry to load and read the data product's schema. One example where the schema is required is when generating the domain model, which the data product requires to serialize and deserialize data. Ultimately the sequence diagram demonstrates how code generation is crucial to bootstrapping the data product, and it is a key enabler for the data product to achieve its goal of being configurable and flexible.

## 7.7 Features and Capabilities

A total of 12 features were planned for the prototype, aimed at addressing the requirements for the prototype, as identified in ch. 5. Of these, eight were implemented, one was partially implemented, and three were not implemented. This section will introduce these features, but first, a brief overview of the limitations of the features is given, as it helps set the prototype's scope and context.

### 7.7.1 Limitations of the Features

The author makes some assumptions about the requirements listed in ch. 5, and deliberately limits the prototype's scope to a subset of the requirements. The goal is to demonstrate a proof of concept with limited available time and not create a production-ready system.

For this purpose, the author decided to implement the prototype as a monolithic application instead of with a microservice architecture. The author believes the prototype will be easier to implement as a monolithic application, as avoiding the added complexity of the microservice architecture will allow a deeper focus on the prototype's features rather than how well it will perform and scale in a production environment. Special care has been taken to ensure the modularity and extensibility of the prototype such that, eventually, migrating to a microservice architecture will be possible. Code implementing the features can be reused and moved to microservices following a refactoring pattern like the strangler fig pattern [60].

A few features are prioritized over others, meaning there was not enough time to implement authentication, authorization, data egestion, and functions in the prototype. The features were planned, and designs have been considered, so this chapter will still describe the design and planned implementation of these features.

The lack of functions is a significant limitation of the prototype, as it dramatically limits the data product's flexibility. Without some means of implementing business logic to transform the data or perform custom operations, the data product is not very useful in a real-world scenario. The author believes this is the most significant limitation of the prototype. However, it is also one of the more challenging features to implement, as it requires many dependencies on infrastructure. There are few and far between good alternatives to non-proprietary **Function as a Service (FaaS)** solutions. OpenFaaS [61] is the closest thing to a non-proprietary **FaaS** solution, but it is very limited in its free tier, so it might not be the best choice. Alternate solutions like loading **DLLs** at runtime or using hosted services in .NET could also be considered. However, ultimately the author believes implementing functions would be the most scalable and flexible solution. It also matches well with an app store in data spaces, where data apps would share the same purpose as functions.

Lastly, most features focus on a single use case or a subset of use cases. As such, the prototype is nowhere as flexible and interoperable as the envisioned data product. The author believes reaching this level of flexibility is an iterative process of prioritizing, prototyping, testing, and refactoring features. This process will be one of the primary tasks if the prototype is to be developed further. There is also a discussion on whether the base data product should provide all implementations or whether it should be possible to extend the data product with custom implementations instead. Implementing a default stack for the data product will improve usability at the cost of flexibility. The author firmly believes in open-source and thinks improving support for third-party libraries could enable a community to form and thus help define and implement the data product, which would benefit the project. If a community-based implementation becomes widespread, it could be adopted by the base data product as a stable implementation, and thus everyone would benefit from the community's work.

### 7.7.2 Apis

**APIs** play a crucial role in data products, allowing users to access and manipulate the product's underlying data. The prototype supports two types of **APIs**: a **REST API** and a GraphQL **API**. The **REST API** uses ASP.NET Core's **Model-View-Controller (MVC)** framework, and the GraphQL **API** is implemented using Hot Chocolate [62], a popular GraphQL framework. This subsection will discuss the design and implementation of both **APIs** and how they contribute to the data product's functionality and flexibility.

#### REST Api

The **REST API** is a crucial component of the data product, providing a standardized interface for users to interact with the data. By implementing a **REST API**, various clients can easily consume the data product, including web applications, mobile apps, and other services. The **REST API** in the prototype is implemented using ASP.NET Core's **MVC** framework [63], which offers a robust and well-supported platform for creating **API** endpoints. This subsection will discuss the design and implementation of the **REST API** and how it contributes to the overall functionality of the data product.

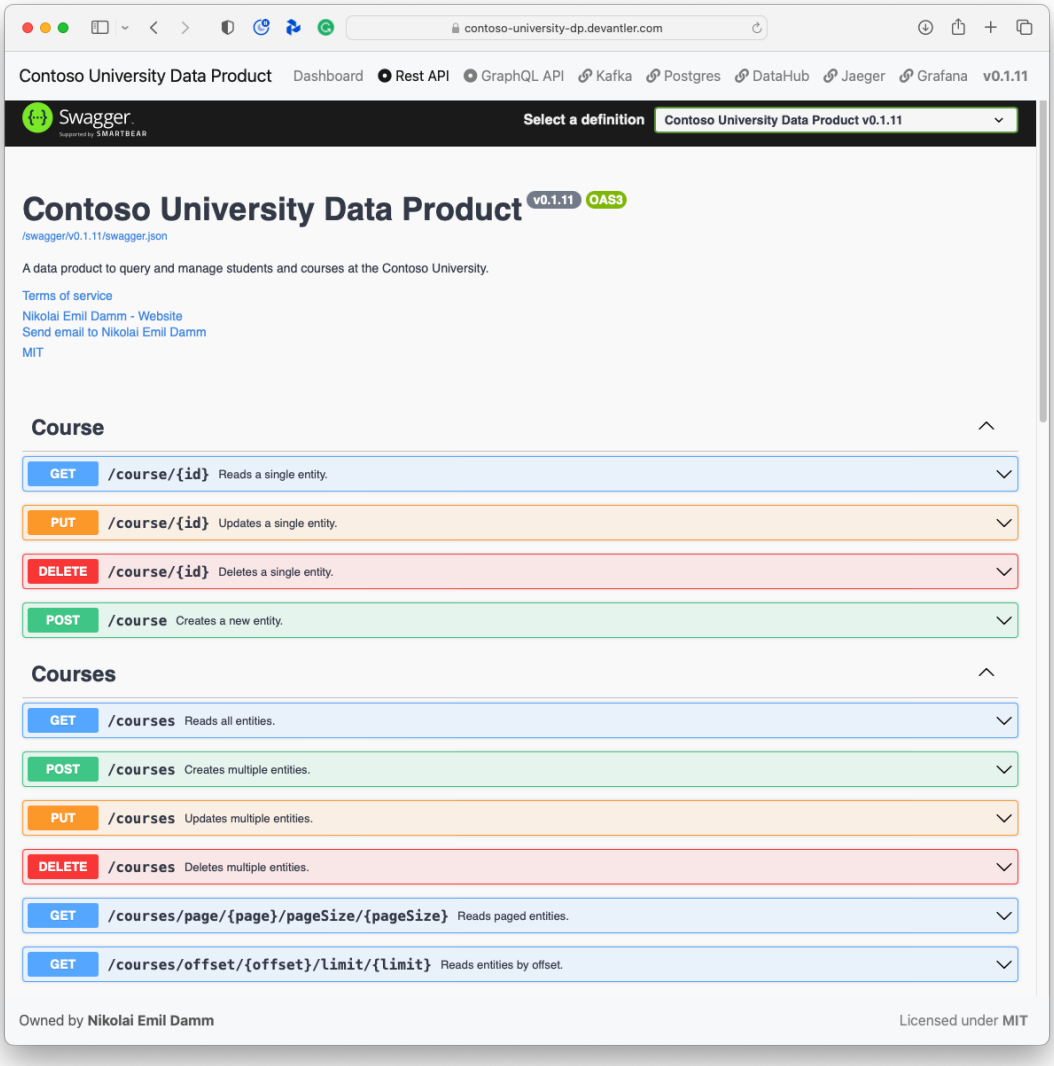


Figure 7.6: An embedded swagger **UI** page for the Contoso University Data Product's OpenAPI Specification.

A controller class is created and decorated with the `ApiController` attribute to implement endpoints

with the framework. The attribute tells the framework that the class is a controller and should be used to handle requests. The controller class contains methods decorated with the `HttpGet`, `HttpPost`, `HttpPut`, and `HttpDelete` attributes. The attributes tell the framework which **Hypertext Transfer Protocol (HTTP)** method the method should handle. The methods or the class can also be decorated with the `Route` attribute to specify the route one or multiple methods should handle. The framework will then automatically map the method to the route and **HTTP** method. An example of the generic controller responsible for handling single **CRUD** operations for different schemas can be seen in lst. 7.8.

**Listing 7.8** An example a method in the generic controller responsible for handling single CRUD operations.

```
namespace Devantler.DataProduct.Features.Apis.Rest.Controllers;

[ApiController]
[Route("[controller]")]
public abstract class RestController<TKey, TSchema> : ControllerBase where TSchema : class, ISchema<TKey>
{
    readonly IDataStoreService<TKey, TSchema> _dataStoreService;

    protected RestController(IDataStoreService<TKey, TSchema> dataStoreService)
        => _dataStoreService = dataStoreService;

    [HttpPost]
    public async Task<ActionResult<TSchema>> PostAsync(TSchema model, CancellationToken cancellationToken
    ↪  = default)
        => await _dataStoreService.CreateSingleAsync(model, cancellationToken);

    [HttpGet("{id}")]
    public async Task<ActionResult<TSchema>> GetAsync(TKey id, CancellationToken cancellationToken =
    ↪  default)
    {
        var result = await _dataStoreService.ReadSingleAsync(id, cancellationToken);
        return Ok(result);
    }
}
```

Besides utilizing the ASP.NET Core's **MVC** framework, the **REST API** has been implemented according to the OpenAPI specification [64]. .NET has excellent support for the OpenAPI specification and allows the specification to be generated using the `Swashbuckle.AspNetCore` library [65]. Endpoints and documentation are generated from the attributes and documentation comments added to the controller and its methods. A friendly **UI** can allow the user to interact with the **API** and see the documentation. The swagger **UI** can be accessed by navigating to the `/swagger` endpoint or the dashboard's embedded swagger **UI**. The **UI** is shown in fig. 7.6.

As the data product supports single and bulk endpoints, a decision was made to have the single endpoints mapped to the `/{modelName}/` route and the bulk endpoints mapped to the `/{modelName}s/` route. The decision was made because having two endpoints share the same route and determining the implementation based on properties was not supported. The best practice is to have single endpoints use the `/{modelName}s/` route and use, for example, `/{modelName}s/{id}/` to get a single entity. However, doing so cluttered the **UI** and mixed the single and bulk endpoints. So to keep the **UI** clean and consistent, the above decision was made.

### GraphQL Api

In addition to the **REST API**, the data product prototype also includes a GraphQL **API** to provide an alternative, more flexible interface for clients to interact with the data. GraphQL **APIs** allows clients to request the needed data, leading to more efficient data retrieval and reduced bandwidth usage. By implementing both **REST** and GraphQL **APIs**, the data product can cater to a broader range of use cases and client preferences. This subsection will describe the GraphQL **API** implementation using the Hot Chocolate framework, its benefits, and its role in enhancing the data product's functionality.

The GraphQL **API** is implemented using Hot Chocolate [62], a popular GraphQL framework for .NET. The framework is a powerful tool for creating GraphQL **APIs** and allows the creation of the GraphQL schema from C# types. Typically, the schema is created by defining a schema in GraphQL **Schema**

**Definition Language (SDL)** or with custom resolvers. However, Hot Chocolate enables schemas to be defined by annotating C# types with attributes that tell the framework how to resolve the type. The approach simplifies using GraphQL with .NET a lot. It allows for a more maintainable and readable codebase, where extensive knowledge of the intricacies of GraphQL is not required to create an **API**. As Hot Chocolate supported this approach, it was the chosen GraphQL framework for the prototype.

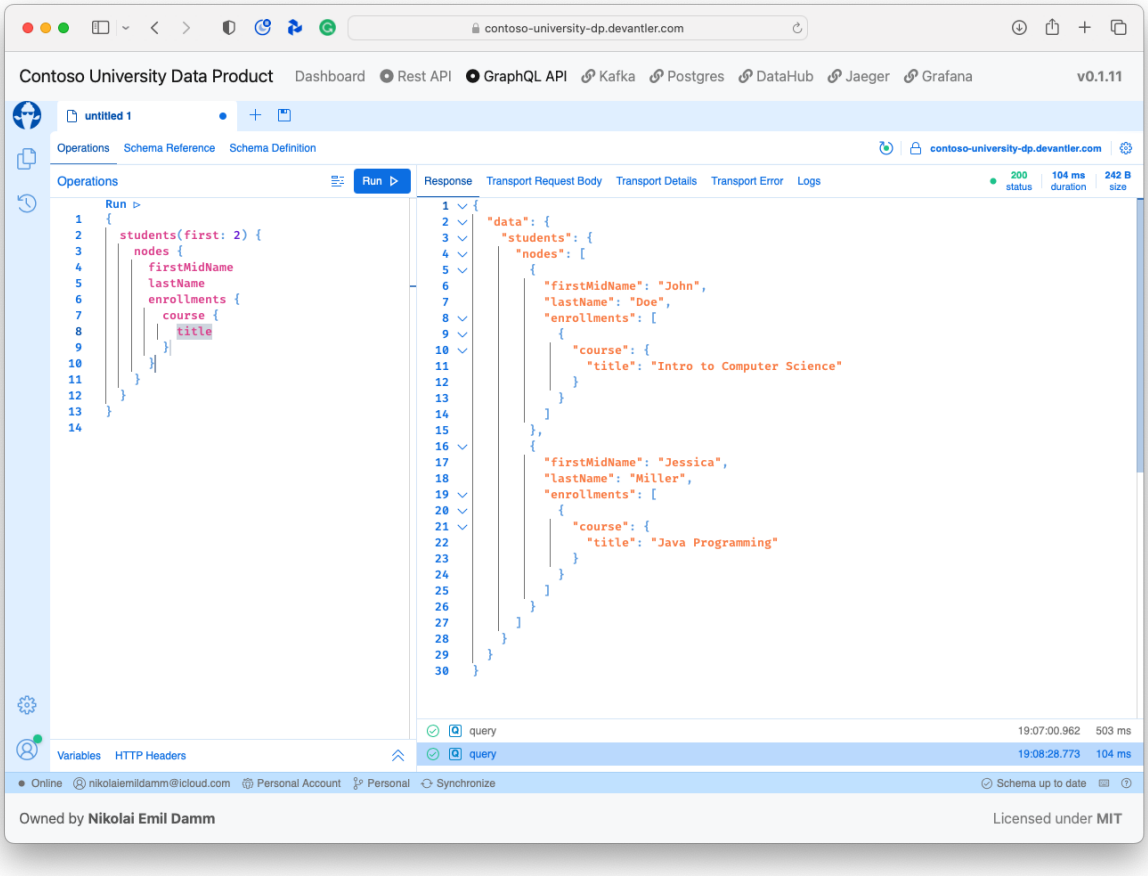


Figure 7.7: An embedded Banana Hot Chocolate Web **UI** to interact with the GraphQL server.

GraphQL supports queries, mutations, and subscriptions, where queries allow reading filtered content from the **API**, mutations allow creating, updating, and deleting content, and subscriptions allow the client to subscribe to changes in the data. As a proof of concept, the prototype only implements queries, but supporting mutations and subscriptions would be a natural next step. In lst. 7.9, an example of a generated query to read students is presented.

**Listing 7.9** An example of a generated query to read students.

```
namespace Devantler.DataMesh.DataProduct.Features.Apis.GraphQL;
public partial class Query
{
    [UsePaging]
    [UseProjection]
    [UseFiltering]
    [UseSorting]
    public async Task<IEnumerable<Student>> GetStudents([Service] IDataStoreService<int, Student>
    ↪  dataStoreService, CancellationToken cancellationToken)
        => await dataStoreService.ReadAllAsync(cancellationToken);
}
```

The different attributes added to the method enable advanced query options. Paging enables the client to

specify the number of items to return or to use cursors to query content from a specific offset. Projection enables optimal queries by allowing the client to construct queries with **Language-Integrated Query (LINQ)** expressions, which can be translated to, e.g., **Structured Query Language (SQL)** queries. Filtering enables the client to filter the content based on the schema's properties. It includes options like `where` and `select` clauses. Sorting lets the client sort the content based on the schema's properties. The advanced query options are all optional and can be omitted in queries if not needed. Likewise, the advanced options can be combined to create more advanced queries. As the code generation system generates the query classes, the different features can be enabled or disabled based on the configuration, and if disabled, the attributes are not generated.

### 7.7.3 Authentication and Authorization

Ensuring secure access to resources and protecting sensitive data is crucial for a data product. Implementing robust authentication and authorization mechanisms is essential for this purpose. However, developing these systems can be complex, requiring comprehensive solutions that can be difficult and time-consuming to implement. Unfortunately, the prototype does not include an authentication and authorization system due to time constraints. Nonetheless, this section discusses a potential approach for implementing such a system.

The initial plan was to implement authentication and authorization using the OpenID Connect [66], OAuth 2.0 [67], and WebAuthN [68] protocols. OpenID Connect extends OAuth 2.0 to provide user authentication, while OAuth 2.0 is a widely adopted authorization framework, and WebAuthn is a modern web standard for secure, passwordless authentication. Using an open-source identity provider called Authentik [69], presented in fig. 7.8, it would be possible to support the various authentication methods, enabling authentication with username and password, token-based authentication, and hardware-based authentication. Moreover, the system would allow the creation of policies to restrict access to specific data products or features, ensuring that only authorized users can access the resources.

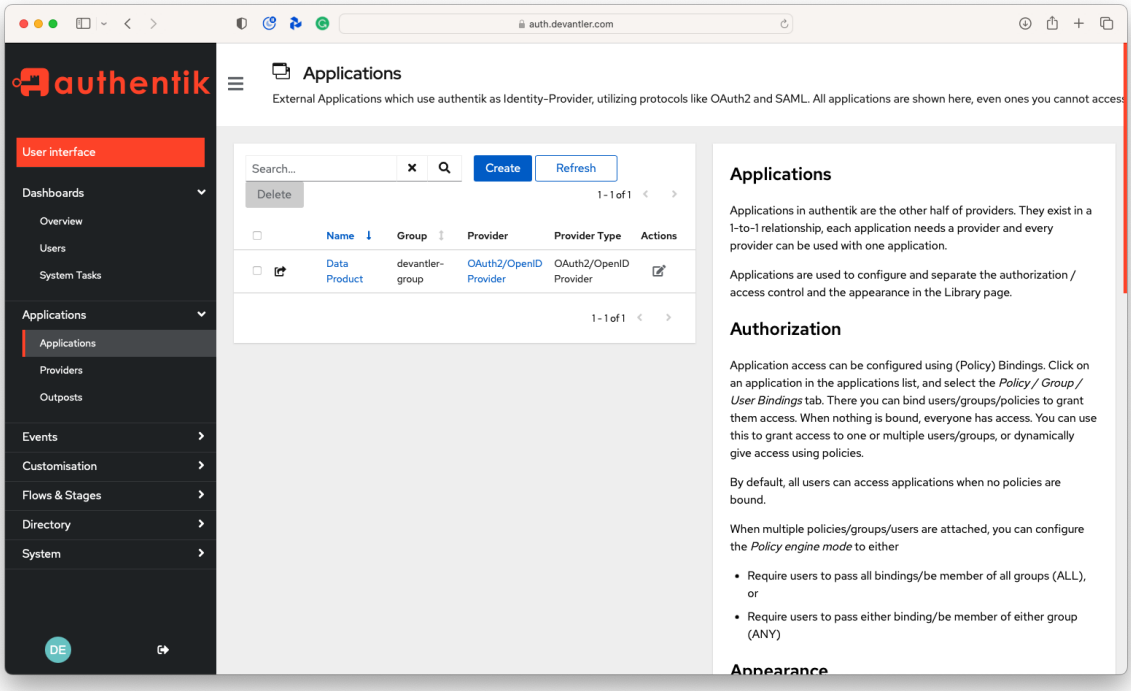


Figure 7.8: An image of Authentik setup with the OpenID/OAuth 2.0 that was planned to be used for authenticating user logins and requests for data products.

### 7.7.4 Caching

Improving performance by reducing the time required to access data is essential in the context of data products. Caching is a technique that can significantly enhance the efficiency of data retrieval. The prototype implements support for both in-memory and distributed caching.

The in-memory caching is implemented using the `Microsoft.Extensions.Caching.Memory` library [70], allowing data to be cached within the application's memory. On the other hand, distributed caching is achieved using the `StackExchange.Redis` library [71], which enables caching across multiple instances with Redis [72]. Both implementations are of type `ICacheStoreService`, making it possible to use the caching system that best fits the use case.

Generally, the caching strategy involves caching data after it has been retrieved and invalidating the cache when data is updated. However, caching bulk reads can be more complex, requiring constructing a unique cache key based on the data. In the prototype, a more straightforward approach is adopted: bulk reads are executed on the data store to retrieve all identifiers for the data, and the actual data is fetched from the cache using these identifiers. While this approach balances performance and complexity, it may require further optimization for production use to avoid requiring a database roundtrip per bulk request.

### 7.7.5 Mapping

In data products, mapping is crucial in transforming data between representations, such as models and database entities. The prototype implements the `AutoMapper` library [73] to handle mapping efficiently, simplifying mapping between different types.

The mapping system is not designed as an abstraction, making it impossible to switch, disable, or enable without changing the code; this could be considered for future iterations. However, since mapping is regarded as a core functionality of the system, the need to disable it was not identified during development.

`AutoMapper` operates by defining profiles that instruct the library on how to map between different types. As these profiles depend on the schema of the data product, they are generated by the code generation system. An example of a generated `AutoMapper` profile can be seen in lst. 7.10.

**Listing 7.10** An example of a generated AutoMapper profile.

```
public class AutoMapperProfile : Profile
{
    public AutoMapperProfile()
    {
        _ = CreateMap<Student, StudentEntity>().ReverseMap();
        _ = CreateMap<Enrollment, EnrollmentEntity>().ReverseMap();
        _ = CreateMap<Course, CourseEntity>().ReverseMap();
    }
}
```

The choice of `AutoMapper` was based on the author's previous experience with the library and its user-friendly **API**. While it may not be the fastest available library, other factors like maintainability and usability also influenced its selection. Both are areas in which `AutoMapper` excels. However, if performance becomes a concern, benchmarking other libraries could reveal better alternatives, and such a decision should be made with care, considering the trade-offs involved.

### 7.7.6 Validation

Validation is crucial for providing a good user experience and ensuring data pipelines exit gracefully and early when encountering invalid data. Although the initial plan was to implement validation using the `FluentValidation` library [74] it was postponed due to time constraints and other priorities.

Not implementing validation might cause unforeseen consequences, such as increased error rates, difficulties in debugging, and a higher likelihood of data inconsistencies. Therefore, it is a priority to implement validation in future iterations to improve the system's overall robustness and user experience.

The planned validation system would be similar to the mapping system. Instead of defining profiles, the code generation system would define and generate validators. This approach would enable the system to

validate requests based on default validation rules and rules extracted from the data product's schema, ensuring a consistent and reliable validation process.

### 7.7.7 Dashboard

A dashboard has been implemented to ease navigation and provide an overview of the data product and its capabilities. The dashboard uses Blazor [75] and Blazorise [76]. Blazor is a framework for building web **UIs** using C# instead of JavaScript, and it works through WebAssembly [77] to run native code in the browser. Blazorise is a component library for Blazor that provides a set of components that can be used to build web applications. Blazor and Blazorise allow for a consistent development experience using C# across the entire project and take advantage of Blazorise's rich set of components. The dashboard is implemented as a server-side Blazor application, meaning that the application is hosted on the server, and the client only receives the rendered **Hypertext Markup Language (HTML)**. Implementing a Blazor application as a client-side application is also possible, which would be an option if time was not a constraint.

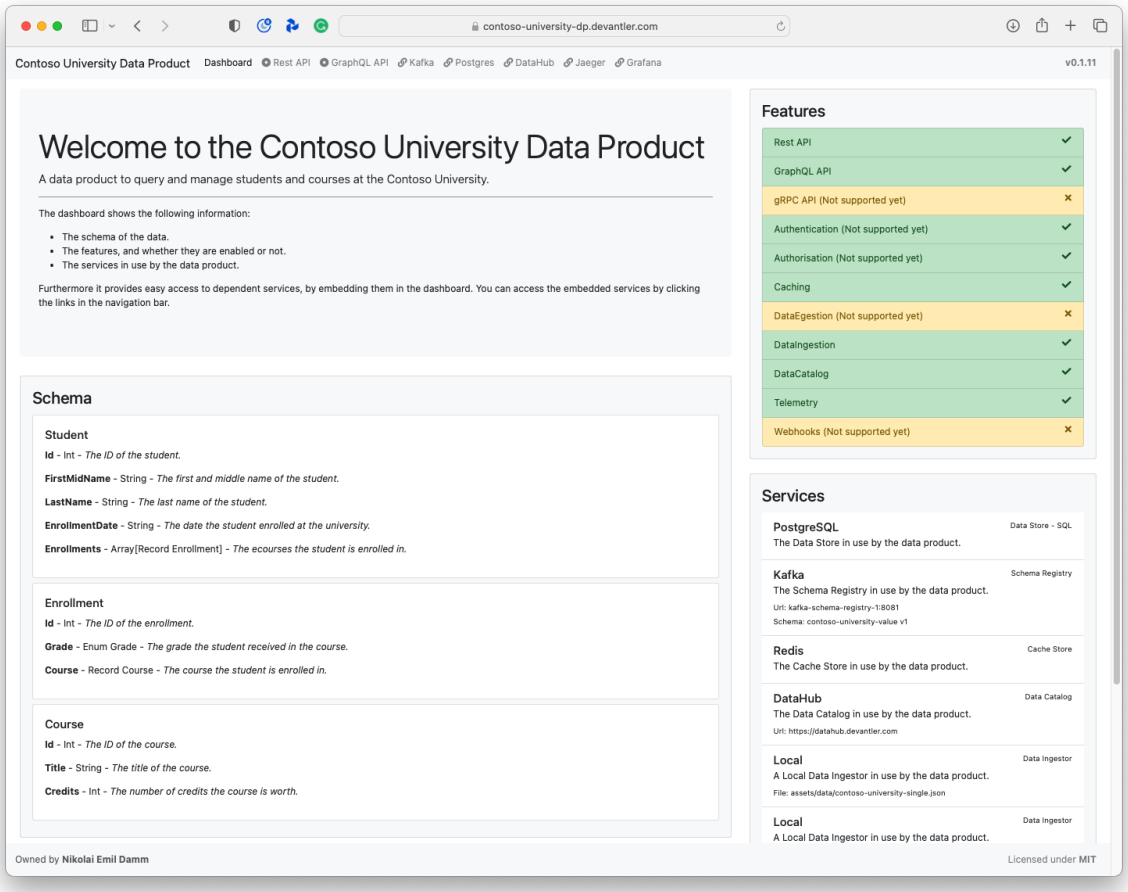


Figure 7.9: A dashboard that provides an overview of the data product.

The dashboard provides four main features:

1. **Overview of the data product** by providing a title, description, and owner.
2. **Schema overview** by providing an overview of the schema and its entities.
3. **Feature overview** by providing an overview of which features are enabled and disabled.
4. **Service overview** by providing a list of dependent services and a glance at their configurations.
5. **Embedded applications** provide a seamless experience when navigating between the data product and its external dependencies.

The first four features are self-explanatory and visible in fig. 7.9. However, the last feature requires some explanation. When configuring the data product, it is possible to specify which applications should be embedded in the dashboard. Embedding applications is done by specifying the **URL** of the application and a label used for the navigation. The dashboard then renders an iframe with the specified **URL** and allows the user to navigate the embedded application seamlessly from the dashboard. The feature was deemed valuable, as the author experienced poor user experience when the data product had many external dependencies. The user was required to navigate between many web apps to configure and work with the data product.

Embedding applications in iframes can pose security issues, and most modern web browsers have limitations for iframe usage to circumvent this. An iframe can enable clickjacking attacks[16], so the embedded applications and their outgoing requests must be trusted. A clickjacking attack could, for example, entail embedding a page and overlaying it with malicious elements so that when the user clicks the page, the attacker can redirect them. Because of security issues, iframes should only embed applications hosted on the same domain as the embedding application. Furthermore, `X-Frame-Options` should be set to disallow other websites from embedding one's page [78].

Another limitation experienced is that outgoing requests to external identity providers are most likely blocked by the provider's **Cross-Origin Resource Sharing (CORS)** policy. Therefore the best experience is achieved if the same organization owns the identity provider. Ownership of the identity provider will allow defining custom **CORS** policies to allow specific domains to authenticate with the identity provider through an iframe.

The limitations to iframes are not necessarily an issue, as most data products are expected to be hosted alongside their infrastructural dependencies. Additionally, applications can be added as external links to support other workflows, redirecting the user to the application in a new tab instead of providing a seamless experience. However, it does pose some challenges, as embedded applications are very dependent on the environment in which they are hosted. As such, embedding applications in all environments might not be possible.

### 7.7.8 Data Catalog

The data catalog feature has been designed to enable easy storage and retrieval of metadata for datasets. It has been built using C# and interfaces seamlessly with DataHub [79], an open-source metadata management platform from LinkedIn. DataHub expects a specific metadata model, and the implementation recreates this model with C# classes and interfaces to ease the creation and manipulation of metadata. Key metadata aspects include `InstitutionalMemoryAspect`, `DatasetPropertiesAspect`, and `SchemaMetadataAspect`, which represent links, datasets, and schema metadata, respectively. There are many other aspects and variations of these aspects, which can be used to create other types of metadata.

The data catalog feature is implemented as a hosted service, allowing the service to run in the background, processing metadata without impacting the application's main functionality. The data catalog service would need to be extended and injected into the request pipeline to handle more metadata use cases, such as the frequency of specific requests. This extension would involve intercepting incoming requests, extracting relevant metadata, and sending this information to DataHub for storage and analysis. Furthermore, the configuration system would need to be extended to allow data product owners to determine which metadata they want to extract and store. This control is essential as it helps prevent unnecessary processing and storage of metadata, which can become costly if managed improperly.

The data catalog feature includes several helper classes, extension methods, and a client, which help create and manipulate metadata. Notably, the `UrnHelper` class eases the creation of **Uniform Resource Names (URNs)**, DataHub's unique identifiers for metadata entities, and the client provides a simple interface for interacting with DataHub's **APIs**.

[16] Clickjacking is a malicious technique of tricking a user into clicking a button that redirects them to a different page than the one they think they are redirected to [78].

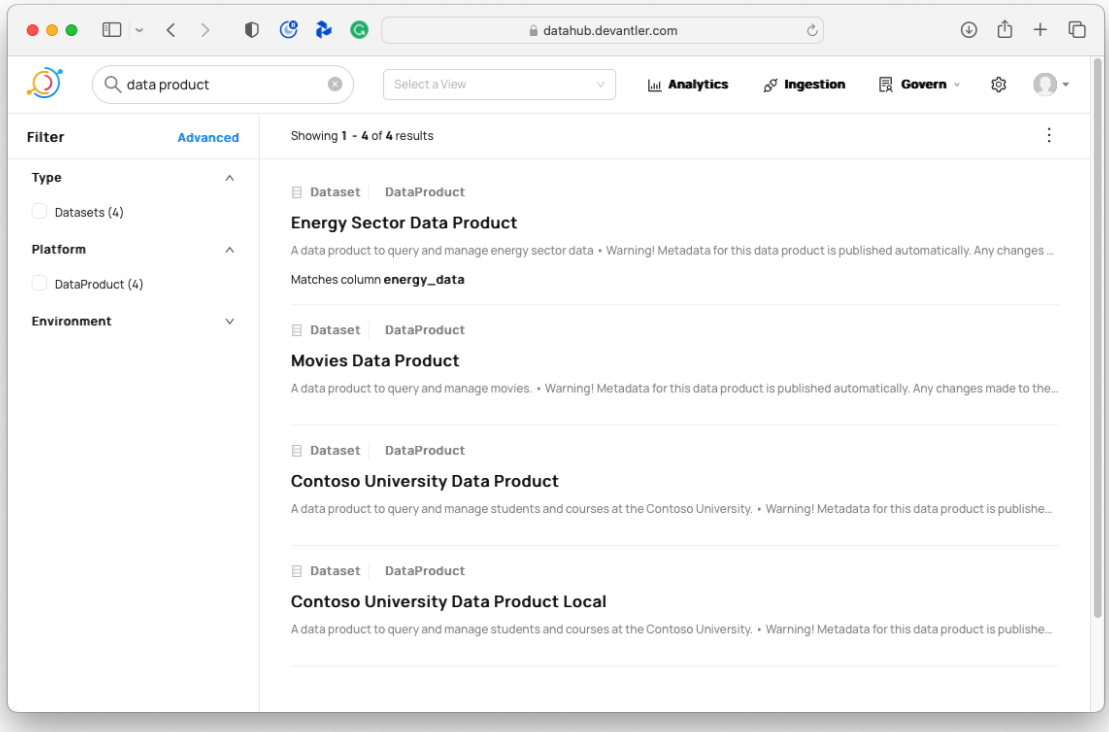


Figure 7.10: The DataHub search page showing the search results for the query “data product”.

