

Consent Management in Data Spaces

A Reference Architecture for Consent Management in Data Spaces

Marc van Duyn

`marc.vanduynduyn@student.uva.nl`

Wednesday 2nd December, 2020, 71 pages

Academic supervisor: Paola Grosso, Uva, `P.Grosso@uva.nl`

Academic supervisor: Lu Zhang, Uva, `l.zhang2@uva.nl`

Daily supervisor: Harrie Bastiaansen, TNO, `harrie.bastiaansen@tno.nl`

Daily supervisor: Maarten Kollenstart, TNO, `maarten.kollenstart@tno.nl`

Host organisation/Research group: TNO, <https://www.tno.nl>



UNIVERSITEIT VAN AMSTERDAM

FACULTEIT DER NATUURWETENSCHAPPEN, WISKUNDE EN INFORMATICA

MASTER SOFTWARE ENGINEERING

<http://www.software-engineering-amsterdam.nl>

Abstract

Data spaces provide data access and data usage control techniques to enforce data usage permissions and obligations for the exchanged data between actors. Data owners can use these techniques to ensure data sovereignty over their data. Using these data control systems introduces the challenge of realizing the processes and components needed to implement the definition, administration, negotiation and management mechanisms of data usage permissions and obligations. Additionally, data spaces commonly distinguish between the role of the data owner and the data provider, where providing the data done by the data provider and defining the permissions and obligations is responsibility of the data owner. This distinction between data owners and data providers results in 'many-to-many' relationships, where a data provider provides data for multiple data providers and a data owner will have multiple data providers providing its data. This also enables a data provider to provide data sets that are composed of data entries that originate from different data owners. This means that data owners often do not know in which data set their data is contained. Therefore, data owners are requesting for data control capabilities and systems that prevent their shared data from being misused and enforce their usage constraints no matter in which context their data is used.

Consent management can be used to facilitate processes, where consent management as a system or a set processes can be used to define, exchange and manage data usage permissions and obligations. In data spaces, consent management can be seen as an abstraction layer on top of the data access and data usage control systems. This research identifies and solves the challenges of implementing consent management in data spaces. This work provides the requirements of implementing consent management in data spaces, components that realize these requirements and a reference architecture for consent management in data spaces.

Contents

1	Introduction	1
1.1	Access and Usage Control in Data Spaces	1
1.2	Consent Management in Data Spaces	2
1.3	Challenges in Consent Management in Data Spaces	2
1.4	Research Outline	3
1.4.1	Research Questions	3
1.4.2	Research Method	4
1.5	Contributions	5
1.6	Outline	5
2	Background	6
2.1	Data Exchange	6
2.2	Data Spaces	6
2.2.1	Data Access and Usage Control in Data Spaces	7
2.3	Consent Management	8
2.3.1	Consent and Data Management Model	9
2.3.2	Requirements for Implementation of Consent Management	10
2.3.3	Processes of Consent Management	10
2.4	Permissions and Obligations in Consent Management	10
2.5	Right Expressions Languages (RELs)	11
2.5.1	Open Digital Rights Language (ODRL)	11
2.5.2	Extensible Access Control Markup Language (XACML)	11
2.5.3	Policy Types	11
2.6	Technical Enforcement of Policies in DRM Systems	12
2.6.1	Architectural Elements for Policy Control in Data Spaces	12
3	Problem Statement	15
3.1	Consent Management in Data Spaces	15
3.1.1	Policies Creation and Orchestration	15
3.1.2	Composed Data Sets and Distinction of Data Owner and Data Provider	15
3.2	Problems in Consent Management Tackled by this Thesis	16
3.3	Problems Related to the Use Case	16
4	System Design for Consent Management in Data Spaces	18
4.1	Problem and Motivation	18
4.2	Objectives of a Solution	18
4.2.1	Processes in Consent Management	18
4.2.2	Defining of Permissions and Obligations of Consent	19
4.2.3	Registration of Permissions and Obligations and Linking with a Data Set	19
4.2.4	Negotiation of Permissions and Obligations	20
4.2.5	Forming of Policies of Consent-Based of the Permissions and Obligations Given	21
4.2.6	Registration and Exchanging of Policies of Consent	22
4.2.7	Revoking of Policies of Consent	22
5	System Implementation for Consent Management in Data Spaces	23
5.1	Consent Manager	23
5.1.1	Intent	23

5.1.2	Motivation	23
5.1.3	Requirements Implementation	24
5.1.4	Applicability	25
5.1.5	Participants	25
5.1.6	Collaborations	25
5.1.7	Consequences and Benefits	27
5.1.8	Implementation Considerations	27
5.2	Policy Catalogue	28
5.2.1	Intent	28
5.2.2	Motivation	28
5.2.3	Requirements Implementations	28
5.2.4	Applicability	29
5.2.5	Collaborations	29
5.2.6	Consequences and Benefits	32
5.2.7	Implementation Considerations	32
5.3	Policy Broker	33
5.3.1	Intent	33
5.3.2	Motivation	33
5.3.3	Requirements Implementation	33
5.3.4	Applicability	33
5.3.5	Participants	33
5.3.6	Collaborations	34
5.3.7	Consequences and Benefits	35
5.3.8	Implementation Considerations	35
6	System Demonstration for Consent Management in Data Spaces	36
6.1	Basic Data Space Architecture Without Consent Management	36
6.2	Scenario Reflected in the Proposed Architecture	37
6.3	Processes of Consent Management Reflected in the Proposed Architecture	37
6.3.1	Proposed Architecture of Consent Management	37
6.4	Process Interactions	38
6.4.1	Defining of Permissions and Obligations of Consent	38
6.4.2	Registration of Permissions and Obligations and Linking with a Data Set	39
6.4.3	Forming of Policies of Consent-Based of the Permissions and Obligations Given	40
6.4.4	Registration and Exchanging of Policies of Consent	42
6.4.5	Revoking of Policies of Consent	43
6.4.6	Searching for Policies with Permissions and Obligations of Intent	43
6.4.7	Implementation Concerns	44
7	Use Case Demonstration	45
7.1	Setup	45
7.2	Representation of the Reference Architecture	45
7.2.1	Data Provider Component	46
7.2.2	Consent Manager Component	48
7.2.3	Policy Catalogue Component	53
7.2.4	Data Consumer Component	53
7.2.5	Policy Broker Component	56
7.2.6	Permissions and Obligations Mapping Mechanism	58
7.2.7	Context Attributes	58
7.3	Applicability to the Use Case	59
8	Discussion	60
8.1	Findings of the Research	60
8.2	Evaluation and Limitations of the Prototype	61
9	Conclusions	62
9.1	Answers to Research Questions	62
9.2	Research Limitations	63
9.2.1	Proposed Components	63

9.2.2 Architecture Validation	63
10 Future Work	64
10.1 Consent Management in Different Data Spaces	64
10.2 Permissions and Obligations Definitions	64
10.3 Permissions and Obligations Mapping	64
10.4 Obligations and Permissions Legal Validity	64
10.5 Policies Legal Validity	65
10.6 Context Specification	65
Bibliography	67
Appendix A Terminology	70
A.1 Data Spaces	70
A.2 Consent Management	71

List of Figures

2.1	Interaction of Technical Components in Data Spaces [2]	7
2.2	Semantic Model of Consent as Proposed in [28]	8
2.3	Consent and Data Management Model as Proposed in [28]	9
2.4	Single Network Node of a PEP Interacting with a PDP	12
2.5	XACML Policy Enforcement Components and Interaction Flow [32]	14
5.1	Interactions of Data Owner with Consent Manager	25
5.2	Interactions of Data Provider with Consent Manager	26
5.3	Interactions of Data Consumer with Consent Manager	26
5.4	Revoking of Consent Policy by the Consent Manager	29
5.5	Policy Registrations Interactions of Consent Manager with Policy Catalogue	30
5.6	Interactions of Data Consumer with Policy Catalogue	31
5.7	Interactions of Data Provider with Policy Catalogue	32
5.8	Interactions of Consent Manager with Policy Broker	34
5.9	Interactions of Data Consumer with Policy Broker	35
6.1	Basic Data Space Architecture Without Consent Management	36
6.2	Proposed Architecture with Consent Manager, Policy Catalogue and Policy Broker Components	38
6.3	Sequence Diagram of Data Owner Interactions with Consent Manager	39
6.4	Sequence Diagram of Data Provider Interactions with Consent Manager	40
6.5	Sequence Diagram of Data Consumer Interactions with Consent Manager	41
6.6	Sequence Diagram of Intent of Usage Permissions and Obligations Validation and Negotiation	41
6.7	Sequence Diagram of Retrieval of Policies	42
6.8	Sequence Diagram of Revoking of Consent by a Data Owner	43
6.9	Sequence Diagram of Get a Selection of Consent Manager with Intent of Usage Permissions and Obligations	44
7.1	REST Endpoints for Data Owner-Related Services at the Data Provider	46
7.2	REST Endpoints for Data Set-Related Services at the Data Provider	46
7.3	Dashboard of Registration and Listing of Data Owners	47
7.4	Dashboard of Registration and Listing of Data Sets	47
7.5	Dashboard Overview of the Data Provider	48
7.6	REST Endpoints for Data Provider Related Services at the Consent Manager	48
7.7	REST Endpoint for Data Set Related Services at the Consent Manager	49
7.8	REST Endpoints for Data Owner Related Services at the Consent Manager	49
7.9	REST Endpoints for Data-Permission Related Services at the Consent Manager	49
7.10	Permission Registration Request Body	49
7.11	REST Endpoints for Obligation-Related Services at the Consent Manager	50
7.12	Obligation Registration Request Body	50
7.13	REST Endpoints for Policy-Related Services at the Consent Manager	50
7.14	Policy Request Post Object	50
7.15	Policy Request Response Object	51
7.16	Dashboard of Listing of Data Sets of a Selected Data Provider	51
7.17	Dashboard of Defining of Permissions and Obligations of Data Owners	52
7.18	REST Endpoints for Policy Related Services at the Policy Catalogue	53

7.19	Dashboard Overview of the Policy Catalogue	53
7.20	REST Endpoints for Data Set Related Services at the Data Consumer	54
7.21	REST Endpoints for Data Policy Related Services at the Data Consumer	54
7.22	Dashboard of Data Set Related Services at the Data Consumer Component	55
7.23	Dashboard of Obtained Data Set Related Services at the Data Consumer	56
7.24	REST Endpoints for Registration Related Services	56
7.25	REST Endpoints for Selection Related Services	57
7.26	Selection Request Body	57
7.27	Dashboard of the Policy Broker	58
A.1	Reference Architecture of International Data Spaces [2].	71

Chapter 1

Introduction

This thesis aims to present an architecture for the implementation of consent management in data spaces.

Data spaces are an implementation of a network approach for data sharing to provide a solution for a trusted, controlled, and secure way to exchange data in a bilateral manner[1], [2] [3]. In data spaces techniques are used that focus on data integration, data management and privacy preserving data exchange. Data spaces and other data exchange techniques contribute the goal of Europe's digital transformation over the next five years to establish effective frameworks to ensure trustworthy technologies, and to give businesses the confidence and means to digitise[4].

The key criteria for organizations to participate in Data Spaces is the potential of structured exchange, collection, and analysis of data with their stakeholders and customers[1]. Additionally, the decentralized characteristic of data spaces makes it a good alternative to the more centralized hub models. However, the widespread adoption of data spaces has still some challenges. These challenges can especially be seen in sharing an increasing amount of data with a growing number of data consumers, whilst adhering to regulations (e.g., General Data Protection Regulation (GDPR)[5]), and whilst maintaining data control interests (e.g., managing access and usage of their data).

Data spaces aim to solve the request for data control by utilizing data control techniques similar to digital rights management systems (DRM). The data control systems can enforce data usage control based on the permissions and obligations of the data owners. Using these data control systems introduces the challenge of realizing the processes and components needed to implement definition, administration, negotiation and management of data usage permissions and obligations. To facilitate these processes data spaces can make use of consent management to orchestrate the permissions and obligations. Where consent management is a system or set of processes that support the dynamic creation, management and enforcement of consumer, organizational and jurisdictional privacy directives[6].

1.1 Access and Usage Control in Data Spaces

In the context of access and usage control, data owners express their usage constraints over their data by controlling the permissions and obligations.

Data spaces commonly distinguish between the role of the data owner and the data provider. The responsibilities of specifying the permissions and obligations for data usage are part of the role of a data owner. Forming data sets from different data owners, making the data available, and exchanging it with data consumers are the responsibilities of the data provider. Traditionally, a participant acting as Data Owner automatically assumes the role of the Data Provider as well. However, there may be cases in which the Data Provider is not the Data Owner. Hence, the roles of the data owner and the data provider are viewed as two separate entities in this research. The view, that data owners and data providers are separate entities implies, that different data providers could potentially provide data of the same data owner. It also implies that the data owner may need to provide its permissions and obligations to multiple providers. The distinction between data owners and data providers results in 'many to many' relationships, in which a data owner can have multiple data providers and a data provider can have multiple data owners of its data sets. This enables a data provider to provide data sets to data

consumers that are composed of data entries that originate from different data owners. This means that data owners often do not know in which data set their data is contained. Therefore, data owners are requesting for data control capabilities and systems that prevent their shared data from being misused. In the context of access and usage control systems, the distinction follows that for each data owner in the data set, its permissions and obligations need to be enforced.

There is no commonly used system for access and usage control in data spaces. Data spaces can implement access and usage control similar to that of DRM systems. DRM systems focus on aspects such as languages for rights expression, digital objects declaration and protection information declaration [7]. However, DRM systems are commonly used for prevention of unauthorized redistribution of digital media, where data spaces focus on the prevention of unauthorized usage of data. In the context of DRM systems, permissions and obligations are specified in rights expressions language (REL). Hence, REL can be used to specify the permissions and obligations about the usage of data. The permissions and obligations which are written in rights expressions language can, in turn, be interpreted and enforced by the technical components in DRM systems. Within this thesis, consent management is viewed as a process on top of the DRM system.

An example of access and usage control in a data space similar to a DRM system is that of International Data Spaces (IDS) [2]. The IDS expresses policies that are based on permissions and obligations of the data owner, in RELs. These RELs are then exchanged as sticky policies to enforce data control within the IDS [8] [9].

1.2 Consent Management in Data Spaces

In the context of data spaces, this research defines consent management as a system or a set processes to define, exchange and manage data usage permissions and obligations. In data spaces, consent management can be seen as an abstraction layer on top of the access and usage control systems. The abstraction layer can be used to facilitate the process in which a common agreement is formed between the data owner and the data consumer. The agreement encompasses the permissions and obligations of users of the data, and enforcement of consumer, organizational, and jurisdictional privacy policies.

In the scope of this research, the term data spaces refer to a network to facilitate data exchange as a solution for the need for a trusted, controlled, and secure way to exchange data. The most concrete example of this definition would that of the reference architecture of International Data Spaces (IDS) [2] [10] [2]. As stated in the previous sections this research assumes that data access and usage control is expressed by defining permissions and obligations for the exchanged data. And where these permissions and obligations are enforced at the data consumer infrastructure[3]. With these specifications, consent management encompasses in this research:

- Defining of obligations and permissions for data usage.
- Negotiation of permissions and obligations for data usage.
- Forming of policies in right expression languages based on the permissions and obligations given.
- Registration and exchange of policies.

So far, there is no reference architecture or model for implementing consent management in data spaces, yet.

1.3 Challenges in Consent Management in Data Spaces

For this research, a case study is leveraged to make a use case for the energy production project of the Dutch Ministry of Economic Affairs and Climate "VIVET: betere informatie voorziening energie transitie"[11]. The VIVET project is aimed at improving the information provisioning of the energy transition. The Netherlands is facing the challenge of gathering reliable information about the current and foreseen development of the energy system in real-time and at low-cost to fulfil the objective of making data-driven decisions on the contributions to the energy transition. The outcome of the project recommended the establishment of a virtual data platform that connects data owners and data users, to tackle the current challenges in availability and usability of data.

The case study helps to outline the complexity of Consent Management in data spaces for the energy sector. One central topic of energy information provisioning is the use of smart meters in households. The smart meters allow households to measure and share data of their usage, production and storage of energy. If the number of households with smart meters grows, the accumulation of data becomes a valuable resource for commercial and research organizations. For example, the measured data can reveal information about households, such as energy usage, energy storage and energy generation. These kinds of data and information could potentially be provided by a multitude of data providers and shared with a large number of data consumers. Controlling the ownership of data and preventing misuse is therefore of great importance.

Consent management in this context would give a household the option to assign a data provider to share their data with interested or selected parties. Then a household, being the owner of the data it produces, could make use of data usage and access control to enforce their permissions and obligations at the data consumer level.

However, the process of specifying permissions and obligations, and eventually orchestrating these specifications across all data consumers becomes more complex, if the data is provided by a multitude of data providers. On top of this, it becomes even more complex, if the data is part of a composed data set, in which the data within the data set belongs to different data owners that select different kinds of permissions and obligations.

On the other hand, Consent Management could be introduced to automate some steps in process of requests and approval. Data consumers have a growing demand for large quantities of data. For example, an organization, as a data consumer, could be interested in gathering information on the average times a household uses large amounts of energy to predict the times its residents are at home. Consequently, the data consumer would have to go through the time-consuming process of requesting explicit approval of usage from each data owner that is part of the data set. Some of these steps could be automated with the aid of consent management.

The challenges in regulations, ownership and control over exchanged data create the need for a controlled, recurring, and a centralized way of managing and providing the permissions and obligations of data usage. The process of orchestrating the permissions and obligations is within the scope of consent management.

1.4 Research Outline

The following sub-chapters outline the underlying research, the research questions, and the methodology used to answer the research questions.

1.4.1 Research Questions

The objective of this thesis is to understand how consent management can be realized in data spaces. In the given context, the research focuses on defining an architecture that realizes consent management in data spaces. For the realization of this goal, this research uses a design science research (DSR) paradigm to come up with a new artefact. Consent management is not an entirely new concept in systems that facilitate the exchanging of data between different actors. However, as described in the introduction, data spaces introduce complex situations that make consent management more difficult. Additionally, judging from the current state of data spaces, implementing consent management in data spaces is a new concept [2] [8] [12] [13].

The research contributes to solving the challenges in consent management by analyzing what is needed for consent management in data spaces and providing an architectural model that realizes this. To create the architecture, three different types of concepts should be researched: 1) a set of requirements that defines consent management, 2) the creation of a reference architecture and 3) the introduction of components that realize this architecture. These three types of concepts are researched through the following research questions:

”How can Consent Management be realized in Data Spaces?”

To answer the main research question, the following sub-questions need to be answered:

1. *"What are the requirements that define Consent Management?"*

The objective is to create a mapping of the requirements that define consent management, and how this is translated into data spaces.

2. *"How do data spaces need to be changed in order to realize consent management?"*

The objective is to evaluate the requirements against the current state of data spaces. From this evaluation new artifacts for data spaces can be proposed to implement consent management in data spaces.

1.4.2 Research Method

This sub-chapter explains the research methodologies that were applied in this research. The general research paradigm used for this thesis is design science research applied to a use case.

Case Setting and Description

This research develops a reference architecture and components to realize consent management in data spaces. The solution, that is the outcome of this research, applies to a wide variety of use cases. For this research, an example use case was chosen to validate the results. The selected use case describes an exchange of sensitive data between multiple actors of smart energy grids.

Case Study Design

The case study performed focuses on a single case, and therefore can be categorized as a single-case study[14], [15].

The case description is representative of the energy production project described in "VIVET: betere informatie voorziening energie transitie"[11] that was proposed by the Dutch government. In the domain of energy grid management, actors can economize in generating, storing, and saving energy through the exchange of data with other parties. However, these same actors are reluctant to share their data, if they may lose data sovereignty in the process. A data space that implements data usage control, and that provides the components to realize consent management, could be the foundation to enable the willingness of actors to share their data.

Literature states that focusing one's attention on a single case provides the opportunity to study a complex phenomenon in-depth and with greater focus as opposed to multiple cases[16]. Consent management is especially characteristic for a smart grid network data space, because of the large number of data owners in data sets that contain data from a multitude of data owners. However, the findings of this thesis can also be leveraged as a basis for multiple case studies. Therefore, it is assumed that consent management for data spaces can be applied to other use cases as well.

Design Science Research

The design science research (DSR) paradigm is used for the realization and evaluation of the architecture and component options of the smart grid data space use case. The research was executed along the guidelines proposed in [17]. The reason for DSR is the creation of a new IT-artifact in the form of a set of architectural models and components. Also, DSR has gained increasing attention in the information systems research community who proposes DSR a suitable paradigm for the development of novel IT-artifacts [18]. Deliverables of DSR research projects are such as constructs, models, methods and instantiations [19]. The proposed architecture options fall into the second category of DSR contribution types provided by [17], as they classify as nascent design theory[20].