One notable limitation of the data catalog feature is its lack of flexibility, primarily because it does not provide an abstraction for different data catalogs. The design is tailored to DataHub, which may hinder the future integration of alternative catalog systems. Without a more generalized approach or a well-defined interface, it becomes difficult to adapt the current implementation to accommodate different data catalog platforms or to switch between them as needed. A more extensible design would involve creating a higher-level abstraction to encapsulate the various data catalog functionalities, ensuring seamless and flexible integration with multiple catalog platforms while minimizing the need for extensive code modifications. However, the current landscape of data catalog platforms is not well aligned, so implementing a generalized approach is expected to be challenging.

### 7.7.9 Input/Output

The input and output system is responsible for ingesting and egesting data from and to external data sources. Of the two, only the input system has been implemented due to time constraints. The input system has two implementations, one for ingesting data from local **JSON** files and another for stream ingesting data from Kafka [80]. Both implementations are implemented as background services that can continuously ingest data from their respective sources and add it to the data store.

The local implementation works by reading files from the `DataIngestors` configuration. It deserializes all the data for each file and adds it to a list. When all files have been read, the list is filtered for distinct entities, and the entities are added to the data store with a bulk operation. This process minimizes the number of database roundtrips required to ingest data and ensures the data store is not polluted with duplicate data. The local implementation can be extended to support other file formats like **YAML** or **Comma-Separated Values (CSV)**, but currently, it only supports **JSON** files.

The Kafka implementation reads one or more configurations for `DataIngestors`. Each configuration contains a group id and topic that determines the source of the data and whether the Kafka consumer should be part of a consumer group. A background service that creates the Kafka consumers and subscribes to the specified topics is started to manage the consumers. When a message is received, the message is deserialized and written to the data store. Unlike the local implementation, the Kafka

consumer does not use bulk operations, as doing so would impose a delay on the data pipeline, as it would have to wait for messages to accumulate. The author did not find this desirable, so instead, a Kafka consumer is designed to write each received message to the data store.

Though the output system has not been implemented, it is expected to be designed similarly to the input system. Furthermore, if the functions feature were implemented, the input and output systems would likely use it to enable the user to transform and enrich the data as part of the ingestion and egestion pipeline. Functions could also be utilized to implement the input and output systems, such that configuring an ingestor as a function would use a specified function to ingest the data. This approach would be more scalable than the current implementation, as it runs the background service in a thread and thus scales with the instances of the data product. Functions can be scaled independently of the data product.

### 7.7.10 Data Store

The data store is responsible for storing a data product's data and providing an interface for accessing the data. It enables the data product to operate decentralized while governing access and usage of the data. In this context, a data store is a means of storing data, such as a database, a file system, or a message queue.

The data store feature currently supports relational databases through Entity Framework Core [81], an **Object-Relational Mapping (ORM)** that enables the implementation of a database using C# classes. The data store is implemented using the code-first approach, enabling configuring and controlling the database from code. Entity Framework also allows for a semi-database-agnostic approach, where the framework can generate the **SQL** and remove slight deviations between databases. However, this abstraction is only supported for a limited number of relational databases.

Entity Framework also supports migrations to update the database schema. Although the prototype does not currently support migrations, auto migrations are planned. Traditionally, migrations require creating migration files for rolling changes back and forth. However, Entity Framework can auto-generate migrations with third-party libraries to keep the database in sync with entity classes. Although this approach can be dangerous and potentially lead to data loss, the data product's use of Kafka schemas imposes limitations on schema changes, ensuring either backward or forward compatibility. The limitations restrict the evolution of the schema and will not allow breaking changes. As such, it can be safe to enable auto migrations to make data products self-sufficient and less dependent on their data store.

Data stores are implemented using the repository pattern, with each entity having a repository that provides an interface for accessing the entity. The repositories are implemented using the `IRepository` interface, which provides a generic interface for accessing entities. This abstraction enables data access logic to vary between repositories, allowing for implementing different **ORMs** for different data stores and optimizing each implementation. For example, one repository could interact with Entity Framework, while another could interact with a MongoDB client. This flexibility is beneficial, as some data structures, such as time series, are not well suited for relational databases.

### 7.7.11 Schema Registry

In the context of the data product, a schema registry is an external service that provides a schema for a given data source. The data product's prototype relies on knowing the data's schema to operate effectively. The prototype has been implemented with local and Kafka schema registries, supporting Avro [82] and schema versioning. Versioning is vital for data products, enabling controlled migration to new schemas.

The local implementation reads an Avro schema from a folder on the host machine, while the Kafka implementation retrieves the schema from Kafka's schema registry. Both implementations parse the Avro schema into an abstract model, which is then used to generate C# classes and methods as described in sec. 7.5. Kafka consumers and producers also use the schemas to serialize and deserialize data when reading and writing to Kafka.

The data product has been designed considering other schema registries as potential options. One such possibility is using introspection to read and generate a schema from a Postgres database. While not as flexible as a full-fledged schema registry, this approach would demonstrate the adaptability of the data

product in supporting various schema sources. However, the author has not implemented this feature due to time constraints.

### 7.7.12 Telemetry

The telemetry feature plays a crucial role in the data product by providing observability through tracing, logs, and metrics. The telemetry feature leverages cutting-edge technologies, including Jaeger [83], OpenTelemetry [84], OpenTelemetry Collector [85], Prometheus [86], and Grafana [87], and uses these technologies to gather, visualize and manage telemetry data.

Instrumentation libraries [88] are employed to automatically create telemetry for use cases that support it, making it straightforward to collect telemetry from a data product. The telemetry feature can gather vital information about a data product's performance and resource usage by integrating with these libraries. OpenTelemetry is advancing quickly with instrumentation and provides a wide range of preview libraries to enable instrumentation for many use cases, for example, **HTTP**, GraphQL, and Entity Framework Core.

Tracing and logs offer observability for requests, making it easier to understand data flow within the system and identify potential bottlenecks or issues. With the help of Jaeger and OpenTelemetry, traces and logs are collected and visualized, allowing insights into the data product's inner workings and enabling users to make informed decisions about performance improvements. In fig. 7.11, a trace shows the data flow through the data product, demonstrating how a **REST** bulk operation requires a single database round trip to get **Identifiers (IDs)** before retrieving the data from the cache.

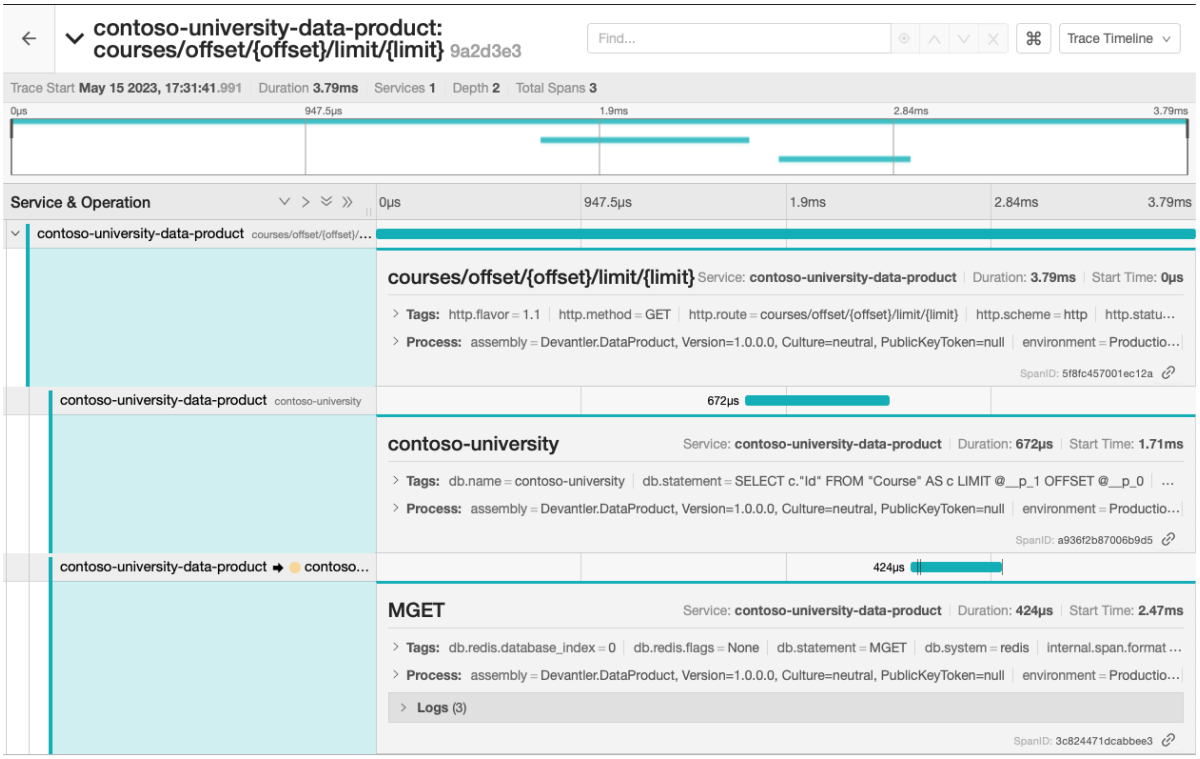


Figure 7.11: A trace that shows the flow of data through the data product.

Metrics provide quantitative data on a data product's performance and resource usage, which can be invaluable in assessing its overall efficiency and effectiveness. Prometheus, a robust monitoring and alerting toolkit, is used to gather and store these metrics. Grafana, a popular open-source analytics platform, is then employed to visualize the collected metrics, enabling a clear understanding of the data product's performance over time. In fig. 7.12, a Grafana dashboard displays the data product's **Central Processing Unit (CPU)**, memory, disk, and network usage.

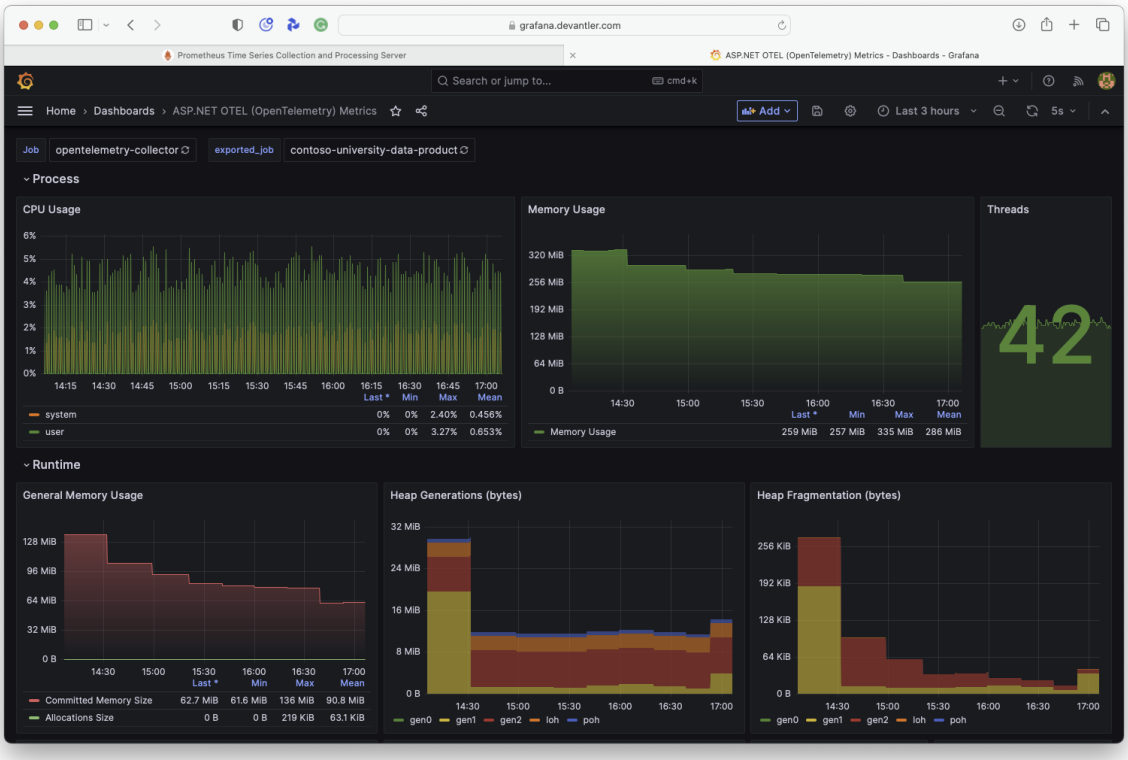


Figure 7.12: A Grafana dashboard displaying the data product's **CPU**, memory, disk, and network usage.

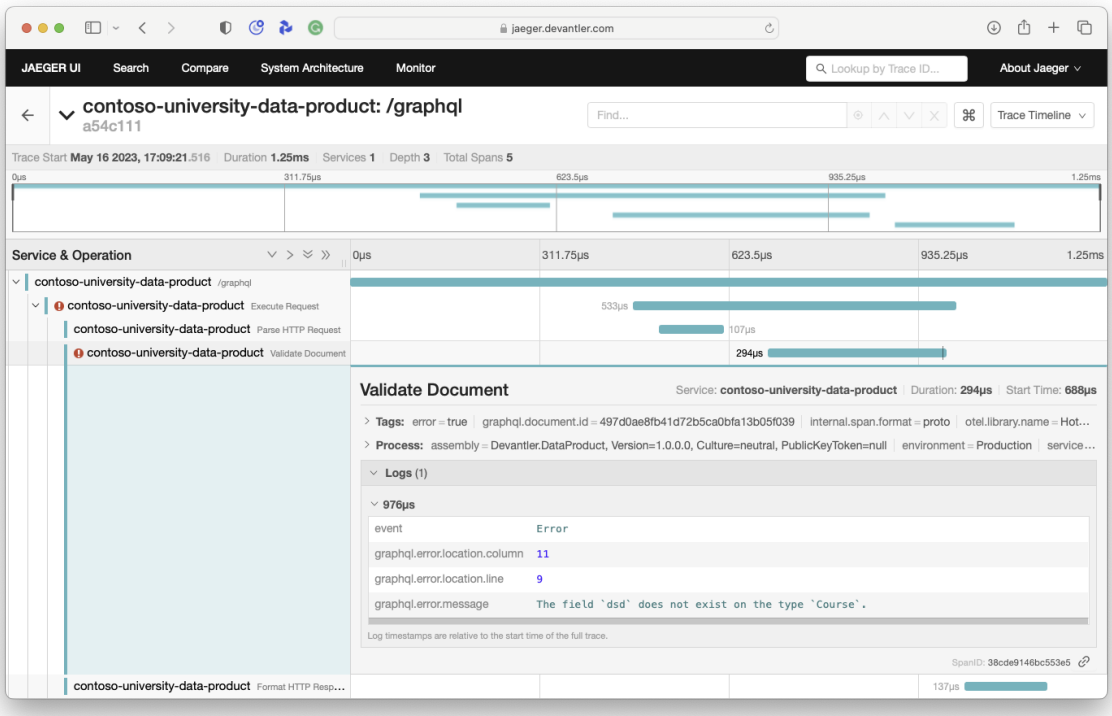


Figure 7.13: A trace showing an error occurring in the data product.

Errors encountered within the data product are displayed in traces, significantly simplifying debugging. By identifying and analyzing errors within the data flow context, developers can quickly pinpoint the root cause of the issue and implement targeted solutions to ensure the data product's continued reliability and efficiency. For example, in fig. 7.13, a trace shows a GraphQL validation error in the data product, as it cannot find a field named `dsd` on the existing type `Course`.

As demonstrated, the telemetry features are vital to monitoring and understanding what happens within or between data products. The telemetry feature is also essential in optimizing and debugging the data product by providing insights into its performance and behavior.

## 7.8 Infrastructural Dependencies

The current iteration of the prototype has a few infrastructural dependencies. The dependencies differ according to the data product configuration. Assuming a complex prototype configuration results in the following service and infrastructure dependencies:

- **A Containerization Environment.** As the prototype is packaged as a Docker container, it can be deployed to any significant containerization environment like Kubernetes [89], Nomad [90], Docker [91], or even cloud providers like AWS [92], Azure [93], and Google Cloud [94].
- **Internal and Public Networking Capabilities.** The prototype relies on being able to communicate with other services and the outside world, so it requires networking capabilities that the containerization environment or cloud provider can provide.
- **Local vs. online requirements.** Depending on the configuration, the prototype might require internet access to external services not hosted in its local environment. The prototype can be configured to use local services instead of external services. However, it will require the availability of local services, which might not be possible in all edge environments. In these cases, one can consider a more straightforward setup where some less critical features are disabled.
- **PostgreSQL.** The prototype relies on PostgreSQL to store data. The prototype can also be configured to use other **SQL** databases, but they have not been tested.
- **Kafka.** In most scenarios, the prototype will require a Kafka cluster. The prototype relies on Kafka to store data and schemas and provide streaming capabilities.
- **LinkedIn DataHub.** The prototype relies on LinkedIn DataHub to store data product metadata and provide a **UI** for the data catalog.
- **Authentik.** If authentication and authorization were implemented, the prototype would rely on Authentik to authenticate and authorize users.
- **Redis.** The prototype relies on Redis to cache data.
- **Jaeger.** The prototype relies on Jaeger to visualize traces and logs.
- **Grafana.** The prototype relies on Grafana to visualize metrics data.
- **Prometheus.** The prototype relies on Prometheus to collect and store metrics data.
- **OpenTelemetry Collector.** The prototype relies on OpenTelemetry Collector to collect, export, and expose telemetry data to Jaeger and Prometheus.
- **Elasticsearch.** Jaeger and LinkedIn DataHub rely on Elasticsearch to store telemetry and metadata.

The prototype has no provisioning service, so the infrastructural dependencies must be provisioned manually. The author has relied on a Docker environment to host the infrastructural dependencies, but hosting them in other containerization environments or cloud providers should be possible. A production-ready system is also expected to include a self-serve data platform, as described in sec. 2.2.1, to enable users to provision and manage data products without involving an **IT** department.

# 8 | Evaluation

This chapter evaluates the research findings and the prototype developed. The evaluation encompasses multiple aspects, such as assessing the theoretical validity of the research by examining its alignment with the **DSI** and **DMA**, as well as evaluating the practical validity of the prototype through an acceptance test to determine whether it meets the functional requirements and is applicable in real-world scenarios.

## 8.1 Theoretical Validity of the Study

In this section, the theories (T1 to T17 in Appendix N.1) derived from the **GT** method are evaluated on how well they align with the **DSI** and **DMA**. This evaluation aims to determine the theoretical validity of the research by comparing its findings with the theoretical **SOTA** of the **DSI** and **DMA**.

Theories that exhibit alignment with both approaches are considered valid (✓), while theories that do not align with either approach are considered invalid (✗). Theories that align with only one approach are considered partially valid (✓). Lastly, some theories are irrelevant to the evaluation, as they touch upon aspects that are not directly related to the **DSI** or **DMA**. These theories are considered to be neutral (✗).

**T1 The Business Ecosystem** ✓ The theory on the business ecosystem emphasizes the need to improve sector coupling and information availability in the energy sector such that collaboration can be enhanced. It also touches upon some of the critical challenges that slow down the development of the energy sector, such as the lack of harmonized roles and differing levels of digitalization. These properties align somewhat with the **DSI**, which, in many ways, is about improving collaboration and information availability by breaking down the barriers that limit collaboration between sectors and thus liberating information. However, it does not align with the **DMA** as it does not directly address any of the critical aspects of the **DMA**.

**T2 The Role of Centralization and Decentralization** ✓ The theory on the role of centralization and decentralization emphasizes the need to balance centralization and decentralization, as some aspects of the data infrastructure must be centralized to enable standardization, trust, and collaboration. In contrast, others must be decentralized to enable performance and a decentralized communication model. This theory aligns well with the **DSI** and **DMA** as both approaches emphasize the importance of decentralization while acknowledging the need for centralization in some aspects of the data infrastructure. For example, a data mesh is a decentralized infrastructure with a mesh of data products, but it also requires a self-serve data platform that is a centralized component.

**T3 The Importance of Collaboration** ✓ The theory on the importance of collaboration emphasizes modernizing collaboration processes to enable decentralized communication, like transaction-based communication. Furthermore, it mentions that data silos should be broken down, and data sharing should be encouraged. The theory aligns well with the **DSI** and **DMA** as both approaches emphasize the importance of collaboration and provide a decentralized communication model with modern collaboration processes that enable data sharing and aim to break down data silos.

**T4 Data Management in an Evolving Ecosystem** ✓ The theory on data management in an evolving ecosystem emphasizes how data management requires addressing data quality, storage, ingestion, egestion, and integrations. Furthermore, it stresses the significance of domain-specific data and the need for a common language for data representation. It also emphasizes aspects of governance with retention policies. All these aspects are vital aspects of the **DSI** and **DMA**, and thus the theory aligns well with both approaches as it is a prerequisite for establishing better communication and understanding within and across domains, fostering seamless data exchange.

**T5 The Perception of a Data Mesh** ✓ The theory on the perception of a data mesh emphasizes many critical aspects, such as its decentralized nature and focus on discoverability and observability. It also mentions data products and how they must expose **APIs** and provide metadata to make them discoverable in a data catalog. Overall the theory represents a good understanding of the **DMA**, and albeit it does not mention the specific design principles of a data mesh, it does cover the critical aspects of a data mesh. As such, the theory aligns well with the **DMA**. It also aligns with the **DSI**, as both approaches are much alike, and although this theory is specifically about data meshes, it does not contradict the **DSI** in any way.

**T6 The Perception of Data Spaces** ✓ The theory on the perception of data spaces describes data spaces as a journey that must start with simple data discovery platforms that later can become capable decentralized systems that enable efficient data exchange. It emphasizes many vital aspects of data spaces, like governance, security, compliance, and observability. As such, the theory aligns well with the **DSI**, and as **T5**, it does not contradict the **DMA** and thus aligns with it as well.

**T7 The Role of Domain Modelling** ✓ The theory on the role of domain modeling emphasizes the importance of domain modeling for data management and how it serves as a foundation for data exchange with a common language. It also mentions that domain models should be split according to their sub-domains, making data management more approachable. It is assumed that domain modeling is a vital aspect of the **DSI** and **DMA**, as it is a prerequisite for establishing a common language for data representation and thus enabling seamless data exchange. As such, the theory aligns well with both approaches.

**T8 The Complexities of Flexibility and Grid Balance** ✗ The theory on the complexities of flexibility and grid balance presents some challenges and possible solutions in balancing the grid. It provides insight into some of the challenges in the energy sector and how solutions like power-to-x and sector coupling can address them. However, it does not mention much that directly aligns with the **DSI** or **DMA**, and thus it is hard to justify that it is relevant other than it touches upon some of the capabilities that the **DSI** aims to provide.

**T9 The Perception of Governance** ✓ The theory on the perception of governance emphasizes how the energy sector requires balancing confidentiality, transparency, and information management, as the sector needs transparency to know what is going on but also needs confidentiality to ensure that sensitive information is protected. This theory touches upon some aspects that make the energy sector unique and how it requires tailored solutions. However, it also touches upon elements of governance like residency, access control, and data cleansing. As such, it aligns well with the **DSI** and **DMA** as both approaches emphasize the importance of governance and how it must be part of the foundation of the data infrastructure to enable key aspects like data sovereignty, data ownership and essentially trust.

**T10 The Digital Infrastructure** ✗ The theory of digital infrastructure talks about the need for convergence, orchestration, and adaptability. Neither of these aspects is directly mentioned in the **DSI** or **DMA**; however, they are related. Nevertheless, as it is not directly mentioned in either approach, this theory does not align.

**T11 The Physical Infrastructure** ✗ The theory on the physical infrastructure mentions the need for decentralization and modernized communication technologies. However, it is mainly about the physical infrastructure and how the developing digital infrastructure underpins it; as such, it is not relevant to the **DSI** or **DMA**.

**T12 Legislation and Regulation** ✓ The theory on legislation and regulation touches upon some interesting aspects of technology innovation, as it mentions the importance of innovating before regulating and how regulation can be a barrier to innovation. It also mentions the importance of standardization and how it is critical in pushing innovation forward. Regulation and standardization are vital aspects of the **DSI** as it aims to create an infrastructure legally governed by standards. However, the data mesh is more freely formed and does not have the same focus on legal regulation. The theory aligns well with the **DSI** but not the **DMA**.

**T13 Metadata and its Future Role** ✓ The theory on metadata and its future role describe its purpose as providing context for data and describing the actual data while emphasizing how metadata is vital to provide discoverability and trust in data. Metadata is a vital aspect of the **DSI** and **DMA**, as it is a prerequisite for enabling discoverability in decentralized data environments. As such, the theory aligns well with both approaches.

**T14 The Balance Between Open-Source and Proprietary Software** ✗ The theory on the balance between open-source and proprietary software discusses that the balance may vary depending on the specific requirements of a project or organization. Carefully evaluating the context and goals is necessary to determine the most appropriate balance. However, this discussion is not directly related to the **DSI** or **DMA** other than it being an important decision to make when designing implementations of either approach.

**T15 The Roles and Actors** ✗ The theory on roles and actors concerns the different roles operating in the energy sector. As such, it is irrelevant to the **DSI** and **DMA**.

**T16 Prioritized Software Qualities** ✗ The theory prioritized software qualities mentions performance, functionality vs. usability, stability, robustness, security, and integrity. Neither of these qualities is unique to the **DSI** or **DMA**, although they are just as crucial in data spaces and data meshes as other systems. As such, the theory is evaluated as invalid, as it does not contradict either approach but does not provide any new insight.

**T17 Business Users and Data Scientists** ✗ The theory on business users and data scientists is much like the **T16** as it is unique to the **DSI** and **DMA** but is just as crucial in data spaces and data meshes as other systems. As such, this theory is also evaluated as invalid, as it provides no new insight into users of data spaces and data meshes.

The results show that eight theories align with the **DSI** and **DMA**, while three do not. Two are partially satisfied and align with the **DSI** but not the **DMA**. Lastly, four are considered irrelevant to the **DSI** and **DMA**. From this result, it is clear that the research has amounted to relevant knowledge on not only what the **DSI** and **DMA** are but also how developing following these approaches is beneficial to the energy sector. The information on both the **DSI** and **DMA** align well with the overall idea and goals of the approaches, but without synthesizing the information with the theoretical **SOTA**, it would not be easy to obtain a complete understanding of what these approaches entail, and how they can be implemented. As such, the research has provided an excellent foundation for developing a prototype that implements a data space as a data mesh with all the needs and challenges of the energy sector in mind.

However, without the synthesis of the theoretical **SOTA**, it would be difficult to argue that the prototype is an exemplary implementation of the **DSI** or **DMA**, as not synthesizing the research could easily have led the prototype in a different direction that aims to solve specific problems in the energy sector, but in doing so risks contradicting critical aspects of the **DSI** and **DMA**. For instance, the prototype could have been designed as a data lake, which would have been an excellent approach to solving many of the problems in the energy sector, but it would no longer be a valid implementation of the **DMA**. As such, it has been valuable to validate the research with the theoretical **SOTA** and let it be a part of the definition of the prototype.

## 8.2 Practical Validity of the Prototype

The practical validity of the prototype will be tested with an acceptance test. The prototype will be evaluated against the functional and non-functional requirements. As the functional requirements are derived from the non-functional requirements, whether the prototype satisfies them will also determine to what extent the non-functional requirements are satisfied. As such, the section list functional requirements and evaluate whether the prototype satisfies them. Satisfied requirements will be marked with a green checkmark (✓), requirements that are partially satisfied will be marked with a yellow checkmark (✓), and requirements that are not satisfied will be marked with a red cross (✗).

**FR1 Schema Management** ✓ The prototype satisfies the schema management requirement as it can use local and Kafka schema registries to manage and interact with schemas. By supporting multiple schema registries, the system provides flexibility in storing and retrieving schemas, catering to different use cases and preferences. This approach ensures the data product can work with various data sources and adapt to changing schema requirements.

**FR2 Dynamic Domain Models** ✓ The prototype satisfies the dynamic domain models requirement through flexible code generation, facilitating class and method construction at compile time. This process adjusts to diverse data structures per configuration and schema, increasing adaptability. Generating C# classes and methods from the Avro schema's abstract model can handle changing data requirements and various data structures, enhancing its versatility.

**FR3 Data Discovery and Search** ✓ The prototype satisfies the data discovery and search requirement by integrating with LinkedIn DataHub to provide a data catalog where data products can be discovered and understood. This integration enables users to explore the available data products, understand their schemas, and gain insights into their usage and relationships with other data products. By offering a comprehensive and searchable catalog, the data product fosters better collaboration and data-driven decision-making among teams and organizations, promoting more efficient use of the available data assets.

**FR4 Data Transformation** ✗ The current prototype does not satisfy the data transformation requirement. The data product must integrate with serverless functions or a distributed framework like Apache Spark to achieve this functionality. Such integration would enable data transformation and enrichment at appropriate stages in the data product's pipeline, making it more versatile and capable of handling various data processing tasks. Data transformation would be a valuable addition to the data product, allowing for more advanced data processing and analysis capabilities.

**FR5 Data Cleansing** ✗ The prototype does not satisfy the data cleansing requirement, as it lacks built-in support for many governance features. To fully satisfy this requirement, the data product must incorporate more extensive data cleansing capabilities by integrating with external tools or extending the data processing capabilities to listen for or be triggered by events that require data cleansing. This functionality would allow data owners to enforce whether their data should be accessible.

**FR6 Data Ingestion** ✓ The prototype satisfies the data ingestion requirement, as it can ingest data from local files and Kafka streams. However, for it to be truly usable and versatile, it will require integrations with serverless functions to allow transforming or enriching data as it is ingested. Serverless functions will enable the data product to process and adapt incoming data according to the system's and its users' specific needs, making it more powerful and flexible.

**FR7 Data Egestion** ✗ The prototype does not satisfy the data egestion requirement, as it currently lacks support for egesting data to external systems or destinations. This functionality is crucial for sharing and exporting data, enabling collaboration and integration with other services.

**FR8 Integrate with CNCF tools and technologies** ✓ The prototype satisfies the requirement of being built on proven technologies by relying on **CNCF** technologies and tools. The tools and technologies provided by **CNCF** are widely used, well-maintained, and have a thriving community behind them. In the prototype, the following **CNCF** tools and technologies are used:

- **Kafka**: A distributed streaming platform for building real-time data pipelines and streaming applications.
- **Redis**: A distributed cache used for caching and storing data.
- **Jaeger**: A distributed tracing system for monitoring and troubleshooting services in distributed systems.
- **OpenTelemetry**: A set of **API** libraries, agents, and instrumentation to provide application observability.
- **Prometheus**: A monitoring and alerting toolkit designed for reliability and scalability.
- **Grafana**: A platform for visualizing and analyzing metrics from various data sources.
- **Elasticsearch**: A distributed **REST**ful search and analytics engine.

Using **CNCF** tools over other alternatives ensure the prototype benefits from these well-established technologies' ongoing development, community support, and best practices. Besides using **CNCF**

tools, the data product also uses a few popular open-source tools when no **CNCF** alternative exists. The open-source tools and technologies used by the prototype are:

- **LinkedIn DataHub**: A data catalog enabling metadata discovery, search, and governance.
- **Authentik**: An open-source identity and access management solution for authentication and authorization.

**FR9 Modern APIs** ✓ The prototype partially satisfies the modern **APIs** requirement as it supports a **REST API** and GraphQL **API**, but the GraphQL **API** can still be extended with mutations and subscriptions. Likewise, a data product should have **APIs** for retrieving metadata about the data product, but this is not yet implemented primarily because such an approach does not align with how data cataloging is done in LinkedIn DataHub. Nonetheless, the **APIs** enables the data product to interact with a wide range of clients and external systems, ensuring that it can be easily integrated with other services and applications. This approach promotes interoperability and makes the data product more accessible to different users and systems.

**FR10 Modularity and Extensibility** ✓ This requirement is satisfied through the feature pattern, which allows the data product to be organized into modular components that can be developed and maintained independently. By building abstractions for most parts of the data product, the architecture promotes the separation of concerns, making it easier to understand, modify, and extend. The data product is designed to be extensible, as most features contain an abstraction for which new implementations can be developed. This design approach enables seamless integration of new features, enhancements, or updates to existing features, ensuring that the data product can evolve to meet future needs.