In [18] an outlining of the DSR model is given with the individual steps necessary toward generating an IT-artifact. The model consists of the following six steps: 1) identify problem and motivation,

2) define objectives of a solution, 3) design and development, 4) demonstration, 5) evaluation and 6) communication.[17] In the upcoming chapter ?? Research, these six steps are leveraged as the foundation to realize the architecture options.

1.5 Contributions

This research is the first effort to introduce architectural solutions to the challenges in consent and data rights management that were highlighted in section 1.3.

The preliminary research has found that there is no reference architecture or model to implement consent management in data spaces. Therefore, the architectural solutions that are the outcome of this thesis in the form of requirements, processes, and components help with implementing consent management in data spaces. Consequently, this thesis can form a reference for future and current data spaces.

This thesis makes the following contributions:

1. A translation of ontology's and semantic models of consent management into requirements that can be implemented in data spaces. This is primarily done in chapter 2 and chapter 4. An overview is made of all literature regarding consent management and data usage control in data spaces. From this, a set of requirements are identified that realize consent management.
2. A set of components that implement the requirements of consent management in data spaces. The components are introduced in chapter 5.
3. A reference architecture for realization of consent management with the proposed components. The reference architecture that is introduced in chapter 5 are implemented in chapter 6 and 7.
4. A prototype of data spaces with consent management implemented inline with the proposed reference architecture, where chapter 7 provides insights in how the prototype is realized.

1.6 Outline

In chapter 2 an explanation is given of the background to this thesis. It evaluates different literature resources and explains some key concepts

In chapter 3 a summarization is given of the challenges of consent management and data spaces that form the motivation to this thesis. This chapter is in line with the identification of the problem and the motivation for design science research.

Chapter 4 provides an set of requirements for realization of consent management. This chapter is in line with defining objectives of a solution in the form of requirements of design science research. Chapter 5 defines components that realize the requirements. This chapter is inline with design of design science research. Chapter 6 is the development chapter, where a reference architecture is proposed an overview of key processes is given. This chapter also contributes to the demonstration part of design science research.

The demonstration part of design science research is given in Chapter 7.

Chapter 8 discusses the results that are highlighted in chapter 7 and proposes a solution to the given use case Chapter 8. This chapter functions as the evaluation and communication part.

Chapter 10 provides future topics that a related to this thesis.

Finally, the concluding remarks of this thesis are made in Chapter 9.

Chapter 2

Background

2.1 Data Exchange

Data is a valuable asset in the emerging data economy. Therefore, organizations are gaining awareness of their value and evaluate how they plan to exchange data with other organizations or stakeholders [21], [22]. For example, changing market dynamics can force organizations to start sharing data in an and throughout supply chains [10].

The pressure and value to exchange data eventually creates the need for a trusted, controlled, and secure way to exchange data as a prerequisite for organizations to share their data with another party. This is also outlined in [23], where potential challenges have presented that need to be addressed to achieve privacy-preserving data integration and exchange. Companies could exchange information to boost productivity, but are prevented by fear of being exploited by competitors or antitrust concerns. Therefore, there is an emphasis on a solution that enables widespread integration and sharing of data with customers and other organizations while allowing easy and effective privacy control for the data owners.

Currently, data sovereignty concepts are mainly provided in communities. Those communities each offer their own specific data sovereignty solutions. This faces data providers with a threat of lock-in and major integration efforts in case of data sharing with a multitude of data consumers. As an alternative, a network-model approach for providing generic infrastructural data sovereignty can overcome these challenges [10]. These network-model approaches enable users to establish bilateral data exchange relations, making data exchange decentralized.

2.2 Data Spaces

This research focuses on data spaces. Data spaces are defined in this research as a peer-to-peer network of shared technical components that facilitate a trusted, controlled, and secure way to exchange data.

Data spaces are an implementation of a network approach for data exchange. Reference architectures and implementations of data spaces share common characteristics, such as a focus on the bilateral exchange of data between actors in the network and data control technologies to implement data sovereignty.

International Data Spaces (IDS) is an example of a data space solution and is mostly used in the European market. IDS is the closest implementation of a data space as defined in the terminology of this thesis A. IDS comes with a set of components and a reference architecture[2]. The main components that form the IDS data space are the Connector, the Broker, and the App Store. How these components interact with each other is depicted in Figure 2.1.

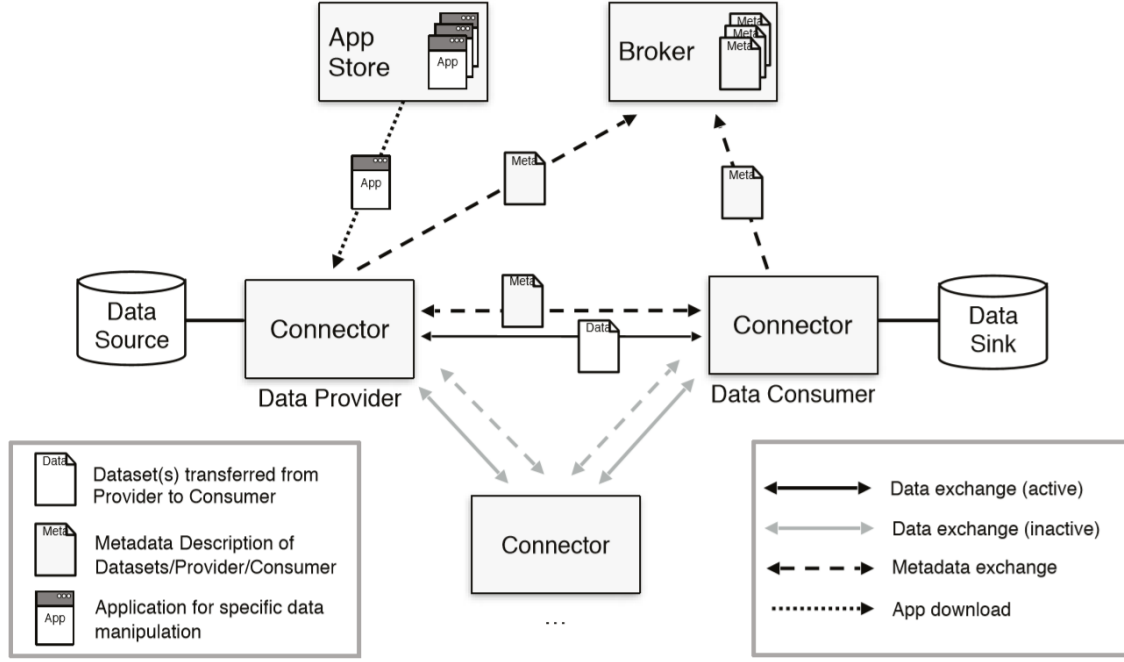


Figure 2.1: Interaction of Technical Components in Data Spaces [2]

Figure 2.1, taken from the IDS reference architecture [2], visualizes the base components and their interactions. The central component in this diagram is the connector. The connector is the technical component that connects the data provider with the data consumer. Connectors and their data objects are catalogued at a broker. This allows for searching of different data providers. Connectors can make use of app store applications that will add extra functionality to the data exchanges such as anonymization and pseudonymization. The Connector, the Broker, and the App Store in IDS are supported by four additional components. These are the identity provider, the Vocabulary Hub, the Update Repository and the Trust Repository. For this research, these components are not considered and out of scope. (see chapter A Appendix for more information)

Data spaces commonly make use of data control techniques to ensure data sovereignty. In reference architectures such as IDS, policies are used based on the data usage permissions and obligations a data owner defines for its data sets. These policies enable technical enforcement at the data consumer level and are exchanged between data providers and data consumers. A data consumer must obtain the policies that are linked to the data set to use it. The process of forming and exchanging policies is entirely dependent on the data space implementation in use. However, the process that is used for the creation and exchange of policies needs to be appropriate to the number of actors in the data spaces, to regulations (e.g., GDPR [5]), and to the data control interests of data owners (e.g., managing access and usage of data).

2.2.1 Data Access and Usage Control in Data Spaces

There is no common technique to ensure data sovereignty in data spaces. As stated in chapter 1, data spaces provide their own data control capabilities.

The literature proposes different concepts of data usage enforcement. The solutions range from organizational rules or legal contracts to complete technical enforcement of data access and usage permissions and obligations. Intermediate levels may contain parts of both enforcement manifestations.

In the context of this research, there is a focus on technical enforcement in combination with consent management. In network-based infrastructures, a common technique is to implement data access and usage control by using policy enforcement[24] [2]. These policies are based on the permissions and obligations that data owners define; where a policy is defined as the information and rules which influences the interactions between a subject and a target[25].

For the definition of the usage and access policies, data spaces commonly use Rights Expression Languages (REL)[8]. The most commonly used REL's are XACML [26] and ODRL[27].

2.3 Consent Management

Consent management is a system or process to define, exchange and manage data usage permissions and obligations. Consent management helps a data owner to determine what data is shared, for what purpose and under which circumstances. In consent management, the data consumer needs to get the explicit consent of the data owner to use the data. The purpose of giving consent is to grant permission to perform personal data processing for specified purposes[28].

Consent is a prerequisite to processing one's data, on the authority of the GDPR. According to Art. 6 of the General Data Protection Regulation (GDPR), consent of a data subject is one of the legal bases for processing personal data [5]. Furthermore, the GDPR specifies according to Rec. 32, 42, Art. 4, 6(2), 7 and 8[Gdpr:2018], that consent should be freely given, specific, unambiguous and informed. In addition to these requirements, Art. 7 (2)[Gdpr:2018] stated that the data subject should be able to withdraw the given consent as easily as it was given. Meaning that there should be processes in place to terminate the policies of consent in a system by the data owner.

These restrictions on consent requirements raise the need for data spaces to review their consent model to be compliant with the GDPR. As described in previous sections, it is possible, for systems that focus on exchanging data as well as in particular data spaces, to implement data protection techniques to ensure data sovereignty. However, how the data usage permissions and obligations are defined and exchanged before these data protection techniques can be used, is part of consent management. With GDPR imposing greater restrictions on obtaining consent, where the data controller must be able to demonstrate that consent was validly obtained, implies that failure to provide proof of the validity of obtained consent will be a breach of the legal requirement for GDPR compliance[5].

Generally, consent management implementations are system or use case-specific, which in turn adds complexity to the subject. In [28] an ontology is given to capture the requirements for consent management mechanisms. Where an ontology is commonly defined as: "an explicit specification of a conceptualization" [29]. The ontology proposed by [28] can be seen in Figure 2.3.

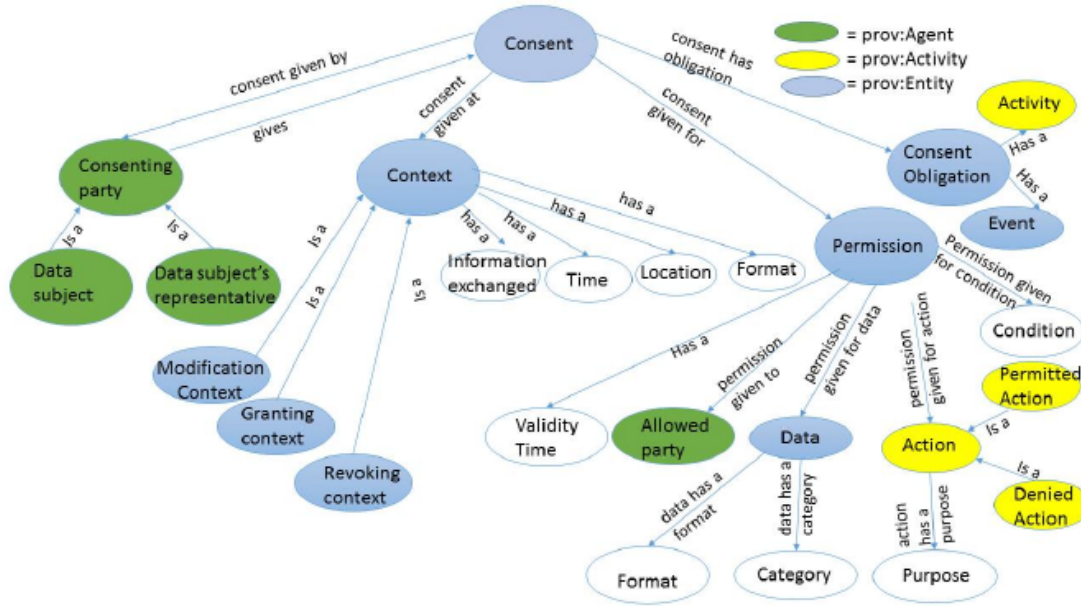


Figure 2.2: Semantic Model of Consent as Proposed in [28]

The semantic model in Figure 2.3 introduces three concepts in consent management, and their relationships: 1) the context in which consent has been given, 2) the permissions and the obligations that

the consent defines, and 3) the consenting party.

The consenting party is the owner of the data or a representative of the data owner. This role needs to give explicit consent for data usage. The role of the consenting party also defines the permissions and obligations that can be used to form policies of consent.

A potential solution for data spaces must also store and monitor the context of the consent for each data exchange between a data owner and a data consumer. The GDPR [5] prescribes that it is the data controller's responsibility to demonstrate that the consent was validly obtained. According to the GDPR [5], if the intended purpose of data processing changes, the controller should provide the data subject with information on that other purpose and other necessary information[5]. The ontology in Figure 2.3 provides support for the provenance of consent by defining a context object. The context object in the ontology includes the time, location, format of gathering consent and as well as information specific for the time when consent was requested[28] (e.g., data provider id, data owner id, data consumer id).

2.3.1 Consent and Data Management Model

In [28], a Consent and Data Management Model (CDMM) is proposed based on the semantic model for a reference architecture. The reference architecture implements consent management according to the requirements defined by the ontology, as can be seen in figure 2.3. However, the CDMM is scoped around a single application infrastructure, which means that it is not indicative for a network or data space.

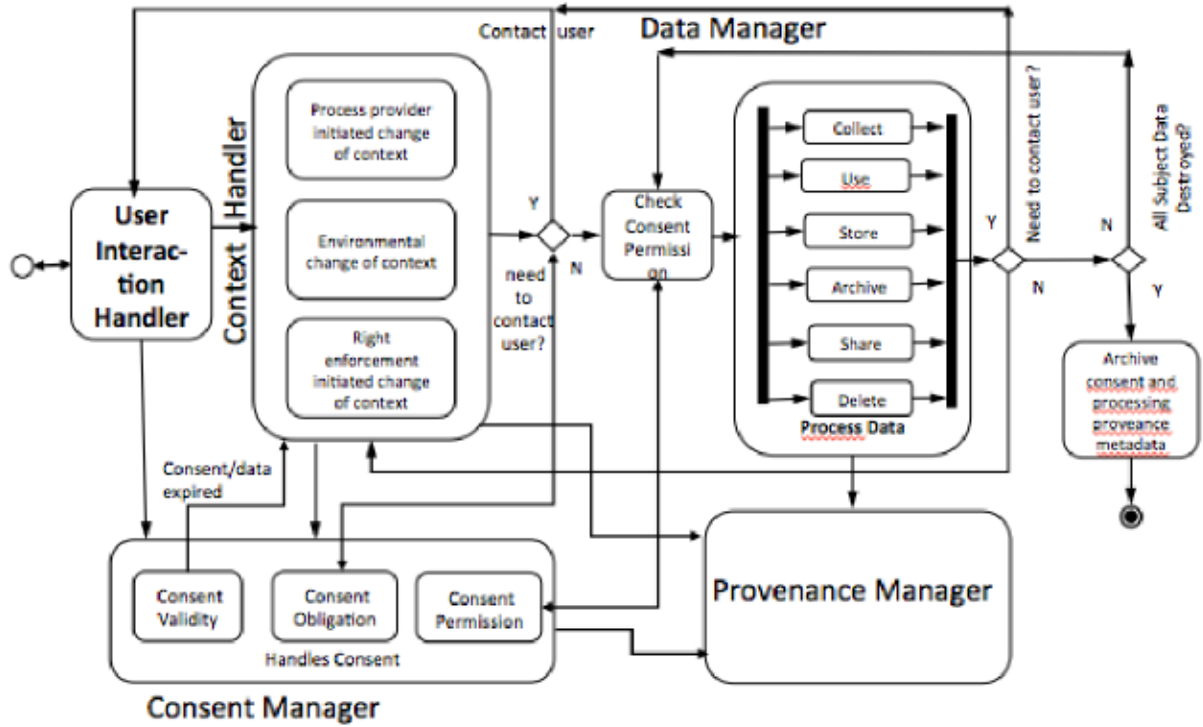


Figure 2.3: Consent and Data Management Model as Proposed in [28]

The architecture in Figure 2.3 introduces five components that realize consent management:

- **User Interaction Handler (UH):** The User Interaction Handler is an interface for interacting with the users. Through the UH, the users can define permissions and obligations of consent, which in turn form the policies for data exchange [28].
- **Context Handler (CH):** The Context Handler is a component for managing and detecting changes of context. If a change is detected by the CH, it informs the Consent Manager and Data Manager of these changes [28].
- **Data Manager (DM):** The Data Manager is a component that is responsible for managing data following the consent given. The Data Manager provides protective control by ensuring that only

authorized processing may occur [28].

- **Provenance Manager (PM):** The Provenance Manager is a component that is responsible for maintaining a provenance log of all activities which involve data and consent [28].
- **Consent Manager (CM):** The Consent Manager is responsible for annotating and storing the consent which was collected by the data owner [28].

2.3.2 Requirements for Implementation of Consent Management

Additionally, the semantic model in Figure 2.3 creates the following requirements for the implementation of consent management:

- Collection of permissions and obligations of consent in a shared vocabulary [28].
- A mapping mechanism that enables mapping of permissions and obligations into machine-readable and machine enforceable format [28].
- Proposing a consent and data management model by taking into account of consent and data lifecycle (e.g., consent is revoked when the permissions and obligations change) [28].
- Identifying and incorporating the change of context into the data processing model [28].

2.3.3 Processes of Consent Management

The reference architecture in Figure 2.3 is used to identify mechanisms that realize consent management. Throughout this thesis, the following mechanisms are used to define consent management, based on the ontology and reference architecture from [28].

- Defining of permissions and obligations of consent.
- Registration of permissions and obligations and linking with a data set.
- Negotiation of permissions and obligations.
- Forming of policies of consent-based of the permissions and obligations given.
- Exchanging of policies of consent.
- Revoking of policies of consent.

2.4 Permissions and Obligations in Consent Management

Permissions and obligations are expressed as parameters that define constraints on data usage. These parameters can be converted into policies that can be used by data access and usage control systems. In general, the parameters for defining permissions and obligations can be split into four categories:

- **Subject Attributes:** Subject Attributes describe the subject (user, organization) by e.g., age, role or clearance.
- **Action Attributes:** Action Attributes describe the action attempted e.g. read, delete or view.
- **Resource (or object) Attributes:** Resource Attributes describe the resource itself e.g. object type, location or classification.
- **Contextual (environment) Attributes:** Contextual Attributes address time, location or other dynamic aspects.

Data Permissions and Data Obligations are defined as follows in the scope of this thesis:

- **Data Permissions:** Data Permissions define what actions are permitted with the data. Data Permissions are useful to define access and usage permissions. However, Data Permissions have drawbacks in terms of obligations that need to be followed by the data consumer.
- **Data Obligations:** Data Obligations define what actions a data consumer must follow when using the data. Data obligations are useful to define the processes and actions that need to be followed. However, Data Obligations have drawbacks in terms of access and usage constraints for the data consumer.

2.5 Right Expressions Languages (RELs)

Policies formed from the permissions and obligations are commonly expressed in Rights Expression languages. These languages are used for rights expression, digital objects declaration, and protection information declaration[30].

These policies are a mapping of the permissions and obligations defined by a data owner that can be mapped into a machine-readable format (policy). The resulting policy can be interpreted by components to enable technical enforcement such as data access and usage control systems. This enables technical enforcement, where under run-time actions are evaluated with the constraints defined by the policies.

There are multiple interpretations of RELs that are technology independent. For this thesis, two languages are considered due to their common usage: ODRL and XACML. The recommendation for a specific REL is out of the scope of this thesis.

2.5.1 Open Digital Rights Language (ODRL)

The Open Digital Rights Language (ODRL) [27] is a proposed language for the DRM community for the standardisation of expressing rights information over digital content. ODRL uses XML-based usage grammar and is an open-source language that helps expressing statements about the usage of digital resources [27].

ODRL policies can define one or more rules regarding permissions, obligations, prohibitions.

ODRL provides no mechanism to implement any access control logic; instead, its policies can describe how an external access control system should behave. These characteristics enable ODRL to be completely neutral about the technology used (if any) to grant access to a certain resource.

2.5.2 Extensible Access Control Markup Language (XACML)

The XACML language is specifically aimed at the specification of authorization policies[26]. XACML is maintained by the OASIS standard organization. The main motivation for the definition of the XACML was the amount of proprietary and application-specific access control policy languages used to define policies and which, once defined, cannot be shared across different systems [30].