**FR11 Telemetry** ✓ The prototype satisfies the telemetry requirement by using a combination of instrumentation, OpenTelemetry, Jaeger, Grafana, and Prometheus to gather, visualize, and manage telemetry data, including logs, metrics, and traces. Implementing these tools and technologies enables the data product to provide a rich set of observability features, allowing users to monitor the performance and health of the system, identify potential issues, and optimize resource usage. This comprehensive approach to telemetry ensures that users have the necessary information and insights to understand the behavior of the data product, maintain its reliability, and improve its performance over time.

**FR12 Caching** ✓ The prototype successfully satisfies the caching requirement, employing both in-memory and distributed caching. Caching improves the system's overall performance by reducing the need for repeated data retrieval or computation. In-memory caching is useful for quickly accessing frequently used data. In contrast, distributed caching enables data sharing across multiple application instances, ensuring consistency and reducing the need for additional data requests.

**FR13 Real-time Data Processing and Querying** ✗ The current prototype does not satisfy the requirements of real-time data processing and querying. Implementing distributed computing with technologies like Apache Spark or incorporating user-defined functions could alleviate this limitation. These solutions can be used for distributed computing enabling real-time data processing and querying capabilities, ensuring that the data product can efficiently handle large-scale data analytics. However, these features might better suit a self-serve data platform (as explained in sec. 2.2.1), enabling users to perform data processing and queries across multiple data products.

**FR14 Validation** ✗ The prototype does not satisfy the validation requirements, as it currently does not support validating data in its data pipeline.

**FR15 Microservice Architecture** ✗ The current prototype does not satisfy the microservice architecture requirement, as it employs a monolithic approach. Although specific features and components are modular, the system's structure remains tightly coupled. The prototype must be restructured to achieve a microservice architecture, separating its components into independent, loosely coupled services that can be deployed, scaled, and maintained independently.

**FR16 Data Storage** ✓ The prototype partially satisfies the data storage requirement by implementing support for relational databases and providing abstractions allowing further iterations to support other databases. Entity Framework Core and the repository pattern allow for a flexible and extensible data storage solution, catering to different storage needs and facilitating future integration with alternative database technologies, such as NoSQL or graph databases. However, as the prototype

currently does not support databases other than relational databases, the requirement is deemed partially satisfied. Especially since the flexibility and extensibility of the data storage solution are not yet proven without this support.

**FR17 Authentication and Authorization** ✗ The prototype does not satisfy the authentication and authorization requirement, as it currently lacks support for securing access to its resources and functionality. The data product must implement a robust security mechanism, such as integrating with an identity provider or OAuth2/OpenID Connect, to provide secure authentication and role-based authorization. Role-based authorization can ensure that only authorized users and systems can access and manipulate the data, thus protecting the integrity and privacy of the information.

**FR18 Dashboard** ✓ The prototype satisfies the dashboard requirement by providing a dashboard that enables users to seamlessly navigate between the data product's functionality and external services. The dashboard incorporates embedded applications, allowing users to access and manage various data product services without switching between tools or **UIs**. This design approach contributes to a more cohesive and user-friendly experience, allowing users to perform their tasks efficiently.

**FR19 Data Visualization and Analytics** ✓ The prototype partially satisfies the data visualization and analytics requirement through Jaeger and Grafana integrations, offering observability for tracing, logs, and metrics to aid performance optimization. Current limitations necessitate broader service integration or specific data visualization feature implementations, such as DataHub's support for lineage and data profiling, to meet the requirement entirely.

The evaluation of the prototype against the functional requirements reveals that, while it does satisfy many of the criteria, there is still room for improvement before it becomes a viable solution within the Danish energy sector, specifically for Energinet. This summary highlights the areas where the prototype excels and those that require further development to function effectively in a production environment, all while considering the unique needs of Energinet and the energy sector.

The study emphasizes the need for a data infrastructure capable of handling the growing volume and complexity of data within the Danish energy sector. Energinet requires a solution to manage, process, and provide insights into the data to support decision-making, promote collaboration, and facilitate the transition to a more sustainable and efficient energy system. In this context, the prototype demonstrates potential in several areas, specifically the areas covered by the 12 functional requirements it satisfies or partially satisfies.

However, to effectively address the problem statement and cater to the unique requirements of Energinet and the Danish energy sector, the prototype needs to improve in several critical areas. For instance, data egestion is essential for sharing and exporting data to external systems or destinations, enabling collaboration with other stakeholders in the energy sector. The lack of real-time data processing and querying capabilities also hinders the system's efficiency in handling large-scale data analytics. Integrating distributed compute technologies, such as Apache Spark, or incorporating user-defined functions, could address this limitation and enhance the system's overall performance.

Moreover, the prototype must ensure compliance with the energy sector's stringent security and data protection regulations. It entails implementing robust security mechanisms for authentication and authorization, for example, integrating with an identity provider through OAuth2 or OpenID Connect to protect the integrity and privacy of the data.

Lastly, the prototype should prioritize data transformation capabilities for more advanced data processing and analysis. Integrating with serverless functions or a distributed framework like Apache Spark can help transform and enrich data at different stages of the data product's pipeline, making it more versatile and capable of handling various data processing tasks.

By focusing on these improvements, the prototype can become a practical and viable solution that effectively addresses the problem statement—building a data product that can become the central component in a data space, enabling decentralized decision-making and communication to improve collaboration in the Danish energy sector.

# 9 | Conclusion

This study sought to determine whether a data space can be designed as a data mesh to improve collaboration between actors in the Danish energy sector. Using constructivism and the **CRA** (sec. 3.1) as the research methodology and conducting fieldwork guided by the **GT** method (sec. 3.2), the study involved five interviews with experts at Energinet (ch. 4) aimed at answering essential research questions (sec. 1.1.2).

The result of the **GT** method was 17 theories (Appendix N.1) and 66 hypotheses (Appendix N.2) about the Danish energy sector, its needs, and challenges, as well as different perceptions of what a data space and a data mesh are. The background and the theoretical **SOTA** of the **DSI** and the **DMA** (chapter 2) were researched to ensure the theoretical validity of the study (sec. 8.1). Determining the theoretical validity entailed synthesizing the theories from fieldwork with the theoretical **SOTA**, which was done as part of the evaluation (ch. 8). The research and the elicited theories contributed to answering the research questions: (1) "What is a data space?" and (2) "What is a data mesh?", as presented in sec. 2.1.1, and sec. 2.2.1, respectively. The third research question (3) "What are the challenges with collaboration in the Danish energy sector?", was answered primarily by the resulting theories from the **GT** method, as presented in Appendix N.1. The hypotheses were utilized as the foundation for eliciting non-functional and functional requirements, as presented in ch. 5, which helped ensure the prototype's practical validity (sec. 8.2).

The study results demonstrated high relevance to the theoretical **SOTA** of the **DSI** and the **DMA**, and ensured the prototype's alignment with these concepts. Moreover, it was apparent that the Danish energy sector requires decentralized solutions for improving collaboration and progressing sustainability goals. Improving collaboration entailed standardizing the means for communication between actors and sectors as it can allow decentralized decision-making and improve the efficiency at which different actors can cooperate towards shared goals. The challenges of collaboration were revealed to be a result of the sector's complexity, as it is composed of many different actors with different needs and capabilities, and thus interoperability, discoverability, observability, and decentralized decision-making are essential to enable collaboration. The level of digitalization proved to be a significant barrier to progress, as many actors in the energy sector cannot participate in complex data infrastructures. As such, it was also emphasized that a solution should be designed to be inclusive and that it should be able to integrate with existing infrastructure to allow for a gradual transition towards a more decentralized data infrastructure.

The prototype set out to demonstrate a proof of concept for a solution that can help push the barriers aside and demonstrate how many of the Danish energy sector's needs can be solved by combining the data space with the data mesh. The prototype demonstrates how a data product can be flexible with a comprehensive configuration system and an adaptable code generation system. Furthermore, the prototype demonstrates how different capabilities of the data mesh can be implemented to enable a data product to function in many different domains. These capabilities include but are not limited to **APIs**, authentication and authorization, functions, input and output, metadata, and telemetry, some of which the prototype failed to demonstrate. The capabilities allow the data product to integrate with centralized services such as data catalogs to provide discoverability into a sector's data products and **CNCF** tools like Jaeger and Grafana to observe what is happening within or between data products. Centralized services were proven to be essential to enable data products' full potential, as they require these services to enable them to be discovered, managed, observed, and governed.

In conclusion, the study proved theoretically and practically valid and thus successfully proves that a data space can be designed as a data mesh to improve collaboration between actors in the Danish energy sector. Furthermore, doing so enables the separation of domains within sectors and provides discoverability, observability, and governance. However, building a production-grade data space as a data mesh is significant, and the prototype currently lacks many required features. Furthermore, building such a system will require developing supporting platforms or services to enable features like provisioning, federated computational governance, and identity management. As such, this study only scratches the surface of what is needed to build a data space as a data mesh, but it does provide a proof of concept for the central component of a data mesh, the data product, and how it can be utilized in realizing a data mesh and a data space.

## 9.1 Contributions

Overall the project has contributed to the ongoing discussion on future-proof data spaces and provides insights into implementing a data space as a data mesh to achieve that goal. Besides that, the project has resulted in quite a few contributions, as it provides a proof of concept for the central component of a data mesh and comprehensive research into the energy sector's needs and challenges. As such, the contributions can be divided into two categories, theoretical contributions, and practical contributions.

### 9.1.1 Theoretical Contributions

- 17 theories about the Danish energy sector.
- 66 hypotheses about the Danish energy sector.
- The concept of a data space as a data mesh.

### 9.1.2 Practical Contributions

- A proof of concept for a data product that is flexible and easy to use.
- An example of how to use code generation to enable dynamicity in software.
- An example of an easy-to-use code generator that can generate high-quality code with a simple **API**.
- An example of how to construct a comprehensive configuration system that is both functional and user-friendly.
- An example of implementing discoverability, observability, and modern data pipelines with **CNCF** tools in a data product.

## 9.2 Future Work

The current research has established a strong foundation for future work to refine and expand the system's capabilities.

First, significant attention is devoted to improving the interoperability of data products. Currently, the system cannot merge domain models or seamlessly source data from one product to another. The goal is to devise a method for a data product to source and integrate data from another, creating a composite domain model that encapsulates both domains. For example, a school data product should be able to draw on data from a student data product and create a single model incorporating both the school and student domain models.

Secondly, several critical features are missing and should be implemented to empower the system's capabilities, making it more secure, adaptable, and user-friendly:

- Authentication and authorization
- Functions
- Output/egestion
- Validation
- Data cleansing
- GraphQL mutations and subscriptions
- More data storage options

  For a more up-to-date list of planned features and changes, please check the issue tracker on the Data Product's GitHub repository.

Testing the prototype in a distributed environment is another important event for future work. The current setup, hosted on a single MacOS machine within a Docker environment in a virtual machine, does not represent real-world conditions. It would be valuable to run load tests with K6 [95] to evaluate **API** performance and scalability when the system is deployed in a distributed environment.

In terms of reliability improvements, the introduction of integration tests for all features and the implementation of end-to-end testing using Playwright [96] are high priorities. These enhancements will ensure the prototype's robustness and seamless operation during infrastructure updates.

Furthermore, as the data product meets its vertical scaling limits, it should gradually transition to a microservice architecture. This process should involve the strangler fig pattern [60] to isolate the module or service that requires scaling, followed by implementing necessary clients to interface with the new microservice.

In conclusion, the outlined future work aims to optimize the system's performance, interoperability, and reliability while incorporating essential features. Furthermore, it aims to transition towards a scalable architecture, contributing significantly to advancing the data product and its ability to function in real-world scenarios.

# References


[1] "CO2e definition." Accessed: May 19, 2023. [Online]. Available: https://www3.epa.gov/carbon-footprint-calculator/tool/definitions/co2e.html

[2] Danish Energy Agency, "Denmark's climate status and outlook," 2022.

[3] P. L. Ingerslev, *Interview 04 with peter lyck ingerslev*. Available at https://github.com/devantler/thesis-monorepo/tree/main/literature/interviews/interview04-peter-lyck-ingerslev; Unpublished, 2022.

[4] United Nations, "Goal 7: Affordable and clean energy." Accessed: May 19, 2023. [Online]. Available: https://sdgs.un.org/goals/goal7

[5] Flexible Energy Denmark, "Flexible Energy Denmark - Creating Denmark's flexible energy system." Accessed: May 19, 2023. [Online]. Available: https://www.flexibleenergydenmark.com/home/

[6] A. B. Alnor, *Interview 03 with andré bryde alnor*. Available at https://github.com/devantler/thesis-monorepo/tree/main/literature/interviews/interview03-andre-bryde-alnor; Unpublished, 2022.

[7] J. H. Schwee, *Interview 01 with jens hjort schwee*. Available at https://github.com/devantler/thesis-monorepo/tree/main/literature/interviews/interview01-jens-schwee; Unpublished, 2022.

[8] Data Spaces, "Data spaces i energi og forsyning: Aktører og initiativer." Accessed: May 19, 2023. [Online]. Available: https://rapport.dataspaces.dk

[9] K. Lukka, "The constructive research approach," in *Case Study Research in Logistics*, 2003.

[10] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research*. New York, NY: Aldine de Gruyter, 1967.

[11] L. Nagel *et al.*, "Design principles for data spaces – position paper." 2021.

[12] European Commission, "European data strategy." Accessed: May 19, 2023. [Online]. Available: https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy_en

[13] OpenDEI, "OpenDEI." Accessed: May 19, 2023. [Online]. Available: https://opendei.eu/

[14] The European Commission, "The data act." 2022. Accessed: May 19, 2023. [Online]. Available: https://ec.europa.eu/newsroom/dae/redirection/document/83527

[15] GDPR Aps, "GDPR." Accessed: May 19, 2023. [Online]. Available: https://gdpr.dk/

[16] EUR-lex, "Regulation on privacy and electronic communications." Accessed: May 19, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52017PC0010

[17] MyData, "MyData." Accessed: May 19, 2023. [Online]. Available: https://www.mydata.org/

[18] International Data Spaces Association, "International data spaces association." Accessed: May 19, 2023. [Online]. Available: https://www.internationaldataspaces.org/

[19] Sitra, "Sitra." Accessed: May 19, 2023. [Online]. Available: https://www.sitra.fi/en/events/data-spaces-technology-landscape-2023/

[20] Gaia-X, "Gaia-x." Accessed: May 19, 2023. [Online]. Available: https://www.gaia-x.eu/

[21] Sitra, "Sitra rulebook." Accessed: May 19, 2023. [Online]. Available: https://www.sitra.fi/en/publications/rulebook-for-a-fair-data-economy/

[22] FIWARE Foundation, "FIWARE." Accessed: May 19, 2023. [Online]. Available: https://www.fiware.org/

[23] EUR-lex, "Data governance act." Accessed: May 19, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52020PC0767

[24] International Data Spaces Association, "IDS RAM 4." Accessed: May 19, 2023. [Online]. Available: https://docs.internationaldataspaces.org/ids-ram-4/

[25] International Data Spaces Association, "Reference architectural model," 2019.

[26] D. S. S. Centre, "Data spaces support centre." Accessed: May 19, 2023. [Online]. Available: https://dssc.eu/

[27] D. S. S. Centre, "Starter kit for data space designers," 1, 2023.

[28] Z. Dehghani, *Data mesh*, First edition. O'Reilly Media, 2022.

[29] Wikipedia, "Service-level objective." Accessed: May 19, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Service-level_objective

[30] K2View, "Data mesh vendors and buyers guide." Accessed: May 19, 2023. [Online]. Available: https://www.k2view.com/top-data-mesh-vendors

[31] K2View, "K2View." Accessed: May 19, 2023. [Online]. Available: https://www.k2view.com/

[32] Talend, "Talend." Accessed: May 19, 2023. [Online]. Available: https://www.talend.com/

[33] Starburst, "Starburst." Accessed: May 19, 2023. [Online]. Available: https://www.starburst.io/

[34] Informatica, "Informatica." Accessed: May 19, 2023. [Online]. Available: https://www.informatica.com/

[35] Denodo, "Denodo." Accessed: May 19, 2023. [Online]. Available: https://www.denodo.com/

[36] IBM, "Data fabric." Accessed: May 19, 2023. [Online]. Available: https://www.ibm.com/topics/data-fabric

[37] G. Dodig Crnkovic, "Constructive research and info-computational knowledge generation," in *Studies in Computational Intelligence*, vol. 314, 1970, pp. 359–380. doi: 10.1007/978-3-642-15223-8_20.

[38] Delve, "What is Open, Axial, and Selective Coding?" Accessed: May 19, 2023. [Online]. Available: https://www.youtube.com/watch?v=6_gZuEm3Op0

[39] J. Arlow and I. Neustadt, *UML 2 and the unified process: Practical object-oriented analysis and design*, 2nd ed. Addison-Wesley Professional, 2005.

[40] P. S. Jørgensen and L. Rienecker, *Den gode opgave: Håndbog i opgave-, projekt- og specialeskrivning*, 6th ed. Samfundslitteratur, 2022.

[41] P. S. Jørgensen and L. Rienecker, *Specielt om specialer: En aktivitetsbog*, 4th ed. Samfundslitteratur, 2020.

[42] N. Mack, C. Woodsonga, K. M. MacQueen, G. Guest, and E. Namey, *Qualitative research methods: A data collector's field guide*, 1st ed. Family Health International, 2005.

[43] H. B. Hamadou, T. B. Pedersen, and C. Thomsen, "The danish national energy data lake: Requirements, technical architecture, and tool selection," in *2020 IEEE international conference on big data (big data)*, 2020, pp. 1523–1532. doi: 10.1109/BigData50022.2020.9378368.

[44] Energinet, "Energinet." Accessed: May 19, 2023. [Online]. Available: https://www.energinet.dk/

[45] B. Therkelsen, *Interview 02 part 1 with bjørn therkelsen*. Available at https://github.com/devantler/thesis-monorepo/tree/main/literature/interviews/interview02-bjoern-therkelsen; Unpublished, 2022.

[46] G. Kim, J. Humble, P. Debois, and J. Willis, *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations*, 2nd ed. IT Revolution Press, 2021.

[47] Microsoft, "Source generators." Accessed: May 19, 2023. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/roslyn-sdk/source-generators-overview

[48] Scriban, "Scriban." Accessed: May 19, 2023. [Online]. Available: https://github.com/scriban/scriban

[49] J. Bogard, "Vertical slice architecture." Accessed: May 19, 2023. [Online]. Available: https://jimmybogard.com/vertical-slice-architecture/

[50] Microsoft, "Microsoft.FeatureManagement." Accessed: May 19, 2023. [Online]. Available: https://github.com/microsoft/FeatureManagement-Dotnet

[51] G. Kim, J. Humble, P. Debois, and J. Willis, *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations*, 2nd ed. IT Revolution Press, 2021.

[52] aaubry, "YamlDotNet." Accessed: May 19, 2023. [Online]. Available: https://github.com/aaubry/YamlDotNet

[53] A. Lock, "NetEscapades.configuration." Accessed: May 19, 2023. [Online]. Available: https://github.com/andrewlock/NetEscapades.Configuration

[54] Microsoft, "IntelliSense." Accessed: May 19, 2023. [Online]. Available: https://code.visualstudio.com/docs/editor/intellisense

[55] J. Schema, "JSON schema." Accessed: May 19, 2023. [Online]. Available: https://json-schema.org

[56] Microsoft, "Reflection." Accessed: May 19, 2023. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/framework/reflection-and-codedom/reflection

[57] Microsoft, "Incremental generators." Accessed: May 19, 2023. [Online]. Available: https://github.com/dotnet/roslyn/blob/main/docs/features/incremental-generators.md

[58] Microsoft, "System.CodeDom." Accessed: May 19, 2023. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/api/system.codedom

[59] Microsoft, "SyntaxFactory." Accessed: May 19, 2023. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/api/microsoft.codeanalysis.csharp.syntaxfactory

[60] M. Fowler, "StranglerFigApplication," 2004. Accessed: May 19, 2023. [Online]. Available: https://martinfowler.com/bliki/StranglerFigApplication.html

[61] OpenFaaS, "OpenFaaS." Accessed: May 19, 2023. [Online]. Available: https://www.openfaas.com/

[62] H. Chocolate, "Hot chocolate." Accessed: May 19, 2023. [Online]. Available: https://chillicream.com/docs/hotchocolate/v13

[63] Microsoft, "ASP.NET core MVC." Accessed: May 19, 2023. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0

[64] Swagger, "OpenAPI specification." Accessed: May 19, 2023. [Online]. Available: https://swagger.io/specification/

[65] domaindrivendev, "Swashbuckle." Accessed: May 19, 2023. [Online]. Available: https://github.com/domaindrivendev/Swashbuckle.AspNetCore

[66] O. Foundation, "OpenID connect." Accessed: May 19, 2023. [Online]. Available: https://openid.net/connect/

[67] IETF, "OAuth 2.0." Accessed: May 19, 2023. [Online]. Available: https://oauth.net/2/

[68] M. Miller, "WebAuthn." Accessed: May 19, 2023. [Online]. Available: https://webauthn.guide

[69] authentik, "Authentik." Accessed: May 19, 2023. [Online]. Available: https://goauthentik.io

[70] Microsoft, "Caching in .NET." Accessed: May 19, 2023. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/extensions/caching

[71] StackExchange, "StackExchange.redis." Accessed: May 19, 2023. [Online]. Available: https://stackexchange.github.io/StackExchange.Redis/

[72] Redis, "Redis." Accessed: May 19, 2023. [Online]. Available: https://redis.io/

[73] J. Bogard, "AutoMapper." Accessed: May 19, 2023. [Online]. Available: https://automapper.org/

[74] J. Skinner, "FluentValidation." Accessed: May 19, 2023. [Online]. Available: https://docs.fluentvalidation.net/en/latest/

[75] Microsoft, "Blazor." Accessed: May 19, 2023. [Online]. Available: https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor

[76] Blazorise, "Blazorise." Accessed: May 19, 2023. [Online]. Available: https://blazorise.com/

[77] WebAssembly, "WebAssembly." Accessed: May 19, 2023. [Online]. Available: https://webassembly.org/

[78] OWASP, "Clickjacking." Accessed: May 19, 2023. [Online]. Available: https://owasp.org/www-community/attacks/Clickjacking

[79] LinkedIn, "DataHub." Accessed: May 19, 2023. [Online]. Available: https://datahubproject.io/

[80] Apache, "Apache kafka." Accessed: May 19, 2023. [Online]. Available: https://kafka.apache.org/

[81] Microsoft, "Entity Framework Core." Accessed: May 19, 2023. [Online]. Available: https://learn.microsoft.com/en-us/ef/core/

[82] Apache, "Avro." Accessed: May 19, 2023. [Online]. Available: https://avro.apache.org/

[83] Jaeger, "Jaeger." Accessed: May 19, 2023. [Online]. Available: https://www.jaegertracing.io/

[84] OpenTelemetry, "OpenTelemetry." Accessed: May 19, 2023. [Online]. Available: https://opentelemetry.io/

[85] OpenTelemetry, "Collector." Accessed: May 19, 2023. [Online]. Available: https://opentelemetry.io/docs/collector/

[86] Prometheus, "Prometheus." Accessed: May 19, 2023. [Online]. Available: https://prometheus.io/

[87] Grafana, "Grafana." Accessed: May 19, 2023. [Online]. Available: https://grafana.com/

[88] Microsoft, "Instrumentation." Accessed: May 19, 2023. [Online]. Available: https://opentelemetry.io/docs/instrumentation/

[89] Kubernetes, "Kubernetes." Accessed: May 19, 2023. [Online]. Available: https://kubernetes.io/

[90] HashiCorp, "Nomad." Accessed: May 19, 2023. [Online]. Available: https://www.nomadproject.io/

[91] Docker, "Docker." Accessed: May 19, 2023. [Online]. Available: https://www.docker.com/

[92] A. W. Services, "Amazon web services." Accessed: May 19, 2023. [Online]. Available: https://aws.amazon.com/

[93] M. Azure, "Microsoft azure." Accessed: May 19, 2023. [Online]. Available: https://azure.microsoft.com/

[94] G. Cloud, "Google cloud." Accessed: May 19, 2023. [Online]. Available: https://cloud.google.com/

[95] k6, "k6." Accessed: May 19, 2023. [Online]. Available: https://k6.io/

[96] Playwright, "Playwright." Accessed: May 19, 2023. [Online]. Available: https://playwright.dev/

[97] Ministry of Foreign Affairs of Denmark, "Denmark is a laboratory for green solutions." Accessed: May 19, 2023. [Online]. Available: https://denmark.dk/innovation-and-design/green-solutions

[98] Energinet, "Roles and responsibilities in the electricity market." Accessed: May 19, 2023. [Online]. Available: https://energinet.dk/el/elmarkedet/roller-pa-elmarkedet/

[99] OpenInfraMap, "Power plants in denmark." Accessed: May 19, 2023. [Online]. Available: https://openinframap.org/stats/area/Denmark/plants

[100] Energinet, "Kontrolstruktur for det kollektive elforsyningsnet." Accessed: May 19, 2023. [Online]. Available: https://en.energinet.dk/media/jhvjyn4j/kontrolstruktur-for-det-kollektive-elforsyningsnet.pdf

[101] Danish Energy Agency, "Danish energy agency." Accessed: May 19, 2023. [Online]. Available: https://ens.dk/en

[102] Energinet, "Rules and regulations in the danish electricity market." Accessed: May 19, 2023. [Online]. Available: https://en.energinet.dk/electricity/rules-and-regulations/

[103] Danish Ministry of Climate Energy and Utilities, "Climate act (the unofficial translation)." 2020.

[104] The United Nations Framework Convention on Climate Change, "The paris agreement." Accessed: May 19, 2023. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement

[105] United Nations, "THE 17 GOALS | Sustainable Development." Accessed: May 19, 2023. [Online]. Available: https://sdgs.un.org/goals

[106] United Nations, "Goal 13: Climate action." Accessed: May 19, 2023. [Online]. Available: https://sdgs.un.org/goals/goal13

[107] Danish Ministry of Climate, Energy and Utilities, "SDG7 energy compact denmark." 2021. Accessed: May 19, 2023. [Online]. Available: https://www.un.org/sites/un2.un.org/files/2021/09/210920_final_energy_compact-denmark.pdf

[108] Kanbanize, "Kanban." Accessed: May 19, 2023. [Online]. Available: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban

[109] K. Beck *et al.*, "Manifesto for agile software development." Accessed: May 19, 2023. [Online]. Available: https://agilemanifesto.org/

[110] GitHub, "GitHub flow." Accessed: May 19, 2023. [Online]. Available: https://docs.github.com/en/get-started/quickstart/github-flow

[111] P. Hammant, "Trunk based development." Accessed: May 19, 2023. [Online]. Available: https://trunkbaseddevelopment.com/

[112] GitHub, "GitHub actions." Accessed: May 19, 2023. [Online]. Available: https://github.com/features/actions

[113] GitHub, "Dependabot." Accessed: May 19, 2023. [Online]. Available: https://dependabot.com/

[114] GitLab, "What is GitOps?" Accessed: May 19, 2023. [Online]. Available: https://about.gitlab.com/topics/gitops/

[115] GitHub, "CodeQL." Accessed: May 19, 2023. [Online]. Available: https://codeql.github.com

[116] Codecov, "Codecov." Accessed: May 19, 2023. [Online]. Available: https://codecov.io

[117] Hadolint, "Hadolint." Accessed: May 19, 2023. [Online]. Available: https://github.com/hadolint/hadolint

[118] Docker, "Build and push docker images." Accessed: May 19, 2023. [Online]. Available: https://github.com/marketplace/actions/build-and-push-docker-images

# A | The Danish Energy Sector

The Danish energy sector is divided into three sub-sectors, electricity, gas, and hydrogen. The electricity and gas sectors are well established, and the hydrogen sector will be established in the future [44]. Understanding how the energy sector's overall responsibilities and roles are involved is essential to understand the problem domain. For this purpose, the electricity sector will be used as an example.

The Danish electricity sector is known for its progressive renewable energy and sustainability approach [97]. Denmark has a **RES-E** between 65% and 93% today and expects it to reach above 109% by 2030 [2, p. 16]. Obtaining 100% **RES-E** means that Denmark is expected to produce more renewable energy than it consumes. It is a considerable achievement that enables new challenges to be tackled. For example, helping underdeveloped countries reach their climate goals or storing renewable energy for later use.

The electricity sector encompasses various roles and responsibilities, from production to transmission, distribution, and consumption. Furthermore, the sector is highly regulated and governed.

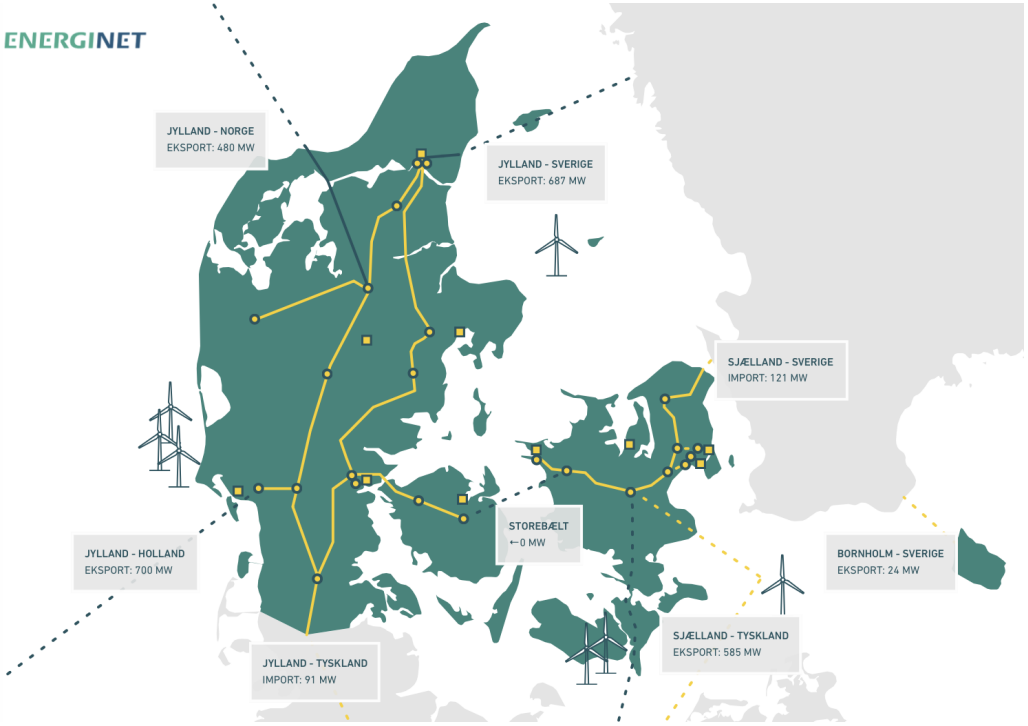


Figure A.1: The Danish Electricity System [44].

## A.1 Electricity Production

Electricity production in Denmark includes power plants, wind farms, solar farms, and biogas facilities. Denmark has been a pioneer in wind energy, with a significant portion of its electricity production coming from offshore and onshore wind farms. Solar farms are also gaining momentum, harnessing energy from the sun to generate electricity [98].

Electricity production is handled by plant owners, who are responsible for the operation and maintenance of power plants. Plant owners are also responsible for ensuring the power plants comply with the relevant regulations and standards. Denmark has about 117 power plants [99] ranging from large- to small-scale facilities.

## A.2 Electricity Transmission

Once electricity is generated, it must be transmitted from the production sites to **Distribution System Operators (DSOs)**. Denmark's electricity transmission responsibility lies with the **Transmission System Operator (TSO)**, Energinet. Energinet is as a **TSO** primarily responsible for planning, operating, and maintaining the transmission grid, including high-voltage power lines and substations [98].

## A.3 Electricity Distribution

**DSOs**, deliver electricity from the transmission grid to end consumers, including households, businesses, and industries [98]. In Denmark, electricity distribution is carried out by several **DSOs**, some serving specific geographical areas [100, Appendix 2]. **DSOs** maintain low-voltage and medium-voltage distribution networks, including power lines, transformers, and substations. They also handle metering, billing, and customer service for electricity consumers [98].

## A.4 Electricity Consumption

End consumers play a vital role in the Danish electricity sector as they are responsible for consuming the electricity produced and distributed. Consumers can be households, businesses, or industries, and consumers are encouraged to adopt energy-efficient practices and use electricity responsibly to minimize their environmental impact. Consumers can choose their electricity supplier, as the electricity market in Denmark is liberalized, allowing for competition among suppliers [98].

## A.5 Rules and Regulations

Laws, regulations, and policies regulate the Danish electricity sector to ensure its smooth functioning and adherence to sustainability goals. The **Danish Energy Agency (DEA)** [101] regulates and oversees the energy sector, including setting energy production, transmission, distribution, and consumption policies. Denmark has set ambitious targets for renewable energy and carbon neutrality, as presented in Appendix B. Policies aimed at promoting the development of renewable energy sources, reducing greenhouse gas emissions, and fostering innovation in the electricity sector have been enacted to achieve these goals [102].

# B | Denmark's Climate Status

According to the Danish Climate Status and Outlook for 2022, Denmark must lower $\mathbf{CO_2e}$ emissions by 50%-54% in 2025 and 70% in 2030 compared to $\mathbf{CO_2e}$ levels in 1990 [2, p. 5]. The targets stem from the **Danish Climate Act (DCA)** that sets the legal requirements for achieving global climate goals in Denmark per the Paris Agreement [103, p. 1].

> "The Paris Agreement is a legally binding international treaty on climate change. It was adopted by 196 Parties at the **UN** Climate Change Conference (COP21) in Paris, France, on 12 December 2015. It entered into force on 4 November 2016.
>
> Its overarching goal is to hold "the increase in the global average temperature to well below 2°C above pre-industrial levels" and pursue efforts "to limit the temperature increase to 1.5°C above pre-industrial levels [104]."

The Paris Agreement resulted in a set of goals dubbed **SDGs** [105]. There are 17 **SDGs** that set the framework for ending poverty, protecting the planet, and ensuring that all people enjoy peace and prosperity by 2030. Each **SDG** has a set of targets with a set of indicators. A target is a sub-goal to the **SDG** that must be met to achieve the **SDG**. An indicator is a measurable value used to determine if a target is met.

Of the 17 **SDGs**, **SDG7** [4], and **SDG13** [106] are the most relevant to the thesis. **SDG7**, depicted in fig. B.1, aims to ensure access to affordable, reliable, sustainable, and modern energy for all. It includes the production, transmission, distribution, and consumption of energy.

**Target 7.1** "By 2030, ensure universal access to affordable, reliable, and modern energy services [4]."

**Indicator 7.1.1** "Proportion of population with access to electricity [4]."

**Target 7.2** "By 2030, increase substantially the share of renewable energy in the global energy mix [4]."

**Indicator 7.2.1** "Renewable energy share in the total final energy consumption [4]."

**Target 7.3** "By 2030, double the global rate of improvement in energy efficiency [4]."

**Indicator 7.3.1** "Energy intensity measured in terms of primary energy and **Gross Domestic Product (GDP)** [4]."

**Target 7.a** "By 2030, enhance international cooperation to facilitate access to clean energy research and technology, including renewable energy, energy efficiency, and advanced and cleaner fossil-fuel technology, and promote investment in energy infrastructure and clean energy technology [4]."

**Indicator 7.a.1** "International financial flows to developing countries in support of clean energy research and development and renewable energy production, including in hybrid systems [4]."

**Target 7.b** "By 2030, expand infrastructure and upgrade technology for supplying modern and sustainable energy services for all in developing countries, in particular least developed countries, small island developing States, and land-locked developing countries, in accordance with their respective programmes of support [4]."

**Indicator 7.b.1** "Installed renewable energy-generating capacity in developing countries (in watts per capita)" [4]

**SDG13** [106], depicted in fig. B.2, aims to take urgent action to combat climate change and its impacts. It includes strengthening resilience and adaptive capacity to climate-related hazards and natural disasters.

**Target 13.2** "Integrate climate change measures into national policies, strategies, and planning [106]."

**Indicator 13.2.2** "Total greenhouse gas emissions per year [106]."

According to the **Danish Ministry of Climate, Energy and Utilities (DMCEU)**, a **Decade Action Agenda (DAA)** is in place to advance **SDGs** [107]. The **DAA** presents the ambitions, actions, outcomes, required resources and support, impact, monitoring, and reporting towards advancing **SDGs**.

However, to keep track of Denmark's progress toward reaching sustainability targets, one can refer to the already-mentioned Danish Climate Status and Outlook report. The **DEA** is legally obliged to write a new report annually [2, p. 5], and the report has more up-to-date info regarding current progress and targets than the **DAA**. The latest available report is from 2022, and it shows that Denmark is not on track to meet its targets by 2025 or 2030. The $\mathbf{CO_2e}$ emissions are expected to be 47% lower than the 1990 level by 2025 and 57% by 2030 [2, p. 10]. It means Denmark is expected to miss the 2025 target by 3%-7% and the 2030 target by 13%. That is a shortfall of 2.4-5.5 million tonnes $\mathbf{CO_2e}$ in 2025 and 10.1 million tonnes $\mathbf{CO_2e}$ in 2030 [2, p. 8].

Not meeting the targets is a huge deal, as the consequences of the increased amount of $\mathbf{CO_2e}$ in the atmosphere can result in crossing the global warming threshold of 1.5°C and resulting in severe impacts of climate change.

> "(...) the UN's Intergovernmental Panel on Climate Change indicates that crossing the 1.5°C threshold risks unleashing far more severe climate change impacts, including more frequent and severe droughts, heatwaves and rainfall [104]."

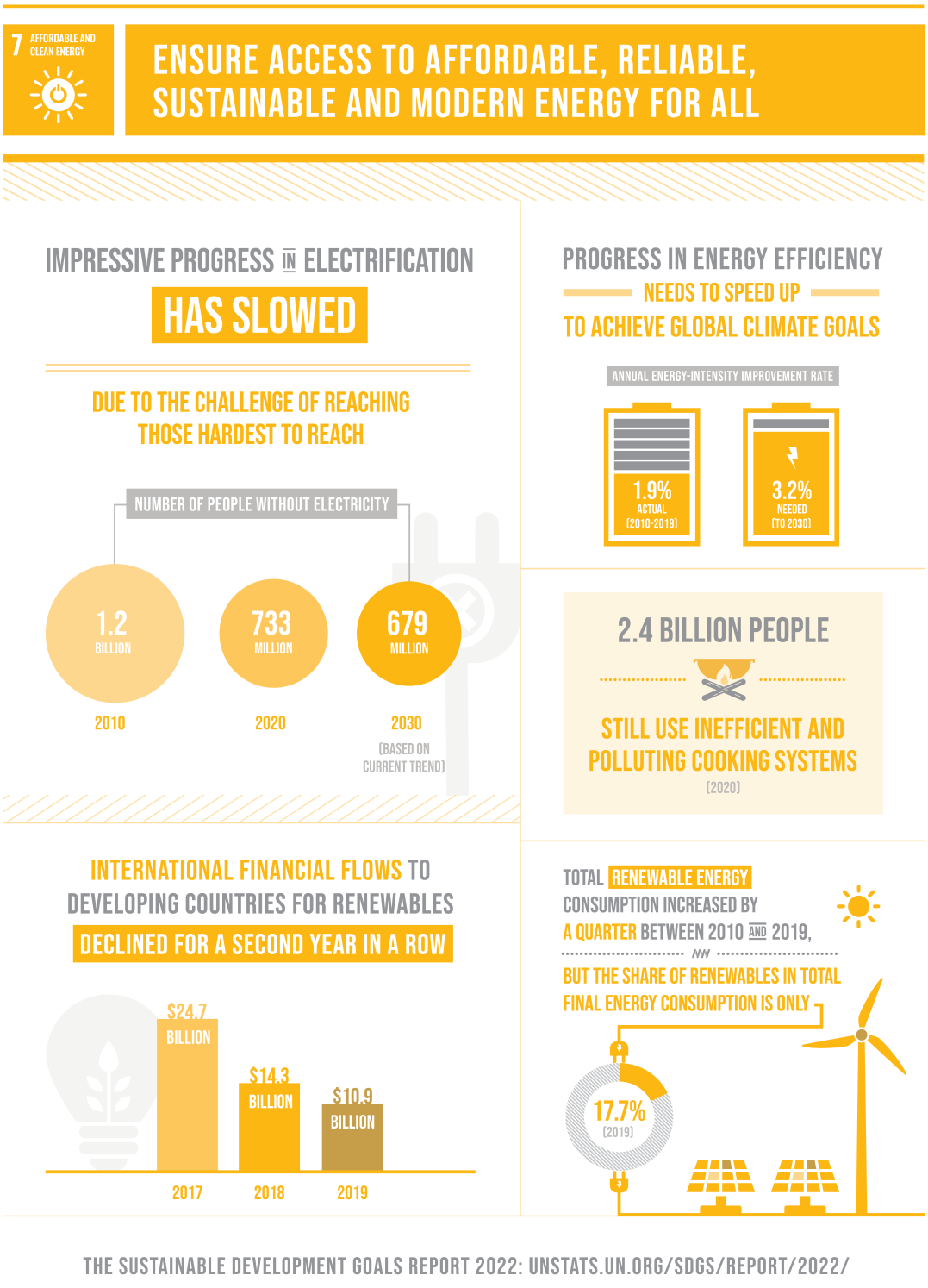


Figure B.1: **SDG7**: Affordable and Clean Energy [4].

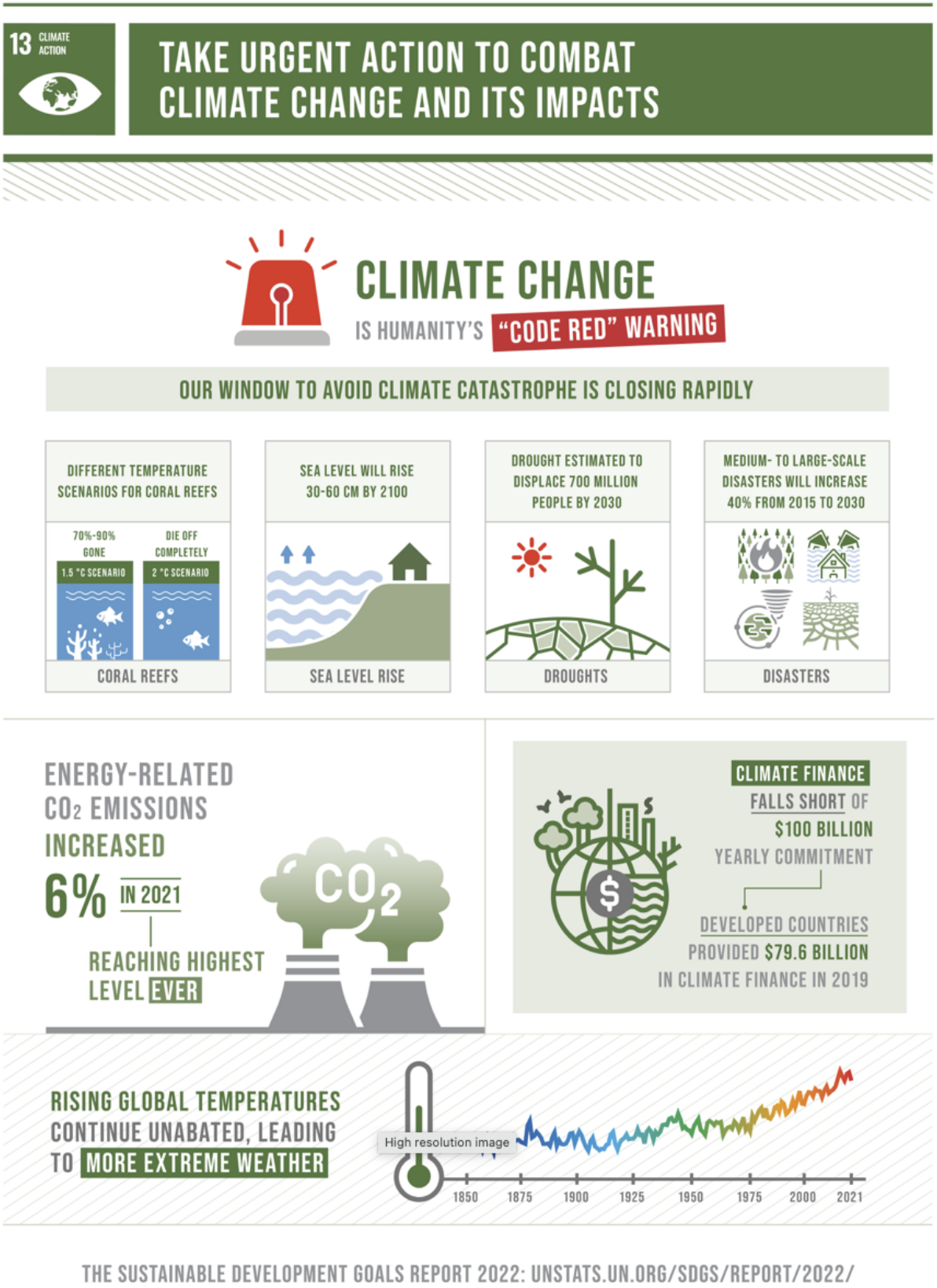


Figure B.2: SDG13: Climate Action [106].

# C | How Kanban Is Utilized

Kanban [108] is a lean workflow management method that is simple yet effective. It is centered around six principles and practices that all draw inspiration from the Agile Manifesto[17]. Each principle and practice provide some value to solo development and project management. As such, they will be used as a project management guideline. How so is described in the following paragraphs, which present the six principles and practices, and how they provide value for managing the project:

**Principle 1 - Start with what you do now** is valuable as it allows starting the process with preferences and practices in mind. Then, gradually improving the process by exercising discipline towards the other principles and practices. Doing so allows the author to practice and master the principles and practices through experience rather than letting the process become an obstacle to the project.

**Principle 2 - Agree to pursue incremental, evolutionary change** is precious for solo development and project management, as pursuing incremental, evolutionary change is a lot easier than attempting a large-scale transformation at once when resources are limited. It also increases flexibility, as doing small changes decreases the risk of taking a wrong turn and allows changing direction quickly. As the project threads new ground, flexibility can become the difference between the project succeeding or failing.

**Principle 3 - Encourage acts of leadership at all levels** does not apply well to solo development and project management, as there is no one else to lead. However, the core of the principle is still valuable as it encourages taking responsibility for the project and taking the initiative to improve the process.

**Principle 4 - Focus on customer needs and expectations** is critical to remember, as it helps focus on the tasks most valuable to the project and the target audience. For projects with deadlines, the value of this principle is even more significant, as helping prioritize tasks decreases the risk of delivering unsatisfactory results due to time constraints.

**Principle 5 - Manage the work, not the workers** is also valuable to solo projects, as it emphasizes the importance of managing the work and empowering the individual's ability to self-organize. This principle perfectly fits as a solo project is all about self-organization.

**Principle 6 - Regularly review the network of services** is about learning from past experiences. If some part of the process is not working, it should be identified and improved promptly and not left to fester. Doing so will continuously improve the process and help avoid repeating mistakes.

**Practice 1 - Visualizing the workflow** refers to using a Kanban board[18], but the author prefers to avoid the overhead of managing a project board. As such, a combination of GitHub issues and GitHub **Pull Requests (PRs)** will be used to visualize and manage tasks while keeping things simple. The process is described in detail in the Appendix D.

**Practice 2 - Limiting work in progress** is one of the critical practices of Kanban. The process does not allow more than X tasks to progress at any given time. The consensus is that it allows a greater focus, improves lead time, and reduces the amount of context switching. The exact value of X is up to the team or individual adopting the practice, but the author will use a value of 1. Non-coding or non-documentation tasks are not considered managed work. The author allows himself to work only on one coding or documentation task at a time, but conducting meetings, interviews, or research in parallel is allowed.

**Practice 3 - Managing flow** is about continuously monitoring the process and adjusting to improve the workflow or address bottlenecks. The author will monitor the process ad-hoc, and if a bottleneck or resistance is identified, the author might stop ongoing work and address the issue. The final decision will be based on the project's current state and priorities. Improving the process is a high priority early on.

[17]The Agile Manifesto [109] is a set of principles and values that define the Agile movement. The manifesto was created in 2001 by a group of software developers and project managers, and it is still widely used today.

[18]A Kanban board [108] is a work log consisting of a table with columns and rows. The columns represent swimlanes that contain all tasks present on the board. The rows represent a task's stages, e.g., To Do, In Progress, and Done. The board is typically visualized as a physical board but can also be visualized as a digital board, e.g., in a spreadsheet or a project management tool.

However, the work might become more critical as the project progresses, requiring the author to focus on the work and ignore potential improvements to the process.

**Practice 4 - Making process policies explicit** is a critical practice when working in a team, but as a solo developer and project manager, it is unnecessary to define policies explicitly. The chosen processes and practices already define the framework of the process, and given that the author is the only one having to follow the process, defining the policies becomes a time-consuming task that does not provide much value.

**Practice 5 - Implementing feedback loops** greatly interests the author. Resources are limited as a solo developer and project manager, so implementing feedback loops is a great way to improve productivity. The author will use feedback loops by utilizing code analysis and code review tools, which will help identify issues early and provide insights into improving the code, quality, and the project's stability. Furthermore, **CI** / **CD** pipelines will automate otherwise manual tasks, which will help reduce the time spent on repetitive work and the risk of human error. It will also reduce the lead time, as implementing, testing, and releasing new features will be faster.

**Practice 6 - Improving collaboratively** does not apply to solo development and project management. However, the author plans to invite the supervisor and external stakeholders to provide feedback on the process and the project. Feedback will help prioritize tasks and emphasize the importance of experimenting with new ideas and concepts.

# D | How GitHub Flow Is Utilized

The GitHub Flow [110] is a lightweight, branch-based workflow that uses short-lived branches to develop features, fix bugs, and perform experiments (fig. D.1). It is closely related to trunk-based development[19] [111] and enables teams to collaborate on projects while enabling rich feedback loops with GitHub Actions[20] [112] official and third-party integrations.

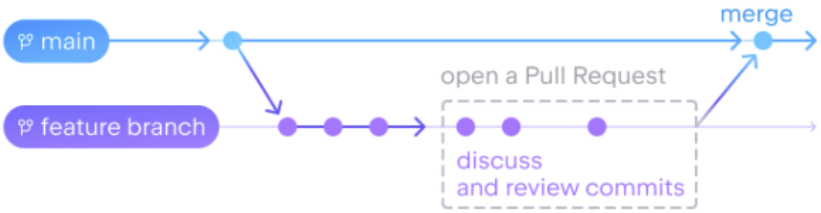


Figure D.1: The GitHub Flow [110]

The process centers around a 6-step cycle:

1. Create a branch
2. Make changes
3. Create a **PR**
4. Address review comments
5. Merge your **PR**
6. Delete your branch

The process has enabled the author to create an automated workflow where many bugs and issues have been caught early. Each **PR** runs GitHub Actions, ensuring Unit Tests and Integration Tests pass, code coverage stays high, and no secrets, security issues, or code quality degradation happens. Validation that these prerequisites pass is called checks (fig. D.2).

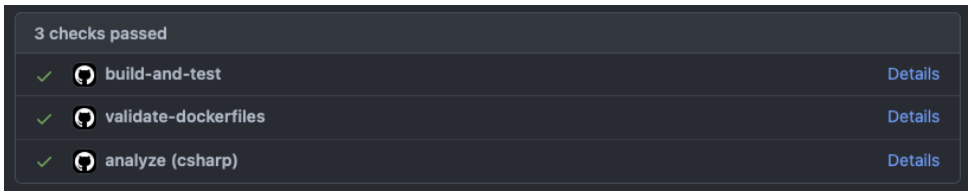


Figure D.2: An overview of three passed checks for a successful **PR**.

Failing checks generate reports as comments on the **PR**, and step four has been chiefly about resolving these comments to make the checks pass. Steps five and six have been automated, as GitHub supports auto-merging **PRs** when all checks pass and removing branches after successfully merging. The process enables the author to develop new features with little overhead and a short lead time while simultaneously adding a layer of quality assurance to the project.

After successfully merging a **PR**, a deployment can be triggered. Deployments are triggered by making a new release in the related repository, which executes a GitHub Action that uploads a new docker image

[19] Trunk-based development is a software development methodology that uses a single branch as the source of truth, and creates feature-branches to develop new features, fix bugs, and perform experiments [111].

[20] GitHub Actions is a **CI / CD** tool that enables developers to automate, customize, and execute their software development workflows defined in their GitHub repository [112].

to an image registry. Then another repository polls for updated image versions with dependabot[21] [113], and if it finds updates, it creates a new **PR** that bumps image versions. Merging the **PR** will trigger the actual deployment through a GitOps[22] [114] agent that watches the repository files for changes. If it encounters any, it will pull and deploy the changes to a containerized environment like Docker [91], Kubernetes [89], or Nomad [90].

The most notable GitHub Actions used in the project are:

**CodeQL** [115]. A code scanning tool built into GitHub that automatically detects common vulnerabilities and coding errors in code.

**CodeCov** [116]. A code coverage tool that tracks how much code is covered by tests and helps identify which parts of the code are not.

**Hadolint** [117]. A linter for Dockerfiles that helps to ensure best practices are followed.

**Docker Build Push Action** [118]. A GitHub Action that builds and pushes Docker images to a container registry.

[21] Dependabot is a GitHub integration that automatically creates **PRs** to update dependencies for various tools and languages [113].

[22] GitOps is a set of practices that aim to automate and standardize software delivery and its infrastructure, using Git as a single source of truth [114]. Changes to declarative files, like docker-compose files, are used to trigger deployments, which are then automatically applied to the environment with the help of an agent.

# E | The Unified Process

**UP** is a software development process that provides a framework for developing software systems, following a traditional software development life cycle, with the major difference being that it is an iterative and incremental process [39, p. 34]. **UP** consists of five project life cycles:

**Inception** A project's inception stage tries to launch the project by determining its viability, developing a business case, gathering crucial needs, and identifying significant risks. Technical prototyping is used to validate technology decisions, and proof of concept prototyping is used to validate business needs. A business case is developed to prove that the project would bring tangible business benefits. The system's essential requirements are gathered to help define its scope and pinpoint its significant risks [39, p. 39].

**Elaboration** A stage of software development called elaboration tries to establish an executable architectural baseline. Unlike a disposable prototype, the executable architectural baseline is the first iteration of the desired system that will transform into the finished delivered system during the construction and transition stages. The risk assessment is one of the objectives of elaboration, along with establishing quality attributes, capturing use cases to the functional requirements, developing a thorough strategy for the building phase, and other things. This stage is regarded as the most crucial because it determines the outcomes of other phases of software development [39, pp. 40–41].

**Construction** The evolution of the architectural baseline created during the elaboration stage into the finished system is the primary focus of the Construction stage of software development. Maintaining the integrity of the system architecture is a crucial problem at this point. To avoid integrity problems, ensuring the system architecture is respected during the development process is crucial [39, p. 42].

**Transition** The transition stage begins when the system is considered ready for deployment and is the last stage of the software development life cycle. This stage's key objectives are to deploy and maintain the software solution to users [39, p. 43].

Although **UP** defines a project's different stages, it is not a linear process. Each stage consists of one or more iterations with five core workflows [39, p. 36]:

1. Requirements
2. Analysis
3. Design
4. Implementation
5. Test

The workflows are traditional to software engineering, where requirements are gathered and analyzed before a design is created. The design is then implemented and tested to ensure it meets the requirements. The process is not exclusive to programming and can be considered a general problem-solving process that might include other workflows, like planning and assessment.

The transition between stages is seldom clear-cut, and it is common for a project to be in multiple stages at once or go back and forth between stages as new information is discovered [39, p. 38].

Each stage is concluded with a set of deliverables, where most are used as input for the next stage. Due to the iterative nature of the project, these deliverables are not final but works in progress that are continuously refined and improved throughout the project. However, as they reach a certain level of maturity, they allow the project to transition to the next stage gradually.

# F | Supervisor Contract

## §1 General

**§1.1** The supervisor expects the student to be proactive.

**§1.2** The supervisor expects feedback on the supervision. Ideally, mid-way evaluation and project closure evaluation.

**§1.3** If there are disagreements or further requirements for the collaboration, this contract will be revised.

## §2 Supervisory meetings

**§2.1** Meetings happen at minimum once every 14th day.

**§2.2** Meetings must be scheduled with Calendly.

**§2.3** There is no limit to how many meetings can be scheduled.

**§2.4** An agenda for meetings must be provided 3-4 days before the meeting.

## §3 Feedback

**§3.1** Feedback will be provided verbally during meetings on a case-by-case basis.

**§3.2** Textual feedback is limited due to the limited supervisor time available.

**§3.3** The supervisor expects the student to ask for feedback selectively.

**§3.4** The supervisor expects material that should be reviewed to be mostly finished.

**§3.5** The student does not expect the supervisor to check grammar.

**§3.6** The student expects the supervisor to be honest and direct and inform the student if a piece of work is not satisfactory.

## §4 Personal issues

**§4.1** The supervisor and the student expect honesty and directness regarding issues on a personal level.

## §5 Confidentiality

**§5.1** The thesis is not expected to be confidential.

**§5.2** The student must sign an **NDA** contract to access and work at Energinet.

**§5.3** The supervisor expects total confidentiality to be maintained if any company or national secrets are shared during on-premise access at Energinet.

**§5.4** In connection with interviews, the student must announce that the student is not under an **NDA** contract (because the thesis is decided not to contain confidential information).

## §6 Planning and progress

**§6.1** The supervisor will intervene if the student is not respecting deadlines and agreements or is not working effectively.

**§6.2** Progress is monitored with bi-weekly meetings and issue tracking on GitHub.

## §7 Roles

**§7.1** The supervisor will provide academic guidance (methods, research, tools, scope and framework of the project) and feedback on the student's work.

**§7.2** The primary contact person at Energinet will take responsibility for guiding questions, interviews, etc., throughout the company.

# G | General Interview Guide

**Before starting an interview:**

1. Introduce the background of the project and why it is important.
2. Introduce the method used for conducting the interview and how data will be processed.
3. Explain the purpose of the interview and how it relates to the project.
4. Inform the interviewee(s) that I am not under an **NDA**, so confidential information should not be shared.
5. Ask for consent to record the interview from all participants. (both before and after the interview)

**During an interview:**

1. Introduce myself.
2. Thank the interviewee(s) for taking the time to participate in the interview.
3. Ask the interviewee(s) to introduce themselves.
4. Conduct the interview

# H | Interview 01 with Jens Hjort Schwee

## H.1 Interview Guide

**Type:** Semi-Structured Interview
**Method:** Grounded Theory
**Date:** 2022-09-08
**Interviewer:** Nikolai Emil Damm
**Interviewee:** Jens Hjort Schwee
**Observer:** Jakob Hviid

**Subject 1- What are the current issues in the field of flexible energy in Denmark concerning collaboration between sectors?**

- What does collaboration between sectors mean?
- What issues are there with collaboration between sectors?
- Can data spaces help solve problems concerning collaboration between sectors? If so, how?

**Subject 2 - What are data spaces, and what types of data spaces exist? Can they be categorized?**

- What is a data space?
- What types of data spaces exist?
- What are the benefits and disadvantages of each type?
- What are the common characteristics of data spaces?
- What are the common pitfalls of data spaces?
- Can data spaces be categorized? If so, how?

**Subject 3 - What does it mean for a Data Space to be centralized or decentralized? What are the benefits and disadvantages of each?**

- What is centralization regarding data spaces?
- What are the benefits of centralization in data spaces?
- What are the disadvantages of centralization in data spaces?
- What is decentralization regarding data spaces?
- What are the benefits of decentralization in data spaces?
- What are the disadvantages of decentralization in data spaces?

## H.2 Open Coding

Disclaimer - Nikolai Emil Damm has translated all transcripts from Danish to English. In this process, some alterations to the transcripts have been made to clarify the context.

Table H.1: Open coding of Jens Hjort Schwee's statements in Interview 01.

| ID | Time | Code | Statement |
|---|---|---|---|
| 1 | 00:20 | Introduction | My name is Jens I am employed as a digital business developer in digitalization with a primary focus on data spaces. |
| 2 | 00:30 | Introduction | PhD in Software Engineering from **SDU** with an expertise in Privacy, Data Sharing, Risk Identification and Governance |