The standard comes with a reference architecture and processing model to interpret XACML policies, which enables different systems to make use of the XACML REL.

XACML uses the W3C XML-Signature Syntax and Processing Standard [31] for providing authentication and integrity protection for XACML policies [31].

In contrast to the ODRL policy language, XACML comes with architecture and a processing model that describes how to evaluate access requests according to the rules defined in policies.

2.5.3 Policy Types

Policies can be just as with permissions and obligations categorized in types. In [25] policies are categorized into the following categories:

- **Authorisation Policies (Permissions):** Authorisation Policies define what activities a subject is permitted to do in terms of the operations it is authorised to perform on a target object. Authorisation Policies are formed from data permissions.
- **Obligation Policies:** Obligation Policies define what activities a subject must (or must not) do. Obligation Policies are formed from data obligations.

Also, as described in [25], each policy category can be sub-categorized in activity and stated-based policies. Activity-based policies define what can and must be done in a certain activity; State-based policies define what can and must be done in a certain state. The following chapters will give some examples of authorization and obligation policies.

Authorisation Policies

Authorisation Policies are formed by the permissions that a data owner defines. An example set of activity- and state-based authorisation policies can be seen below:

- Activity-based Authorization Policy Example
 - Organization X is permitted to the read energy usage data set.
 - Organization X is prohibited to use the energy usage data set for commercial projects.
- State-based Authorization Policy Example
 - Organization X permitted to read energy usage data, if $(\text{currentdate} - 7 \text{ days}) < \text{creation_date}$.
 - Organization X is prohibited to read energy usage data, if $\text{storage date data} + 10 \text{ days} < \text{date}$.

Obligation Policies

Obligation policies are formed by the obligations that a data owner defines. An example set of activity- and state-based obligation policies can be seen below:

- Activity-based Obligation Policy Example
 - Organization X must mention the usage of public data in the license if the data is used in a commercial project.
- State-based Obligation Policy Example
 - Organization X must notify the data owners of data usage, if $(\text{current_date} - 7 \text{ days}) > \text{creation_date}$.

2.6 Technical Enforcement of Policies in DRM Systems

This section explains how policies can be technically enforced in DRM systems.

2.6.1 Architectural Elements for Policy Control in Data Spaces

In [24], an overview is provided of a policy-based admission control framework that can be used by access and usage control systems on an individual network node or application infrastructure. The paper introduces two main architectural elements for policy control: 1) the Policy Enforcement Point (PEP) and 2) the Policy Decision Point (PDP) [24].

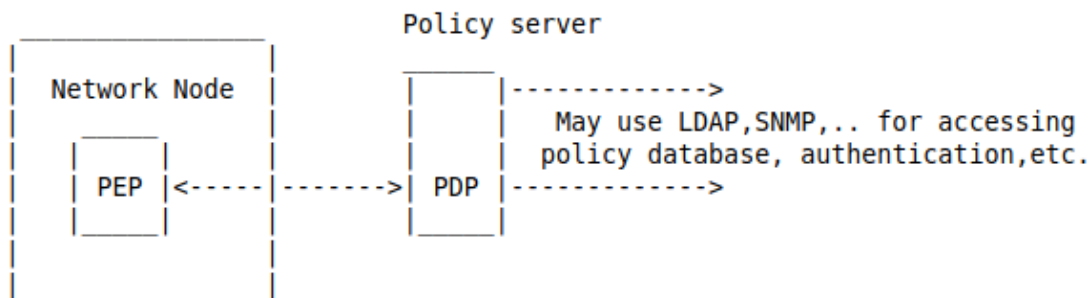


Figure 2.4: Single Network Node of a PEP Interacting with a PDP

In [24] multiple examples are given of access and usage control enforcement configurations. A simple network configuration involves a central PDP entity that may reside at a policy server and a PEP component for each network node. The concept network configuration is visualized in Figure 2.4. The architecture in Figure 2.4 shows that the PEP represents the component that always runs on the policy-aware node, where it is the point at which policy decisions are enforced.

Further, the paper [24] emphasises that the policy specification language should ensure unambiguous mapping of control attributes to policy definitions.

The authors of [8] define two ways where usage restrictions can be placed: 1) sticking usage restrictions to the data and 2) storing policies independently from the data.

- **Sticking Usage Restrictions to the Data:** Usage restrictions that adhere to the data are also called sticky policies. Sticky policies are one way to cope with the distribution of usage restrictions. In this approach, machine-readable usage restrictions (policies) stick to data when it is exchanged. There are different possibilities of realizing sticky policies. Usually, the data is encrypted and can only be decrypted, if the adherence to the usage restrictions is guaranteed.
- **Storing Policies independently from the Data:** Policies can be stored independently from the data, for instance, as a central component. In this case, the management component has to take responsibility to exchange the access and usage restrictions between different systems.

Policy Enforcement Processes

Systems that make use of RELs often also make use of processes and components that will automatically enforce these policies. Figure 2.5 illustrates an example of the interaction of components that are needed for policy enforcement by using XACML. In this example, XACML was chosen because it is a well-established policy language for enforcement of access and usage control. Implementations of data spaces are not forced to choose XACML, therefore this figure functions more as an illustration of a potential data space implementation.

The main components to implement policy enforcement are the Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Information Point (PIP), and Policy Administration Point (PAP).

Policy Enforcement Point: A PEP is a control point that intercepts the information flow and enforces the policies. The PEP will issue a request to the PDP for the evaluation of the intended actions by the client [conf/codaspy/FerraioloCKH16].

Policy Decision Point (PDP): A PDP is a component of a policy-based access control system that determines whether or not to authorize a user's request. The PDP judges authorization based on the available information (attributes) given by the PEP, and applicable security policies of the PAP [conf/codaspy/FerraioloCKH16].

Policy Information Point (PIP): A PIP provides missing information for the decision-making. The PDP can use this component to get contextual information about the intercepted system action (e.g., data flow information or geo-location of the requesting device) [conf/codaspy/FerraioloCKH16].

Policy Administration Point (PAP): A PAP is the entry point for the specification of usage policies. A PAP is often accessible via a user-friendly graphical interface [conf/codaspy/FerraioloCKH16].

Policy Management Point (PMP): A PMP administers the access and usage restrictions. Hence, the component is concerned with the policy life cycle. The policy life cycle includes the instantiating, negotiation, deployment and revocation of usage restrictions, as well as conflict detection and resolution [conf/codaspy/FerraioloCKH16]. In the case of dynamic definition (e.g., a negotiation for custom access and usage policies) the PMP provides and instantiates the policy for the PDP corresponding to the actor that uses the data.

Policy Execution Point (PXP): A PXP is used to perform additional actions based on the policy rules. Such as sending an email when data is used or writing to a specific log system. [conf/codaspy/FerraioloCKH16]

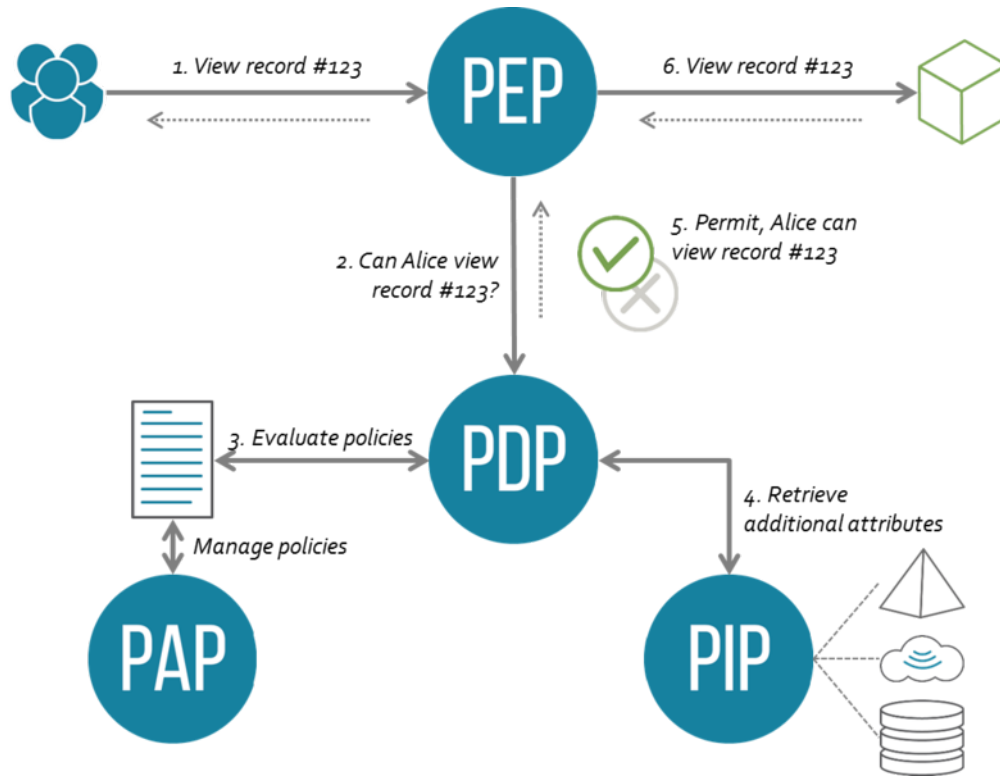


Figure 2.5: XACML Policy Enforcement Components and Interaction Flow [32]

Figure 2.5 visualizes that data flows need to be monitored and intercepted by a PEP for enforcing access and usage restrictions. These intercepted data flows are then translated to a decision request and routed to the PDP for requesting permission or denial of the data flow. The PDP evaluates the request and may invoke the PIP for additional information. After evaluation, the PDP notifies the PXP to execute additional actions (e.g., logging). At the end of the process, the PDP sends an authorization decision to the PEP. The PEP enforces this decision on the intercepted data flow.

Concluding from Diagram 2.5, the components outlined are primarily responsible for policy enforcement, definition, and administration processes. Data space implementations can also choose to implement negotiation and management of policies. XAMCL for these mechanisms the Policy Management Point (PMP) and the Policy Execution Point (PXP).

The XACML Policy Enforcement Components and Interaction flow would have to be adjusted for the use of data spaces. The XACML enforcement framework visualized in Figure 2.5 is primarily intended for a single application infrastructure. Consequently, for a network approach, such as data spaces, this introduces some challenges. However, the outlined enforcement components can be used as a reference architecture, as done by the IDS [8]. Implementations in data spaces need to take into consideration not only enforcement but also all the other processes of consent management (definition, administration, negotiation). Orchestrating these processes throughout a data space introduces a set of challenges that need to be solved before consent management is fully implemented.

Chapter 3

Problem Statement

3.1 Consent Management in Data Spaces

As discussed in the introduction 1, the challenges of regulations, ownership and control over the exchanged data, open the need for access and usage control systems with the use of policies. Orchestrating and managing these policies forms the basis of consent management data spaces. However, the realization of consent management in data spaces introduces a set of challenges that this research tries to overcome.

Currently there is no reference architecture in place for implementing consent management in data spaces. Reference architectures of other domains for consent management state that the creation of an exhaustive ontology covering all domains is not feasible; and instead, propose that others should extend ontology's of consent management for specific application scenarios or application domains[28].

3.1.1 Policies Creation and Orchestration

Data spaces could potentially become a solution for a structured decentralized system for the exchange of data. Currently, reference architectures such as IDS, provide access and usage control systems with the use of sticky policies formed from the permissions and obligations of data owners[8]. However, current data spaces don't provide functionality to support the dynamic creation, management and enforcement of consumer, organizational and jurisdictional policies. As an example, IDS currently states regarding the negotiation process: "The IDS parties bargain over a set of policy parameters in the Policy Negotiation process. As future work, we will investigate these policy parameters and in addition, the feasible approaches of acquiring an Agreement policy"[8].

Legal Obligations

According to the General Data Protection Regulation (GDPR), consent of a data subject is one of the legitimate basis for processing personal data[5].

With GDPR imposing greater restrictions on obtaining consent, the data controller must be able to demonstrate that consent was validly obtained, implies that failure to provide proof of the validity of obtained consent will be a breach of the legal requirement for GDPR compliance [5]. This restriction on consent requirements raises the need for data spaces to base their consent model on ontology's that are compliant with the GDPR.

3.1.2 Composed Data Sets and Distinction of Data Owner and Data Provider

Data spaces introduces the separation of data owner and data provider. This separation entails that a data provider can make data available for multiple owners. Also, this means that a data owner can have multiple data providers that provide its data.

This many-to-many relationship opens the possibility of composed data sets, in which a data set is composed out of data from different data owners. This adds a layer of complexity to consent management. Each data provider and in-turn each data consumer needs to have access to the definition of the

permissions and obligations of the data set. Also, this means that the policies formed of a data set are unique for the combination of a data provider, data owner and a data consumer, where a data consumer could potentially have multiple policies for the same data set with different data providers. This add a lot of complexity to consent management. It is therefore required that a consent management solution for data spaces will unburden the different actors in the data space. A potential solution must look into centralized mechanisms to manageme consent management.

Additionally, the policies that are formed from composed data sets, are based on permissions and obligations of different data owners. Therefore, there must be a mapping mechanism that can translate the permissions and obligations in to a REL and the mapping mechanism must map the different parameters into one encompassing policy.

Another challenge is the process of negotiation where a data owner and data consumer must agree on the permissions and obligations that form the consent policies. If an agreement is made on the parameters, a custom set of policies are created for that particular data consumer. These policies will exist next to the predefined static set of policies.

In the context of data spaces, this leads to the following challenges:

- Storage and exchange of data usage control permissions and obligations.
- Forming of policies based on permissions and obligations of single data sets and composed data sets.
- Storage and exchange of consent policies.
- Revoking of formed policies, based on consent context.
- Revoking of formed policies based on data owner intention.
- Negotiation of permissions and obligations between data owners and data consumers.

3.2 Problems in Consent Management Tackled by this Thesis

This thesis implements consent management in data spaces by providing a set of requirements, a set of components that implement these requirements and a reference architecture that makes use of the proposed components to realize consent management. This research focuses on the limitations of consent management for the general definition of data spaces as described in the terminology A.

3.3 Problems Related to the Use Case

The use case was primarily selected because it has a set of problems that are characteristic for consent management. Current data spaces, as defined in the terminology section, can't provide a solution for these findings. Therefore this research has outlined these problems in the following findings:

Finding 1: Data providers provide data for a large number of data owners, where each data owner has their own permissions and obligations for data usage. Managing and orchestrating these permissions and obligations requires consent management.

Finding 2: Data providers will provide data sets that contain data of different data owners (composed data sets). Therefore providing composed data sets forces data providers to combine the permissions and obligations of the data that is contained in the data set.

Finding 3: Different data providers will provide data for the same data owner. This introduces the difficulty that there must be shared access to the permissions and obligations of the data owner. Each data provider must therefore have access to the permissions and obligations.

Finding 4: The data space is a dynamic environment where (previously unknown) data consumers will request data of the data providers. Also, the intentions for data usage will vary from data consumer and is unknown beforehand. This means that obtaining consent must be an automatic process and that there must be processes in place that will provide dispute management.

Chapter 4

System Design for Consent Management in Data Spaces

In this chapter, the findings of this research are proposed. This chapter is divided into six sub-chapters which represent the six steps of design science research (DSR) paradigm. During the research, the DSR steps were followed along the guidelines proposed in [17].

4.1 Problem and Motivation

In Chapter 3, the problems are outlined that are currently present in data spaces and consent management.

As outlined in the Introduction (see Chapter 1), this research focuses on realizing consent management in data spaces. To achieve this goal, a set of novel IT components are introduced and applied in a reference architecture. These IT components are based on requirements that were derived from literature research and from professional experience in working with data spaces.

4.2 Objectives of a Solution

In this section the problems that were outlined in the problem statement are transformed into system objectives, also referred to as meta-requirements [33] or requirements [34]. These requirements will form the basis for a solution that implements consent management in data spaces.

The identified problems (see Chapter 3), do not necessarily translate directly into objectives for a solution. Therefore, the first objective is to define the requirements for all processes of consent management. These requirements form a basis for a solution. After the requirements are identified, components or processes that implement the requirements are defined. Eventually, these components or processes are applied in a reference architecture.

4.2.1 Processes in Consent Management

As described in section 3.1, the literature research on consent management resulted in the following six main processes that define consent management.

- Defining of permissions and obligations of consent.
- Registration of permissions and obligations and linking with a data set.
- Negotiation of permissions and obligations.
- Creation of policies of consent-based based on the permissions and obligations given.
- Registration and exchanging of policies of consent.
- Revoking of policies of consent.

In the following sub-sections, the requirements are introduced that were specified for the above-mentioned mechanisms. The requirements were obtained from literature, regulations and from analyzing current

data spaces.

Also the main processes were evaluated according to the MoSCoW definition. Where all the processes were identified as MUST. Additionally this research also identified some processes that could be categorized as should and could processes. These were:

- Searching of permissions to select potential data sets.
- Searching of obligations to select potential data sets.

These processes were seen as additional requirements and were therefore not seen as required for consent management in data spaces. However, this research still processes a reference architecture that will take these requirements into account.

4.2.2 Defining of Permissions and Obligations of Consent

As can be seen in the ontology defined in [28], consent is expressed as permissions and obligations defined by the data owner and converted to a machine-readable format by the use of a REL (e.g. XACML, ODRL), which can be interpreted by enforcement frameworks.

Requirements for the Definition of Permissions and Obligations of Consent

The requirements for the definition mechanism are indicated with a capital D. The following requirements were defined for the implementation of consent management, specifically the definition of permissions and obligations of consent:

- D.1: Creation of policies are based on the permissions and obligations linked to the data set. Consent must be taken from the data subject (data owner) by informing the nature of the processing. Informing of the data owner must be done in an intelligible format where the purpose of processing is clear. Additionally, the identification of the controller must be given[5].
- **Permissions related requirements**
 - D.2: The consenting party (data owner) is responsible for defining permissions which can be used to form consent policies[28].
 - D.3: There must be an open vocabulary for forming permissions of consent[28].
 - D.4: Permissions must be specified for a specific data set before a data consumer can use the data set[28].
- **Obligations related requirements**
 - D.5: The consenting party (data owner) is responsible for defining obligations that can be used to form consent policies[28].
 - D.6: There must be an open vocabulary for forming obligations of consent[28].
 - D.7: There must be definitions of obligations for a specific data set before the data is exchanged and processed by a data consumer.

4.2.3 Registration of Permissions and Obligations and Linking with a Data Set

In the reference architecture of [28], the permissions and obligations are directly stored in the same application infrastructure as in which they are enforced.

The decentralized characteristic of data spaces forces this research to introduce additional requirements. Additional requirements are required because the permissions and obligations need to be shared with a large number of actors. Therefore, neither the data owner nor the data provider should be responsible for the storage and accessibility of the obligations and permissions. Furthermore, policies of composed data sets are an aggregation of all the permissions and obligations of the owners in the data set; and because data sets are treated as separate entities, the permissions and obligations need a linking mechanism to the data set it describes.

Based on the problem statement and background research, the following requirements are to be considered in the linking mechanism:

- It must be possible to link permissions and obligations of different data owners to a single data set.
- The same permissions and obligations of a specific data owner can be linked to a multitude of data sets,

To implement these requirements, this research introduces a data category attribute. The data category attribute will be defined with the registration of permissions and obligations. The data category is matched with the data category of the data set.

With this approach, the research assumes that data domains come with their own specific data categories for the indication of data sets. Where a data domain focuses on specific data sets that will be exchanged (e.g. energy grid data). Example data categories for the energy transformation use case could be:

- **Household energy usage:** The energy usage of a specific household that is identified as a data owner in the data space.
- **Household energy storage:** The energy storage of a specific household that is identified as a data owner in the data space.
- **Household energy generation:** The energy generation of a specific household that is identified as a data owner in the data space.

Each created permission and obligation will be assigned to one of the three proposed data categories. Then each permission or obligation can be linked to data sets with the same category.

Requirements for Registration of Permissions and Obligations and Linking with a Data Set

The resulting requirements for the permission and obligations registration mechanism are indicated with a capital R. The following requirements were defined:

- **Permissions**
 - R.1: There must be a central point that each actor of the network can access to retrieve the permissions.
 - R.2: A permission must be identified with a unique ID.
 - R.3: Registration of permissions must be done with either a data category, data set, data consumer or data provider attribute.
 - R.4: Registration of permissions with a direct reference to a specific data set must be possible.
 - R.5: The registration of permission requires a reference to the data owner.
- **Obligations**
 - R.6: There must be a central point that each actor of the network can access to retrieve an obligation.
 - R.7: An obligation must be identified with a unique ID.
 - R.8: Registration of obligations must be done with either a data category, data set, data consumer or data provider attribute.
 - R.9: Registration of an obligation with a direct reference to a specific data set must be possible.
 - R.10: The registration of an obligation requires a reference to the data owner.
- **Composed Data Sets**
 - R.11: Composed data sets are linked to all the permissions and obligations of the owners.
- R.12: Data owners should not be traceable from the consent permissions and obligation administration point unless the trace is purposely allowed.