| ID | Time | Code | Statement |
|---|---|---|---|
| 3 | 01:00 | Introduction | Responsible for the technical aspects of the transverse efforts |
| 4 | 01:58 | Collaboration | Isolated from Energinet's perspective there is a need for collaboration to manage energy consumption from water, heat and gas. |
| 5 | 02:25 | Collaboration | Energinet need to communicate the different needs digitally that different companies have in the different sectors. |
| 6 | 02:35 | Collaboration | From another perspective, Flexibility is not in Energinet's favor concerning collaboration. |
| 7 | 02:52 | Context | Other companies' digitalisation levels differ from Energinet's. |
| 8 | 02:57 | Context | Energinet is dealing with a lot of actors. |
| 9 | 03:27 | Collaboration | In the gas sector, it is somewhat easier to collaborate as it is mostly one company that controls all gas. |
| 10 | 03:38 | Collaboration | If we look at the district heating sector, it has a regional division, so, collaboration is between Energinet and each region. |
| 11 | 04:11 | Context | Energinet and companies in the different sectors have conflicting interests and goals. |
| 12 | 04:21 | Context | Energinet is interested in running its core businesses. |
| 13 | 04:25 | Context | Energinet's core businesses will to some degree, overlap with other companies' interests. |
| 14 | 04:37 | Collaboration | If we need to tell the water sector they must turn down the energy consumption because they use too much power on water pumps, we must come up with a counter offer. |
| 15 | 05:00 | Collaboration | There is no automatic process for handling collaboration and counter-offers today. |
| 16 | 05:50 | Context | Ideologically, we can solve collaboration with data spaces. In reality, there are a lot of prerequisite agreements that must be made. |
| 17 | 06:00 | Context | Energinet has a maturing phase before data spaces can be implemented, and this phase is different for all sectors. |
| 18 | 06:15 | Collaboration | Energinet must collaborate better. |
| 19 | 06:26 | Context | Whether we will solve the issues with data spaces is not fully defined yet, but phones are not very scalable. |
| 20 | 06:41 | Collaboration | Energinet needs a better infrastructure to exchange information between sectors. |
| 21 | 06:57 | Data Spaces | With data spaces, we could ensure exchanging of information, but only the information needed to be exchanged. |
| 22 | 07:13 | Governance | Information is business critical, and we cannot just tell that the solar system around the corner produces X amount of energy. It is not our information to share. |
| 23 | 07:22 | Governance | To some degree, we need to be able to share business-critical information, like when we need to share that we have surplus energy in a specific sector. |
| 24 | 07:37 | Governance | Energinet will have to share some information that parts of the organization will deem sensitive. |
| 25 | 07:44 | Governance | Energinet will need to find an information model and communication model that allows responsible sharing of sensitive information. |
| 26 | 07:59 | Data Spaces | Data spaces is a potential solution, but it is one of many. Data spaces are the most well-defined solution. |
| 27 | 08:17 | Data Spaces | If you find the definition for data spaces, I would like to know it. |
| 28 | 08:22 | Data Spaces | Ideologically, a data space is a way to automatically exchange information between actors, in a secure, transparent, and visual way. |
| 29 | 08:43 | Data Spaces | When looking a data spaces from a practical perspective people have conflicting ideas on what it is, primarily conflicting subparts and interests |
| 30 | 09:50 | Data Spaces | Data spaces are not well-defined, but we can define the overall concepts and components in a data space |
| 31 | 10:16 | Data Spaces | We must accept that there are many different interpretations of data spaces and many different ways to act in data spaces. There is not just one data space but many that collaborate in unity. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 32 | 10:35 | Data Spaces | European data space with core components like identification, trust, certificates and authority. |
| 33 | 10:50 | Data Spaces | National data space that uses the European data space and defines international needs. |
| 34 | 11:20 | Data Spaces | We must differentiate between global to local data spaces. |
| 35 | 11:48 | Data Spaces | From a marketing analysis perspective, a data space is called a marketing platform, where data owners get permission to sell their data. |
| 36 | 12:02 | Data Spaces | Some would call a data space an information bus that allows exchanging information securely and with a high level of trust. |
| 37 | 12:22 | Data Spaces | We need a data space because we need to know who we are. We need metadata to understand each other and to find and use each other's resources. |
| 38 | 13:02 | Data Spaces | There are five main characteristics, trust, identification, sharing (Jens forgot the last one) |
| 39 | 14:05 | Centralisation | Centralisation is part of the solution. Some of the elements must be centralized, but not all. |
| 40 | 14:14 | Centralisation | We need to have some centralized components in our data space because how would we otherwise create trust in the things we create? |
| 41 | 14:22 | Centralisation and Decentralisation | We can decentralize services that represent a centralized component like identification. |
| 42 | 14:26 | Centralisation | We need some central standard, and that is an aspect of centralization. |
| 43 | 14:34 | Data Spaces | I see data spaces as a part of a large collection of sub-groupings of data spaces. |
| 44 | 14:47 | Data Spaces | Creating a centralized or decentralized data space depends on the qualities that we want. |
| 45 | 14:50 | Decentralisation | If we need fast response time, we need decentralization. |
| 46 | 15:04 | Centralisation | Pros of centralization is that you can top manage things. |
| 47 | 15:11 | Decentralisation | In a decentralized solution, you must decouple elements and have much more exchanging and overhead. There is a price for decentralizing; it is not necessarily a bad price to pay, but it is a price. |
| 48 | 15:11 | Centralisation | Some actors need to centralize their solution, as that is what they know. |
| 49 | 15:57 | Decentralisation | If we introduce too much complexity, with decentralization, on top of all the things they need to integrate with, the price for buying into these paradigms will be too great. |
| 50 | 16:51 | Metadata | We need an overall metadata infrastructure before we can agree on anything |
| 51 | 17:10 | Metadata | There is a huge amount of legacy, systems are not built for data spaces. |
| 52 | 17:50 | Metadata | In our DataHub, we collect data in a specific way. We have a schema that we save data into. If we have to decentralize that component, we will meet much resistance. |
| 53 | 18:24 | Metadata | There is a buy-in period that limits what is possible in the beginning; moving too fast will make things too complex. (to begin with, it is not possible to go fully decentralized) |
| 54 | 19:02 | Data Spaces | We need a national data space that can be built on top of. |
| 55 | 19:38 | Centralisation and Decentralisation | We need aspects of centralization and decentralization. |
| 56 | 20:55 | Decentralisation | As we distribute locally, solutions will become more decentralized. |
| 57 | 21:17 | Data Spaces and Governance | Concerning data spaces and the **DSI**, it is defined in the data governance act. |
| 58 | 21:48 | Governance | There exists law text that defines the legal framework for that defines what must be adhered to locally. |
| 59 | 22:00 | Centralisation | To collaborate, we must have central decision-makers. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 60 | 22:00 | Centralisation and Decentralisation | Concerning software architecture, the balance of centralization and decentralization will be decided based on what the data should be used for, and whether issues can be solved on the client-side. |
| 61 | 22:51 | Centralisation and Decentralisation | If we must balance the energy grid, we need both centralization and decentralization, depending on the issues we look into it. |
| 62 | 23:54 | Centralisation and Decentralisation | The vision is to decentralize as much as possible, but practically it will be a mix of both centralization and decentralization. |
| 63 | 24:12 | Centralisation and Decentralisation | We must accept that some actors have a centralized solution that integrates into decentralized solutions and vice versa. |
| 64 | 24:55 | Discoverability | Discoverability is essential for data spaces. |
| 65 | 24:55 | Data Spaces | Discoverability, trust, identity, authorization, and authentication are essential for data spaces. |
| 66 | 26:00 | Software Qualities | There is a long list of software qualities that data spaces should adhere to. |
| 67 | 26:10 | Software Qualities | Performance might not be a problem, depending on the data. |
| 68 | 26:52 | Software Qualities | If your signal on when you need to balance the electrical grid is 5 seconds delayed, then IT IS a problem. |
| 69 | 27:02 | Context | The solution must be defined from the problem |
| 70 | 27:13 | Data Spaces | We need to determine how much logging there should be in the data space. |
| 71 | 27:36 | Governance | When I demand that you cannot use my data any longer, I want to be able to see the data disappear on the other end. We need that transparency. (Delegation and Access-control) |
| 72 | 28:35 | Data Spaces | The fundamental differences between needs for the **EU** data space and the national data spaces are: - As a writer, the answer can be sharing of data. - As database admin, the answer can be that we do as we usually do. - As a member of digitalization, the answer can be that it is unclear. So we have defined three data types: real-time, historical, and future. - It does not state much about delegation and access control. - It mentions digital trust. It is not well-defined in the European context. We look into moving digital trust into edge devices. |
| 73 | 30:13 | Context | The problem with what is stated in the European context is that it is not well-defined. It is written too compactly, so information is lost. |
| 74 | 32:45 | Governance | Delegation is important to Governance. |
| 75 | 33:00 | Governance | I want to share my yearly consumption of energy with someone. I want to share it with you, and you can use it for 10 minutes, when the 10 minutes are up, you cannot access it anymore. That is a Governance layer. We need that. |
| 76 | 33:28 | Governance | We need protocols to enable us to do delegation and access control and to define what you are authorized to do with data. |
| 77 | 33:49 | Governance | We need to secure the compliance that is needed. We need to be able to see that you did what you said you did. |
| 78 | 34:12 | Collaboration | We need to define the framework for how we collaborate, what transaction models we will use, what costs we have, on what terms we trade, and what you can get in return for a trade. |
| 79 | 34:23 | Context | We need to define actors' roles. There are many definitions of roles, and many of the roles are not well-defined yet. |
| 80 | 34:55 | Metadata | Without metadata, we can not understand the data. |
| 81 | 34:50 | Metadata | There are two roles to metadata, (1) the context the data is collected in, and (2) what the data is. When we have this, we can begin to make Discoverability on top of it. |
| 82 | 36:15 | Metadata | It is an open question of how to handle different metadata contexts in different domains or the same domain. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 83 | 36:55 | Metadata | We can put a data model with sufficient metadata into a discoverability **API** that allows us to specify a query that finds data. |
| 84 | 37:16 | Metadata | Someone or something must create and define metadata to make data discoverable in a data space. |
| 85 | 37:52 | Metadata | In practice, there will be multiple different metadata standards within each domain. There must be an aggregator that can transform data from one standard to another. |
| 86 | 38:43 | Metadata | We must accept that some data is lost when aggregating data. |
| 87 | 38:51 | Metadata | ISE61 and ISE150 are standards that span across sectors. |
| 88 | 39:25 | Metadata | There will be a metadata agency and aggregators that can convert between data. |
| 89 | 40:08 | Data mesh | Data meshes are a sequence of concepts that overlap with the data space initiation. If we go with the data mesh strategy, the buy-in will move because we get governance, data structure, and data discoverability. If we strive to go with data mesh, then the migration to data spaces will become easier, and the maturity level where we start will be much different. |

# I | Interview 02 with Bjørn Therkelsen

## I.1 Interview Guide

**Type:** Semi-Structured Interview
**Method:** Grounded Theory
**Date:** 2022-09-20
**Interviewer:** Nikolai Emil Damm
**Interviewee:** Bjørn Therkelsen

**Subject 1 - What is a Data Space?**

- How do you define a data space?
- What are the typical components/layers in a data space?
- What are the advantages and disadvantages of a data space?
- Are you familiar with Data Warehouses, Data Lakes, Data Lakehouses, and data meshes?
  - What are the components of a Data Warehouse?
  - What are the advantages and disadvantages of a Data Warehouse?
  - What are the components of a Data Lake?
  - What are the advantages and disadvantages of a Data Lake?
  - What are the components of a Data Lakehouse?
  - What are the advantages and disadvantages of a Data Lakehouse?
  - What are the components of a data mesh?
  - What are the advantages and disadvantages of a data mesh?
- Are you familiar with the term data landing zones?
  - What is a data landing zone?
  - Why do we need landing zones?
  - How many and which landing zones are required?

**Subject 2 - What is data ingestion?**

- How do you define data ingestion?
- Are you familiar with the term ETL (Extract, Transform, and Load)?
  - What does Extract mean?
  - What does Transform mean?
  - What does Load mean?
  - What are the different combinations of ETL?
  - What are the advantages and disadvantages of different combinations of ETL?
- Are you familiar with Real-time data ingestion, Batch-based data ingestion, and Lambda architecture-based data ingestion?
  - What are the advantages and disadvantages of Real-time data ingestion?
  - What are the advantages and disadvantages of Batch-based data ingestion?
  - What are the advantages and disadvantages of Lambda architecture-based data ingestion?

**Subject 3 - What is Data Storage?**

- How do you define data storage?
- Are you familiar with Object, File, and Block Storage?
  - What is Object Storage?
  - What are the advantages and disadvantages of Object Storage?
  - What is File Storage?
  - What are the advantages and disadvantages of File Storage?
  - What is Block Storage?
  - What are the advantages and disadvantages of Block Storage?

**Subject 4 - What is Data Discovery?**

- How do you define Data Discovery?
- Why do we need metadata?
- What are the possible ways to query data from data spaces today?
- What possible ways to query/get data from external actors are there today?
- Why are the possible ways to query data today insufficient?
- What reasonable expectations can we have of external actors?
  - Can we expect them to provide Schemas/APIs for their data?
  - Can we expect them to allow us to retrieve their data periodically?
  - Can we expect them to let us read their data on demand?
  - Can we expect them to conform their schemas to a specific standard?

**Subject 5 - Centralized Data Query Platform!**

- Would it be possible to define generic data interfaces with YAML?
- Would it be possible to extend the generic data interfaces with YAML?
- Would it be possible to generate clients from the extended data interfaces?
- Would it be interesting to build a platform where generic data interfaces can be shared, implemented, and published to expose data?
- What would be the advantages and disadvantages of such a platform?
- What would be essential to consider when building such a platform?

## I.2 Open Coding

Disclaimer - Nikolai Emil Damm has translated all transcripts from Danish to English. In this process, some alterations to the transcripts have been made to clarify the context.

Table I.1: Open coding of Bjørn Therkelsen's statements in Interview 02 Part 1.

| ID | Time | Code | Statement |
|---|---|---|---|
| 1 | 00:16 | Introduction | I am Bjørn Therkelsen, I am an Enterprise Information Architect at Energinet in the department **IDA**. |
| 2 | 00:31 | Introduction | We handle how we transact data between microservices, how we store data, and how to save data across **Operational Technology (OT)**) and **IT**. |
| 3 | 01:15 | Data Spaces | Data spaces, for me, are of an indistinct size. |
| 4 | 01:25 | Data Spaces | What I read is that it is exchanging of data with a fixed format, which makes sense for a flexible energy market |
| 5 | 01:45 | Data Spaces | When we talk about data spaces internally, it is still very fluffy **EU** text about how we share data. |
| 6 | 01:53 | Data Spaces | So it must be about the format of data, how we share data, and how we make data discoverable, especially how we enable data to make intelligent sense. |
| 7 | 02:28 | DataHub | Energinet has a product (DataHub) that shares data and contracts. |
| 8 | 02:50 | DataHub | Works with APIs and a web platform. |
| 9 | 02:58 | APIs | The technologies an API is created with today will not support the amount of data we will have to exchange in the future. |
| 10 | 03:10 | Technology | We have to look into technologies that can exchange data more appropriately. |
| 11 | 03:18 | Data Mesh | We expose APIs for every single Data Product, that we can gather and make queryable. |
| 12 | 03:35 | Data Mesh | A data product can query another data product. |
| 13 | 03:37 | Data Mesh Context | If the data product is at an internal or external partner does not matter as we are part of the same data ecosystem. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 14 | 03:44 | Data Mesh Context | Many technologies do not support the right vision of a data mesh, and if they exist, they are emerging. |
| 15 | 03:59 | Data Lake | In a data lake, we typically either save data in ORC, Avro, or Parquet, depending on whether it should be real-time or compressed. |
| 16 | 04:12 | Avro | Avro is a good format for real-time data, as you can query on Avro immediately. |
| 17 | 04:16 | Avro | Avro is not good for range queries; for this, we should use Parquet. |
| 18 | 04:54 | Spark | If we need to query with Spark on a distributed data set stored in files, how can we then get the performance we know from a classic SQL server? |
| 19 | 05:04 | Introduction | I come from a classic CHR; I am used to indexes, and that I can tune my things. |
| 20 | 06:00 | General | Data Warehouse, Data Lake, Data LakeHouse, and data mesh are technologies and terms on top of other technologies. |
| 21 | 06:09 | Data Lake | A Data Lake is an implementation of a Hadoop cluster with many types of files that have been made analytically queryable. |
| 22 | 06:23 | Data Spaces | A Data Lake is not a data space. |
| 23 | 06:50 | Azure Data Lake | Azure Data Lake is an implementation of a Hadoop cluster that has been marketing branded and put inside Azure. |
| 24 | 07:10 | Data Lake | Ten years ago, a data lake was described as a Hadoop cluster where we can throw things into. |
| 25 | 07:17 | Data Swamp | A data lake can turn into a data swamp if there is not implemented a proper metadata catalog on top of it because then the data is not discoverable. |
| 26 | 07:25 | Data Lake | A Data Lake is a storage object with analytical query capabilities. |
| 27 | 07:40 | Data Warehouse | A Data Warehouse is classical **Business Intelligence (BI)** and stems from the 1960s. It is a model where you have remodeled data, so it is easy to query for query tools. |
| 28 | 07:52 | Data Warehouse | A Data Warehouse can be anything where data is remodelled to the First Normal Form or more. |
| 29 | 08:35 | Relational Databases | A relational database is typically made somewhere between the Third and the Fifth Normal Form. |
| 30 | 08:40 | Data Warehouse | Data Warehouses are typically on the First or Second Normal Form. |
| 31 | 08:45 | Normal Forms | At the First and Second Normal Form, we have denormalized data, but we still have relations between data, where we have collapsed the data into dimensions and facts. |
| 32 | 08:52 | Data Warehouse | They are good at answering business-centric questions. |
| 33 | 09:10 | Data Warehouse | An Enterprise Data Warehouse is a Data Warehouse where different data has been gathered to answer more **BI** questions with conformed dimensions. |
| 34 | 09:17 | General | We have used the last 30 years on Data Warehouses. |
| 35 | 09:20 | Introduction | I have virtually not done anything else in my professional life concerning **BI** |
| 36 | 09:28 | Data Warehouse | We do it for performance and because it has been hard for users to grasp the terms related to data. |
| 37 | 09:42 | Data Warehouse | The problem is Data Warehouses support classical business users; it does not support data scientists. |
| 38 | 09:48 | General | There is conflict right now about whether we should use domain models or data vaults. |
| 39 | 10:00 | Domain Model | A domain model is a model at the Third Normal Form that describes a specific domain. |
| 40 | 10:15 | Domain Model | We have source systems which we populate our domain models from. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 41 | 10:20 | Data Mart | We can create Data Marts on top of domain models. |
| 42 | 10:37 | Domain Model | External actors can populate domain models if they own a part of the domain. |
| 43 | 12:02 | Domain Model | Domain models are important for Energinet as Energinet is regulated by **EU**, both concerning gas and electricity. |
| 44 | 12:14 | Domain Model | Energinet has domain models on the electro-technical, the gas-technical, and the information-specific aspects. |
| 45 | 12:42 | Data Scientists | Data Scientists are not happy with domain models, they prefer data vault modelling. |
| 46 | 13:40 | Data Vault | Take the original data models, place them next to each other, and give each entry a key. |
| 47 | 14:10 | Data Scientists | With Data Vaults, you do not change the source models, and data scientists prefer pure data. |
| 48 | 14:18 | Domain Model<br>Data Vault | We want to utilize domain models as these make more sense to the business users, plus we then have the mapping at the source owners. |
| 49 | 14:28 | Domain Model<br>Data Vault | Enabling source owners to define the mapping gives us a pseudo data vault as we do not change the source models. |
| 50 | 14:32 | Domain Model | We always save or plan to save a historical copy of the source data which we then map into the domain model. |
| 51 | 14:48 | Domain Model | We built query models on top of the domain models. |
| 52 | 15:56 | Data Lakehouse | A Data Lakehouse creates a query layer with business entities, that hide the complexity in the data lake system. |
| 53 | 18:00 | Stock Exchange | Energinet is also a stock exchange. Every day, producers report how much they can produce, and actors report how much they expect to use. The info is combined to provide the best prices for the customers. |
| 54 | 18:33 | Stock Exchange | We get the offers for exchange through XML, but depending on where in the process of exchange we are, the XML file must be exposed differently, until the mapping of production and consumption is complete. |
| 55 | 18:50 | Stock Exchange<br>Data Lakehouse | Because of the process for stock exchange it makes sense not to alter data, but instead store it and create query models on top of it. |
| 56 | 19:05 | Technology | We use whatever technology that matches our needs. |
| 57 | 19:17 | Data Mesh | It is a contract between two parties, so what you want as a consumer, I should be able to expose as a provider. |
| 58 | 19:31 | Data Mesh | Here, you should probably be a bit moderate and define the possibilities that you want to provide for, e.g., APIs and streams. |
| 59 | 19:49 | General | The framework for transactions between parties should be defined in data contracts. |
| 60 | 20:40 | Business User<br>Data Scientist | A business user often has repeating query patterns on what they want to be provided, whereas data scientists do not. |
| 61 | 20:50 | Business User<br>Data Scientist | A business user uses query models to access data, where a data scientist can access data from the domain model and/or the raw data. |
| 62 | 21:07 | Incubator Area | We want to build an area where data scientists can ingest data, analyze it, and operationalize it. |
| 63 | 22:00 | Landing Zones | When we ingest data, it lands in a landing zone, no one will ever get access to this, as this is the raw data. |
| 64 | 22:12 | General | We will expose data in a curated manner. |
| 65 | 22:38 | Delta Lake | Delta Lake has Azure control and temporal support. Control is that we have control over our insert, update and delete. Temporal support is that we save how a record looked previously, and instead of changing the original, we add a new record. |
| 66 | 23:34 | Temporal Storage | It can be hefty on storage demands. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 67 | 23:45 | General | I want to save all our data points. We have 870.000 data points in electricity and gas production, and we want to save from these every second, and in some cases, every milli-second. |
| 68 | 24:26 | General | The frequency is high, and we save in peta-, hexa-, and soon yotta-byte. |
| 69 | 25:00 | Data Mesh | Data mesh is a collection of existing technologies. |
| 70 | 25:06 | Data Mesh | The reason I got fond of data mesh is that it combines some of the best practices from the world of programming and the data world. |
| 71 | 25:32 | Data Mesh | It provides a model to solve some of the shortcomings in the **BI** world, e.g., data ownership. |
| 72 | 25:50 | Data Mesh | Fundamentally, data mesh moves back to the Data Warehouse, as a part of the Data Product. |
| 73 | 26:00 | Data Mesh | We have a data product and an operational product, and they are part of the same. |
| 74 | 26:10 | General | Compared to today, we have operational systems and analytical systems, and these two worlds have nothing to do with each other. |
| 75 | 26:20 | General | When an operational system needs to report something to an analytical system, data is thrown over the fence, with no regard to problems this may cause. |
| 76 | 26:43 | General | We want to combine the operational and analytical systems into one system. |
| 77 | 27:00 | Data Mesh | Data mesh is a cultural change more than it is a technical change. |
| 78 | 27:20 | Data Mesh | Data products should, via endpoints, expose data that can be consumed. |
| 79 | 27:35 | Data Mesh | The data mesh Experience takes our **Site Reliability Engineering (SRE)** area, and our data catalogue and blends them to provide insights into our data and operational quality. |
| 80 | 27:56 | Data Mesh | The quality of a data product falls if it is not properly maintained. |
| 81 | 28:18 | Data Mesh | Data quality is whether we can deliver in time (e.g., data surveillance) |
| 82 | 28:27 | Data Mesh | The social aspect to data mesh is about whether we have the skills required to execute the changes we want. |
| 83 | 28:33 | Data Mesh | With data mesh, we have data products, their descriptions, whether they provide what they are supposed to, and if the data quality is sufficient. From this, we can create trust. |
| 84 | 28:53 | Data Mesh | If you provide data to me, then I tend not to trust you, so I make my data quality checks because I need to be certain that you deliver quality. If we have trust, if we have a contract that defines the data that you deliver, then I trust you. |
| 85 | 29:11 | Data Mesh | If I get an error on my end, then I report that error back to you because I am certain that it is not on my end. |
| 86 | 29:39 | Contracts | We have always had contracts for data, but never as strict as data meshes define them. |
| 87 | 30:00 | Data Mesh | Data mesh describes the vision of what we want, but very little on how to get there. |
| 88 | 31:07 | Ingestion | The lambda architecture is about recognizing data comes as a stream or as a batch load. |
| 89 | 31:23 | Ingestion | When we talk about **Extract, Transform and Load (ETL)** and **Extract, Load and Transform (ELT)**, it is the same, there is no difference, only where we place our compute layer. |
| 90 | 31:37 | Ingestion | With **ETL**, we ingest data to a compute layer, and send them back to a storage layer. |
| 91 | 32:16 | Ingestion | With **ELT**, we extract data to the storage layer and compute from there. |
| 92 | 32:46 | Ingestion | With the cloud, we have one unified storage, a lot of computing, fast reading from storage, and high **Input/Output (IO)**. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 93 | 33:20 | Microsoft Roadmap | Microsoft is working on simplifying their storage engine, by combining all their storage engines into one |
| 94 | 33:00 | Microsoft Roadmap | They talk about their storage engine being a data lake with Parquet files, where data is exposed via Spark, **SQL** or RDX G5 DBaaS. |
| 95 | 34:42 | Microsoft Roadmap | With three compute engines, e.g., Spark, Parquet, and a real-time compute engine, you cover about 97%-98% of use-cases. |
| 96 | 35:04 | Microsoft Roadmap | I have seen drawings that indicate they also contemplate NoSQL databases. With that in mind, I am not aware of there being any engines that they do not support. |
| 97 | 36:12 | Problem | Providing possibilities to do the same thing in different ways, can be a problem as it adds complexity and needs better administration. |
| 98 | 37:20 | Actors | There are different actors. We have the old actors **TSOs**'s and **DSOs**, and the work that happens between these is regulated by **EU** and conveyances according to the **International Electrotechnical Commission (IEC)** 61-850 standards. |
| 99 | 37:55 | ENTSO-E | ENTSO-E is responsible for the conveyances that define the work between **TSOs** and **DSOs**. |
| 100 | 38:01 | S-557 | There is a Danish work unit that is responsible for the S-557. They all work under the **IEC** 61-850 standard. |
| 101 | 38:40 | Actors | All the old actors are used to communicate with well-defined data exchange methods. |
| 102 | 39:21 | Actors | Work between the professional actors is fully handled today, but all the decentralized actors are not handled at all. |
| 103 | 39:43 | Actors | It is exactly here that we are fucked; there is no integration to decentralized systems. |
| 104 | 40:43 | Actors | Some of the decentralized data is at the actors, and some of it is in the DataHub, but the data is lacking, and we need to be able to get this data faster. |
| 105 | 42:00 | Problem | The question is how can we create an incitement structure and a communication protocol that allows decentralized partners to act according to what is best for the grid. |
| 106 | 43:42 | Actors | We have data exchange somewhat under control in the north, but we also need to have this under control in the rest of the **EU**. However, it is the private actors that are the issue. |
| 107 | 45:23 | Vision | Where we want to get to is that the systems (if developed by us) should expose their data products through a YAML file, so whenever we create a new version of our code, we must also update our YAML file to tell how the data catalog looks and how we communicate internally. |
| 108 | 45:55 | DataHub | We make the data products discoverable in the DataHub through our data catalogue. |
| 109 | 46:17 | Vision | What you say is that I, as a data producer, should expose the data you require, and that is the wild version, because there is a lot that plays into if this is even possible. |
| 110 | 46:37 | Vision | It is a fascinating mindset because if we have our domain model, which could be exposed, then we have a description of what we want within a domain. |
| 111 | 47:16 | Vision | If you have made a domain model in which you can select then, you can define this model, and then the easiest would be to discuss what structure the data should take. |
| 112 | 48:26 | Vision | A contract is always between two parties, so you might be able to expose your domain model, and then the other party can decide if they want to provide the data. |
| 113 | 49:22 | Vision | I think it would be cool if we can make these data available on these protocols, but that they do not need to replicate it to themselves, such that these data spaces could be data lakes that could be used by others in a safe manner. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 114 | 50:25 | Vision | I want to say in this context that it is an entity that owns a data set. |
| 115 | 51:13 | Open-source | Open-source on three levels, design, code, and at an application level. Jakob and I have different opinions here. Jakob believes it should be on every layer; I think you should think about how many layers it makes sense to make it open-source. |

Table I.2: Open coding of Bjørn Therkelsen's statements in Interview 02 Part 2.

| ID | Time | Code | Statement |
|---|---|---|---|
| 1 | 00:55 | Open-source | From a perspective of progress, it might be a good idea to consider open-sourcing the solutions, while still using proprietary tools, technologies, and programming languages that are easy to understand, to build the solutions. |
| 2 | 01:00 | Open-source | Open-source enables others to learn from the solutions. |
| 3 | 01:10 | DataHub | We are required to do this on the data platform, where we do open-source on the design and application level. |
| 4 | 04:40 | Data structure | Can you ignore normal forms in a contract? |
| 5 | 05:53 | Data Structure | It should be up to the customer what normal form data should have. |
| 6 | 06:43 | Data structure | Normal forms and how data is stored is not relevant when ingesting or consuming data. |
| 7 | 07:53 | Data Mesh | If you and I trust each other, then you can read my data catalog, and I can read yours. |
| 8 | 08:30 | Data model | A data model internal in a data domain is yours and only yours, and then you can choose to expose it as you want. |
| 9 | 12:50 | Zones | Another interesting thing is how to make data sets available in multiple zones. |
| 10 | 13:25 | Governance | The customer should be able to control which zones their data is available in. |
| 11 | 13:43 | Governance | From a cosmetic perspective, I do not want a data set available in a zone where it is not used, but I also want to be able to distribute data sets to the zones closest to the customer if allowed. |
| 12 | 18:45 | Bicep | Microsoft keeps new features to Bicep exclusive just long enough that new features are not generally available to third-party **Infrastructure as Code (IaC)** tools. |
| 13 | 27:10 | Data discovery | You need an overview of what data product you have, who uses them, what contracts are there, and more. |
| 14 | 28:12 | Data Mesh | We need a structure and a metadata description, so we can understand them and enrich them in ways that provide us with new abilities. |

# J | Interview 03 with André Bryde Alnor

## J.1 Interview Guide

**Type:** Semi-Structured Interview
**Method:** Grounded Theory
**Date:** 2022-10-03
**Interviewer:** Nikolai Emil Damm
**Interviewee:** André Bryde Alnor

**Subject 1 - Energinet**

- What is the role of a **TSO**?
- How does Energinet fulfill the role as a **TSO**?
- Does Energinet have any other roles?
- Has Energinet always had these roles?
- How does Energinet promote sustainable energy?

**Subject 2 - Data**

- What is the role of data in Energinet?
- What kind of data does Energinet rely on?
- What kind of data does Energinet produce?
- What kind of data does Energinet provide?
- What is sensitive data?

**Subject 3 - Processes and procedures for collaboration**

- What processes and procedures does Energinet have in place today? Collaboration from the broad perspective to the narrow.
- Are any of these processes and procedures not digitalized?
- Can some of these processes and procedures be digitalized?
- What role does digital maturity play concerning collaboration?
- How do you work with other actors that have a low digital maturity?
- How do you work with other actors that have a high digital maturity?
- How do you motivate other actors to increase their digital maturity?

**Subject 4 - Ethics, privacy, and legal regulations**

- What is the role of ethics in Energinet?
- Are there any ethical dilemmas that Energinet faces today?
- Is it ok to save data that is not yours? Under what circumstances can this be allowed?
- If you provide a data contract to a third party, how do you ensure that they will not misuse the data?
- Do you need to own data? Can data be held and provided by a third party?
- Can you rely on data that is not yours, and if so, under what circumstances?

**Subject 5 - Data Mesh and Data Spaces**

- What is a data space? What is the initiative behind it?
- Are you familiar with the concept of a data mesh?

- Do you think a data mesh fits well with the **DSI**?
- What benefits and challenges would a data mesh present for Energinet?

## J.2 Open Coding

Disclaimer - Nikolai Emil Damm has translated all transcripts from Danish to English. In this process, some alterations to the transcripts have been made to clarify the context.

Table J.1: Open coding of André Bryde Alnor's statements in Interview 03.

| ID | Time | Code | Statement |
|---|---|---|---|
| 1 | 00:27 | Introduction | My name is André. I am a department manager in digitalization at Energinet. |
| 2 | 00:53 | General | Energinet is not a state-owned monopoly but an independent public enterprise energy supply company. |
| 3 | 01:05 | General | We look into all the things that need to be done to solve the green transition. |
| 4 | 01:07 | Digital Maturity | A big part of the digital infrastructure that must tie the energy sector together is missing, e.g., component control (both centralized and decentralized), orchestration between actors, settlement, calculation of tariffs and calculation of taxation. |
| 5 | 01:52 | General | If we must tie together the energy sector in such a way that it can have dependencies to each other and enable decentralized control, then we must think more in transaction-based communication, that is disconnected from manual processes. |
| 6 | 02:08 | Data Spaces | In a European context, the **DSI** is focused on being an alternative to big data companies owning data, such that companies can trade data. |
| 7 | 02:40 | Data Spaces | We see it as a digital infrastructure that allows actors to identify, integrate, and share data, with or without money being involved. |
| 8 | 02:51 | Data Spaces | Energinet is dependent on the **DSI** to succeed to be able to tie together the energy sector and to succeed with Energinet's priorities for the energy system. |
| 9 | 03:33 | **TSO** | The role of a **TSO** is the system operation responsibility of balancing both the electrical and gas grids and the transmission operation responsibility, meaning, we must build the infrastructure to transport the energy and gas nationally. Energinet does not maintain the infrastructure to transport energy and gas in local municipalities; this is the responsibility of distribution companies. |
| 10 | 05:05 | Physical Infrastructure | The infrastructure that supports the tasks of the TSO primarily consists of many electrical wires under or above ground, many gas pipes, and few gas storages. |
| 11 | 05:23 | General | One thing that is a big issue is that we are reorganizing from an energy infrastructure with many big central power plants to an energy infrastructure running solely on green energy. |
| 12 | 05:48 | General | Wind energy and solar energy are, funnily enough, only produced when the wind blows or the sun shines, so the task of ensuring energy is available at all times is much harder. We can solve this by ensuring there are other forms of energy available, e.g., bio-gas or power-to-x. |
| 13 | 06:20 | Power-to-x | Power-to-x is the process of using electrical power as one of the primary sources to produce other forms of energy that are better suited to be stored or transported, e.g., hydrogen or ethanol. |
| 14 | 06:43 | **TSO** | Everything Energinet does is under the role of the TSO. |
| 15 | 06:49 | Energinet | Energinet is also responsible for the design of the energy system, such that it is coherent, it is worth paying for, it is climate-friendly, and it creates maximal value for the society. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 16 | 07:06 | Energinet | Energinet is responsible for the design of Denmark's markets for energy trade, and Energinet also plays a role with all 27 European countries in designing markets for energy trade in a European context. |
| 17 | 07:18 | Energinet | Energinet is involved in legislation in a national and European context. |
| 19 | 07:27 | Energinet | We refer to the **DEA**, which is the agency responsible for establishing the legislation, and Energinet is then responsible for carrying out the law within the **TSO** domain. |
| 20 | 08:16 | Energinet | Energinet promote sustainability well. Denmark is one of the countries at the forefront of the green transition, but much digital maturity is missing within the energy industry to bring Energinet closer to the **SDGs**. |
| 21 | 09:37 | General | There is not an energy sector that has been involved in the **IoT** 4.0 perspective as the industry has been. Likewise, Energinet has dependencies on the existing energy system that hinders progress. |
| 22 | 10:11 | General | In the energy sector, we must be careful implementing changes, but sometimes this can mean resistance to change, which in the larger picture has set the energy sector behind other industries. |
| 23 | 10:50 | General | We can build a lot of new physical infrastructures, but it does not provide much value if we do not become better at orchestrating our energy system. |
| 24 | 12:08 | General | No matter what initiatives we take, we still have a 40-year-ish old infrastructure that we must integrate with. |
| 25 | 13:12 | Data | There is a huge amount of data to ensure that units, systems, and persons, can communicate and trust each other. |
| 26 | 13:35 | Legacy | Energinet needs new standards to be implemented so systems can understand each other or software solutions are capable of translating legacy procedures and standards to newer procedures and standards. |
| 27 | 14:10 | Data | Data that is gathered from sensors is not the same as the data that is used in a business context. |
| 28 | 15:35 | Domain Models | One should be careful not to enforce domain models through a value chain. The market context does not correlate well with the technical context. |
| 29 | 15:50 | Data | I think the most important part for me about data is to think of data in a decentralized context. We cannot just think of one huge domain model and enforce everything into it, but we must look at each separate system such that it can work autonomously, with a generally well-defined interface. |
| 30 | 16:45 | Hosting | Systems should not be hosted decentralized, as the amount of information gets so big, that it does not make sense to have some central register that stores the data, especially concerning sensed data; sensed data should be local. |
| 31 | 17:25 | Metadata | Metadata is data that says something about other data. |
| 32 | 18:15 | Metadata | Metadata allows us to define what facility we are talking about quickly. |
| 33 | 18:38 | Metadata | We use metadata for searching functionality but also to establish trust. |
| 34 | 19:13 | Metadata | Where metadata should be stored is dependent on a lot of things, like usage, complexity, and price. |
| 35 | 20:48 | Vision | We need to work towards a place where everything works in a decentralized system. |
| 36 | 22:28 | Vision | A decentralized system will allow users to communicate independently of each other. |
| 37 | 24:04 | Decentralised Systems | A lot of elements are missing to establish a decentralized system; I think we will end up with one, but I also think there is a lot of prerequisites, where we must first combine and use existing central registers to create trust. |
| 38 | 27:01 | Energinet | In a European context, we work a lot in the regulatory domain, which means that we help define required changes to meet new **EU** regulations. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 39 | 28:50 | Energinet | Energinet collaborates with other TSOs in the **EU**, which harmonize with European legislation. |
| 41 | 29:15 | Energinet | The collaboration processes within the **EU** are digitalized; however, the processes are in some cases dated, e.g., communication by email, ingestion of CSV files, no focus on decentralization. |
| 42 | 30:08 | Centralisation | When you think of centralization in the context of the energy sector, where you have approximately 450 million sensors in a European context, then you have a huge need to reduce complexity. In some cases, this can mean that we let the lowest common denominator be the deciding factor, which is not necessarily good. |
| 43 | 30:38 | Centralisation | When you need a system that is flexible and can handle differences and local contexts in a European context, it is almost impossible with a centralized approach. |
| 44 | 31:25 | Centralisation | When you have a technological world that is changing as fast as it does, then you need a very different market setup. |
| 45 | 31:40 | Centralisation | There is huge collaboration in the European context on the market models that have proved to work well but that we can now see beginning to falter. Collaboration on how to accommodate a decentralized approach is missing. |
| 46 | 32:27 | Context | There is no reasonable solution on how to create a generic solution in the European context. |
| 47 | 32:28 | Context | Today, it is accepted that incremental development is the way to go. Technical standards implement new developments; it can take 10-15 years to create a new standard and another 10-15 years to implement it. |
| 48 | 33:27 | Context | We need the functionality that comes with new technology. We are at a place where we need it so much that the distance between research and proof of concept must be minimal to enable us to take advantage of new functionality as fast as possible. |
| 49 | 33:57 | Context | We cannot let legislation come first, but we need to let technology guide us and use learning to solve issues in the energy sector. At that point, we can begin to legislate. |
| 50 | 35:19 | Standards | Typically, standards are created incrementally in global contexts. When a standard gets to a point where it is broadly acknowledged, it will be published as an **IEC** standard for broad usage. At that point, the **EU** commission or national regulation can select the standard and use it as a basis for regulation. |
| 51 | 36:21 | Standards | Sometimes standards will be defined further in a local context, but typically, this is not the case. |
| 52 | 37:04 | Standards | The maturity for implementing standards is not high. It is assumed that operators or developers creating systems have thought of standards. Likewise, it is assumed that suppliers live up to required standards. |
| 53 | 37:37 | Standards | There is no consequence for not following standards other than the lack of advantages implementing a standard might contribute. There are many standards in the energy sector that are not followed or implemented improperly. |
| 54 | 38:36 | Standards | If the law states that you must implement a standard, it is with a high leeway. It is often defined in terms that can be interpreted in a local context. |
| 55 | 39:50 | Standards | The flexibility towards regulated standards is both good and bad. It is good because it is near impossible to enforce one standard in a multitude of contexts. On the other hand, it is a problem in that there is no consistency in domains. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 56 | 41:24 | Software | We can use software to move complexity from manual processes by transferring processes into software that can simplify and abstract the process. Such an approach can allow us to better integrate where we differ. |
| 57 | 43:40 | Software | If you have a lot of features in the software, users might begin to misuse your software, and create a mess. Likewise, with many features, each feature needs to be maintained, and this can increase the cost. |
| 58 | 44:01 | Software | It is important to decide what to implement and what not to implement, such that we bridge in collaboration in meaningful ways that simplify processes. |
| 59 | 44:41 | Data Spaces | Energinet puts a lot of focus on increasing digital maturity in the energy sector. The **DSI** is one of the projects aiming to improve digital maturity in the supply and energy sector. |
| 60 | 45:14 | Data Spaces | The **DSI** is about determining what digital infrastructure is needed to enhance the energy and supply sector now and in the future. |
| 61 | 45:37 | Data Spaces | The **DSI** started with a lot of partners to try to make it part of the national digitalization strategy, and thus a political assignment. Now we are working on the harder part of making it a reality. |
| 62 | 46:00 | Data Spaces | The main aims of data spaces are to convert manual or legacy processes to smarter processes, where the focus is on decentralization, streaming, automatization, and integrability. |
| 63 | 46:50 | Data Spaces | There is a European understanding of what data spaces are, and then there are Energinet's understanding of what it should be. |
| 64 | 47:34 | Data Lake / Data Mesh | Data Lakes is about ensuring we can store a huge amount of data, where a data mesh is concerned with metadata, so we do not need to store all the data at a central place, but that we can find it when we need it. |
| 65 | 48:16 | Data Spaces | With a data space, I think of it as a data ecosystem, where data is stored decentralized. Then we have keys (protected by the data owner) to determine the structure of data and who can access it, among other things. |
| 66 | 49:52 | Key | A key in this context is just a hash; it is needed to authorise and authenticate oneself with services. |
| 67 | 51:20 | Contracts | Contract-based trade is not just a term when trading in a market but is a term for when two parties must communicate. |
| 68 | 51:47 | Data | Data rarely has value if it is not used together with other systems or other data. |
| 69 | 52:16 | Metadata | Using metadata in a data mesh, as data that we can trust and use, is what for makes it extremely complex. |

# K | Interview 04 with Peter Lyck Ingerslev

## K.1 Interview Guide

**Type:** Unstructured Interview
**Method:** Grounded Theory
**Date:** 2022-10-03
**Interviewer:** Nikolai Emil Damm
**Interviewee:** Peter Lyck Ingerslev

For this interview, I have decided to choose an unstructured approach, as I know Peter Lyck Ingerslev is a very experienced person that can tell me a lot based on little context.

**The context:**

- Data mesh concerning the **DSI**
- Focus on Data Storage
- Data Discovery
- Availability
- Performance
- Real-time vs. batch, and when to use which

## K.2 Open Coding

Disclaimer - Nikolai Emil Damm has translated all transcripts from Danish to English. In this process, some alterations to the transcripts have been made to clarify the context.

Table K.1: Open coding of Peter Lyck Ingerslev's statements in Interview 04.

| ID | Time | Code | Statement |
|---|---|---|---|
| 1 | 00:58 | Introduction | My name is Peter Ingerslev. I am the chief principal architect for Energinet's digitalization and innovation initiative. |
| 2 | 01:21 | Introduction | My angle of entry is to make technology apply through data. |
| 3 | 01:29 | Introduction | I have worked with data most of my life in all kinds of facets. Everything from case management systems, the state, the telecom sector, and fiber network; I have pretty much been allowed to work with what I want, and now I have clashed with Jakob to work on the green transition. |
| 4 | 01:54 | Green transition | The green transition presents unique challenges, and the challenges are diverse. Our energy infrastructure has been designed around a monoculture, meaning it is generally thought of as separate pillars and sectors and only optimized within its own domain. |
| 5 | 02:30 | Context | The green transition is in dispute with the monoculture, as it requires more diversity in the energy sector. |
| 6 | 02:37 | History | In old times, we built large coal factories and ensured coal, oil, or gas was delivered to ensure the supply chain. Fossil fuels allow us to store huge amounts of resources for later production needs. We are not in this situation today. We want to get rid of fossil fuels. |
| 7 | 03:11 | Context | From all the energy Denmark uses today, only 20% is electricity. The rest is from combustible substances. 8% is liquid or gas-based, and this 8% stems from 40% of the 20% electricity. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 8 | 03:47 | Context | The more sustainable energy we produce in our energy system, the more fluctuating it becomes. The closer to 100% sustainable energy we get, the more stochastic the energy systems act. |
| 9 | 04:23 | Context | A 680 Mega Watt solar cell factory can deliver 90% one minute, three minutes later it might deliver 40%, and 30 seconds later it might deliver 100%. Clouds obstruct the sun's rays. |
| 10 | 05:15 | Context | Our foreign connections are not worth much in these crisis times because each nation focuses on its wellbeing. |
| 11 | 06:23 | Context | We cannot use Himmelbjerget because it is not a mountain, and we cannot use old coal mines as reverse waterfalls. So what can we do? We can increase the amount of available information. |
| 12 | 06:48 | Context | The Danish energy sector bets on two racehorses sector coupling and flexibility. |
| 13 | 06:53 | Context | We do not have nuclear or coal power plants, as we demolish them. Luckily we have not demolished them all, so we can start some of them up again. And we do that! |
| 14 | 07:27 | Context | Sector coupling and flexibility require the sectors to be connected with information. |
| 15 | 07:41 | Context | The closer we get to 100% sustainable energy, the more information we need. |
| 16 | 07:48 | Context | My gut tells me that if we double the amount of sustainable energy, then we quadruple the fluctuation in the energy system. Still, we need 8x times as much information, so it is a factor of 32. |
| 17 | 08:10 | Energy Systems | The classical energy system is very hierarchical, with the consumers at the bottom and the markets, the actors, and the TSO at the top. The consumers demand something, and the market responds with production. It is a vertical implementation within one sector, e.g., electricity or gas. |
| 19 | 08:39 | Energy Systems | When we talk about energy systems, we also talk about district heating, water, and more. |
| 20 | 09:08 | Context | At some point, we will recognize that we cannot transfer information fast |
| 20 | 09:08 | Context | enough in the current market design. |
| 21 | 09:20 | Context | The closer we get to 100% sustainable energy unless we invent the fusion reactor during 2023, we will have to scale out to meet the information transfer demands. |
| 22 | 09:36 | Context | The energy system must become diverse; it has to become more complex. Adding complexity is a good idea because the more diverse an energy system is, the fewer larger parts can fail. |
| 23 | 10:41 | Context | When corona hit, Energinet sent 5000 people home, but the system did not crash. |
| 24 | 10:58 | Context | A stable complex system is notoriously stable. |
| 25 | 11:20 | Complex system | We migrate from the complex systems, which is a just-in-time based system. |
| 26 | 11:35 | Complex system | A just-in-time-based system works by creating forecasts that allow us to foresee how future scenarios might look. |
| 27 | 12:25 | Context | Because of the energy crisis and the high prices right now, we get flexibility. Denmark uses approximately 13% less electricity this year and moves their electricity usage to times of the day when the prices are lower. |
| 28 | 13:14 | Context | If a kilowatt-hour only costs 12 cents, what if then save that power with a battery that loses 10%, and then use it in the evening when the prices are 8 kroner. It requires information to be available to make these forecasts. |
| 29 | 13:40 | Context | The new reality of our energy systems are all about increasing the amount of information available, e.g., the amount, quality, availability, and diversity. |
| 30 | 14:57 | Context | I think we will have to include information in our forecasts that we have never dreamed of using before. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 31 | 15:04 | Vision | In a complex world, we will have to create simulations from models that we train with an individual household or company to be able to provide the insights we need. Then we might have 3.3 million simulations running. |
| 32 | 15:53 | Context | It costs 52% energy loss to convert electricity to synthetical methane. It means we do power-to-x, but first, we convert water to hydrogen, then hydrogen to ethane, and then ethane to methane. Why is methane, then, interesting? Methane is our gas net, and we can move enormous amounts of energy with our existing gas net. Because of the prices, this can be very advantageous. |
| 33 | 17:42 | Context | We need more information to be able to know if we have surplus electricity. We can market surplus electricity and convert it no matter the conversion rate. It allows us to store the electricity, which has a high probability of being worth it considering the savings and fluctuations. |
| 34 | 19:30 | Context | Danish gas storage can provide energy for approximately a year. |
| 35 | 19:48 | Context | 10 Gigawatt which our gas system can move, corresponds to seventeen 400 kilovolts wiring harness. |
| 36 | 20:06 | Context | Most of Denmark's electricity is consumed in Zealand but is produced in west Jutland. So I do not think Funen would like that we create seventeen 400 kilovolts wiring harness that traverses over Funen. |
| 37 | 20:56 | History | In the old days, when we had too much electricity in our energy system, we could downregulate the electricity grid. We could call the producer and tell them we did not need as much, and then they would ask us to pay to downregulate. This meant we ended up paying double for electricity, and to combat that, we installed industrial electric kettles to consume the surplus energy to not pay for downregulation. |
| 38 | 22:54 | Context | Cogeneration plants can use natural gas, oil, wood chips, and electricity, and they can store surplus energy by heating up their district heating systems above what they usually do. It gives them a capacitive characteristic. |
| 39 | 23:34 | Context | 40% of the flexibility we want to achieve, we can get from the consumers. |
| 40 | 25:06 | Context | We could use **IoT** devices or **Long Range Wide Area Network (LoRaWAN)** as a technology to send and receive a price or **$CO_2e$** forecasts to help balance the grid. |
| 41 | 25:48 | Context | The electricity market is planned for 48 hours. |
| 42 | 26:00 | Context | Danish windmills cost society 17 million a year because they produce both when the prices are positive and negative. |
| 43 | 26:26 | Context | Three years ago, we spent 680 million on energy regulation; this year, we will probably spend around 1,3-1,6 billion. |
| 44 | 27:29 | Context | The electricity price you see when you look at your electricity bill consists of a set of subproducts; generation of power, consumption, non-consumption, inertia, frequency stabilizing services. |
| 45 | 29:19 | Context | Electrical vehicles (Lithium batteries) can deliver synthetic inertia well, as they can provide synthetic inertia in microseconds. |
| 46 | 29:46 | Context | Synthetic inertia can be obtained by adjusting the charging power if we need more electricity elsewhere. |
| 47 | 31:30 | Context | Synthetic inertia is about ensuring consumed electricity lasts longer, e.g., with regenerative braking. |
| 48 | 33:36 | Content | The energy system should be shielded on closed networks - it is not accessible for malicious intents. |
| 49 | 33:55 | Context | Energinet could register itself as an **Mobile Virtual Network Operator (MVNO)** to ensure they own a network where it can communicate safely. |
| 50 | 34:15 | Context | Every week, a thousand charging stands are installed in Denmark. These charging stands typically communicate over 4g. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 51 | 35:05 | Context | If we as a state-owned organization registered as a **MVNO** as a charging stand network, and told all telecom operators that it is free to use, and added the data to our DataHub, then we could gain a real-time overview of where electricity is consumed, and then use the real-time data to micro-balance the grid. |
| 52 | 35:57 | Context | Brownouts are if we strategically choose to turn off the electricity grid in some regions to save the grid. |
| 53 | 36:50 | Context | Packetised energy prioritization is a project in England where units publish what their intent is, and a central system then responds if it is possible, and when it is possible. It showed that if a household can introduce 2,5-kilowatt hours in flexibility, then North England can balance their grid with the sustainable energy they have available today. |
| 54 | 38:40 | Context | The need to increase the amount of available information, and because the speed of fiber is 0,61 times the speed of light, and not fast enough to move data; we need decentralization in terms of a data mesh, Data spaces or distributed information setup. |
| 55 | 40:04 | Citation | "It does not matter what speed we communicate with, just as long as the information is available when we need it." - Nicholas Negroponte |
| 56 | 41:16 | Term | For the supply sector, we have a term coined robustness, which means that we must build out infrastructure, so it does not have a single point of failure. |
| 57 | 44:43 | Context | We cannot use non-robust standards; we can use something like **IEC**-61850 instead, as it is robust. |
| 58 | 45:34 | Context | We do not use classical HTTP standards for communication internally, as we need standards that can be used for 30 or 40 years. |
| 59 | 46:38 | Context | In the supply law, we must choose well-established standards, which means something like **American National Standards Institute (ANSI)**, **International Organization for Standardization (ISO)**, or **IEC** standards. If these standards do not exist, then we can choose a de-facto standard. |
| 60 | 50:36 | Context | **ANSI**, **ISO**, and **IEC** standards are used in all parts of the energy sector digital infrastructure. |
| 61 | 50:55 | Context | Every time a 400 kilovolts wiring harness is installed, then we install 96 fibers along with it, it allows us to have a closed network from the public. |
| 62 | 51:27 | Context | We run something called **OT** instead of **IT**. 80% is the same; we use the same computers, we use the same programming languages, and we use the same development tools, but on one point, we differ. In the **IT** world, you use end-to-end encryption in the supply sector, is is contra intuitive. If we encrypt everything in our network, we become blind. |
| 63 | 52:15 | Context | **Confidentiality, Integrity and Availability (CIA)** is a security principle, which means we have to encrypt, secure integrity, and do authentication and authorization. |
| 64 | 52:40 | Context | In the **OT** world, accessibility (we can access data), integrity (data is kept intact), and confidentiality (who can access the data) are the key parameters. |
| 65 | 53:02 | Context | The worst scenario for the energy system is not if someone hacks a station and destroys it, but our worst fear is if someone makes our infrastructure lie. |
| 66 | 54:18 | Context | To enable integrity, we must digitally sign messages instead of encrypting them, so we can make sure the messages are immutable. |
| 67 | 61:31 | Context | It is important to consider how data can be made accessible where it is needed. |

# L | Interview 05 with Jakob Hviid

## L.1 Interview Guide

**Type:** Semi-Structured Interview
**Method:** Grounded Theory
**Date:** 2022-10-24
**Interviewer:** Nikolai Emil Damm
**Interviewee:** Jakob Hviid

**Context**

- Data Spaces
- Data Mesh
- Contract-based provisioning

**Subject 1 - Data Spaces**

- What components are vital for a data space to function?
- Can a data space be considered a data product?
- Can a data space be considered a service? E.g., a service that can access, manipulate and retrieve data from external sources.
- What does it mean for a data space to be decentralized? Who hosts the data space?
- What input and output ports should a data space have? E.g., for data ingestion, data retrieval, and management.
- On what hardware or cloud platform should a data space function?

**Subject 2 - Storage**

- How much data is expected to be stored in a data space?
- Can there be a retention time on data in a data space?
- Where/when should data be stored? In the data space or somewhere else? In-memory vs. on-disk?

**Subject 3 - Data**

- Does a data space require variety in data formats? Can it be limited to data stored in a relational database?
- Is data in a single data space expected to be within a single domain? What defines a domain? When is a domain too large to be stored in a single data space?
- Should/could data spaces support data schemas? If so, what are the requirements for a schema?

**Subject 4 - Metadata**

- What metadata is essential to describe a data space?
- Why is metadata important for data spaces?
- What is the expected outcome of utilizing metadata for data spaces?

**Subject 5 - Data discovery**

- What are the requirements for a data discovery platform?
- How much control should a data discovery platform have over the data in a data space?
- Who should be able to access a data discovery service?

**Subject 6 - Management**

- How are data spaces expected to be managed?
- Who should be able to manage a data space?
- Do you think a file-based approach can manage a data space? E.g. manifest files (YAML).
- Does data space need data migration capabilities? How so?
- How is metadata expected to be managed?
- Who should be able to manage metadata?
- Do you think a file-based approach can add metadata to a data space? E.g. manifest files (YAML).

**Subject 7 - Provisioning**

- What is vital to consider if wanting to automate the creation of data spaces?
- Do you think a file-based approach can provision a data space? E.g. manifest files (YAML).
- What should be included in a contract?
- What should be excluded from a contract?
- Any thoughts on ensuring the provisioning is flexible?

## L.2 Open Coding

Disclaimer - Nikolai Emil Damm has translated all transcripts from Danish to English. In this process, some alterations to the transcripts have been made to clarify the context.

Table L.1: Open coding of Jakob Hviid's statements in Interview 05.

| ID | Time | Code | Statement |
|---|---|---|---|
| 1 | 02:00 | Data Storage | For a data space to function, you need data storage. |
| 2 | 02:17 | Data Storage | The chosen data storage for a data space can vary by domain. |
| 3 | 02:30 | Data Storage | The specific need for data storage in a data space is that it supports real-time data and historical data. |
| 4 | 02:50 | Data Storage | There are some types of data that are better suited for specific data storage solutions. For example, graphs should be stored in a graph database. |
| 5 | 03:20 | Data Storage | In data mesh, the choice of data storage is determined by domain, but in data spaces the choice of data storage is chosen based on business needs. |
| 6 | 03:50 | Ownership | Something interesting to consider is to determine when a data space belongs to a team or an organization. Both can be true. |
| 7 | 04:10 | Provisioning | The more automatized the provisioning process is when considering federation, the more it makes sense to keep data spaces small and owned by individual teams. |
| 8 | 04:40 | Data Spaces | A data space today is more or less data discovery built on top of existing data infrastructure, with interfaces that expose the data to external actors |
| 9 | 05:40 | Data Spaces | If considering scale and interaction on a more decentralized level, a team can create a data space, that can interact with another data space. |
| 10 | 06:00 | Data Spaces | In the software, many components might not be needed for a data space. |
| 11 | 06:10 | Software Components | Some of the components that will repeat are authentication, authorization, the governance layer, data formats, standardized APIs, usage, logging, tracing, and more. |
| 12 | 07:50 | Data Spaces | It is not well-defined how data spaces should be implemented. |
| 13 | 08:00 | Data Access | The interface to data can move from a generic standard/data format to a more specific standard/data format. |
| 14 | 10:25 | Data Spaces | A data space is as a concept so generalised. Many sectors talk about data spaces as data liberation. In other sectors, data spaces are interesting for real-time analyses. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 15 | 12:00 | Data Spaces | In the supply sector, it is a requirement that data spaces are decentralized. In other sectors, it is not required. |
| 16 | 12:40 | Decentralisation | Interoperability between data spaces allows the systems to function as a decentralized system regardless of which nature the data space takes (monolithic vs. service-oriented). For example, a data lake can be part of a by being interoperable with other systems. |
| 17 | 14:30 | Electricity Sector | In the electricity sector, we have **DSOs** and **TSOs**. **TSOs** distribute energy and balance the grid, and **DSOs** distribute the energy to households. |
| 19 | 15:04 | Data Spaces | **DSOs** and **TSOs** want to use data spaces internally, e.g., collecting and analyzing data. Some of this data should be classified, and some can be shared. |
| 20 | 16:08 | Water Sector | The water sector consists of more than a hundred small water factories. If every water factory had a data space, it would be an enormous expenditure because they would need to hire IT personnel. What they would do, is hire an external partner, to create a central data space for the whole sector. |
| 21 | 18:34 | Provisioning | An automatic provisioning system for whole data spaces is more relevant where provisioning must happen often, e.g., smaller teams or where it is decentralized. |
| 22 | 20:00 | Provisioning | There are no requirements to where decentralized data spaces are hosted, but it should be up to the owner. |
| 23 | 20:14 | Data Spaces | Data spaces must adhere to requirements for the governance layer, e.g., provenance, tracing, interoperability and extensibility. |
| 24 | 22:50 | Data Spaces | Data spaces require data storage. |
| 25 | 24:43 | Retention | In some areas, it makes sense to store data with no retention, and in other areas retention must be set. |
| 26 | 24:54 | Data Storage | Most storage systems that exist decouple the compute layer from storage, and many of the storage systems can auto-scale. |
| 27 | 25:14 | Data Storage | It is ok that external storage solutions are used to store data. |
| 28 | 25:29 | Data Storage | There can be a storage tiering problem, where relatively new data is stored in expensive but performant storage and historical data is stored on cheaper and less performant storage. |
| 29 | 26:11 | Policies | Policies should set where data is stored, how long it is stored, and more. |
| 30 | 26:46 | Policies | Policies should be able to trigger, e.g., transferring data when the |
| 30 | 26:46 | Policies | data retention is reached. |
| 31 | 27:09 | Tiering | Data should be moved from slow storage to faster storage when requested. |
| 32 | 27:40 | Tiering | Data tiering is not supported by all storage solutions. |
| 33 | 31:14 | Data Storage | Data is always domain driven, so the data storage solution should be flexible. |
| 34 | 35:40 | General | Only a few scenarios should be selected, but be built with flexibility in mind, such that it is extensible for future iterations. |
| 35 | 37:24 | General | You can build something very versatile, but I do not believe that you can build something that covers all scenarios. |
| 36 | 38:57 | Domains | Domains should be split into sub-domains where it is logical and brings value. |
| 37 | 42:34 | On-the-edge | In on-the-edge scenarios, it can be interesting to deploy a subset of functionality to edge devices. |
| 38 | 43:12 | Ingestion | Data spaces should support both push and pull (e.g., a pub-sub). |
| 39 | 44:20 | Metadata | Metadata is domain-specific and should be configurable. But generally, all datasets should be documented with metadata. For example, ownership, schema, data quality, lineage, context, data access, and glossaries. |
| 40 | 48:20 | Metadata | We must support modern approaches, like auto-provisioning and DataOps. |

| ID | Time | Code | Statement |
|---|---|---|---|
| 41 | 51:10 | Provisioning | Provisioning data products from a YAML-based contract that abstract complexity away from the user have a large value, as the alternative is managing many manifest files to provision infrastructure and code needed to run data spaces. |

# M | Axial Coding of Interviews

To make the Axial Coding tables more concise, a key that combines the interviewee and the code ID is constructed. The key uses the interviewee's initials followed by the code ID. For example, for Jens Hjort Schwee's seventh statement, the key is `JHS7`. The key is named `ID` in the tables.

## M.1 Business Ecosystem

| ID | Sub-category | Statement |
|---|---|---|
| JHS7 | Digitalization | Other companies' digitalisation levels differ from Energinet's. |
| JHS11 | Goals and Interests | Energinet and companies in the different sectors have conflicting interests and goals. |
| JHS13 | Goals and Interests | Energinet's core businesses will to some degree, overlap with other companies' interests. |
| JHS79 | Roles | We need to define actors' roles. There are many definitions of roles, and many of the roles are not well-defined yet. |
| ABA20 | Maturity | Energinet promote sustainability well. Denmark is one of the countries at the forefront of the green transition, but much digital maturity is missing within the energy industry to bring Energinet closer to the **SDGs**. |
| ABA22 | Resistance | In the energy sector, we must be careful implementing changes, but sometimes this can mean a resistance to change, which in the larger picture has set the energy sector behind other industries. |
| ABA24 | General | No matter what initiatives we take, we still have a 40-year-ish old infrastructure that we must integrate with. |
| ABA38 | Regulations | In a European context, Energinet works a lot in the regulatory domain, which means that Energinet helps define required changes to meet new **EU** regulations. |
| ABA39 | Energinet | Energinet collaborates with other **TSOs** in the **EU**, which harmonise with European legislation. |
| ABA47 | Incremental Development | Today, it is accepted that incremental development is the way to go. Technical standards implement new developments; it can take 10-15 years to create a new standard and another 10-15 years to implement it. |
| ABA48 | Need | We need the functionality that comes with new technology. We are at a place where we need it so much that the distance between research and proof of concept must be minimal to enable us to take advantage of new functionality as fast as possible. |
| PLI4 | Green Transition | The green transition presents unique challenges, and the challenges are diverse. Our energy infrastructure has been designed around a monoculture, meaning it is generally thought of as separate pillars and sectors and only optimised within its own domain. |
| PLI5 | Green Transition | The green transition is in dispute with the monoculture, as it requires more diversity in the energy sector. |
| PLI6 | Green Transition | In old times, we built large coal factories and ensured coal, oil or gas was delivered to ensure the supply chain. Fossil fuels allow us to store huge amounts of resources for later production needs. We are not in this situation today. We want to get rid of fossil fuels. |
| PLI7 | Climate Status | From all the energy Denmark uses today, only 20% is electricity. The rest is from combustible substances. 8% is liquid or gas-based, and this 8% stems from 40% of the 20% electricity. |
| PLI8 | Green Transistion | The more sustainable energy we produce in our energy system, the more fluctuating it becomes. The closer to 100% sustainable energy we get, the more stochastic the energy systems act. |
| PLI9 | Green Transition | A 680 Mega Watt solar cell factory can deliver 90% one minute, three minutes later it might deliver 40%, and 30 seconds later it might deliver 100%. Clouds obstruct the sun's rays. |
| PLI11 | General | We cannot use Himmelbjerget because it is not a mountain, and we cannot use old coal mines as reverse waterfalls. So what can we do? We can increase the amount of available information. |

| ID | Sub-category | Statement |
|---|---|---|
| PLI12 | Sector Coupling and Flexibility | The Danish energy sector bets on two racehorses sector coupling and flexibility. |
| PLI13 | Nuclear and Coal Power Plants | We do not have nuclear or coal power plants, as we demolish them. Luckily we have not demolished them all, so we can start some of them up again. And we do that! |
| PLI17 | Energy Systems | The classical energy system is very hierarchical, with the consumers at the bottom and the markets, the actors, and the **TSO** at the top. The consumers demand something, and the market responds with production. It is a vertical implementation within one sector, e.g. electricity or gas. |
| PLI19 | Energy Systems | When we talk about energy systems, we also talk about district heating, water and more. |
| PLI20 | Limitations | At some point, we will recognise that we cannot transfer information fast enough in the current market design. |
| PLI41 | Electricity Market | The electricity market is planned for 48 hours. |
| PLI42 | Windmills | Danish windmills cost society 17 million a year because they produce both when the prices are positive and negative. |
| PLI43 | Energy Regulation | Three years ago, we spent 680 million on energy regulation; this year, we will probably spend around 1,3-1,6 billion. |
| PLI44 | Electricity Market | The electricity price you see when you look at your electricity bill consists of a set of subproducts; generation of power, consumption, non-consumption, inertia, frequency stabilising services. |
| JH17 | Electricity Sector | In the electricity sector, we have **DSOs** and **TSOs**. **TSOs** distribute energy and balance the grid, and **DSOs** distribute the energy to households. |

## M.2 Centralisation and Decentralisation

| ID | Sub-category | Statement |
|---|---|---|
| JHS39 | Centralisation | Centralisation is part of the solution. Some of the elements must be centralized, but not all. |
| JHS41 | Decentralisation | We can decentralize services that represent a centralized component like identification. |
| JHS42 | Centralisation | We need some central standard, and that is an aspect of centralization. |
| JHS45 | Decentralisation | If we need fast response time, we need decentralization. |
| JHS46 | Centralisation | Pros of centralization is that you can top manage things. |
| JHS47 | Decentralisation | In a decentralized solution, you must decouple elements and have much more exchanging and overhead. There is a price for decentralizing; it is not necessarily a bad price to pay, but it is a price. |
| JHS48 | Centralisation | Some actors need to centralize their solution, as that is what they know. |
| JHS49 | Decentralisation | If we introduce too much complexity, with decentralization, on top of all the things they need to integrate with, the price for buying into these paradigms will be too great. |
| JHS53 | Decentralisation | There is a buy-in period that limits what is possible in the beginning; moving too fast will make things too complex. (to begin with, it is not possible to go fully decentralized) |
| JHS55 | Centralisation and Decentralisation | We need aspects of centralization and decentralization. |
| JHS56 | Decentralisation | As we distribute locally, solutions will become more decentralized. |
| JHS59 | Centralisation | To collaborate, we must have central decision-makers. |
| JHS60 | Centralisation and Decentralisation | Concerning software architecture, centralization, and decentralization will be decided based on what the data should be used for, and whether issues can be solved on the client-side. |
| JHS61 | Centralisation and Decentralisation | If we must balance the energy grid, we need both centralization and decentralization, depending on the issues we look into it. |

| ID | Sub-category | Statement |
|---|---|---|
| JHS62 | Centralisation and Decentralisation | The vision is to decentralize as much as possible, but practically, it will be a mix of both centralization and decentralization. |
| JHS63 | Centralisation and Decentralisation | We must accept that some actors have a centralized solution that integrates into decentralized solutions and vice versa. |
| ABA25 | Data | There is a huge amount of data to ensure that units, systems, and persons, can communicate and trust each other. |
| ABA26 | Legacy | Energinet needs new standards to be implemented so systems can understand each other or software solutions are capable of translating legacy procedures and standards to newer procedures and standards. |
| ABA35 | Goal | We need to work towards a place where everything works in a decentralized system. |
| ABA36 | Collaboration | A decentralized system will allow users to communicate independent of each other. |
| ABA37 | Decentralization | A lot of elements are missing to establish a decentralized system; I think we will end up with one, but I also think there is a lot of prerequisites, where we must first combine and use existing central registers to create trust. |
| ABA42 | Centralisation in the Energy Sector | When you think of centralization in the context of the energy sector, where you have approximately 450 million sensors in a European context, then you have a huge need to reduce complexity. In some cases, this can mean that we let the lowest common denominator be the deciding factor, which is not necessarily good. |
| ABA43 | Centralisation Limitations | When you need a system that is flexible and can handle differences and local contexts in a European context, it is almost impossible with a centralized approach. |
| ABA44 | Centralisation Limitations | When you have a technological world that is changing as fast as it does, then you need a very different market setup. |