4.2.4 Negotiation of Permissions and Obligations

The GDPR [5] prescribes that it is the controller's responsibility to demonstrate that the consent was validly obtained. Therefore, the implementation of consent management in data spaces should provide support for the provenance of obtained consent.

In obtaining consent, a data consumer can express its intent of usage in permissions and obligations. Acceptance of a request for consent requires a custom context wherein consent has been given. Obtaining

consent must therefore consist out of the consenting party and context. The context shall include the time, location, format of gathering consent as well as the information that was provided to the consenting party when the consent was requested.

However, these requirements not only apply to the negotiation but also apply to the acceptance of the default permissions and obligations of a data owner. The reference architecture described in [28] does not contain a mechanism for negotiation. However, the model defines a fallback to the data owner whenever the usage of the data falls outside of the boundaries of consent. This notification process can be expanded in a negotiation interaction in which a data owner can reject or accept the request.

Requirements for Negotiation of Permissions and Obligations

The resulting requirements for the negotiation of permissions and obligations mechanism are indicated with a capital N. The following requirements are defined:

- N.1: The intent of usage from the data consumer shall be presented in a manner which is distinguishable and understandable. Additionally, it must be defined in shared vocabulary and must inform the nature of the processing in an intelligible format. This format must contain the purpose of processing, identification of the controller, and information about withdrawing consent[5].
- N.2: A data consumer must be able to demonstrate that the consent was validly obtained[5].
- N.3: When the intended purpose of data processing changes, the data consumer should provide the data owner with information on that other purpose and other necessary information[5].
- N.4: When the intent of the purpose of data processing conflicts with the permissions and obligations for the data set, the data consumer must be able to start a negotiation with the data owners.
- N.5: When the data owner complies with the intent of usage policy parameters of the data consumer, a custom policy is created.

4.2.5 Forming of Policies of Consent-Based of the Permissions and Obligations Given

Consent is expressed in the permissions and obligations that are defined by the data owner. The permissions and obligations are converted into a machine-readable format such as RELs (e.g. XACML, ODRL), which can be interpreted by the enforcement frameworks that are part of the access and usage control systems.

Data spaces require the introduction of new requirements on top of the mapping mechanisms requirements defined by [28]. The ontology and reference architecture that is defined in [28], contains requirements that focus on the mapping mechanism. The mapping mechanism introduced by the authors of [28] converts the permissions and obligations into machine-readable policies (See requirement F.1). Data spaces require additional requirements, given the requirement that in a data space, neither the data owner nor the data provider should be responsible for the forming of policies (see requirement F.3). Further requirements are needed about the fact that policies from composed data sets are an aggregation of all the permissions and obligations of the owners in the data set (see requirement F.2).

Requirements for Forming of Policies of Consent-Based Permissions and Obligations Given

The resulting requirements for the negotiation of permissions and obligations mechanism are indicated with a capital F. The following requirements are defined:

- F.1: There must be a process for mapping of permissions and obligations of consent into machine-readable policy [28].
- F.2: Composed data set policies are formed on the aggregated permissions and obligations of all the data owners.
- F.3: Policies should be formed by an authorized actor, process, or component to ensure the validity of the resulting policy.

4.2.6 Registration and Exchanging of Policies of Consent

In the reference architecture of [28], there is no mechanism in place for the administration of policies based on the permissions and obligations of consent. For a network approach, such as in data spaces, these policies need to be stored, because the data consumer, data owner, and the data provider are separate actors in the network. Administration, storage and exchange of the policies must be done in a way that each actor can access these policies. To achieve this, the researchers assume that central authorities in the network are responsible for the process of registration and exchange of policies. The resulting requirements were therefore a direct result of this assumption. Furthermore, the literature research concluded that the context in which consent has been given must be stored for each defined policy that belongs to a data consumer. Additionally, according to GDPR [5], a data owner must be able to withdraw consent at any time.

Based on these requirements, the underlying research assumes that mechanisms of registration and exchanging of policies are managed by a central component.

Requirements for Registration and Exchanging of Policies and Consent

The resulting requirements for the process of registering and exchanging policies are indicated with a capital G. The following requirements are defined for the mechanism of registering and exchanging policies in consent management:

- G.1: Before collecting or using the data set, the data provider should obtain consent from the data owner [28].
- G.2: When a policy is registered, a context object is created, that defines the context when the consent was given.
- G.3: Data owners should not be traceable from the policy definition or administration point unless this is purposely allowed.
- G.4: The data owner must be able to access the policy definition at any moment.
- G.5: The data consumer must be able to access the policy definition at any moment.
- G.6: A policy must have a single point of truth regarding storage.
- G.7: A consent policy must have a reference to the data owner, data set, and data provider when administrated [28].
- G.8: Retrieval of a specific policy must be done with an ID that is unique for the combination of data consumer, data owner, data provider and data set.

4.2.7 Revoking of Policies of Consent

The requirements that were defined for the process of revoking policies are primarily based on the specification defined by the GDPR [5]. GDPR states that at any time consent can be revoked by the data owner. On top of these specifications, consent must be revoked as soon as the context changes in which consent has been given [28]. This means that as soon as one of the attributes change that belongs to the context (e.g. data provider ID), the consent will be revoked.

Requirements for Revoking Policies of Consent

The resulting requirements for the revocation of consent are indicated with a capital H. The following requirements are defined for the process of revoking policies of consent:

- H.1: In case a data owner changes the permissions and obligations that formed the policy, the policy will automatically be revoked as a consequence of the context change.
- H.2: The data owner must be able to withdraw consent for a specific data consumer or data set:
 - If the consent was not freely given [5].
 - If the data owner has no genuine and/or free choice [5].
 - Is unable to refuse or withdraw consent easily [5].
 - if there is a "clear imbalance" of power between the controller and the data subject [5].
- H.3: A change of the context in which consent was obtained revokes the policy of consent (e.g., a change of data consumer identity, permission, obligation, data provider identity, time, location, or the format of gathering)

Chapter 5

System Implementation for Consent Management in Data Spaces

The requirements outlined in the objectives of a solution section (see Chapter 4.2) form the basis for a set of novel IT artefacts. This research proposes three IT components: the consent manager (5.1), policy catalogue (5.2) and the policy broker (5.3). The components are introduced by this research as a result of the defined requirements. Each of the three components is introduced in the sub-chapters in the following manner:

- **Intent:** a summary of what the component tries to achieve.
- **Motivation:** the reasons why the component was introduced, and how it achieves its stated goals.
- **Requirements Implementation:** a list of requirements from the objectives the component implements.
- **Applicability:** a list of scenarios where the component would be useful to be introduced.
- **Participants:** a high-level overview of all the interactions where the introduced component is part of.
- **Collaborations:** list of actors and components the introduced components needs or makes use of.
- **Consequences and Benefits:** a summary of consequences and benefits when the component is introduced.
- **Implementation Considerations:** a list of discussion topics that are implementation-specific and may differ depending on the use case.

The introduction of components is done in a strict manner. This is done for reference usage, where future implementation can use this research as a reference. Diagrams are also used with the introduction of each component. These diagrams are of free form and do not adhere to a certain modelling language. They solely function as an indication of the functionality.

5.1 Consent Manager

The consent manager is a component that primarily focuses on the management of permissions and obligations. This component will implement processes that are categorized as must (MoSCoW) for consent management.

5.1.1 Intent

The consent manager acts as an intermediate to translate the obtained permissions and obligations into a usable machine-readable format of consent management policies; handling of requests for consent and negotiations of the intent of usage between data owners and data providers.

5.1.2 Motivation

The consent manager was introduced from the need for a component that covers the implementation of forming the policies of consent, based on the permissions and obligations that a data owner has defined. Data owners setup their consent permissions and obligations for their data. The permissions

and obligations cover a set of data sets belonging to the data owner. There is thus a distinction between the data subject (data set) and the associated permissions and obligations. To form policies of consent, based on these permissions and obligations for a data consumer, requires a component that implements this process. A consent manager (CM) forms machine-readable policies that the data usage and access control systems can interpret, according to the permissions and obligations defined by the data owner, GDPR-related regulations, and domain regulations. Additionally, the CM provides an interface for defining permissions and obligations, linking of data sets to permissions and obligations, and to facilitate a negotiation process between data owners and data providers.

5.1.3 Requirements Implementation

The CM implements the following requirements out of the requirements that were outlined in Chapter 4.2:

- **Defining of Permissions and Obligations of Consent**
 - D.2, D.4, D.5, D.7: The CM provides an interface for the consenting party (data owner) to define its permissions and obligations. Any data consumer that wished to use a data set must request the consent of usage at the CM. Also, any data set that is registered at the CM must have a set of obligations and permissions before it can be used by a data consumer.
 - D.1, D.3, D.6: The CM forms consent policies from the permissions and obligations that are linked to the data set. It does this by providing a translation mechanism that can interpret the attributes that define the permissions and obligations and in-turn translate them to machine-readable policies. The form in which the attributes of permissions and obligations are expressed are discussed in the implementation considerations section.
- **Registration of Permissions and Obligations and Linking with a Data Set**
 - R.1, R.2, R.3, R.4, R.5, R.6, R.7, R.8, R.9, R.10: The CM provides a central point for storage, definition and retrieval of permissions and obligations. With the registration of permissions or obligations, the CM requires a data owner and a data category, data consumer, data set or data provider attribute for linking of the permissions and obligations to a specific data set. In the registration process, the CM provides a unique ID to the permission and obligations.
 - R.11: The CM allows linkage of multiple permissions and obligations to a single data set.
 - R.12: The CM does not implement any interface functionality that provides a result that contains a reference to a data owner.
- **Negotiation of Permissions and Obligations**
 - N.1: The CM requires the attributes that define permissions and obligations to be defined in a shared vocabulary. The form in which the attributes of permissions and obligations are expressed are discussed in the implementation considerations section.
 - N.4, N.5: If permissions or obligations conflict with the intent of usage of data processing, the data consumer could start a negotiation process through the CM. The CM will contact the data owner to request consent for the given intent of usage. If the data consumer complies with the intent of usage, a custom policy is made by the CM for the data consumer.
- **Forming of Policies of Consent-Based of the Permissions and Obligations are Given**
 - F.1, F.2, F.3: The CM is responsible for the mapping mechanisms that translate the permissions and obligations of consent into a machine-readable policy that can be used by the data usage and access control system in use. This also means that the CM needs to form policies for composed data sets, where mapping needs to be made of a set of permissions and obligations of the different owner into a policy.
- **Registration and Exchanging of Policies of Consent**
 - G.1: A data consumer must obtain consent from a data owner before it can use a data set. The CM provides an interface for a data consumer, where it could apply for consent for a specific data set. In this process, a data consumer could also provide its intent of usage attributes.
- **Revoking of Policies of Consent**
 - R.1: The CM will automatically notify the policy catalogue whenever permissions or obligations are updated that are used in already created policies. The created policies are then

revoked, which forces the data consumer to apply for consent at the CM if it wished to keep using the data set.

5.1.4 Applicability

Use the CM when:

- There must be a central location where consent permissions and obligations must be stored for a multitude of data owners. This does not mean that a CM must be the sole actor in the network with the defined functionality, it is allowed to have a multitude of CMs in a network.
- There must be a service that can translate the obtained consent into usable properties such as consent validity, consent obligation, and consent permission.
- There must be a service that can form policies for specific data consumers based on the defined permissions and obligations for a data set.
- (Domain) Regulations must be enforced on top of the data owner defined permissions and obligations.
- There must be a service that can form a single policy for a multitude of data owners of a single data set with each their own data usage permissions and obligations.
- A central service is needed to regulate interactions with consent permissions and obligations for regulations and privacy of the data owner.

5.1.5 Participants

The participants of the CM are:

- Data Owner: Administrates consent permissions and obligations
- Data Providers: Administrates data sets that it provides. The data provider is required to link all the data owners to the data set it registers. It does this by providing a list of data owners.
- Client (data consumer): Request data policies that function as the consent of the data owner for the usage of the data set.
- Policy Catalogue: Stores the formed policies from the CM that in-turn can be retrieved by clients.

5.1.6 Collaborations

The CM interacts with the following entities:

Data Owner: A data owner will use the CM for the management of permissions and obligations. If the data owner updates, deletes, or creates a permission or obligation, all corresponding data sets will be revoked at the policy catalogue (see Chapter 5.2 for the introduction of the policy catalogue component). These activities can be seen in Figure 5.1.

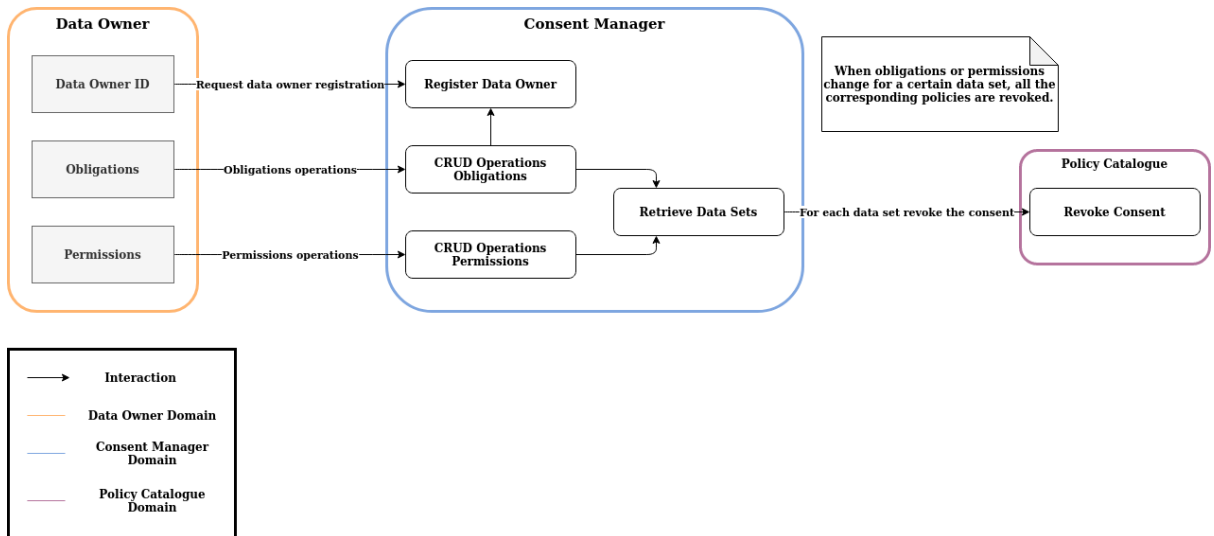


Figure 5.1: Interactions of Data Owner with Consent Manager

Figure 5.1 visualizes the interactions a data owner has with a CM, those are as follows:

- Registration as a data owner at a CM. The registration as a data owner at a CM allows data providers to link the data owner to their registered data sets.
- Registration of permissions at a CM. From these consent permissions, a CM can form policies for clients that want to use data sets of which the data owner is the owner. (The mapping of permissions into machine-readable policies is further discussed in the Implementation Considerations Section)
- Registration of consent obligations at a CM. From these consent obligations, a CM can form policies for a client that want to use data sets where the data owner is the owner of. (The mapping of permissions into machine-readable policies is further discussed in the Implementation Considerations Section)

Data Provider: A data provider will use the CM for the management of data sets and link data owners to these data sets. These activities can be seen in Figure 5.2.

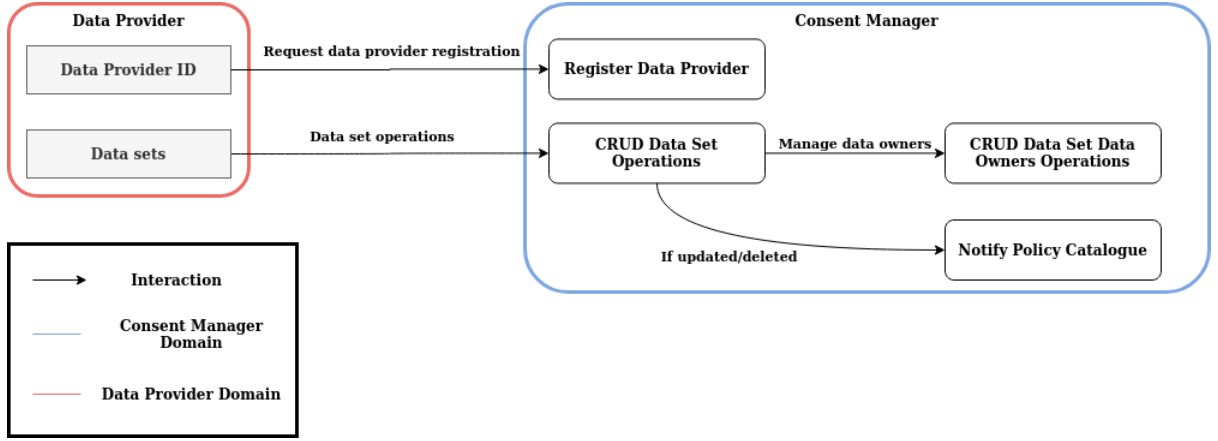


Figure 5.2: Interactions of Data Provider with Consent Manager

Figure 5.2 shows a data provider has the following interactions with a CM:

- Registration as data provider at a CM. This allows data providers to register data sets and link themselves as data provider to this data set.
- Registration of data sets at a CM. The data sets a data provider provides must register a data owner or a set of data owners.

Data Consumer: A data consumer will use the CM for requesting of policies that can be used as the consent of usage of a specific data set by the data owners. These activities can be seen in Figure 5.3

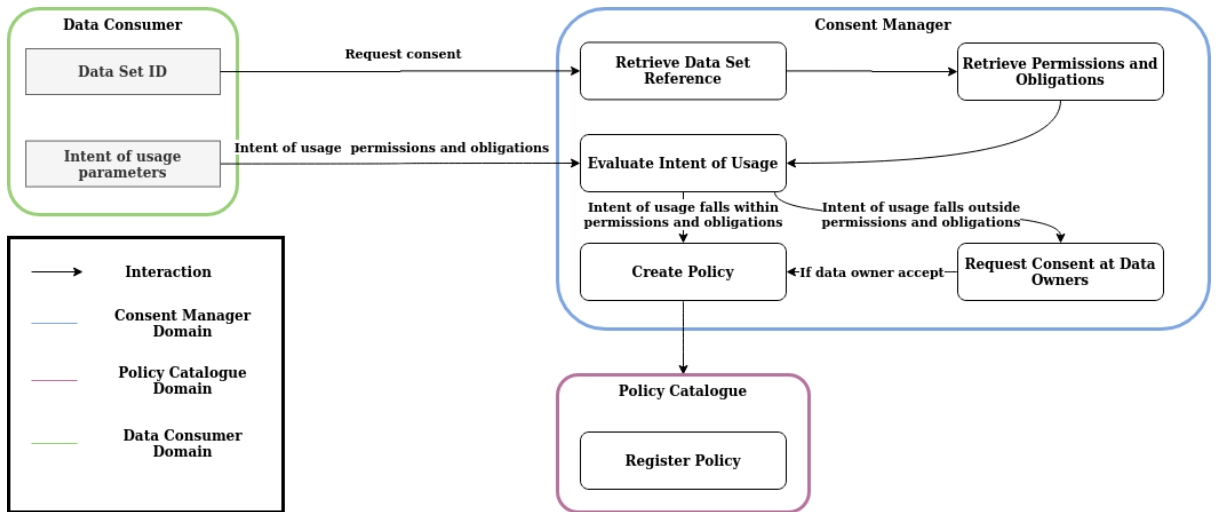


Figure 5.3: Interactions of Data Consumer with Consent Manager

Figure 5.3 shows a data consumer has the following interactions with a CM:

- Requesting of policies for a specific data set. The CM will return a policy that will cover all the permissions and obligations of the data owners.
- Requesting of policies for a specific data set with the intent of usage parameters. The CM will check whether the intent of usage falls within the consent permissions and obligations of the data owners. If this is the case, the policy will be created. If this is not the case, the data owners of the conflicting consent permissions and obligations will be contacted to accept or reject the intent of usage.

5.1.7 Consequences and Benefits

A CM can come with the following consequences and benefits.

Consequences:

- **Vocabulary for Expressing Consent:** The CM must fulfil the requirement of an open vocabulary for expressing consent in the form of permissions and obligations. Therefore, when implementing a CM, it must be known how these permissions and obligations are expressed. The CM must also provide the functionality that supports this vocabulary for clients to interact with the CM.
- **Mapping of Consent to Policy Language:** The CM must fulfil the requirement to provide a mapping of consent permissions and obligations into machine-readable REL. Therefore, when implementing a CM, it must be known which RELs are used. Based on the vocabulary used, a translation must be made into a specific REL. The CM must therefore implement the logic for translating the attributes that define the permissions and obligations into a specific REL.
- **Negotiation Orchestration:** When the intent of usage of a data consumer falls outside the permissions and obligations the CM must contact each conflicting data owner. Each data owner can then accept or reject the request. Orchestrating this process is the responsibility of the CM. This also means that the data owners need to implement the functionality to be contacted by a CM.
- **Context Updates:** When a data owner changes permissions or obligations, the CM must send a context update to the data consumer. This update will notify the data consumer that the consent of usage obtained in the previous context is revoked, and therefore the data consumer must apply for a new consent of usage.

Benefits:

- **Centralized storage:** Allowing a data owner to store and manage its permissions and obligations at a specific CM prevents redundancy and overhead for the data owner. Clients that want to use the permissions and obligations can therefore use a central actor to get the definition of a specific permission or obligation. Central in the context of CMs doesn't mean that only a single CM is allowed in data spaces, however, it does mean that specific permissions or obligations are unique to a CM instance.
- **Regulations:** Regulations that apply for all data sets can be enforced at the consent manager. In the process of the creation of policies, these standard regulations can be applied by default.
- **Controlled Mapping to REL:** Conversion of permissions and obligations into policies occurs in a controlled environment, where the research assumes that a specific CM in use is trusted as the component to translate permissions and obligations into policies.
- **Abstraction and Categorization:** the abstraction of permission and obligation storage allows for categorization of similar permissions and obligations. Policies with similar data sets or policies of composed data sets can be grouped to a specific CM, which allows for ease of use for clients. However, this does mean that actors in the network need to have an overview of all CMs or a selection of CMs that are present in the data space.

5.1.8 Implementation Considerations

The following points are to be considered for the implementation of the consent manager component:

- **Actor References:** There must be a commonly used way for identification of actors and resources. This allows for registration of data owners, data providers and data sets in a way that can be traced back to the source.

- **Permissions and Obligation Specification:** There must be a set of functions that a data owner can use to define its permissions and obligations. In [28] HTML forms are used for defining the permissions and obligations.
- **Shared Vocabulary and Mapping to REL:** There must be a shared vocabulary to define permissions and obligations used in the data space. Also, the chosen vocabulary has to have a direct relation to the REL used, where the CM must be able to map the definitions of permissions and obligations into usable policies.
- **Policy Definitions:** Policy definition entries have a reference to all the data sets they are contained in (many-to-many relationships).

5.2 Policy Catalogue

5.2.1 Intent

The intent of the policy catalogue (PC) component is to separate the storage of policies defined by RELs so that a client can access, store, delete and update these policies at a central location.

5.2.2 Motivation

The policy catalogue (PC) was introduced based on the need for a component to cope with the complexity of policy exchange in networks. Composed data sets, the distinction between the data owner and data provider, and the processes of consent management give rise to complex policy exchange networks. A solution to mitigate these complex networks is to introduce a component that takes the form as a PC. This catalogue functions as a globally accessible actor in the network for different clients to perform CRUD operations on created policies. The PC becomes the single point of truth of policy definitions across all clients. In this way, neither the data owner nor the data provider is responsible for the management of the policies.

5.2.3 Requirements Implementations

The PC primarily covers administration and retrieval requirements. The following requirements, out of the requirements that were outlined in Chapter 4.2, are covered by the PC:

- **Negotiation of Permissions and Obligations**
 - N.5: The PC stores the resulting policies formed from the negotiation process.
- **Registration and Exchanging of Policies of Consent**
 - G.1, G.3, G.7: A data consumer must have a valid policy of consent registered that is unique for the combination of data set, data consumer and data provider to use the data set. A policy is registered with a reference to the data provider, data consumer and data set, where the PC does not provide any functionality or result where the identity of a data owner is given.
 - G.2: A PC creates a context object for a policy of consent the moment it is registered. where consent is given for a certain context, e.g., for a certain data consumer provided by a certain data provider for a certain purpose.
 - G.4, G.5, G.6: A PC is a component in the data spaces that stores a unique policy for the combination of the data owner, data provider and data consumer. This means that a PC is the sole owner of a specific policy, making it the central point of truth for a policy definition. (It is allowed to have multiple PCs in a data space). Also, a data provider, data owner and data consumer can access and retrieve a specific policy at any time.
 - G.8: Retrieval of consent policies at a PC is done with a unique ID for a policy. This is usually only shared between the creator of the policy (CM, data provider or data owner) and the data consumer.
- **Revoking of Policies of Consent**
 - H.1, H.2: A data owner can revoke consent at any time at the PC. This means that the PC provides the functionality for a data consumer to withdraw a specific policy. When revoking a policy, the data consumer will be notified that its consent for usage has lost its validity.

- H.3: A data consumer can revoke its own policies. However, this entails that a data consumer must implement functionality that notifies the PC when the context changes from the data consumer side (call back to the PC). Also, the CM or the data provider can revoke a policy at the PC when the context changes.

5.2.4 Applicability

Use the policy catalogue when:

- There must be exactly one instance of a policy definition, and it must be accessible to clients from a well-known access point.
- A central actor is needed to notify multiple actors in the network on the revoking of consent.

Participants

The participants of the PC are:

- Consent Manager: Administrates policies and notifies the PC when the consent rules change for a specific policy.
- Data Consumer: Retrieves policies belonging to a data set they obtained consent for as proof for data usage.
- Data Provider: Retrieves policies to check access control for a specific data consumer.

5.2.5 Collaborations

The PC collaborates with the following entities:

Consent Manager: A consent manager will use the PC for registration of policies and notification of revoking of consent when the consent permissions and obligations of a specific policy are changed by the data owner. The revoking of consent based on context change activities can be seen in Figure 5.4. The registration activities can be seen in Figure 5.5.

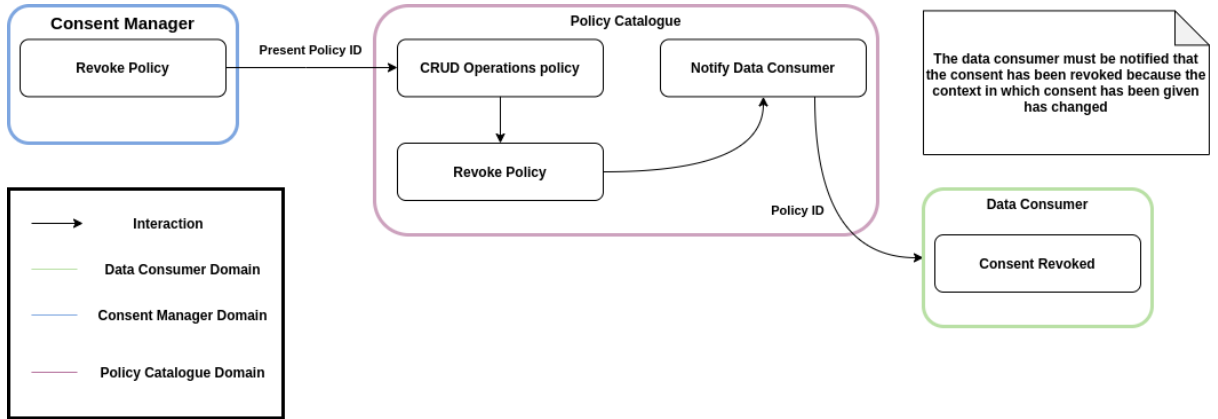


Figure 5.4: Revoking of Consent Policy by the Consent Manager

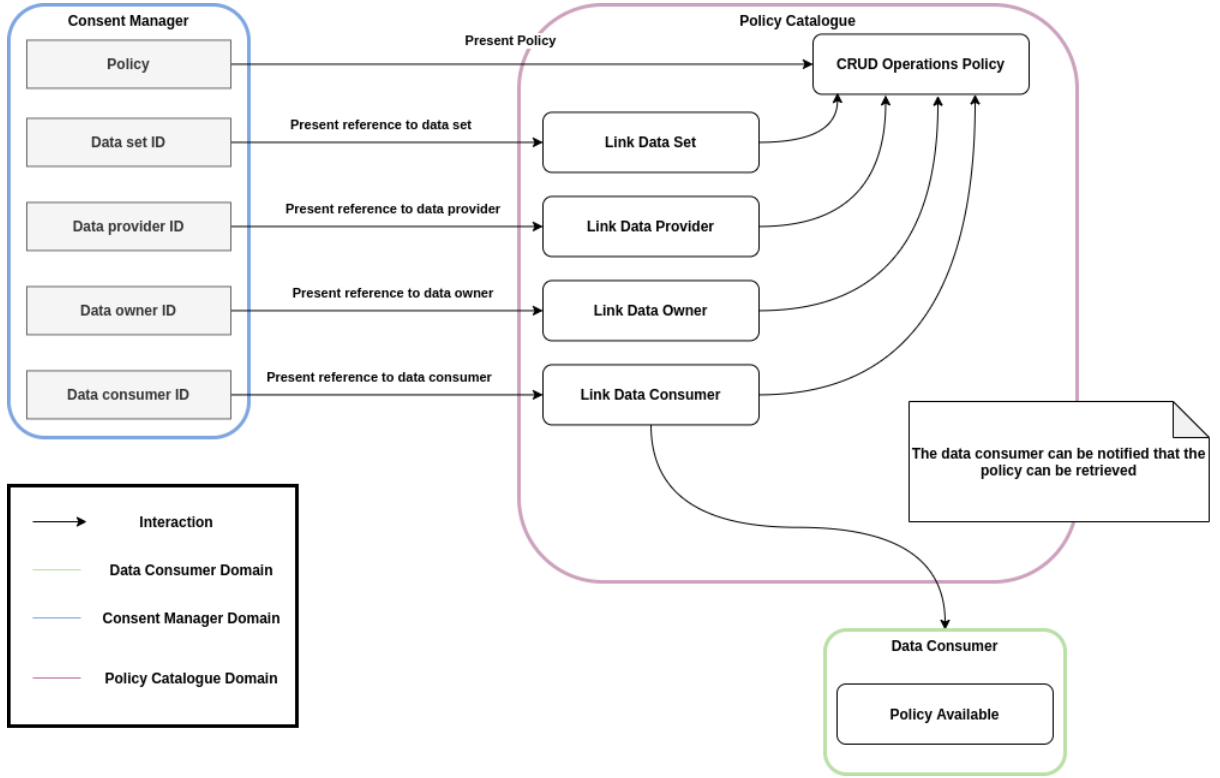


Figure 5.5: Policy Registrations Interactions of Consent Manager with Policy Catalogue

A consent manager has the following interactions with a PC:

- (Figure 5.4) Registration of policies with the corresponding data owner, a data consumer, data provider and data set. The PC makes sure that the context is set wherein consent has been given (time, location, the format of gathering consent, data owner, data provider, data consumer and data set).
- (Figure 5.5) Revoking of consent based on the change of context (change to permissions or obligations).

Data Consumer: A data consumer will use the PC for retrieval of obtained consent from a data owner in the form of policies. These activities can be seen in Figure 5.6.

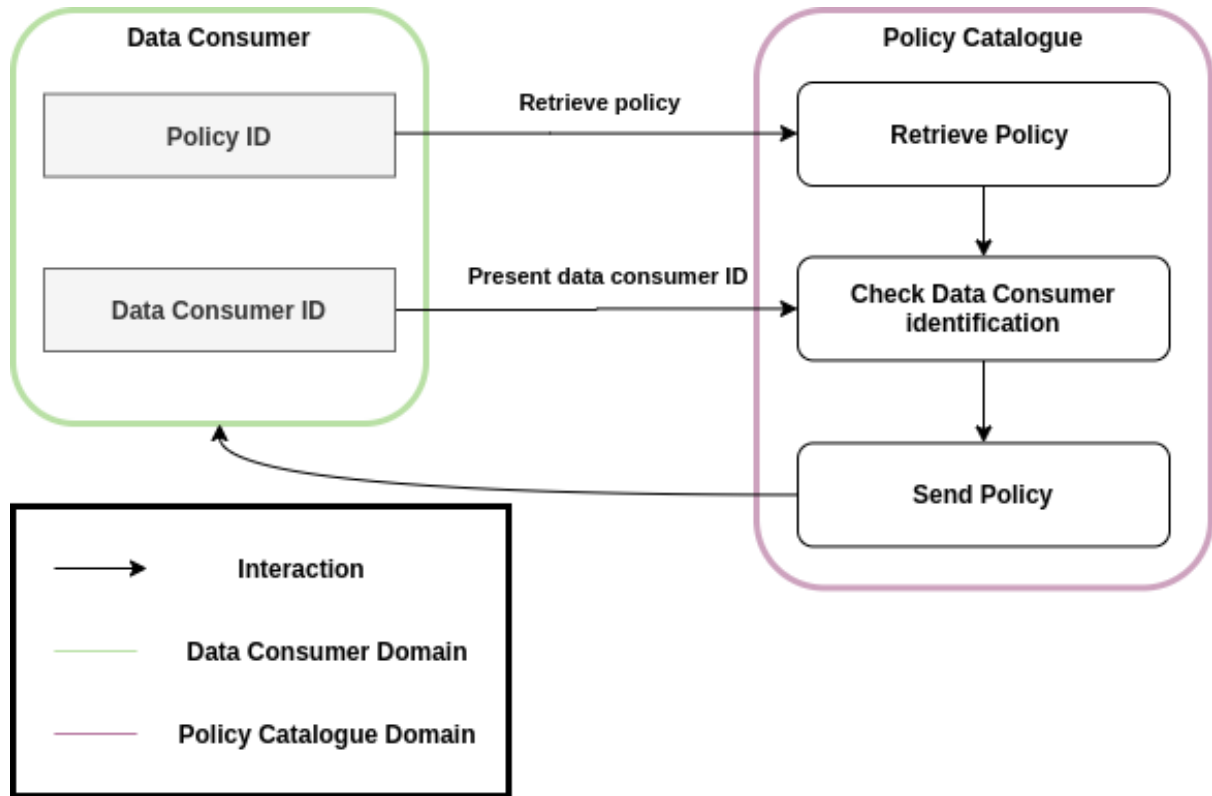


Figure 5.6: Interactions of Data Consumer with Policy Catalogue

A data consumer has the following interactions with a PC:

- Retrieval of policies with policy ID and data consumer identification (Figure 5.6).
- Notification by PC of revoking of consent (Figure 5.4).

Data Provider: A data provider will use the PC for validating the policy before providing the requested data set to the data consumer.

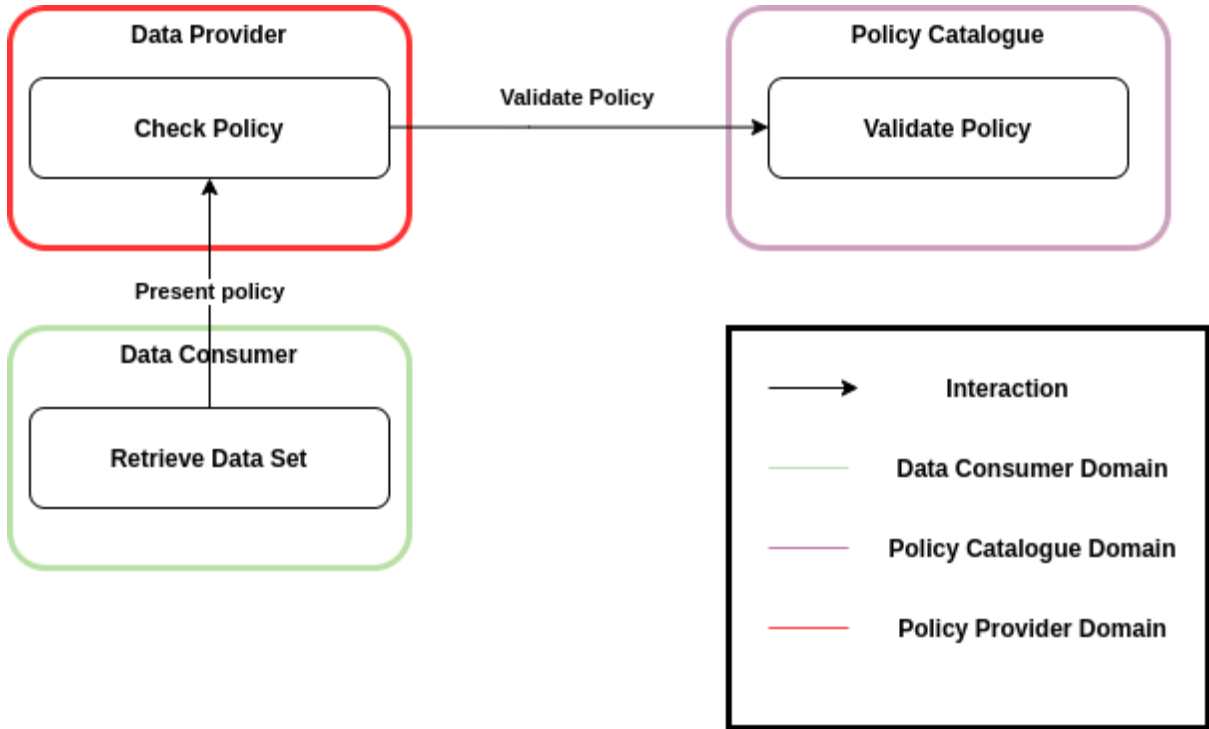


Figure 5.7: Interactions of Data Provider with Policy Catalogue

Figure 5.7 shows a data provider has the following interactions with a PC:

- Validation of the policies that are provided by the data consumer.

5.2.6 Consequences and Benefits

The PC component can come with the following consequences and benefits.

Consequences

- **Sticky Policies:** If the data space implementation makes use of sticky policies, a data provider must retrieve the policies for the data consumer at the policy catalogue. This is a consideration for the data space implementation and how the PC is implemented.
- **Data Consumer Callback Functionality:** When a PC is used, data consumers must implement a callback function that allows the PC to notify them when the consent is revoked.

Benefits

- **Controlled Access:** Controlled access to sole instances of policies. Because the policy catalogue functions as a central actor, it can have strict control over how and when clients in the network can access the policies.
- **Reduced Complexity:** Introducing a single access point for storing, retrieving, updating and deleting access and usage policing, can reduce the complexity of operations in scenarios where multiple policies of different owners are in use.
- **Single Point for Storage:** The central characteristic allows for single storage of the policy parameters between all data providers and data consumers.

5.2.7 Implementation Considerations

The following points are to be considered for the implementation of a policy catalogue component:

- **Policy Reference:** There must be commonly used way for the identification of policies. This will allow actors to interact with a specific policy.

- **Data set, Data Consumer, Data Provider, Data Owner, and Consent Manager Reference:** Policies are registered with a unique combination of IDs. Therefore, to implement a PC, a shared manner of identification of actors needs to be used by all actors in the data space.

5.3 Policy Broker

5.3.1 Intent

The intent of the policy broker is to provide a unified interface to a set of consent managers in the network, to enable categorization and searching of permissions and obligations of consent. The Policy Broker (PB) enables clients to search for a specific set of permissions and obligations, instead of a specific data set. This is a direct reaction to data brokers in data spaces such as IDS [2].

5.3.2 Motivation

The creation of the policy broker component is motivated by the current lack of a capability to search for a specific set of permissions and obligations of consent. Usually, data consumers would search for specific data sets a data provider or data broker, and then obtain consent to use the data set. However, it could also be the case that a data consumer is looking for a specific set of permissions and obligations for its data usage (e.g. having as few restrictions as possible to train a machine learning model). In that context, the data consumer is not interested in a specific data owner or data set but is interested in a set of permissions and obligations that belong to a specific data category. To find the permissions and obligations that are connected to a specific data category, the data consumer would have to know all the consent managers and the permissions and obligations that these consent managers store. Utmost, every data consumer would have to know each consent manager and its permissions and obligations in the network to get an overview of the different policies available. This is a problem that grows with the scale of the network and opens the need for a service actor.

The Policy Broker shall act as a service actor within distributed networks. Data spaces reference architectures such as the IDS [2] encourage the creation of service actors (Data brokers, Clearinghouses) that can be used by all the actors in the network. This distribution of behaviour results in less direct connections because these service actors will provide functionality that will unburden the actors.

In a sum, the Policy Broker as a service actor provides an interface for searching for specific sets of permissions and obligations corresponding to a data category. Consent Managers can register themselves at the PB to show which data categories they facilitate and what kind of permissions and obligations they hold.

5.3.3 Requirements Implementation

The policy broker is an optional component that flows naturally from the implementation of consent. Therefore it does not implement any requirements.

5.3.4 Applicability

Use a policy broker when:

- An interface is needed for a complex network of consent managers. With the introduction of the consent manager and the policy catalogue service, complex networks could occur, where actors store their permissions and obligations and their resulting policies at different consent managers and policy catalogues. A policy broker can provide a default service for such a network.