## M.3 Collaboration

| ID | Sub-category | Statement |
|---|---|---|
| JHS5 | Communication | Energinet needs to communicate the different needs digitally that different companies have in the different sectors. |
| JHS4 | Scope | Isolated from Energinet's perspective there is a need for collaboration to manage energy consumption from water, heat and gas. |
| JHS6 | Flexibility | From another perspective, Flexibility is not in Energinet's favour concerning collaboration. |
| JHS9 | Scope | In the gas sector, it is somewhat easier to collaborate as it is mostly one company that controls all gas. |
| JHS10 | Scope | If we look at the district heating sector, it has a regional division, so collaboration is between Energinet and each region. |
| JHS14 | Process | If we need to tell the water sector they must turn down the energy consumption because they use too much power on water pumps, we must come up with a counter offer. |
| JHS15 | Process | There is no automatic process for handling collaboration and counter-offers today. |
| JHS18 | General | Energinet must collaborate better. |
| JHS20 | Infrastructure | Energinet needs a better infrastructure to exchange information between sectors. |
| JHS78 | Transaction-based Communication | We need to define the framework for how we collaborate, what transaction models we will use, what costs we have, on what terms we trade, and what you can get in return for a trade. |
| BT74 | General | Compared to today, we have operational systems and analytical systems, and these two worlds have nothing to do with each other. |
| BT75 | General | When an operational system needs to report something to an analytical system, data is thrown over the fence, with no regard to problems this may cause. |
| BT101 | Data Exchange | All the old actors are used to communicate with well-defined data exchange methods. |
| BT102 | Decentralized Communication | Work between the professional actors is fully handled today, but all the decentralized actors are not handled at all. |
| BT104 | Data Silos | Some of the decentralized data is at the actors, and some of it is in the DataHub, but the data is lacking, and we need to be able to get this data faster. |

| ID | Sub-category | Statement |
|---|---|---|
| BT105 | Incitement Structure | The question is how can we create an incitement structure and a communication protocol that allows decentralized partners to act according to what is best for the grid. |
| BT106 | **EU** and Private Actors | We have data exchange somewhat under control in the north, but we also need to have this under control in the rest of the **EU**. However, it is the private actors that are the issue. |
| BT109 | Vision | What you say is that I, as a data producer, should expose the data you require, and that is the wild version, because there is a lot that plays into if this is even possible. |
| BT110 | Vision | It is a fascinating mindset because if we have our domain model, which could be exposed, then we have a description of what we want within a domain. |
| BT112 | Vision | A contract is always between two parties, so you might be able to expose your domain model, and then the other party can decide if they want to provide the data. |
| BT122 | Access | If you and I trust each other, then you can read my data catalogue, and I can read yours. |
| ABA4 | Infrastructure | A big part of the digital infrastructure that must tie the energy sector together is missing, e.g., component control (both centralized and decentralized), orchestration between actors, settlement, calculation of tariffs and calculation of taxation. |
| ABA5 | Transaction-based Communication | If we must tie together the energy sector in such a way that it can have dependencies to each other and enable decentralized control, then we must think more in transaction-based communication, that is disconnected from manual processes. |
| ABA41 | Maturity | The collaboration processes within the **EU** are digitalised; however, the processes are in some cases dated, e.g., communication by email, ingestion of CSV files, no focus on decentralization. |
| ABA45 | Market Models | There is huge collaboration in the European context on the market models that have proved to work well but that we can now see beginning to falter. Collaboration on how to accommodate a decentralized approach is missing. |
| ABA67 | Contract-based trade | Contract-based trade is not just a term when trading in a market, but is a term for when two parties must communicate. |
| PLI10 | General | Our foreign connections are not worth much in these crisis times because each nation focuses on its wellbeing. |

## M.4 Data Management

| ID | Sub-category | Statement |
|---|---|---|
| BT9 | Constraint | The technologies an API is created with today will not support the amount of data we will have to exchange in the future. |
| BT10 | Technology | We have to look into technologies that can exchange data more appropriately. |
| BT16 | Avro | Avro is a good format for real-time data, as you can query on Avro immediately. |
| BT17 | Avro vs Parquet | Avro is not good for range queries; for this, we should use Parquet. |
| BT59 | Data Contracts | The framework for transactions between parties should be defined in data contracts. |
| BT62 | Incubator Area | We want to build an area where data scientists can ingest data, analyze it, and operationalize it. |
| BT64 | Data Curation | We will expose data in a curated manner. |
| BT66 | Temporal Storage | It can be hefty on storage demands. |
| BT68 | General | The frequency is high, and we save in peta-, hexa-, and soon yotta- byte. |
| BT81 | Data Quality | Data quality is whether we can deliver in time (e.g., data surveillance) |
| BT88 | Ingestion | The lambda architecture is about recognising data comes as a stream or as a batch load. |
| BT89 | Ingestion | When we talk about **ETL** and **ELT**, it is the same, there is no difference, only where we place our compute layer. |
| BT90 | Ingestion | With **ETL**, we ingest data to a compute layer, and send them back to a storage layer. |
| BT91 | Ingestion | With **ELT**, we extract data to the storage layer and compute from there. |

| ID | Sub-category | Statement |
|---|---|---|
| BT114 | Vision | I want to say in this context that it is an entity that owns a data set. |
| BT120 | Normal Forms | It should be up to the customer what normal form data should have. |
| BT121 | Normal Forms | Normal forms and how data is stored is not relevant when ingesting or consuming data. |
| ABA27 | Data | Data that is gathered from sensors is not the same as the data that is used in a business context. |
| ABA68 | Data | Data rarely has value if it is not used together with other systems or other data. |
| JH4 | Data Storage | There are some types of data that are better suited for specific data storage solutions. For example, graphs should be stored in a graph database. |
| JH25 | Retention | In some areas, it makes sense to store data with no retention, and in other areas retention must be set. |
| JH26 | Compute | Most storage systems that exist decouple the compute layer from storage, and many of the storage systems can auto-scale. |
| JH27 | External Storage | It is ok that external storage solutions are used to store data. |
| JH28 | Tiering | There can be a storage tiering problem, where relatively new data is stored in expensive but performant storage and historical data is stored on cheaper and less performant storage. |
| JH29 | Policies | Policies should set where data is stored, how long it is stored and more. |
| JH30 | Policies | Policies should be able to trigger, e.g. transferring data when the data retention is reached. |
| JH31 | Tiering | Data should be moved from slow storage to faster storage when requested. |
| JH32 | Tiering | Data tiering is not supported by all storage solutions. |
| JH33 | Flexibility | Data is always domain driven, so the data storage solution should be flexible. |

## M.5 Data Mesh

| ID | Sub-category | Statement |
|---|---|---|
| JHS89 | Data Mesh vs Data Spaces | Data meshes are a sequence of concepts that overlap with the **DSI**. |
| JHS89 | Cost | If we go with the data mesh strategy, the buy-in will move because we get governance, data structure, and data discoverability. |
| JHS89 | Migration and Maturity | If we strive to go with data mesh, then the migration to data spaces will become easier, and the maturity level where we start will be much different. |
| BT11 | Data Product | We expose APIs for every single Data Product, that we can gather and make queryable. |
| BT12 | Data Product | A data product can query another data product. |
| BT13 | Collaboration | If the data product is at an internal or external partner does not matter as we are part of the same data ecosystem. |
| BT14 | Constraint | Many technologies do not support the right vision of a data mesh, and if they exist, they are emerging. |
| BT20 | Convergence | Data Warehouse, Data Lake, Data LakeHouse, and data mesh are technologies and terms on top of other technologies. |
| BT57 | Definition | It is a contract between two parties, so what you want as a consumer, I should be able to expose as a provider. |
| BT69 | Convergence | Data mesh is a collection of existing technologies. |
| BT70 | Convergence | The reason I got fond of data mesh is that it combines some of the best practices from the world of programming and the data world. |
| BT71 | Convergence | It provides a model to solve some of the shortcomings in the **BI** world, e.g. data ownership. |
| BT72 | Convergence | Fundamentally, data mesh moves back to the Data Warehouse, as a part of the Data Product. |
| BT73 | Data Product | We have a data product and an operational product, and they are part of the same. |
| BT77 | Definition | Data mesh is a cultural change more than it is a technical change. |
| BT78 | APIs | Data products should, via endpoints, expose data that can be consumed. |
| BT79 | Discoverability and Observability | The data mesh experience takes our **SRE** area, and our data catalog and blends them to provide insights into our data and operational quality. |

| ID | Sub-category | Statement |
|---|---|---|
| BT80 | Maintenance | The quality of a data product falls if it is not properly maintained. |
| BT82 | Maturity Requirements | The social aspect to data mesh is about whether we have the skills required to execute the changes we want. |
| BT83 | **SLOs** | With data mesh, we have data products, their descriptions, whether they provide what they are supposed to, and if the data quality is sufficient. From this, we can create trust. |
| BT84 | Contract-based Trust | If you provide data to me, then I tend not to trust you, so I make my data quality checks because I need to be certain that you deliver quality. If we have trust, if we have a contract that defines the data that you deliver, then I trust you. |
| BT85 | Observability | If I get an error on my end, then I report that error back to you because I am certain that it is not on my end. |
| BT86 | Contracts | We have always had contracts for data, but never as strict as data meshes define them. |
| BT87 | Vague | Data mesh describes the vision of what we want, but very little on how to get there. |
| BT107 | Vision | Where we want to get to is that the systems (if developed by us) should expose their data products through a YAML file, so whenever we create a new version of our code, we must also update our YAML file to tell how the data catalogue looks and how we communicate internally. |
| BT108 | DataHub | We make the data products discoverable in the DataHub through our data catalog. |
| ABA64 | Data Lake / Data Mesh | Data Lakes is about ensuring we can store a huge amount of data, where a data mesh is concerned with metadata, so we do not need to store all the data at a central place, but that we can find it when we need it. |
| ABA69 | Metadata | Using metadata in a data mesh, as data that we can trust and use, is what for makes it extremely complex. |
| JH5 | Data Storage | In data mesh, the choice of data storage is determined by domain (. . . ). |
| JH41 | Provisioning | Provisioning data products from a YAML-based contract that abstract complexity away from the user have a large value, as the alternative is managing many manifest files to provision infrastructure and code needed to run data spaces. |

## M.6 Data Spaces

| ID | Sub-category | Statement |
|---|---|---|
| JHS16 | Collaboration | Ideologically, we can solve collaboration with data spaces. In reality, there are a lot of prerequisite agreements that must be made. |
| JHS17 | Maturity | Energinet have a maturing phase before data spaces can be implemented, and this phase is different for all sectors. |
| JHS19 | Scalability | Whether we will solve the issues with data spaces is not fully defined yet, but phones are not very scalable. |
| JHS21 | Information Exchange | With data spaces, we could ensure exchanging of information, but only the information needed to be exchanged. |
| JHS26 | Potential | Data spaces is a potential solution, but it is one of many. Data spaces is the most well-defined solution. |
| JHS28 | Definition | Ideologically, a data space is a way to automatically exchange information between actors in a secure, transparent, and visual way. |
| JHS29 | Definition | When looking at data spaces from a practical perspective people have conflicting ideas on what it is, primarily conflicting subparts and interests. |
| JHS30 | Definition | Data spaces are not well-defined, but we can define the overall concepts and components in a data space. |
| JHS31 | Definition | We must accept that there are many different interpretations of data spaces and many different ways to act in data spaces. There is not just one data space but many that collaborate in unity. |
| JHS32 | Types: European Data Space | European data space with core components like identification, trust, certificates, and authority. |

| ID | Sub-category | Statement |
|---|---|---|
| JHS33 | Types: National Data Space | National data space that uses the European data space and defines international needs. |
| JHS34 | Types: Global vs Local | We must differentiate between global to local data spaces. |
| JHS35 | Marketing perspective | From a marketing analysis perspective, a data space is called a marketing platform, where data owners get permission to sell their data. |
| JHS36 | Information bus | Some would call a data space an information bus that allows exchanging information securely and with a high level of trust. |
| JHS37 | Metadata and Discovery | We need a data space because we need to know who we are. We need metadata to understand each other and to find and use each other's resources. |
| JHS40 | Centralisation | We need to have some centralized components in our data space because how would we otherwise create trust in the things we create? |
| JHS43 | Types | I see data spaces as a part of a large collection of sub-groupings of data spaces. |
| JHS44 | Centralisation vs. Decentralisation | Creating a centralized or decentralized data space depends on the qualities that we want. |
| JHS51 | Legacy Systems | There is a huge amount of legacy systems that are not built for data spaces. |
| JHS57 | Definition | Concerning data spaces and the **DSI**, it is defined in the data governance act. |
| JHS54 | Types: National Data Space | We need a national data space that can be built on top of. |
| JHS64 | Discoverability | Discoverability is essential for data spaces. |
| JHS65 | Characteristics | Discoverability, trust, identity, authorisation and authentication are essential for data spaces. |
| JHS66 | Software Qualities | There is a long list of software qualities that data spaces should adhere to. |
| JHS70 | Logging | We need to determine how much logging there should be in the data space. |
| JHS72 | Types: European Data Space vs. National Data Space | The fundamental differences between needs for the **EU** data space and the national data spaces are: (1) As a writer, the answer can be sharing of data. (2) As database admin, the answer can be that we do as we usually do. (3) As a member of digitalisation, the answer can be that it is unclear. So we have defined three types of data real-time, historical, and future. (4) It does not state much about delegation and access control. (5) It mentions digital trust. It is not well-defined in the European context. We look into moving digital trust into edge devices. |
| BT3 | Definition | Data spaces, for me, are of an indistinct size. |
| BT4 | Definition | What I read is that it is exchanging of data with a fixed format, which makes sense for a flexible energy market |
| BT5 | Definition | When we talk about data spaces internally, it is still very fluffy **EU** text about how we share data. |
| BT6 | Definition | So it must be about the format of data, how we share data, and how we make data discoverable, especially how we enable data to make intelligent sense. |
| BT22 | Data Lake | A Data Lake is not a data space. |
| BT97 | Problem | Providing possibilities to do the same thing in different ways, can be a problem as it adds complexity and needs better administration. |
| ABA6 | Definition | In a European context, the **DSI** is focused on being an alternative to big data companies owning data, such that companies can trade data. |
| ABA7 | Definition | We see it as a digital infrastructure that allows actors to identify, integrate, and share data, with or without money being involved. |
| ABA8 | Need | Energinet is dependent on the **DSI** to succeed to be able to tie together the energy sector and to succeed with Energinet's priorities for the energy system. |
| ABA59 | Maturity | Energinet puts a lot of focus on increasing digital maturity in the energy sector. The **DSI** is one of the projects aiming to improve digital maturity in the supply and energy sector. |
| ABA60 | Definition | The **DSI** is about determining what digital infrastructure is needed to enhance the energy and supply sector now and in the future. |
| ABA61 | State of Progress | The **DSI** started with a lot of partners to try to make it part of the national digitalisation strategy, and thus a political assignment. Now we are working on the harder part of making it a reality. |
| ABA62 | Capabilities | The main aims of data spaces are to convert manual or legacy processes to smarter processes, where the focus is on decentralization, streaming, automatization and integrability. |
| ABA63 | Definition | There is a European understanding of what data spaces are, and then there are Energinet's understanding of what it should be. |

| ID | Sub-category | Statement |
|---|---|---|
| ABA65 | Definition | With a data space, I think of it as a data ecosystem, where data is stored decentralized. Then we have keys (protected by the data owner) to determine the structure of data and who can access it, among other things. |
| ABA66 | Token-based auth | A key in this context is just a hash; it is needed to authorise and authenticate oneself with services. |
| JH1 | Data Storage | For a data space to function, you need data storage. |
| JH2 | Data Storage | The chosen data storage for a data space can vary by domain. |
| JH3 | Data Storage | The specific need for data storage in a data space is that it supports real-time data and historical data. |
| JH5 | Data Storage | (. . . ) in data spaces the choice of data storage is chosen based on business needs. |
| JH6 | Ownership | Something interesting to consider is to determine when a data space belongs to a team or an organisation. Both can be true. |
| JH7 | Provisioning | The more automatised the provisioning process is when considering federation, the more it makes sense to keep data spaces small and owned by individual teams. |
| JH8 | Definition | A data space today is more or less data discovery built on top of existing data infrastructure, with interfaces that expose the data to external actors |
| JH9 | Interoperability | If considering scale and interaction on a more decentralized level, a team can create a data space, that can interact with another data space. |
| JH10 | Software Architecture | In the software, many components might not be needed for a data space. |
| JH11 | Software Architecture | Some of the components that will repeat are authentication, authorisation, the governance layer, data formats, standardised APIs, usage, logging, tracing, and more. |
| JH12 | Definition | It is not well-defined how data spaces should be implemented. |
| JH13 | Data Format | The interface to data can move from a generic standard/data format to a more specific standard/data format. |
| JH14 | Definition | A data space is as a concept so generalised. A lot of sectors talk about data spaces as data liberation. In other sectors data spaces are interesting for real-time analyses. |
| JH15 | Decentralization vs. Centralization | In the supply sector, it is a requirement that data spaces are decentralized. In other sectors, it is not required. |
| JH16 | Interoperability and Decentralization | Interoperability between data spaces allows the systems to function as a decentralized system regardless of which nature the data space takes (monolithic vs service-oriented). For example, a data lake can be part of a data space by being interoperable with other systems. |
| JH19 | Types: Local | **DSOs** and **TSOs** want to use data spaces internally, e.g. collecting and analysing data. Some of this data should be classified, and some can be shared. |
| JH20 | Cost | The water sector consists of more than a hundred small water factories. If every water factory had a data space it would be an enormous expenditure because they would need to hire IT personnel. What they would do, is hire an external partner, to create a central data space for the whole sector. |
| JH21 | Provisioning | An automatic provisioning system for whole data spaces is more relevant where provisioning must happen often, e.g. smaller teams or where it is decentralized. |
| JH22 | Provisioning | There are no requirements to where decentralized data spaces are hosted, but it should be up to the owner. |
| JH23 | Governance | Data spaces must adhere to requirements for the governance layer, e.g. provenance, tracing, interoperability and extensibility. |
| JH24 | Data Storage | Data spaces require data storage. |
| JH38 | Input/Output | Data spaces should support both push and pull (e.g. a pub-sub). |

## M.7 Domain Modelling

| ID | Sub-category | Statement |
|---|---|---|
| BT38 | Domain Models vs. Data Vaults | There is conflict right now about whether we should use domain models or data vaults. |
| BT39 | Definition | A domain model is a model at the Third Normal Form that describes a specific domain. |
| BT40 | Use-case | We have source systems which we populate our domain models from. |
| BT41 | Data Mart | We can create Data Marts on top of domain models. |
| BT42 | Use-case | External actors can populate domain models if they own a part of the domain. |
| BT43 | Context | Domain models are important for Energinet as Energinet is regulated by **EU**, both concerning gas and electricity. |
| BT44 | Context | Energinet has domain models on the electro-technical, the gas-technical, and the information-specific aspects. |
| BT45 | Preference | Data Scientists are not happy with domain models, they prefer data vault modelling. |
| BT46 | Data Vault | Take the original data models, place them next to each other, and give each entry a key. |
| BT47 | Data Vault | With Data Vaults, you do not change the source models, and data scientists prefer pure data. |
| BT48 | Context | We want to utilise domain models as these make more sense to the business users, plus we then have the mapping at the source owners. |
| BT49 | Pseudo Data Vault | Enabling source owners to define the mapping gives us a pseudo data vault as we do not change the source models. |
| BT50 | Use-case | We always save or plan to save a historical copy of the source data which we then map into the domain model. |
| BT51 | Use-case | We built query models on top of the domain models. |
| BT123 | Ownership | A data model internal in a data domain is yours and only yours, and then you can choose to expose it as you want. |
| ABA28 | Risk | One should be careful not to enforce domain models through a value chain. The market context does not correlate well with the technical context. |
| ABA29 | Bounded Contexts | I think the most important part for me about data is to think of data in a decentralized context. We cannot just think of one huge domain model and enforce everything into it, but we must look at each separate system such that it can work autonomously, with a generally well-defined interface. |
| JH36 | Domains | Domains should be split into sub-domains where it is logical and brings value. |

## M.8 Flexibility and Grid Balance

| ID | Sub-category | Statement |
|---|---|---|
| ABA12 | Green Transition | Wind energy and solar energy are, funnily enough, only produced when the wind blows or the sun shines, so the task of ensuring energy is available at all times is much harder. We can solve this by ensuring there are other forms of energy available, e.g., bio-gas or power-to-x. |
| ABA13 | Power-to-x | Power-to-x is the process of using electrical power as one of the primary sources to produce other forms of energy that are better suited to be stored or transported, e.g., hydrogen or ethanol. |
| PLI14 | Sector Coupling and Flexibility | Sector coupling and flexibility require the sectors to be connected with information. |
| PLI15 | Green Transistion | The closer we get to 100% sustainable energy, the more information we need. |
| PLI16 | Green Transistion | My gut tells me that if we double the amount of sustainable energy, then we quadruple the fluctuation in the energy system. Still, we need 8x times as much information, so it is a factor of 32. |
| PLI21 | Green Transistion | The closer we get to 100% sustainable energy, unless we invent the fusion reactor during 2023, we will have to scale out to meet the information transfer demands. |

| ID | Sub-category | Statement |
|---|---|---|
| PLI22 | Energy Systems | The energy system must become diverse; it has to become more complex. Adding complexity is a good idea because the more diverse an energy system is, the fewer larger parts can fail. |
| PLI27 | Context | Because of the energy crisis and the high prices right now, we get flexibility. Denmark uses approximately 13% less electricity this year and moves their electricity usage to times of the day when the prices are lower. |
| PLI28 | Flexibility By Storing | If a kilowatt-hour only costs 12 cents, what if we then save that power with a battery that loses 10%, and then use it in the evening when the prices are 8 kroner. It requires information to be available to make these forecasts. |
| PLI29 | Context | The new reality of our energy systems are all about increasing the amount of information available, e.g., the amount, quality, availability and diversity. |
| PLI30 | Context | I think we will have to include information in our forecasts that we have never dreamed of using before. |
| PLI31 | Vision | In a complex world, we will have to create simulations from models that we train with an individual household or company to be able to provide the insights we need. Then we might have 3.3 million simulations running. |
| PLI32 | Power-to-X | It costs 52% energy loss to convert electricity to synthetical methane. It means we do power-to-x, but first, we convert water to hydrogen, then hydrogen to ethane, and then ethane to methane. Why is methane, then, interesting? Methane is our gas net, and we can move enormous amounts of energy with our existing gas net. Because of the prices, this can be very advantageous. |
| PLI33 | Power-to-X | We need more information to be able to know if we have surplus electricity. We can market surplus electricity and convert it no matter the conversion rate. It allows us to store the electricity, which has a high probability of being worth it considering the savings and fluctuations. |
| PLI37 | History | In the old days, when we had too much electricity in our energy system, we could downregulate the electricity grid. We could call the producer and tell them we didn't need as much, and then they would ask us to pay to downregulate. This meant we ended up paying double for electricity, and to combat that, we installed industrial electric kettles to consume the surplus energy to not pay for downregulation. |
| PLI39 | Flexibility at the Edge | 40% of the flexibility we want to achieve, we can get from the consumers. |
| PLI40 | Forecasts | We could use **IoT** devices or **LoRaWAN** as a technology to send and receive a price or **$CO_2e$** forecasts to help balance the grid. |
| PLI45 | Synthetic Inertia | Electrical vehicles (Lithium batteries) can deliver synthetic inertia well, as they can provide synthetic inertia in microseconds. |
| PLI46 | Synthetic Inertia | Synthetic inertia can be obtained by adjusting the charging power if we need more electricity elsewhere. |
| PLI47 | Synthetic Inertia | Synthetic inertia is about ensuring consumed electricity lasts longer, e.g., with regenerative braking. |
| PLI51 | Grid Balance | If we as a state-owned organisation registered as a **MVNO** as a charging stand network, and told all telecom operators that it is free to use, and added the data to our DataHub, then we could gain a real-time overview of where electricity is consumed, and then use the real-time data to micro-balance the grid. |
| PLI52 | Brownouts | Brownouts are if we strategically choose to turn off the electricity grid in some regions to save the grid. |
| PLI53 | Packetised Energy Prioritization | Packetised energy prioritisation is a project in England where units publish what their intent is, and a central system then responds if it is possible, and when it is possible. It showed that if a household can introduce 2,5-kilowatt hours in flexibility, then north England can balance their grid with the sustainable energy they have available today. |

## M.9 Governance

| ID | Sub-category | Statement |
|---|---|---|
| JHS22 | Confidentiality | Information is business critical, and we can't just tell that the solar system around the corner produces X amount of energy. It is not our information to share. |
| JHS23 | Transparency | To some degree, we need to be able to share business-critical information, like when we need to share that we have surplus energy in a specific sector. |
| JHS24 | Transparency | Energinet will have to share some information that parts of the organisation will deem sensitive. |
| JHS25 | Information Management | Energinet will need to find an information model and communication model that allows responsible sharing of sensitive information. |
| JHS71 | Data Erasure | When I demand that you cannot use my data any longer, I want to be able to see the data disappear on the other end. We need that transparency. (Delegation and Access-control) |
| JHS74 | Delegation | Delegation is important to Governance. |
| JHS75 | Access-control | I want to share my yearly consumption of energy with someone. I want to share it with you, and you can use it for 10 minutes, when the 10 minutes are up, you cannot access it anymore. That is a Governance layer. We need that. |
| JHS76 | Delegation and Access-control | We need protocols to enable us to do delegation and access control and to define what you are authorised to do with data. |
| JHS77 | Compliance | We need to secure the compliance that is needed. We need to be able to see that you did what you said you did. |
| BT125 | Residency | The customer should be able to control which zones their data is available in. |
| BT126 | Residency | From a cosmetic perspective, I do not want a data set available in a zone where it is not used, but I also want to be able to distribute data sets to the zones closest to the customer if allowed. |
| BT128 | Discoverability | You need an overview of what data product you have, who uses them, what contracts are there and more. |
| BT129 | Discoverability | We need a structure and a metadata description, so we can understand data and enrich data in ways that provide us with new abilities. |

## M.10 Infrastructure: Digital

| ID | Sub-category | Statement |
|---|---|---|
| BT76 | Convergence | We want to combine the operational and analytical systems into one system. |
| BT92 | Cloud Storage | With the cloud, we have one unified storage, a lot of computing, fast reading from storage and high **IO**. |
| ABA23 | Orchestration | We can build a lot of new physical infrastructures, but it does not provide much value if we do not become better at orchestrating our energy system. |
| PLI48 | Security | The energy system should be shielded on closed networks - it is not accessible for malicious intents. |
| PLI49 | Security | Energinet could register itself as an **MVNO** to ensure they own a network where it can communicate safely. |
| PLI58 | HTTP | We do not use classical HTTP standards for communication internally, as we need standards that can be used for 30 or 40 years. |
| PLI62 | **OT** | We run something called **OT** instead of **IT**. 80% is the same; we use the same computers, we use the same programming languages, and we use the same development tools, but on one point, we differ. In the **IT** world, you use end-to-end encryption, in the supply sector it is counter intuitive. If we encrypt everything in our network, we become blind. |
| PLI64 | **OT** | In the **OT** world accessibility (we can access data), integrity (data is kept intact), and confidentiality (who can access the data) are the key parameters. |
| PLI65 | Validity | The worst scenario for the energy system is not if someone hacks a station and destroys it, but our worst fear is if someone makes our infrastructure lie. |
| JH37 | On-the-edge | In on-the-edge scenarios, it can be interesting to deploy a subset of functionality to edge devices. |

| ID | Sub-category | Statement |
|---|---|---|
| JH40 | Provisioning and DataOps | We must support modern approaches, like auto-provisioning and DataOps. |

## M.11 Infrastructure: Physical

| ID | Sub-category | Statement |
|---|---|---|
| ABA10 | **TSO** | The infrastructure that supports the tasks of the **TSO** primarily consists of many electrical wires under or above ground, many gas pipes, and few gas storages. |
| ABA11 | Green Transition | One thing that is a big issue is that we are reorganising from an energy infrastructure with many big central power plants to an energy infrastructure running solely on green energy. |
| PLI35 | Gas System | 10 Gigawatt which our gas system can move, corresponds to seventeen 400 kilovolts wiring harness. |
| PLI34 | Gas Storage | Danish gas storage can provide energy for approximately a year. |
| PLI36 | Context | Most of Denmark's electricity is consumed in Zealand but is produced in west Jutland. So I do not think Funen would like that we create seventeen 400 kilovolts wiring harness that traverses over Funen. |
| PLI38 | Cogeneration Plants | Cogeneration plants can use natural gas, oil, wood chips, and electricity, and they can store surplus energy be heating up their district heating systems above what they usually do. It gives them a capacitive characteristic. |
| PLI50 | Charging Stands | Every week, a thousand charging stands are installed in Denmark. These charging stands typically communicate over 4g. |
| PLI54 | Limitations | The need to increase the amount of available information, and because the speed of fibre is 0,61 times the speed of light, and not fast enough to move data; we need decentralization in terms of a data mesh, Data spaces or distributed information setup. |
| PLI61 | Fibre Network | Every time a 400 kilovolts wiring harness is installed, then we install 96 fibres along with it, it allows us to have a closed network from the public. |

## M.12 Legislation and Regulation

| ID | Sub-category | Statement |
|---|---|---|
| BT99 | ENTSO-E | ENTSO-E is responsible for the conveyances that define the work between **TSOs** and **DSOs**. |
| BT100 | S-557 | There is a danish work unit that is responsible for the S-557. They all work under the **IEC** 61-850 standard. |
| ABA49 | Priority | We cannot let legislation come first, but we need to let technology guide us and use learning to solve issues in the energy sector. At that point, we can begin to legislate. |
| ABA50 | Standards | Typically, standards are created incrementally in global contexts. When a standard gets to a point where it is broadly acknowledged, it will be published as an **IEC** standard for broad usage. At that point, the **EU** commission or national regulation can select the standard and use it as a basis for regulation. |
| ABA51 | Standards | Sometimes standards will be defined further in a local context, but typically, this is not the case. |
| ABA52 | Standards | The maturity for implementing standards is not high. It is assumed that operators or developers creating systems have thought of standards. Likewise, it is assumed that suppliers live up to required standards. |
| ABA53 | Standards | There is no consequence for not following standards other than the lack of advantages implementing a standard might contribute. There are many standards in the energy sector that are not followed or implemented improperly. |
| ABA54 | Standards | If the law states that you must implement a standard, it is with a high leeway. It is often defined in terms that can be interpreted in a local context. |

| ID | Sub-category | Statement |
|---|---|---|
| ABA55 | Standards | The flexibility towards regulated standards is both good and bad. It is good because it is near impossible to enforce one standard in a multitude of contexts. On the other hand, it is a problem in that there is no consistency in domains. |
| PLI57 | Robustness | We cannot use non-robust standards; we can use something like IEC-61850 instead, as it is robust. |
| PLI59 | Standards | In the supply law, we must choose well-established standards, which means something like **ANSI**, **ISO** or **IEC** standards. If these standards do not exist, then we can choose a de-facto standard. |

## M.13 Metadata

| ID | Sub-category | Statement |
|---|---|---|
| JHS80 | General | Without metadata, we can not understand the data. |
| JHS81 | Roles | There are two roles to metadata, (1) the context the data is collected in, and (2) what the data is. When we have this, we can begin to make Discoverability on top of it. |
| JHS82 | Contexts | It is an open question of how to handle different metadata contexts in different domains or the same domain. |
| JHS83 | General | We can put a data model with sufficient metadata into a discoverability **API** that allows us to specify a query that finds data. |
| JHS84 | General | Someone or something must create and define metadata, to make data discoverable in a data space. |
| JHS85 | Standards and Aggregation | In practice, there will be multiple different metadata standards within each domain. There must be an aggregator that can transform data from one standard to another. |
| JHS86 | Aggregation | We must accept that some data is lost when aggregating data. |
| JHS50 | Infrastructure | We need an overall metadata infrastructure before we can agree on anything |
| JHS88 | Agency and Aggregation | There will be a metadata agency and aggregators that can convert between data. |
| ABA31 | Definition | Metadata is data that says something about other data. |
| ABA32 | Capability | Metadata allows us to define what facility we are talking about quickly. |
| ABA33 | Capability | We use metadata for searching functionality but also to establish trust. |
| ABA34 | Storage | Where metadata should be stored is dependent on a lot of things, like usage, complexity and price. |
| JH39 | Definition | Metadata is domain-specific and should be configurable. But generally, all datasets should be documented with metadata. For example, ownership, schema, data quality, lineage, context, data access, and glossaries. |

## M.14 Open Source vs. Proprietary

| ID | Sub-category | Statement |
|---|---|---|
| BT115 | Open-source Levels | Open-source on three levels, design, code, and at an application level. Jakob and I have different opinions here. Jakob believes it should be on every layer; I think you should think about how many layers it makes sense to make it open-source. |
| BT116 | Open-source Solution | From a perspective of progress it might be a good idea to consider open-sourcing the solutions, while still using proprietary tools, technologies, and programming languages that are easy to understand, to build the solutions. |
| BT117 | Learning | Open-source enables others to learn from the solutions. |
| BT118 | DataHub | We are required to do this on the data platform, where we do open-source on the design and application level. |

## M.15 Roles and Actors

| ID | Sub-category | Statement |
|---|---|---|
| JHS8 | Scope | Energinet is dealing with a lot of actors. |
| BT98 | **TSO**, **DSO**, and **EU** | There are different actors. We have the old actors **TSOs**'s and **DSOs**, and the work that happens between these is regulated by **EU** and conveyances according to the **IEC**-61850 standards. |
| ABA2 | Energinet | Energinet is not a state-owned monopoly but an independent public enterprise energy supply company. |
| ABA3 | Energinet | Energinet look into all the things that need to be done to solve the green transition. |
| ABA9 | **TSO** | The role of a **TSO** is the system operation responsibility of balancing both the electrical and gas grids and the transmission operation responsibility, meaning, we must build the infrastructure to transport the energy and gas nationally. Energinet do not maintain the infrastructure to transport energy and gas in local municipalities; this is the responsibility of distribution companies. |
| ABA14 | **TSO** | Everything Energinet does is under the role of the **TSO**. |
| ABA15 | Energinet | Energinet is also responsible for the design of the energy system, such that it is coherent, it is worth paying for, it is climate-friendly, and it creates maximal value for the society. |
| ABA16 | Energinet | Energinet is responsible for the design of Denmark's markets for energy trade, and Energinet also plays a role with all 27 European countries in designing markets for energy trade in a European context. |
| ABA17 | Energinet | Energinet is involved in legislation in a national and European context. |
| ABA19 | **DEA** | We refer to the **DEA**, which is the agency responsible for establishing the legislation, and Energinet is then responsible for carrying out the law within the **TSO** domain. |

## M.16 Software Qualities

| ID | Sub-category | Statement |
|---|---|---|
| JHS67 | Performance | Performance might not be a problem, depending on the data. |
| JHS68 | Performance | If your signal on when you need to balance the electrical grid is 5 seconds delayed, then IT IS a problem. |
| ABA57 | Functionality vs. Usability | If you have a lot of features in the software, users might begin to misuse your software, and create a mess. Likewise, with many features, each feature needs to be maintained, and this can increase the cost. |
| PLI23 | Stability | When corona hit, Energinet sent 5000 people home, but the system did not crash. |
| PLI24 | Stability | A stable complex system is notoriously stable. |
| PLI55 | Performance | "It does not matter what speed we communicate with, just as long as the information is available when we need it." - Nicholas Negroponte |
| PLI56 | Robustness | For the supply sector, we have a term coined robustness, which means that we must built out infrastructure, so it does not have a single point of failure. |
| PLI63 | Security | **CIA** is a security principle, which means we have to encrypt, secure integrity, and do authentication and authorisation. |
| PLI66 | Integrity | To enable integrity, we must digitally sign messages instead of encrypting them, so we can make sure the messages are immutable. |

## M.17 Users

| ID | Sub-category | Statement |
|---|---|---|
| BT60 | Business User vs. Data Scientist | A business user often has repeating query patterns on what they want to be provided, whereas data scientists do not. |
| BT61 | Business User vs. Data Scientist | A business user uses query models to access data, where a data scientist accesses data from the domain model and/or the raw data. |

# N | Selective Coding of Interviews

This appendix summarizes the results of the selective coding process as described in sec. 4.8. A total of 17 theories and 66 hypotheses were identified.