5.3.5 Participants

The participants of the PB are:

- **Consent Manager:** Registers and provides an overview of the data categories it holds permissions and obligations for at the PB
- **Data Consumer:** Queries the PB for permissions and obligations that fall within its intent of usage criteria for a specific data category.

5.3.6 Collaborations

The PB collaborates with the following entities:

Consent Manager: A CM will use the PB to register its data categories. For each data category, the CM will give an overview of the permissions and obligations it stores. These activities can be seen in Figure 5.8.

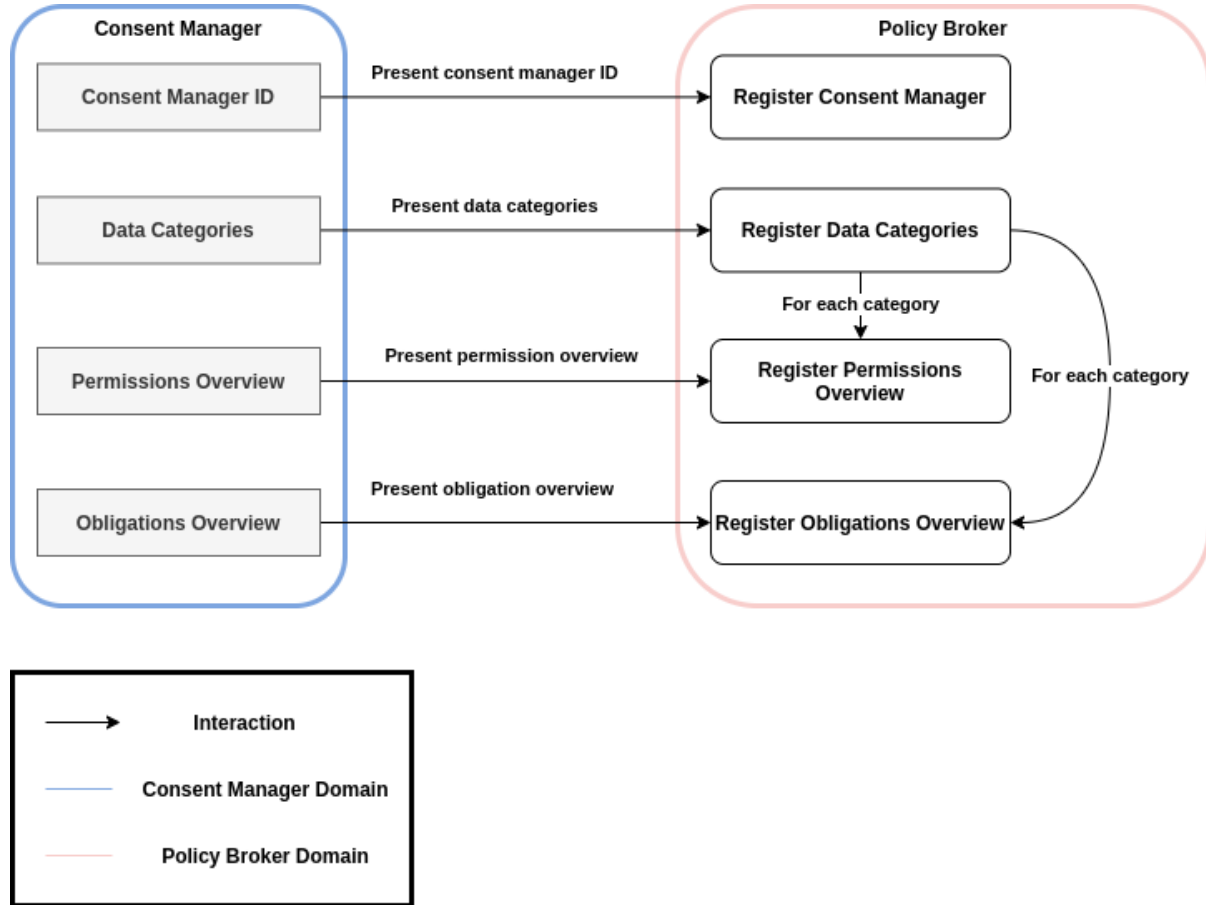


Figure 5.8: Interactions of Consent Manager with Policy Broker

Figure 5.8 visualizes the interactions of a CM with a PB:

- Registration as a CM at the PB.
- Registration of data categories.
- Registration of permissions overview corresponding to data categories.
- Registration of obligations overview corresponding to data categories.

Data Consumer: A data consumer uses the PB to search for permissions and obligations for specific data categories. The PB will return a selection of data providers and their data sets where the permissions and obligations fall within the intent of usage parameters that were specified by the data consumer. These activities can be seen in Figure 5.9.

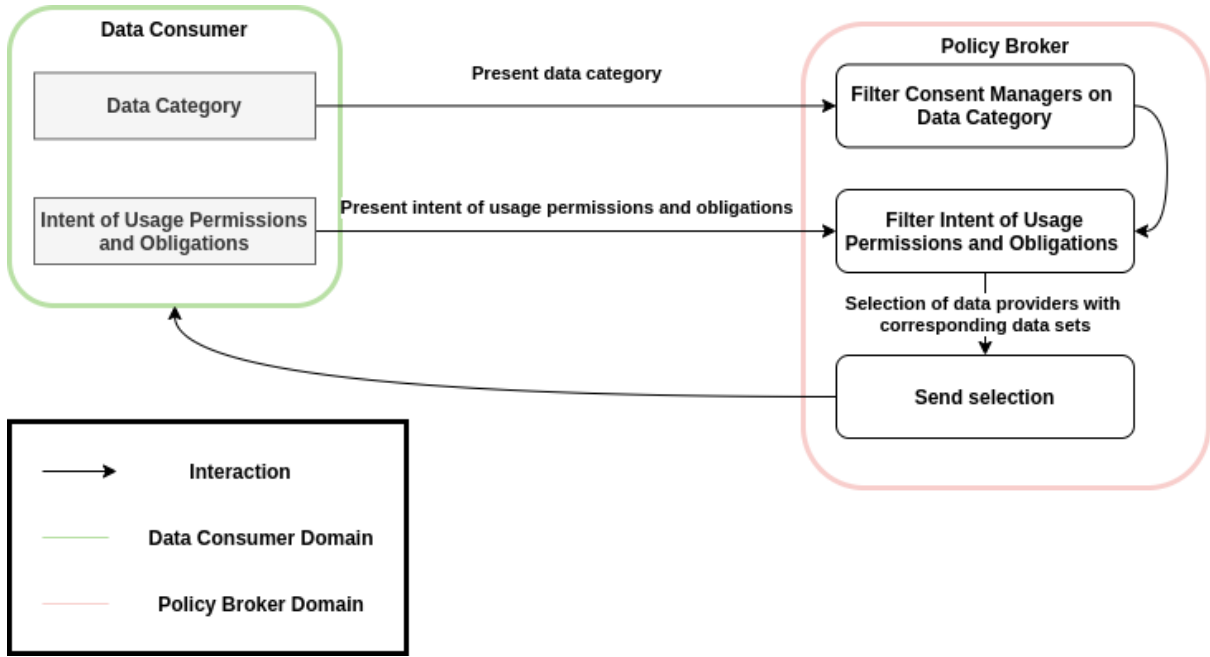


Figure 5.9: Interactions of Data Consumer with Policy Broker

Figure 5.8 visualizes the interactions of a data consumer with a PB:

- Querying for data categories.
- Presenting intent of usage permissions and obligations.
- Retrieval of data providers.

5.3.7 Consequences and Benefits

The PB component can come with the following consequences and benefits.

Consequences

- **Data Categories:** To implement a PB, there must be an enumeration of the data categories used throughout the data space that can be leveraged by a CM to categorize its data sets.
- **Intent of Usage Permissions and Obligations:** Just as with the CM, there must be a shared vocabulary that allows clients to express their intent of usage permissions and obligations. For this research, the vocabulary will be the same as for defining the permissions and obligations of consent by the data owner.

Benefits

- **Reverse Searching:** A PB allows searching for specific policies. Searching for specific policies could be relevant for use cases that focus on acquiring large amounts of data. In such projects, limitations of data should be as little as possible to get the most significant results.

5.3.8 Implementation Considerations

The following points are to be considered for the implementation of a policy broker component:

- **Shared Vocabulary:** The vocabulary for expressing permissions and obligations must be the same as being used by the CM.
- **Data Categories:** To make a CM search-able, all the data categories of the specific CM must be registered at the PB. This allows the PB to categorize the different CM's in the data space.

Chapter 6

System Demonstration for Consent Management in Data Spaces

This research proposes an architecture that leverages the CM, PB, and PC components that were introduced in the previous sections (see Section 5.1, 5.3 and 5.2). In this demonstration a basic architecture is first proposed to provide a reference to a data space without consent management. Then the reference architecture of this research is proposed in combination with a scenario. This demonstration provides multiple UML sequence diagrams to show how the different components implement the processes of consent management.

6.1 Basic Data Space Architecture Without Consent Management

This section introduces a basic architecture of a data space that does not leverage consent management. The objective is to be able to conduct a comparison of each architecture that makes use of the proposed CM, PB, or PC. This basic architecture is based on the definition of data spaces as defined in A.

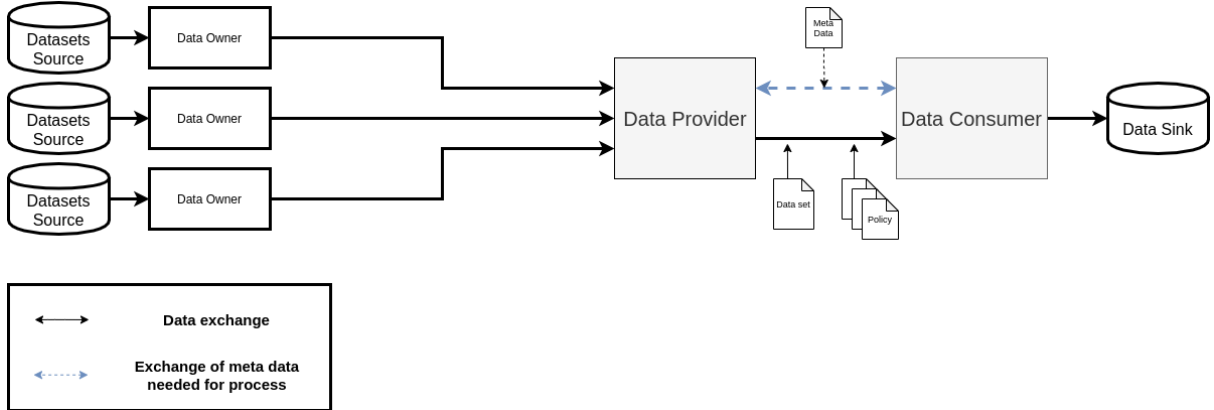


Figure 6.1: Basic Data Space Architecture Without Consent Management

The basic architecture is representing a data space as defined in the terminology section (Appendix A.1). Thus, the basic architecture depicted in Figure 6.1 does not reflect the consent management components that were introduced in this research.

As elaborated in the Problem Statement (Section ??) current data space architectures, as represented in Figure 6.1, do not facilitate consent management. The lack of capability is especially relevant in cases where the data owner and the data provider are separate entities, and as a result, composed data sets can occur.

Nevertheless, a basic architecture of data spaces as represented in Figure 6.1 can be useful to a va-

riety of cases. The main benefit of the basic data space architecture is the predefined policies of consent in use cases where there is no need for dynamic relations between a data provider and data consumer. Consent management is of most relevance in cases that allow dynamic connections to be formed between actors in a data space. In such cases where a data consumer does not know who is going to use their data and where their data is located.

6.2 Scenario Reflected in the Proposed Architecture

This section outlines the actors and processes in consent management that are applied in the proposed reference architecture. The scenario represents the following actors:

- 2 Data Provider
- 3 Data Owners
- 2 Data Consumer

The number of actors has been deliberately chosen to be able to show the capabilities of consent management in a representative case while keeping complexity as low as possible.

6.3 Processes of Consent Management Reflected in the Proposed Architecture

As defined in the Section 4.2.1 the processes of consent management in the scope of this research are:

- Defining of permissions and obligations of consent.
- Registration of permissions and obligations and linking with a data set.
- Negotiation of permissions and obligations.
- Forming of policies of consent-based of the permissions and obligations given.
- Registration and exchanging of policies of consent
- Revoking of policies of consent.

6.3.1 Proposed Architecture of Consent Management

In this section a reference architecture is proposed for consent management. The proposed architecture combines the findings of the previous chapters by implementing the Policy Broker, Policy Catalogue, and the Consent Manager.

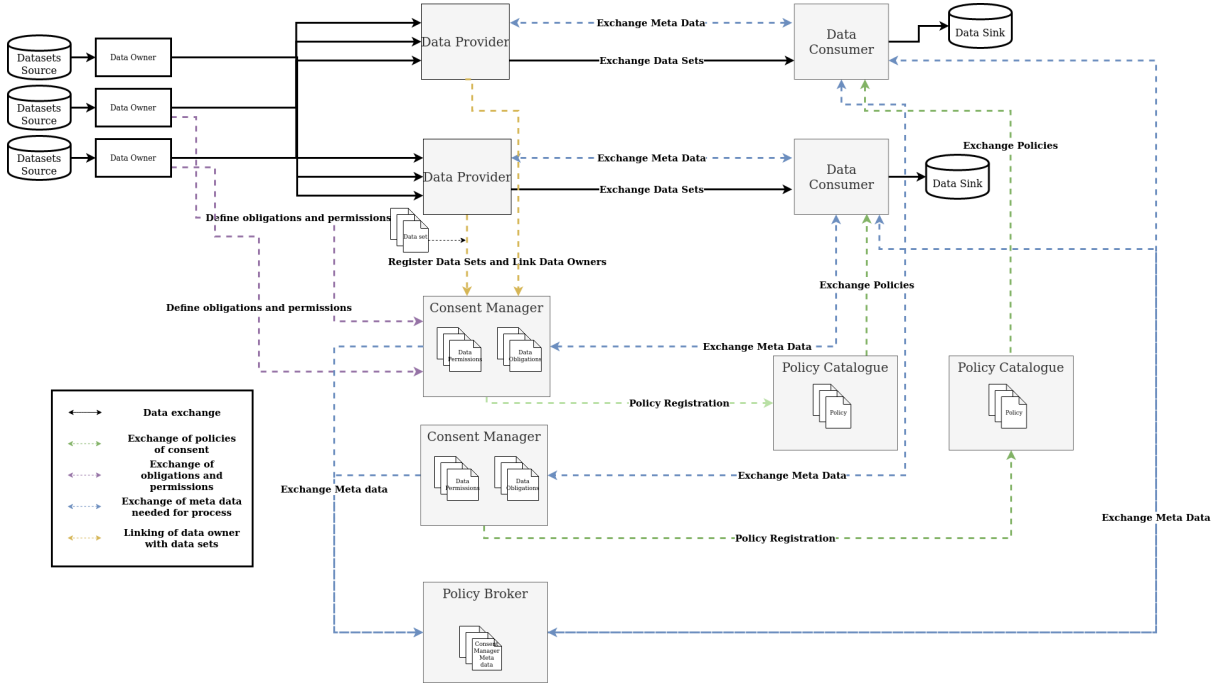


Figure 6.2: Proposed Architecture with Consent Manager, Policy Catalogue and Policy Broker Components

The proposed architecture in Figure 6.2 integrates a consent manager, policy catalogue and a policy broker. The consent manager will be used for storing the permissions and obligations, linking the data owners to the data sets, resolving the requests for the consent of data usage by data consumers and mapping the permissions and obligations into the specific REL in use in the data space. Policies formed by the consent manager are stored at the policy catalogue.

The architecture consists of the following components:

- **Consent Manager:** The CM implements the process of defining permissions and obligations, storing them and converting them to policies of consent. Data providers and data owners will register at the CM, where they can register data sets that will be linked to permissions and obligations.
- **Policy Catalogue:** The PC implements the processes of storing and exchanging policies of consent. Data consumers will retrieve the policies from the PC, that where obtained from the CM. The CM and the data owner can use the PC to revoke policies belonging to a data set.
- **Policy Broker:** The policy broker will have an overview of all the CMs that are present in the data space. Data consumers can use the policy broker to filter CMs that are in line with the intent of usage permissions and obligations.

The goal of the proposed architecture is to implement all processes of consent management a defined in Chapter 4.2.1. The following sections will show, with the use of sequence diagrams, examples of implementations of the processes that define consent management.

6.4 Process Interactions

For the reference architecture, a set of sequence diagrams are made to show the different interactions between components and actors in the network. The functions specified are only for illustration purpose and will differ depending on the implementation.

6.4.1 Defining of Permissions and Obligations of Consent

The proposed architecture reflects the definition of permissions and obligations of consent as an interaction between the data owner and the CM. Through a set of interactions, a data owner will define and register its permissions and obligations of consents, and a CM will store the permissions and obligations.

These interactions are depicted in the sequence diagram in Figure 6.3.

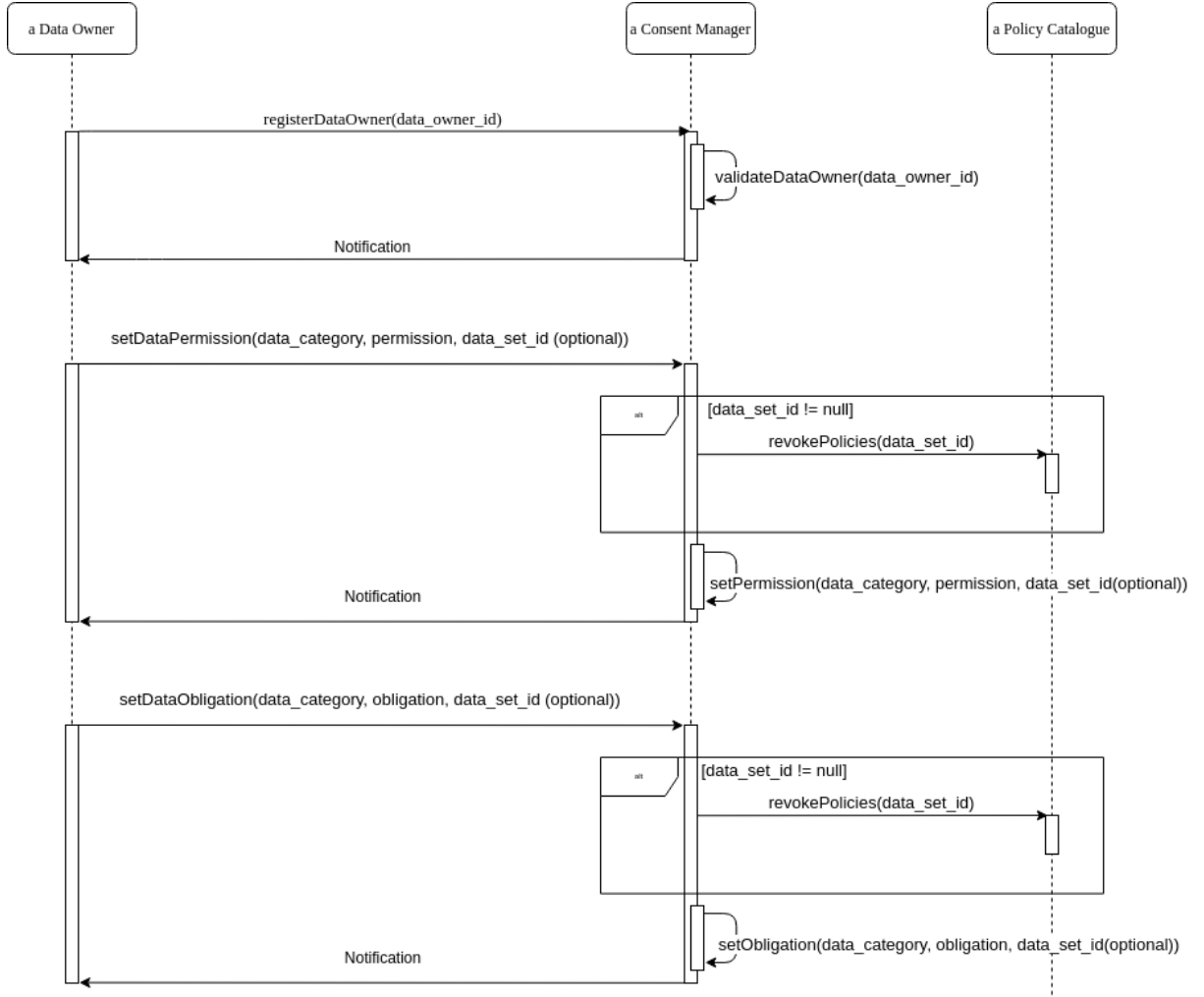


Figure 6.3: Sequence Diagram of Data Owner Interactions with Consent Manager

The sequence diagram in Figure 6.3 illustrates the following sequences:

- Registration as a data owner, if the data owner is not registered at the CM.
- Registration of data permission
 - If data set ID is given, and if there are already policies formed for the data set, all corresponding policies will be revoked at the policy catalogue (context in which consent has been given, has changed).
 - If the permission is linked to a data set that is already defined, all corresponding policies will be revoked at the policy catalogue (context in which consent has been given, has changed).
- Registration of data obligations
 - If data set id is given, and if there are already policies formed for the data set, all corresponding policies will be revoked at the policy catalogue (context in which consent has been given, has changed).
 - If the obligation is linked to a data set that is already defined, all corresponding policies will be revoked at the policy catalogue (context in which consent has been given, has changed).

6.4.2 Registration of Permissions and Obligations and Linking with a Data Set

The proposed architecture reflects the process of linking permissions and obligations of consent with a data set as an interaction between a data provider and the CM. In the proposed architecture, the CM

is responsible for linking all the given data owners to the data set and automatically checking whether a given data owner is registered at the CM. In the linking process that is visualized in Figure 6.4 the obligations and permissions are linked that correspond to the '*data_category*' attribute.

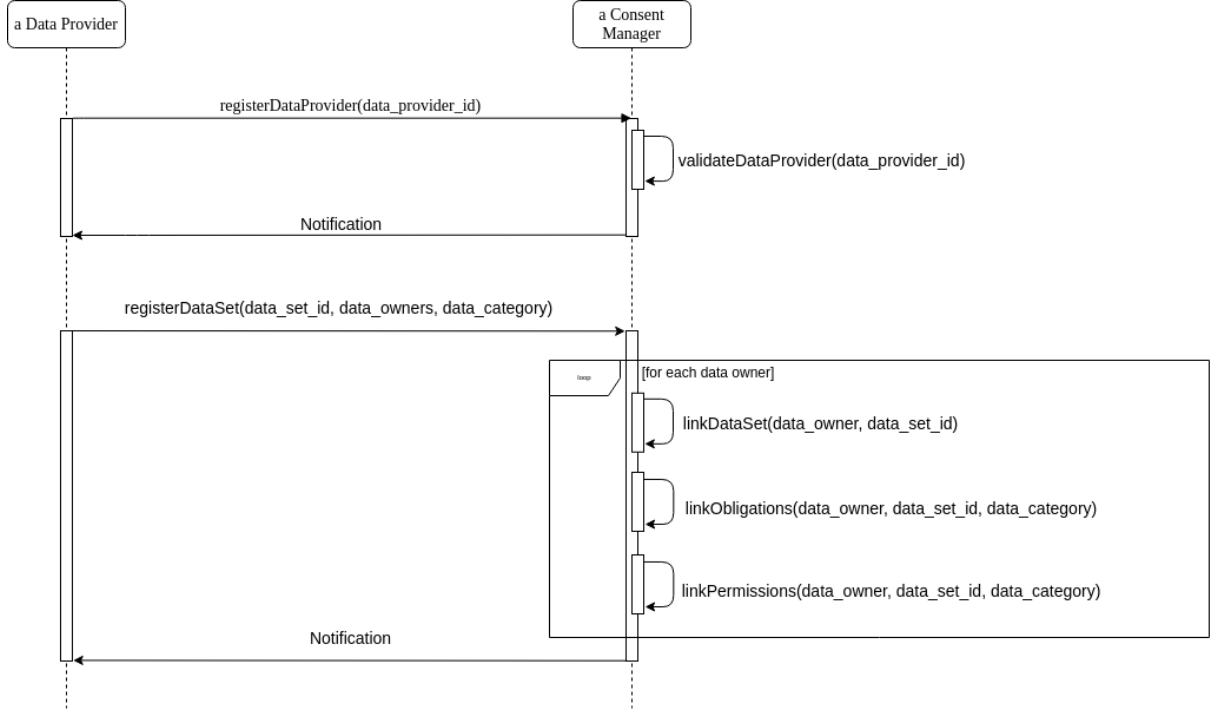


Figure 6.4: Sequence Diagram of Data Provider Interactions with Consent Manager

The sequence diagram in Figure 6.4 shows the following sequences:

- Registration as a data provider, if the data provider is not registered at the CM.
- Registration of data sets
 - For each data owner, the corresponding obligations and permissions are linked to the data set.

6.4.3 Forming of Policies of Consent-Based of the Permissions and Obligations Given

The process of requesting consent of usage and forming of policies is in the proposed architecture an interaction between the data consumer and the CM. Where a CM will form the policies based on the permissions and obligations of the data owners corresponding to the given data set. At the end of the interaction, the policies are stored at the policy catalogue. The data owner interactions with the CM are depicted in the sequence diagram in Figure 6.5.

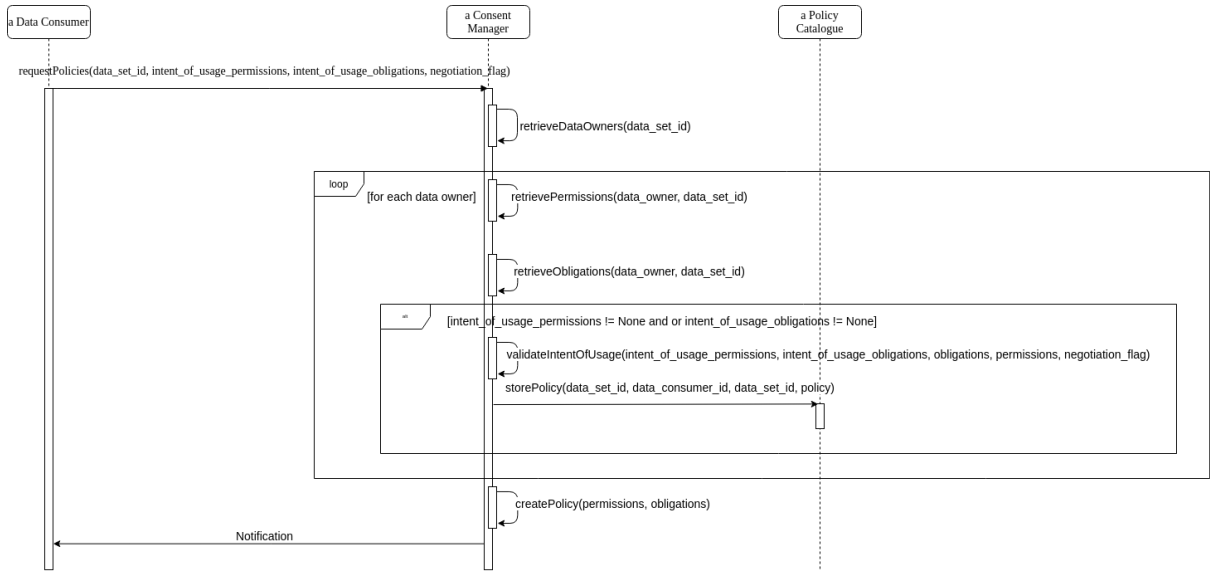


Figure 6.5: Sequence Diagram of Data Consumer Interactions with Consent Manager

A negotiation process can be started, if the data consumer provides intent of usage permissions and obligations that are conflicting with the permissions and obligations of one or more data owners. The negotiation process is only started, if the intent of usage permissions and obligations are given and the negotiation flag is set. This process is triggered by the *'validateIntentOfUsage'* function in Figure 6.5 and the process can be seen in Figure 6.5.

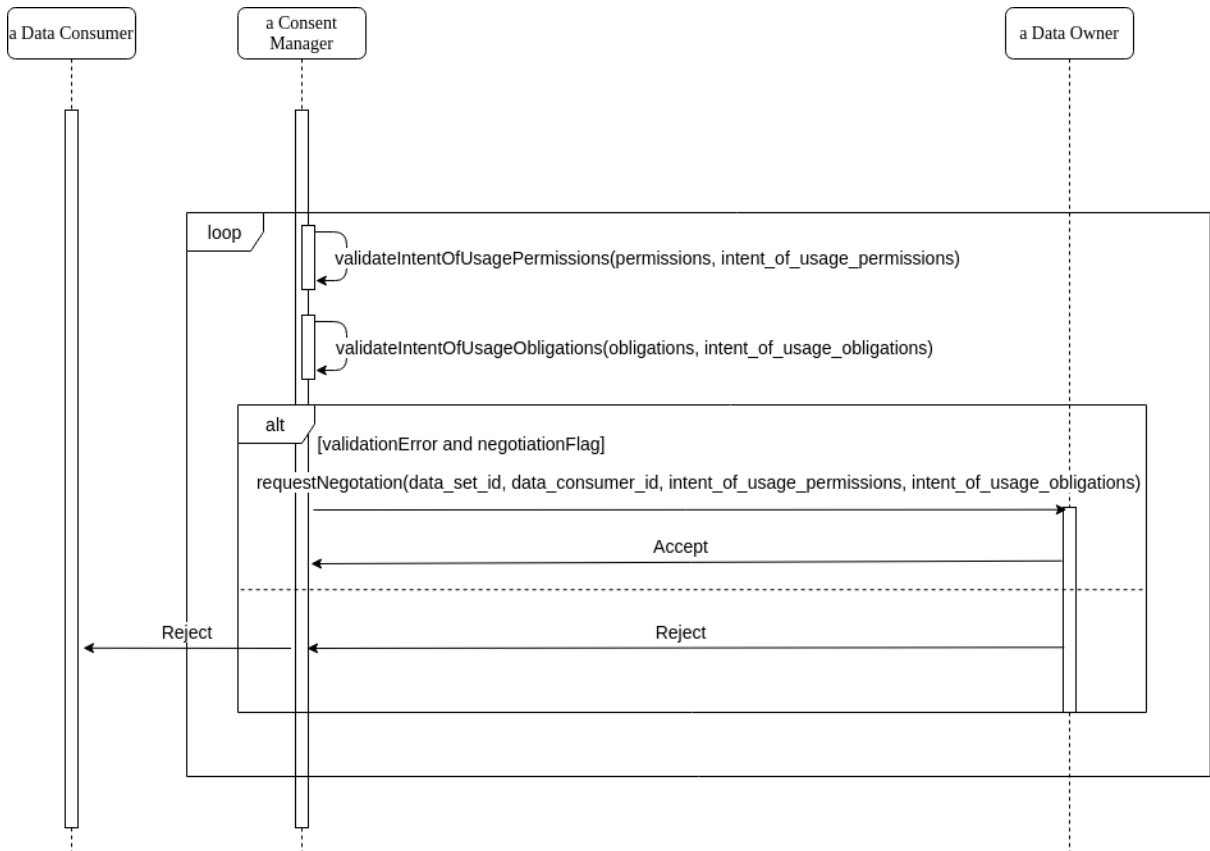


Figure 6.6: Sequence Diagram of Intent of Usage Permissions and Obligations Validation and Negotiation

In the proposed components, if a rejection is given by one of the data owners in the negotiation

process, the process is stopped. This means that consent is not given for the usage of data set for the data consumer. Custom implementations, based on this proposed reference architecture, can look at removing the data entry of the specific data owner that does not give consent. This will not give an overall rejection of consent for the data consumer if a subset of the data owners rejects the negotiation.

The sequence diagrams in the given figures ?? ?? show the following processes:

- Request for the consent of usage for a specific data set
 - If the intent of usage permissions and obligations are given, they will be evaluated with the permissions and obligations of the data owners registered at the consent manager. This evaluation must be done by custom mechanisms specific to the given use case.
 - If the intent of usage permissions and obligations conflict with the given permissions and obligations and the negotiation flag is set by the data consumer, and a negotiation process is triggered.

6.4.4 Registration and Exchanging of Policies of Consent

After consent has been given, a data consumer can interact with the policy catalogue to retrieve the policies. This process can be seen in Figure 6.7.

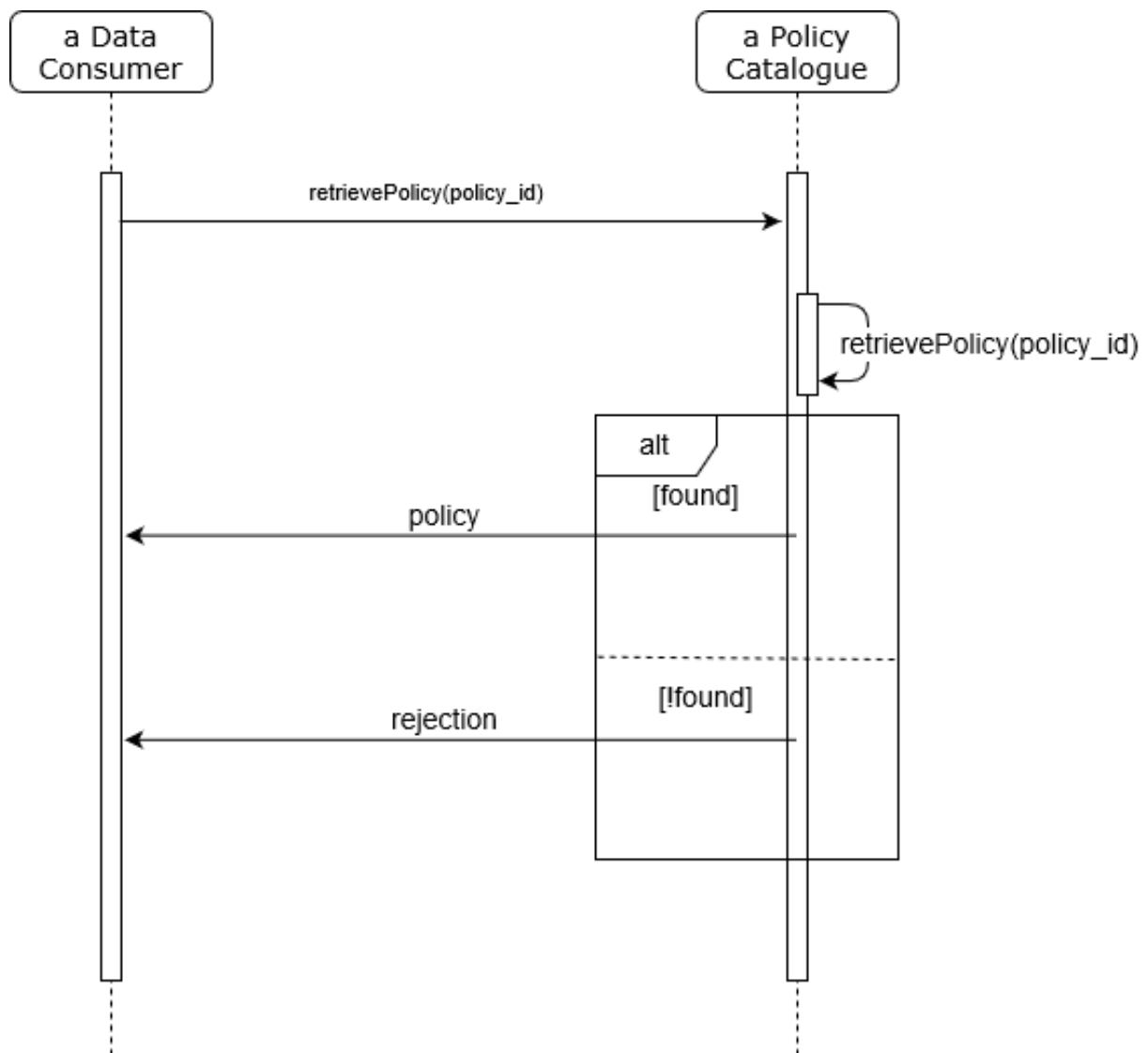


Figure 6.7: Sequence Diagram of Retrieval of Policies

The sequence diagrams in the given figures show the following processes:

- Retrieval of policy with given policy ID
 - If the policy is present, the policy is given.
 - If the policy is not present, the data consumer gets a rejection.

6.4.5 Revoking of Policies of Consent

The revoking of policies process can be triggered by the data owner and the consent manager. The consent manager will make use of this process if one or more permissions or obligations are changed for the given data set that belongs to a policy. In this situation, the context is changed in which consent has been given. Therefore, the consent must be revoked.

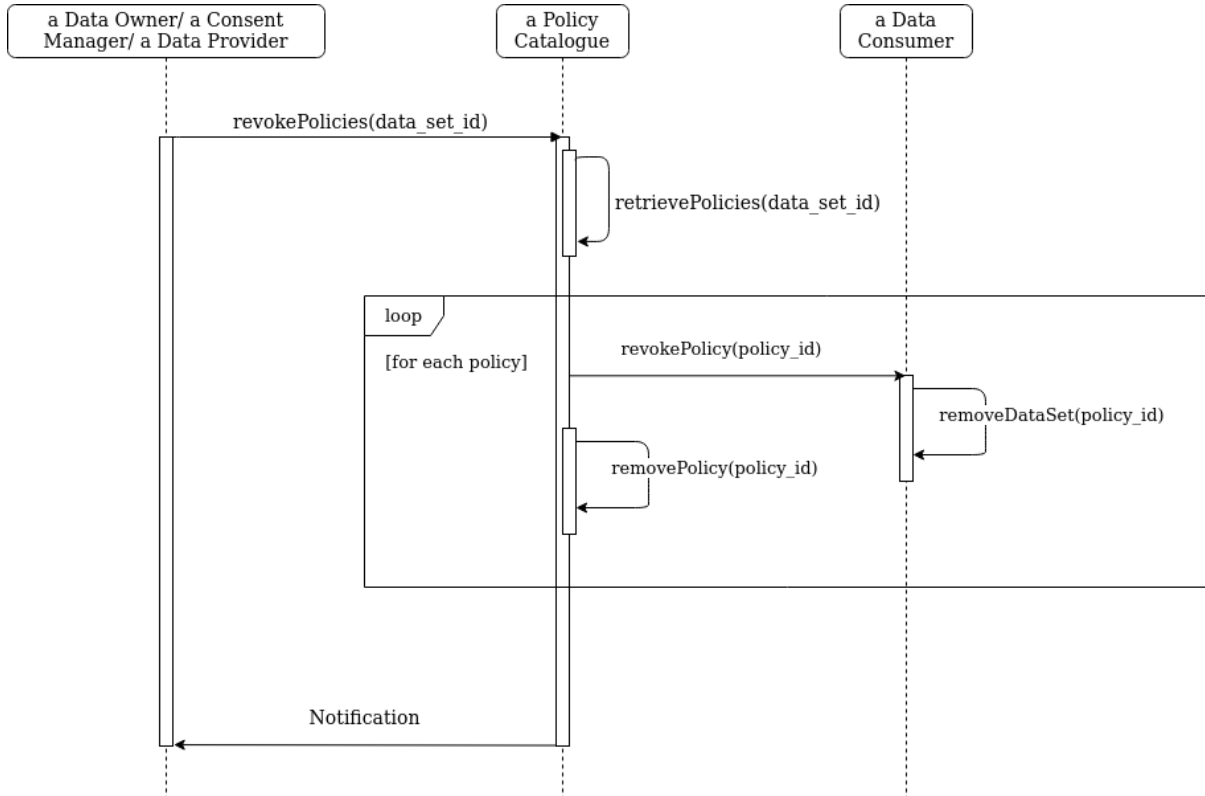


Figure 6.8: Sequence Diagram of Revoking of Consent by a Data Owner

The sequence diagram in figure 6.8 shows the following processes:

- Revoking of policies issued by data owner, data provider or consent manager.
- Revoking of policies with a given data set ID.
- Removal of a data set at the data consumer side.
- Removal of a policy at the policy catalogue.

6.4.6 Searching for Policies with Permissions and Obligations of Intent

This process demonstration belongs to the policy broker. The policy broker does not implement functionality that is considered a must (MoSCoW) for consent management. However this research wants to show how additional functionality can be achieved that can be useful for data spaces.

In figure 6.9 the process can be seen where a data consumer will present its intent of usage permissions and obligations with a given data category. The policy broker will match the permissions and obligations belonging to the given data category to that of the permissions and obligations of its registered consent managers. From this comparison a selection is returned with consent managers where the data sets that match the intent of usage reside.