## N.1 Theories

### T1: The Business Ecosystem

The business ecosystem in the Danish energy sector is undergoing a transformation driven by the green transition. As the different actors in the sector navigate this changing landscape, they face challenges related to conflicting interests and goals. As such, a need to define and harmonize their roles within the ecosystem arises.

The green transition demands greater diversity and flexibility in the energy sector, requiring transitioning away from the traditional monoculture and hierarchical energy systems. The shift requires increased information availability, sector coupling, and adopting sustainable energy sources resulting in a more stochastic energy system. In this regard, digitalization is crucial as digital maturity varies among the different actors and sectors, which might hinder the integration of new technologies and advancements needed to facilitate the green transition.

The regulatory environment, including **EU** regulations and collaborations among **TSOs**, influences the ecosystem's development by shaping the required changes and fostering incremental development. While this approach has its merits, it can also result in resistance to change, setting the energy sector behind other industries.

In conclusion, the Danish energy sector is moving towards a more sustainable and digitalized future where collaboration, communication, and adaptability become central to ensure a well-functioning business ecosystem.

### T2: The Role of Centralization and Decentralization

The effective functioning of the Danish energy sector requires a balanced combination of centralization and decentralization. Centralization is essential for standardization, top-level management, and collaboration, while decentralization enables better performance, local adaptability, and in some regards, reduced complexity. The optimal balance depends on data usage, energy grid balancing, and software architecture.

It is essential to acknowledge that different actors may have different preferences for centralized or decentralized solutions, and an effective system must facilitate both approaches. The ultimate goal is a predominantly decentralized system allowing users to communicate while leveraging existing centralized registers to establish trust. However, the transition towards decentralization will require overcoming numerous challenges and is expected to be gradual.

### T3: The Importance of Collaboration

The key to successful collaboration requires improving the means of information exchange between different actors while considering the distinct needs and constraints of each.

In order to facilitate collaboration, it is necessary to establish a robust digital infrastructure that supports transaction-based communication and data exchange. Doing so will enable decentralized partners to act in the best interests of the grid. Overcoming challenges such as data

silos, outdated collaboration processes, and the lack of incitement structures and standardized communication protocols is an integral part of this process. The energy sector's vision should focus on establishing a decentralized, trust-based infrastructure enabling actors to collaborate.

**T4: Data Management in an Evolving Ecosystem**

The future of data management requires addressing data quality, storage, ingestion, egestion, and integrations. Ensuring data quality involves delivering data on time and maintaining data surveillance. However, prioritizing what to address should consider an organization's technological constraints and needs. A future-proof data management strategy should adopt suitable technologies and data formats to facilitate efficient data exchange, storage, and real-time querying; for example, Avro for real-time serialization and deserialization and Parquet for efficient storage and querying.

Data management should also encompass an incubator area where data scientists can ingest, analyze, and operationalize data. This area should facilitate data curation and ensure that data is exposed meaningfully. In that regard, storage demands, especially for temporal data, should be carefully considered, as these can significantly impact the organization's resources due to the potential amount of time-series data. Furthermore, data storage solutions should be chosen based on the data type, such as graph databases for graph data and **SQL** databases for relational data. Likewise, storage tiering can help optimize resource utilization by storing new data in high-performance and historical data in less expensive, less performant storage.

Furthermore, data storage solutions should support flexibility, allowing domain-driven data management and external storage solutions when necessary. Retention policies must also be established to determine the duration data can be stored and to enable triggers for data cleansing.

In conclusion, an effective data management strategy should be comprehensive, flexible, and adaptable to the organization's evolving needs.

**T5: The Perception of a Data Mesh**

Data mesh is a cultural and technical shift in managing, sharing, and consuming data in and between organizations. It leverages metadata and data products to create a decentralized, trust-based infrastructure emphasizing discoverability, observability, and collaboration between actors. Concerning discoverability, metadata is crucial, enabling data products to be discovered by integrating with a data catalog. Furthermore, metadata provides transparency of the data and its context, which is essential for data consumers to understand the data and its quality.

Interestingly data mesh combines the best practices from the programming and data worlds, solving shortcomings in traditional **BI** approaches, such as data ownership. It also adopts contract-based trust systems, where data providers and consumers must adhere to strict contracts defining data quality and delivery targets.

Data products expose **APIs** in a data mesh, enabling interoperability between different data products. The mesh of data products ensures seamless collaboration and data exchange within its ecosystem. Another thing to note is that data meshes are built on top of existing technologies and converge with concepts like Data Warehouse, Data Lake, Data LakeHouse, and others to provide a comprehensive and decentralized data management solution. In other words, it is the natural evolution of existing data infrastructures.

In summary, data mesh is an innovative approach to managing data in organizations by leveraging metadata, data products, and a contract-based trust system to create a decentralized, discoverable, and collaborative data infrastructure. As such, its success depends heavily on

the organization's maturity, skillset, and ability to adapt to the cultural and technical changes required.

### T6: The Perception of Data Spaces

Data spaces are adaptable, evolving infrastructures aimed at providing seamless interaction and exchange among local and global actors. They are built on existing infrastructure and handle real-time, historical, and future data. Some of a data mesh's crucial features are standard data formats, authentication, authorization, and governance layers. Effective governance is essential to maintaining trust, security, and compliance and involves requirements for provenance, tracing, interoperability, and extensibility.

Data spaces are in their infancy and are expected to evolve gradually from being simple data discovery platforms to offering advanced capabilities. These capabilities are, for example, decentralization, streaming, automatization, and integrability. On this journey, data spaces must be adaptable to specific domain requirements and organizational needs, including data storage choices, decentralization, and degree of automation in the provisioning process.

Finally, data spaces should be designed to scale, concerning the amount of data and the processes they support. Scaling ensures that data spaces can accommodate the needs of organizations and industries while minimizing any negative impact on efficiency and performance. However, scaling is not free and requires considering cost, resource allocation, and provisioning to succeed.

### T7: The Role of Domain Modelling

Domain modeling is a crucial aspect of data management that focuses on creating well-defined models to represent specific domains. The choice between domain models and data vaults depends on organizational preferences, use cases, and historical data needs.

Domain models are typically represented in the third normal form and can be populated from source systems or external actors, provided they own a part of the domain. These models can serve as the foundation for creating data marts, data products, and query models, enabling better understanding and utilization by business users. In organizations like Energinet, which the **EU** regulates, domain models, play a critical role in representing electro-technical, gas-technical, and information-specific aspects.

Data vaults, on the other hand, preserve original data models and provide a more flexible data storage option, and are often preferred by data scientists. Pseudo-data vaults, which do not alter the source models, can compromise domain models and data vaults, mapping source data to domain models while preserving historical copies.

The ownership of data models should be well-defined, allowing entities to expose their models as needed. In that regard, separate systems must be designed to work autonomously with well-defined interfaces representing their domain. Separation is crucial as attempting to enforce a single, massive domain model counteracts the adaptation of a decentralized context. Likewise, it is essential to avoid enforcing domain models through a value chain, as market and technical contexts might not align well.

In conclusion, domain modeling should carefully consider organizational preferences, use cases, and the need for historical data while ensuring a decentralized context and well-defined interfaces. Domains should be split into logical sub-domains that bring value, allowing for better data management and utilization.

**T8: The Complexities of Flexibility and Grid Balance**

The transition to renewable energy sources, such as wind and solar, increases energy systems' complexity and variability. Ensuring grid balance and flexibility is crucial to accommodate these fluctuations and maintain a consistent energy supply.

Power-to-x technologies, such as hydrogen or ethanol production, are vital in converting and storing excess electricity for later use. Despite conversion losses, they allow for better utilization of surplus energy, as the existing gas networks can be used for transportation and storage.

The increase in sustainable energy necessitates gathering more information to manage and predict energy systems effectively. As the proportion of sustainable energy grows, the need for information transfer and the complexity of energy systems increases exponentially. Innovative approaches like simulations may be required to cope with these challenges.

Sector coupling and flexibility rely on the interconnection of various energy sectors with accurate and timely information. Integrating **IoT** devices and communication technologies like **LoRaWAN** can be enablers in this regard and improve grid balance by providing real-time price or $\mathbf{CO_2e}$ forecasts.

Consumer participation is essential for grid flexibility, with up to 40% of the desired flexibility potentially achievable through consumer involvement. Electric vehicles, for instance, can offer synthetic inertia by adjusting charging power and using regenerative braking to extend electricity consumption.

Grid balance can be further improved through innovative solutions such as packetized energy prioritization, which involves households and units communicating their intent and central systems responding with feasibility and timing. This approach has shown promising results in balancing the grid with available sustainable energy in England.

In conclusion, flexibility and grid balance in the era of sustainable energy require innovative approaches, such as power-to-x technologies, sector coupling, advanced information systems, and consumer participation. Integrating these strategies can establish a more diverse, complex, and robust energy system, ensuring grid stability and a sustainable future.

**T9: The Perception of Governance**

Effective governance requires a delicate balance between confidentiality, transparency, and information management. As the energy industry becomes increasingly interconnected and data-driven, responsibly handling and sharing sensitive data is crucial for business operations and compliance with regulations.

Confidentiality is essential to protect business-critical information. At the same time, transparency is necessary for sharing energy information in specific sectors to maintain grid balance and promote collaboration.

Organizations like Energinet must develop information and communication models that allow for the responsible sharing of sensitive information to navigate this balance. The models include delegation and access control mechanisms that grant time-limited and role-based access to data, ensuring that information is available only to authorized parties.

Data cleansing and compliance are also critical components of governance. Users should be able to request data deletion and have transparency in the process. Organizations must ensure compliance with data protection regulations and maintain records of actions taken.

Residency and discoverability of data contribute to effective governance. Residency allows customers to control the zones in which their data is available and ensures that datasets are

optimally distributed. Discoverability is obtained by providing an overview of data products, their usage, and contracts to help maintain an organized and compliant data environment.

Finally, establishing a structure and metadata description system can facilitate data understanding and enrichment. Such a system can enable organizations to harness their data for new insights and capabilities while maintaining governance standards.

In conclusion, effective governance in the energy sector involves balancing confidentiality and transparency, implementing robust information management systems, ensuring data cleansing and compliance, and promoting data residency, discoverability, and enrichment. Organizations can establish a robust governance framework that promotes responsible data sharing by addressing these aspects.

**T10: The Digital Infrastructure**

Digital infrastructure in the energy sector should focus on convergence, orchestration, security, and adaptability to modern approaches. Developing a unified, flexible, and secure digital infrastructure as the industry evolves is crucial to support long-term growth and innovation.

Convergence is essential for consolidating operational and analytical systems into a coherent platform. This approach facilitates efficient data storage, computing, and access, enabling organizations to leverage improved performance and resource management.

Orchestration plays a crucial role in maximizing the value of physical infrastructure investments. Organizations can drive innovation and ensure the seamless integration of new technologies and solutions by effectively coordinating energy systems and optimizing their usage.

Security is a foundational aspect of digital infrastructure in the energy sector. As the risk of cyber attacks grows, protecting energy systems using closed networks or private communication channels, such as registering as a **MVNO**, is critical. Additionally, adopting **OT** standards that prioritize accessibility, integrity, and confidentiality of data can help safeguard against threats and maintain the reliability of the energy grid.

Adaptability is vital for maintaining robust digital infrastructure in the long term. It includes embracing non-traditional communication standards that ensure longevity and incorporating on-the-edge technology to deploy functionalities to edge devices. Embracing modern approaches like auto-provisioning and DataOps further streamlines infrastructure management and ensures that digital systems are agile and responsive to industry changes.

In conclusion, a digital infrastructure in the energy sector emphasizes the importance of convergence, orchestration, security, and adaptability. Organizations can build a digital infrastructure to drive innovation for decades by focusing on these core principles.

**T11: The Physical Infrastructure**

In the energy sector, the physical infrastructure must adapt to green energy, leverage existing systems, and incorporate decentralization and modern communication technologies. By focusing on these core aspects, Denmark can develop a resilient, efficient, and sustainable physical infrastructure that satisfies evolving needs of the energy industry.

The green transition is a central challenge for physical infrastructure in the energy sector. As the industry shifts from relying on central power plants to green energy sources, it is crucial to redesign and reorganize existing infrastructure to accommodate these changes. It requires strategic planning and investment to ensure a seamless transition and maintain the stability of the energy grid.

Leveraging existing systems, such as gas networks, storage, and cogeneration plants, can optimize physical infrastructure. Gas networks and storage facilities can provide an alternative energy source during periods of high demand, while cogeneration plants can offer flexibility through their ability to utilize various fuel types and store surplus energy in district heating systems. These systems can enhance the overall capacity and resilience of the energy infrastructure.

Decentralization is an essential aspect of modern physical infrastructure. Due to the limitations in data transmission speed, it is necessary to adopt a decentralized approach that incorporates data mesh, data spaces, or other distributed information setups. Doing so can improve performance, help organizations manage data more effectively, and reduce reliance on centralized systems.

Modern communication technologies play a significant role in the development of physical infrastructure. As the deployment of charging stands for electric vehicles increases, organizations must ensure that these stations can communicate effectively using technologies like 4G. Additionally, installing fiber networks alongside power lines can create closed networks that improve data transmission and security.

In conclusion, improvements to the physical infrastructure in the energy sector emphasize the importance of adapting to green energy, leveraging existing systems, incorporating decentralization, and integrating modern communication technologies. Organizations can develop a robust and sustainable physical infrastructure by focusing on these core principles.

**T12: Legislation and Regulation**

In the energy sector, legislation and regulation should balance allowing technology innovation to drive progress and ensuring robust standards to maintain consistency and interoperability across the industry. By adopting this approach, a reliable regulatory environment that supports the ongoing evolution of the energy sector can be established.

Technology should guide the development of legislation and regulation in the energy sector. By allowing technology innovation to lead the way, organizations can address pressing issues and drive advancements in the industry. Once these innovations have been proven and adopted, legislation can be developed to formalize and standardize best practices.

Standards are crucial in maintaining consistency and interoperability across the energy sector. These standards are often developed incrementally globally, eventually becoming widely acknowledged and published as **IEC**, **ISO**, or **ANSI** standards. At this point, regional or national regulators can adopt these standards as the basis for regulation.

However, the maturity for implementing standards in the industry is not always high. It is often assumed that operators, developers, and suppliers will consider and adhere to these standards, but this is not always true. As a result, many standards are not followed or are implemented improperly, leading to inconsistencies and potential interoperability issues.

To address this, regulators should encourage the adoption of well-established and robust standards like **IEC**-61850. By focusing on widely recognized and proven standards, regulators can help ensure more compliance and consistency across the industry. Additionally, using well-established standards in supply laws can further promote adherence to these best practices.

Flexibility is also essential when implementing regulated standards. While maintaining consistency is crucial, enforcing a single standard across various contexts is often impossible. As such, regulators should allow for some interpretation and adaptation of standards at the local level. This flexibility can help organizations apply standards to best suit their needs and circumstances.

In conclusion, legislation and regulation in the energy sector emphasize the importance of striking a balance between technology innovation, the implementation of robust standards, and flexibility

in interpreting these standards. It is a prerequisite for establishing a regulatory environment to ensure a consistent and interoperable foundation for industry growth.

**T13: Metadata and its Future Role**

Metadata is essential for understanding, discovering, and trusting data in various domains, and it serves as a contextual bridge between raw data and its insights.

Metadata plays two primary roles: (1) providing context for the data and (2) describing the actual data. With these roles in mind, metadata supports discoverability, enabling users or systems to find relevant data based on specific queries or criteria.

In practice, multiple metadata standards exist within each domain. Metadata aggregators must be employed to transform data from one standard to another, which can help ensure seamless interoperability and discovery. This process may involve data loss but is a necessary trade-off for effective data integration and usage.

Metadata is domain-specific and should be configurable according to the needs of each domain. At a minimum, all datasets should be documented with metadata covering ownership, schema, data quality, lineage, context, data access, and glossaries. This level of documentation enables efficient searching and establishing trust in the data.

Metadata storage depends on various factors such as usage, complexity, and price. Organizations must carefully consider these factors when determining where to store metadata to ensure optimal performance and accessibility.

In summary, a unified theory of metadata emphasizes the crucial role of metadata in understanding, discovering, and trusting data across multiple domains. By developing a robust and flexible metadata infrastructure, organizations can ensure seamless data management, transformation, and discovery, ultimately enabling them to derive valuable insights and drive informed decision-making.

**T14: The Balance Between Open-Source and Proprietary Software**

The debate between open source and proprietary approaches in technology development revolves around the level of openness and collaboration desired in different aspects of a project. Open source enables knowledge sharing, collaboration, and learning across various project layers, while proprietary solutions often provide specific tools or technologies that are easy to understand and implement.

It is essential to consider the context and goals of each project or solution, especially in striking a balance between the two approaches. Open-sourcing the design, code, and application layers may be beneficial for fostering innovation and promoting transparency. However, the extent of openness should be determined on a case-by-case basis, as different layers might require different levels of openness.

A hybrid approach can be adopted, where open-source principles are applied to solution development while utilizing proprietary tools, technologies, and programming languages that facilitate understanding and ease of implementation. This combination encourages progress by promoting knowledge sharing and collaboration while leveraging the benefits of proprietary resources.

For example, in the context of a data platform, open-source principles can be applied at the design and application levels to ensure transparency and foster collaboration. At the same time, proprietary tools and technologies can streamline development and improve efficiency.

In summary, open source vs. proprietary emphasizes the need to balance the two approaches by carefully considering the context and goals of each project. Organizations can foster innovation,

collaboration, and learning while maximizing the efficiency and effectiveness of their technology development efforts by adopting a hybrid approach that combines the benefits of both open-source and proprietary solutions.

### T15: The Roles and Actors

In the context of energy systems and the green transition, there is a complex interplay of various roles and actors, such as **TSOs**, **DSOs**, **EU**, and **DEA**. Each actor has specific responsibilities and functions contributing to the overall design, operation, and regulation of energy infrastructure and markets.

Energinet is critical to this landscape. Its responsibilities include designing a coherent, cost-effective, and climate-friendly energy system that maximizes societal value. Energinet operates under the role of a **TSO**, which involves balancing both electrical and gas grids and managing transmission operations on a national level. However, Energinet does not maintain the energy infrastructure for distribution to consumers, as this responsibility falls upon **DSOs**.

In addition to its design and operational roles, Energinet is also responsible for shaping Denmark's energy markets and collaborating with European countries to create energy trading markets at the regional level. The tasks involve engaging with legislative bodies such as the **DEA** in national and European contexts.

The **TSOs** and **DSOs**, meanwhile, operate within a regulatory framework established by the **EU** and adhere to standards such as **IEC**-61850. This framework governs the interactions and coordination between these actors and ensures the proper functioning of the energy system.

In summary, roles and actors emphasize the interconnectedness and interdependence of various stakeholders in the energy sector. The complex interplay between these actors is crucial for designing, operating, regulating, and transitioning energy systems toward a more sustainable future.

### T16: Prioritized Software Qualities

In developing and maintaining software systems, particularly in critical infrastructure like energy systems, it is essential to consider various software qualities:

1. Performance: The efficiency of software depends on the data being processed and the system's requirements. In some cases, a slight delay in processing might not impact the system, but in others, it can be critical, such as delayed signals for balancing electrical grids.
2. Functionality vs. Usability: Striking a balance between features and usability is crucial. While adding numerous features can enhance software functionality, it may also lead to user misuse, increased maintenance costs, and decreased usability.
3. Stability: Software systems must be resilient and capable of handling unexpected situations, such as the rapid shift to remote work during the COVID-19 pandemic. A stable system is known for maintaining functionality even under challenging circumstances.
4. Robustness: In the supply sector, robustness refers to building infrastructure that avoids single points of failure. A robust system can continue functioning even if individual components fail or are compromised.
5. Security: Adhering to the **CIA** principles, software must prioritize confidentiality, integrity, and availability. Doing so involves encrypting sensitive data, ensuring data integrity, and implementing authentication and authorization mechanisms.
6. Integrity: Data integrity is essential for maintaining trust in software systems. One way to achieve this is by digitally signing messages rather than encrypting them, which helps ensure that messages remain immutable and verifiable.

In conclusion, software qualities highlight the importance of specific needs that underpin the energy sector. Balancing these qualities ensures that software systems can effectively meet the needs of users and stakeholders.

#### T17: Business Users and Data Scientists

Considering users' differing needs and behaviors, particularly business users and data scientists, is essential in data-driven systems. Recognizing the distinctions between these user categories allows for developing software systems that cater to their specific requirements and preferences.

1. Query Patterns: Business users typically exhibit repetitive query patterns, seeking specific information to support their decision-making processes. In contrast, data scientists often explore data more flexibly to uncover novel insights, correlations, and trends.
2. Data Access: Business users rely on predefined query models, allowing them to access and retrieve data efficiently without needing extensive technical knowledge. On the other hand, data scientists often require access to raw data or domain models, as their work involves deeper data analysis and manipulation.

In conclusion, understanding business users' and data scientists' distinct needs and behaviors are highlighted. By catering to these differences, software systems can be designed and developed to provide tailored experiences that better support the specific requirements of each user group.

## N.2 Hypotheses

**H1:** Increasing information availability and sector coupling will lead to a more diverse and flexible energy sector (**T1**).

**H2:** Companies with higher levels of digital maturity will be more successful in integrating new technologies and advancements (**T1**).

**H3:** Resistance to regulatory changes may slow down the energy sector's transformation compared to other industries (**T1**).

**H4:** Improved collaboration, communication, and adaptability among stakeholders are necessary for a well-functioning business ecosystem in the energy sector (**T1**).

**H5:** A balanced combination of centralization and decentralization is crucial for the effective functioning of the energy sector (**T2**).

**H6:** Different actors in the energy sector may have varying preferences for centralized or decentralized solutions (**T2**).

**H7:** A predominantly decentralized system will enable users to communicate independently and establish trust through existing centralized registers (**T2**).

**H8:** Overcoming challenges related to legacy procedures, data management, and local contexts is essential for transitioning towards a more decentralized energy sector (**T2**).

**H9:** Establishing a robust digital infrastructure will facilitate seamless information exchange and improved collaboration in the energy sector (**T3**).

**H10:** Overcoming data silos, outdated collaboration processes, and developing incitement structures and communication protocols will enhance collaboration in the energy sector (**T3**).

**H11:** Exposing domain models and fostering trust between parties will lead to more efficient and transparent data sharing (**T3**).

**H12:** Embracing a more decentralized approach and digitalized processes will improve collaboration efforts in the European energy sector (**T3**).

**H13:** Adopting suitable technologies and data formats will facilitate efficient data exchange and real-time querying (**T4**).

**H14:** Creating an incubator area for data scientists will improve data analysis and operationalization (**T4**).

**H15:** Ensuring data quality and appropriate data storage solutions are essential for effective data management (**T4**).

**H16:** Retention policies and storage tiering

can optimize resource utilization and support a flexible data management strategy (**T4**).

**H17:** Data Mesh implementation fosters a decentralized, trust-based ecosystem emphasizing discoverability, observability, and collaboration (**T5**).

**H18:** Contract-based trust systems can enhance data ownership and enable efficient error reporting and resolution (**T5**).

**H19:** Data Meshes built on existing technologies can provide a comprehensive data management solution (**T5**).

**H20:** The success of a data mesh depends on the organization's maturity, skillset, and ability to adapt to cultural and technical changes (**T5**).

**H21:** Data spaces enable seamless interaction and exchange among various actors and systems by incorporating interoperability features (**T6**).

**H22:** Effective governance is crucial for maintaining trust, security, and data-spaces compliance (**T6**).

**H23:** Data spaces must be adaptable to specific domain requirements and organizational needs (**T6**).

**H24:** Scalability considerations in data spaces are essential for accommodating evolving needs and minimizing negative impacts on efficiency and performance (**T6**).

**H25:** The choice between domain models and data vaults is influenced by organizational preferences, use cases, and historical data preservation needs (**T7**).

**H26:** Pseudo-data vaults can compromise domain models and data vaults, offering flexibility and historical data preservation (**T7**).

**H27:** Ensuring a decentralized context and well-defined interfaces is essential for effective domain modeling in the energy sector (**T7**).

**H28:** Power-to-x technologies are crucial for converting and storing excess electricity from renewable energy sources (**T8**).

**H29:** The increase in sustainable energy necessitates more information and innovative approaches for managing and predicting energy systems (**T8**).

**H30:** Integrating **IoT** devices and communication technologies can enable better grid balance and sector coupling (**T8**).

**H31:** Consumer participation can significantly improve grid balance in sustainable energy systems (**T8**).

**H32:** Balancing confidentiality and transparency is essential for effective governance in the energy sector (**T9**).

**H33:** Implementing robust information and communication models can facilitate the responsible sharing of sensitive information (**T9**).

**H34:** Compliance with data protection regulations and ensuring data erasure are critical components of governance in the energy sector (**T9**).

**H35:** Promoting data residency, discoverability, and enrichment can contribute to a robust governance framework that supports the evolving needs of the energy industry (**T9**).

**H36:** Convergence of operational and analytical systems into a coherent platform is essential for efficient data storage, computing, and access (**T10**).

**H37:** Orchestration is crucial for maximizing the value of physical infrastructure investments and driving innovation (**T10**).

**H38:** Adopting **OT** standards prioritizing accessibility, integrity, and data confidentiality can improve energy sector security (**T10**).

**H39:** Embracing non-traditional communication standards and modern approaches, like auto-provisioning and DataOps, can ensure a more flexible and responsive digital infrastructure (**T10**).

**H40:** Redesigning and reorganizing existing infrastructure to accommodate green energy sources is crucial for a seamless transition to sustainable energy (**T11**).

**H41:** Leveraging existing systems, such as gas networks and cogeneration plants, can optimize physical infrastructure and enhance capacity and resilience (**T11**).

**H42:** A decentralized approach, incorporating data mesh or distributed information setups, is essential for effective data management and reducing reliance on centralized systems (**T11**).

**H43:** Integrating modern communication tech-

nologies, such as 4G and fiber networks, can improve data transmission and security in physical infrastructure (**T11**).

**H44:** Allowing technology innovation to guide the development of legislation and regulation can address pressing issues and drive advancements in the energy sector (**T12**).

**H45:** Implementing well-established and robust standards, like **IEC**-61850, can ensure more compliance and consistency across the industry (**T12**).

**H46:** Many standards in the energy sector are not followed or are implemented improperly, leading to inconsistencies and potential interoperability issues (**T12**).

**H47:** Flexibility in interpreting and adapting standards at the local level is essential for organizations to apply standards that best suit their needs and circumstances (**T12**).

**H48:** Metadata aggregators are essential for seamless interoperability and data discovery across multiple metadata standards (**T13**).

**H49:** Domain-specific metadata, configurable according to the needs of each domain, can enhance data searching and trust (**T13**).

**H50:** Establishing a metadata agency and employing aggregators to convert between different data standards facilitate seamless data discovery, transformation, and usage (**T13**).

**H51:** Careful consideration of factors like usage, complexity, and price is necessary for determining optimal metadata storage (**T13**).

**H52:** A hybrid approach that combines open-source principles with proprietary tools and technologies can maximize innovation and efficiency in technology development (**T14**).

**H53:** The extent of openness in a project should be determined on a case-by-case basis, with different layers requiring different levels of openness (**T14**).

**H54:** Open-sourcing the design, code, and application layers can foster innovation and promote transparency in a data platform (**T14**).

**H55:** Utilizing proprietary tools and technologies can streamline development and improve efficiency in an open-source project (**T14**).

**H56:** Energinet's responsibilities include designing a coherent, cost-effective, and climate-friendly energy system, balancing electrical and gas grids, and shaping Denmark's energy markets (**T15**).

**H57:** **DSOs** are responsible for maintaining local energy infrastructure (**T15**).

**H58:** The **TSOs** and **DSOs** operate within a regulatory framework established by the **EU**, adhering to standards such as **IEC**-61850 (**T15**).

**H59:** The successful design, operation, regulation, and transition of energy systems towards a more sustainable future depends on the interconnectedness and interdependence of various stakeholders in the energy sector (**T15**).

**H60:** Balancing performance, functionality, usability, stability, robustness, security, and integrity is essential for software systems, particularly in critical infrastructure sectors like the energy sector (**T16**).

**H61:** Striking a balance between features and usability can help prevent user misuse, decrease maintenance costs, and enhance software effectiveness (**T16**).

**H62:** Robust systems that avoid single points of failure can continue functioning even if individual components fail or are compromised (**T16**).

**H63:** Ensuring data integrity is crucial for maintaining trust in software systems and can be achieved through methods such as digitally signing messages (**T16**).

**H64:** Catering to business users and data scientists' distinct needs and behaviors is essential for designing software systems that effectively support their specific requirements (**T17**).

**H65:** Business users typically exhibit repetitive query patterns and rely on predefined query models, while data scientists often explore data more flexibly and require access to raw data or domain models (**T17**).

**H66:** Recognizing the distinctions between business users and data scientists allows for developing tailored software systems that enhance their respective user experiences (**T17**).