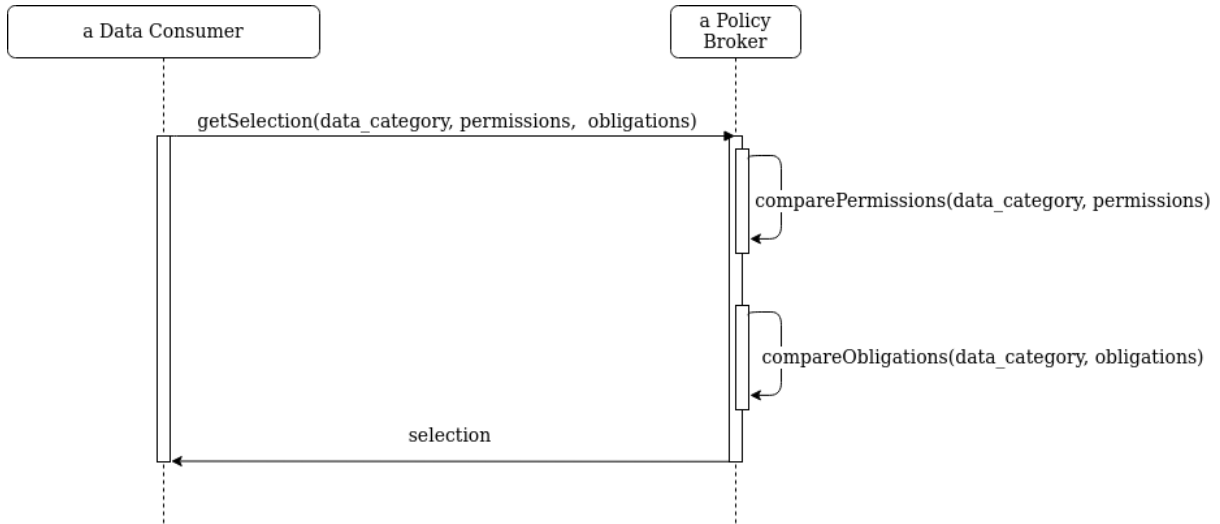


Figure 6.9: Sequence Diagram of Get a Selection of Consent Manager with Intent of Usage Permissions and Obligations

The sequence diagram in figure 6.9 shows the following processes:

- Providing a data category with intent of usage permissions and obligations by the data consumer to the policy broker.
- Comparison of permissions to the permissions of the registered consent managers
- Comparison of obligations to the obligations of the registered consent managers
- Providing a selection of matching consent managers with the intent of usage permissions and obligations.

6.4.7 Implementation Concerns

This research has chosen to not show every process that is part of the proposed components. Mainly the processes of registration, updating and retrieving are left out. The reason for this decision is that these processes are rather straightforward and also use case specific (e.g. working with the unique identification mechanisms of the actors in the data space)

- **Linking of Obligations and Permissions to Data Sets:** The linking of a data owner to a data set and finding the corresponding permissions and obligations requires a mapping mechanism. In the interactions in Figure 6.3 this is done by data categories, where a data category specifies how a data set can be linked to a specific set of permissions and obligations. However, these categorizations must be very specific. An example with the given use case would be for a household with a smart meter installed, where data categories could be identified as:
 - **Household energy usage:** A category that covers all data sets related to energy usage.
 - **Household energy storage:** A category that covers all data sets related to energy storage.
 - **Household energy generation:** A category that covers all data sets related to energy generation.
- **Mapping Mechanism:** As described by the implementation concern of the CM, there must be a mapping mechanism in place that can convert the set of permissions and obligations applicable for a specific data set into a REL policy. There is currently no framework in place for this. Therefore, implementing consent management with the use of RELs to form policies that are used in an access and usage control systems require a custom mapping mechanism.
- **Matching of intent of usage permissions and obligations to that of the registered consent managers** at the policy broker can be very computing intensive. With this statement it is assumed that every permissions and obligations of a certain data category from a consent manager will be compared. Real implementations could make use of caching mechanisms or efficient sorting algorithms to make this process more efficient. This was out of scope for this research to make any recommendations for.

Chapter 7

Use Case Demonstration

In this chapter, the results of implementing a prototype are presented. The prototype is based on the proposed reference architecture (Chapter 6.3.1). The source code of the prototype can be found at: https://github.com/MDUYN/tno_consent_management_data_spaces-

7.1 Setup

For the prototype, a Kubernetes [35] cluster is used. In the cluster a set of applications are defined that represent the following components of the proposed reference architecture:

Service	Language of Implementation	Description
Data Provider	Python	Python REST API with functionality according to the specifications of the data provider. The data provider contains a set of data sets with corresponding data owners references.
Consent Manager	Python	Python REST API with functionality according to the proposed specification of the consent manager.
Policy Catalogue	Python	Python REST API with functionality according to the proposed specification of the policy catalogue.
Data Consumer	Python	Python REST API with functionality according to the proposed specification of the data consumer.
Policy Broker	Python	Python REST API with functionality according to the proposed specification of the policy broker.

The Kubernetes cluster was run on a single node with the use of minikube. This allowed the researchers to run the services locally on their computers. Each REST API was implemented with the Flask web framework [36]. Additionally, each service has access to an individual PostgreSQL[37] database instance for storage of models. For the orchestration of the PostgreSQL databases in Kubernetes, the Postgres operator of Zalando was used [38].

7.2 Representation of the Reference Architecture

The combined set of components give an idea of how consent management should work in data spaces. This means that this prototype does not include the functionality expected in a real use case (authentication, authorization, notifications, etc.). The objective of this prototype was to show the interactions of the different components as clearly as possible. Each component that is introduced is reflected in a dashboard.

In the following sections, for each component, the most important REST endpoints are explained. Also, an impression is given of the dashboard that was used to communicate with the components.

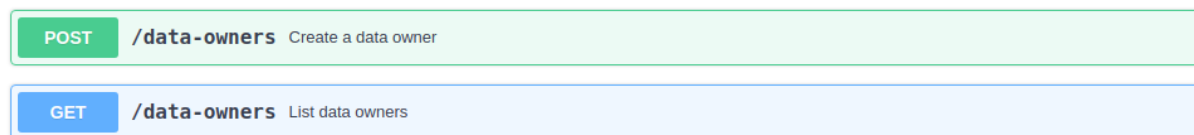
7.2.1 Data Provider Component

The data provider component implements the most basic functionality that is required in a data space. The data provider component implements two endpoints for data owner operations and two endpoints for data set operations (where an endpoint is a specific URL of the REST API). The endpoints and functionalities are introduced in the subsections below.

Data Owners Functionality of the Data Provider Component

Data owners are registered with their unique ID. The data provider is then allowed to provide the data of the registered data owner to data consumers.

The data provider component in the prototype provides the following two REST endpoints for the management of data owners:



POST	/data-owners	Create a data owner
GET	/data-owners	List data owners


Figure 7.1: REST Endpoints for Data Owner-Related Services at the Data Provider

Figure 7.1 shows the two endpoints for data owner-related services at the data provider component: the first API endpoint `/data-owners` helps to register a data owner in a post request. The second API endpoint `/data-owners` lists all data owners in a get request.

Data Sets Functionality of the Data Provider Component

A data set, similar to a data owner, is identified with a unique ID. Additionally, a data set needs a list of data owners and a data category. The list of data owners is a representation of the owners of the data. The data category is later used by the consent manager to map the data set to the obligations and permissions of the data owners.

The data provider component in the prototype provides the following three endpoints for data sets operations.



POST	/data-sets	Create a data set
GET	/data-sets	List data sets
GET	/data-sets/{dataSetId}/policy/{policyId}	Retrieve a data set with an obtained policy

Figure 7.2: REST Endpoints for Data Set-Related Services at the Data Provider

Figure 7.2 shows the three endpoints of the data set-related services at the data provider component: the first API endpoint `/data-sets` creates a data set in a post request. The second API endpoint `/data-sets` lists all data sets in a get request. The third API endpoint `/data-sets/dataSetId/policy/policyId` retrieves a data set with an obtained policy from the data provider component.

Dashboard Implementation of the Data Provider Component

The dashboard that interacts with the REST endpoints focuses on registration of data owners (7.3) and registration of data sets (7.4).

Data Owners

Data Owner ID

3bc29a59-6475-4b3e-a025-f043d6b8058e

Data Owner Registration

Data Owner Identification

Fill in data owner identification

The identification of the data owner

REGISTER

Figure 7.3: Dashboard of Registration and Listing of Data Owners

The top component of Figure 7.3 provides an overview of all registered data owners.

As can be seen in the second component of Figure 7.3, a data owner is registered by providing an unique UUID that is representative of the data owner.

Registered Data Sets

Data Set ID

9d5973b5-5748-4544-837c-e76c24b64dd4

Data Set Registration

Data set Identification

Fill in data set identification

The identification of the data set

Data Category

energy_generation_data

Data Owners

☒ 3bc29a59-6475-4b3e-a025-f043d6b8058e

REGISTER

Figure 7.4: Dashboard of Registration and Listing of Data Sets

As can be seen in Figure 7.4 the top component lists all registered data sets of the data provider.

The registration of a data set is reflected in the second component of Figure 7.4. In the process of registering a data set, a unique UUID for the data set must be given, a data category must be specified, and the data owners must be selected that are part of the data set.

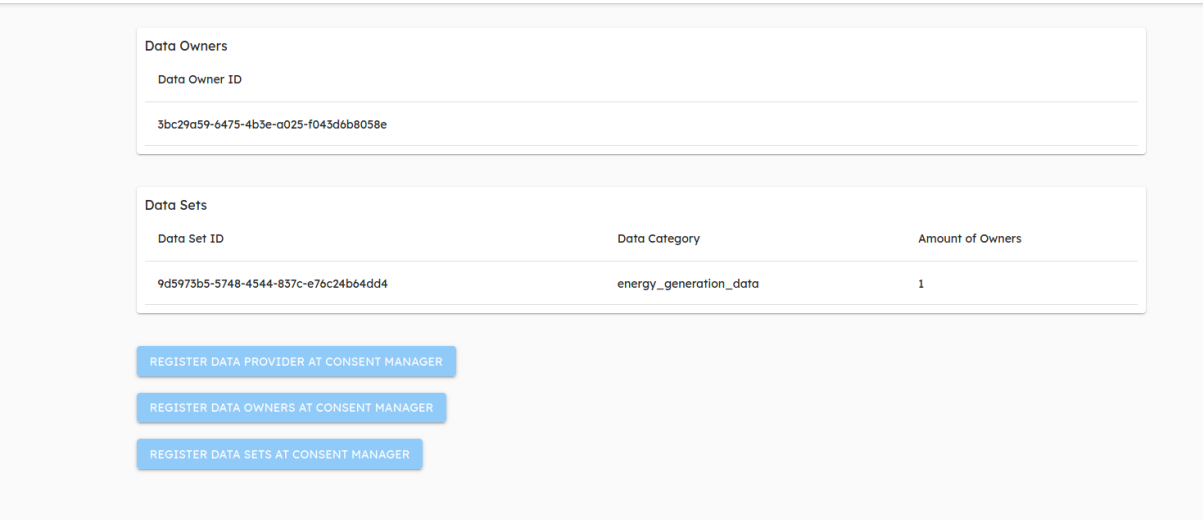


Figure 7.5: Dashboard Overview of the Data Provider

The dashboard that the researchers implemented also provides an overview page that shows all the registered data owners and registered data sets. This can be seen in Figure 7.5. On this overview page, there are also a set of buttons available. These buttons can be used to register the resources of the data provider at a consent manager.

7.2.2 Consent Manager Component

The consent manager implements all the functionality needed for forming of policies and handling requests for consent. The consent manager also makes sure to remove any policies that are formed with outdated permissions and/or obligations. The endpoints and functionalities of the consent manager prototype are introduced in the subsections below.

Data Provider Functionality of the Consent Manager Component

A data provider is registered at a consent manager with its unique ID (UUID). At the consent manager component, a data provider can then register its data sets and data owners. Additionally, the consent manager implements endpoints for listing of data providers that are registered at the consent manager.

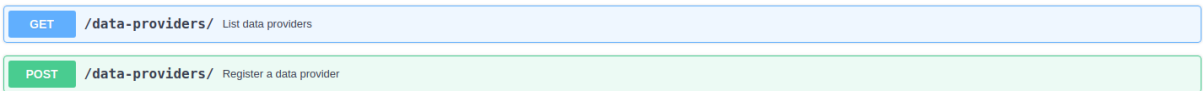


Figure 7.6: REST Endpoints for Data Provider Related Services at the Consent Manager

Figure 7.6 shows the REST endpoints for data provider-related services at the consent manager. The first API endpoint `/data-providers` lists all data providers that are registered at a consent manager. The second API endpoint `/data-providers` helps to register a data provider at a consent manager.

Data Set Functionality of the Consent Manager Component

Data sets are registered at a consent manager with a unique ID in the same way they are registered at a data provider. Further, for the registration of a data set, it is required to provide not only the data set, but also a list of data owners, and a data category. The data category is used to match the permissions and obligations of each data owner that is part of the data set. At the consent manager, a data set is registered with a reference to the data provider.

POST	<code>/data-providerId/data-sets</code>	Register a data set
GET	<code>/data-providerId/data-sets</code>	List data sets
GET	<code>/data-owners/{dataOwnerId}/data-sets</code>	List data owner data sets

Figure 7.7: REST Endpoint for Data Set Related Services at the Consent Manager

Data Owner Functionality of the Consent Manager Component

The consent manager component comes with a set of endpoints that are used for the registration and listing of data owners. A data owner will not need a direct reference to a data provider, because at a consent manager a data owner can be linked to data sets of different data providers.

POST	<code>/data-owners</code>	Register a data owner
GET	<code>/data-owners</code>	List data owners

Figure 7.8: REST Endpoints for Data Owner Related Services at the Consent Manager

Figure 7.8 shows the REST endpoints for data owner-related services at the consent manager. The first API endpoint `/data-owners` helps to register a data owner at a consent manager. The second API endpoint `/data-owners` lists all data owners that are registered at a consent manager.

Permissions Functionality of the Consent Manager Component

The endpoints related to permissions are used for the registration and listing of data permissions. Data permissions are registered with a data category. The data category is used to link data permissions to a data set. Data permissions are directly linked to a data owner, this means that a data owner can have multiple data permissions for different data categories. Also, for a single data category, a data owner can create multiple permissions.

POST	<code>/data-owners/{dataOwnerId}/data-permissions</code>	Register a data permission
GET	<code>/data-owners/{dataOwnerId}/data-permissions/{dataCategory}</code>	List all data permissions for a data category

Figure 7.9: REST Endpoints for Data-Permission Related Services at the Consent Manager

Figure 7.9 shows the REST endpoints for data permission-related services at the consent manager. The first API endpoint `/data-owners/dataOwnerId/data-permissions` helps to register a data permission at a consent manager. The second API endpoint `/data-owners/dataOwnerId/data-permissions` lists all data permissions that are registered at a consent manager.

The registration of a data permission is done with the request body that can be seen in Figure 7.10.

```
{
  "attribute_id": "string",
  "attribute_constraint": "string",
  "data_category": "string"
}
```

Figure 7.10: Permission Registration Request Body

Obligations Functionality of the Consent Manager Component

The data obligations endpoints are almost identical to those of the data permissions. The primary reason for this is that permissions and obligations have the same attributes.

The endpoints are used for registration of data obligations and listing them. A data obligation is registered with a data category. The data category is used to link a data obligation to a data set. A data obligation is directly linked to a data owner, this means that a data owner can have multiple obligations

for different data categories. Also, for a single data category, a data owner can create multiple data obligations.

POST	/data-owners/{dataOwnerId}/data-obligations	Register a data permission
GET	/data-owners/{dataOwnerId}/data-obligations/{dataCategory}	List all data obligations for a data category

Figure 7.11: REST Endpoints for Obligation-Related Services at the Consent Manager

Figure 7.9 shows the REST endpoints for data obligation-related services at the consent manager. The first API endpoint `/data-owners/{dataOwnerId}/data-obligations` helps to register a data obligation at a consent manager. The second API endpoint `/data-owners/{dataOwnerId}/data-obligations/{dataCategory}` lists all data obligations for a data category that are registered at a consent manager.

The registration of a data obligation is done with the request body that can be seen in Figure 7.10.

```
{
  "attribute_id": "string",
  "attribute_constraint": "string",
  "data_category": "string"
}
```

Figure 7.12: Obligation Registration Request Body

Policy Functionality of the Consent Manager Component

Policy request is done by the data consumer. The data consumer specifies the data set it requests for usage. In this request, a data consumer can optionally provide a set of permissions and obligations to state its intentions with the data. This request object can be seen in Figure 7.14. Requesting custom permissions and/or obligations are called custom policy requests.

POST	/policy/{dataProviderId}/data-sets/{dataSetId}	Create policy request
GET	/policy/{policyId}	Get the status of a policy request
PUT	/policy/{policyId}	Accept or reject custom policy request

Figure 7.13: REST Endpoints for Policy-Related Services at the Consent Manager

Figure 7.13 shows the REST endpoints for policy-related services at the consent manager. The first API endpoint `/policy/{dataProviderId}/data-sets/{dataSetId}` creates a policy request at a consent manager. The second API endpoint `/policy/{policyId}` can be used to get the status of a policy request. The third API endpoint `/policy/{policyId}` can be used to accept or reject a custom policy request.

```
{
  "data_owner_id": "b5908551-b047-4256-934d-27de7e2a023b",
  "permissions": [
    {
      "attribute_id": "string",
      "attribute_constraint": "string",
      "data_category": "string"
    }
  ],
  "obligations": [
    {
      "attribute_id": "string",
      "attribute_constraint": "string",
      "data_category": "string"
    }
  ]
}
```

Figure 7.14: Policy Request Post Object

In the case of a custom policy request, intent of usage permissions and obligations can be added to the policy request body. This functionality can be seen in figure 7.14. With a custom policy request, the provided intended permissions and obligations are compared to the default permissions and obligations of each data owner. If there is a conflict, the data owner will be requested to give consent for the data usage request. The data owner will be notified, and can then accept or reject the request.

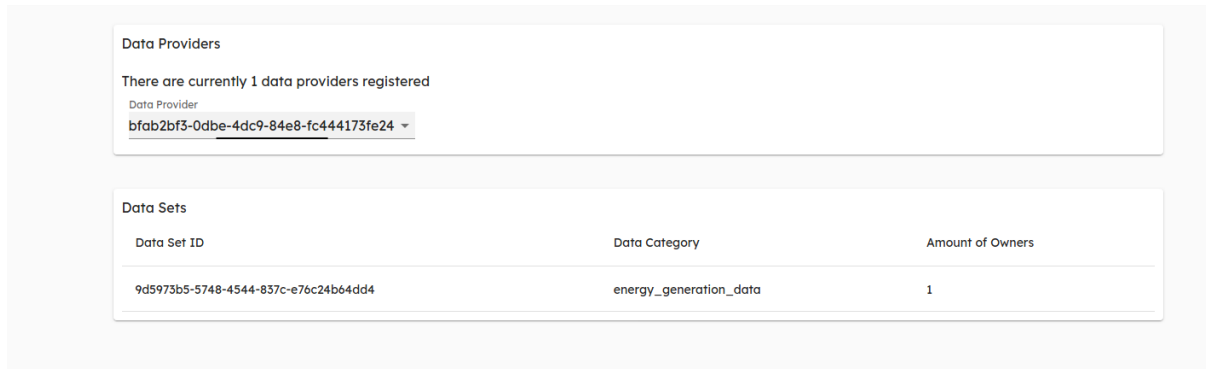
```
{
  "status": "created",
  "policyCatalogue": "a94a07c1-8010-417b-8afc-d26a9f791dc9"
}
```

Figure 7.15: Policy Request Response Object

After the policy request, the data consumer receives a response with the status of the policy request and the corresponding policy catalogue location where the policies will be stored, if they are approved. This response object can be seen in Figure 7.15.

Dashboard Implementation of the Consent Management Component

The dashboard introduced for the consent manager provides two views.



The screenshot shows a dashboard with two main sections. The top section, titled 'Data Providers', states 'There are currently 1 data providers registered' and shows a dropdown menu for 'Data Provider' with the selected value 'bfab2bf3-0dbe-4dc9-84e8-fc444173fe24'. The bottom section, titled 'Data Sets', contains a table with three columns: 'Data Set ID', 'Data Category', and 'Amount of Owners'. The table has one row with the following data: '9d5973b5-5748-4544-857c-e76c24b64dd4', 'energy_generation_data', and '1'.

Data Providers		
There are currently 1 data providers registered		
Data Provider		
bfab2bf3-0dbe-4dc9-84e8-fc444173fe24 ▾		

Data Sets		
Data Set ID	Data Category	Amount of Owners
9d5973b5-5748-4544-857c-e76c24b64dd4	energy_generation_data	1

Figure 7.16: Dashboard of Listing of Data Sets of a Selected Data Provider

The first view focuses on data providers. In this view a data provider can be selected (see top component of Figure 7.16). When a data provider is selected, all its data sets will be shown that are registered at the consent manager (see bottom component of Figure 7.16).

Data Owner
Data Owner
3bc29a59-6475-4b3e-a025-f043d6b8058e ▾

Data Sets

Data Set ID	Data Category	Amount of Owners
9d5973b5-5748-4544-837c-e76c24b64dd4	energy_generation_data	1

Pending custom policy requests

Data Consumer ID	Data Set ID
<input checked="" type="checkbox"/> 84e51bbf-cdc5-46fb-8ef9-b4895ae44brr	59df3462-3e76-4056-ac35-ad340c44a9c3

ACCEPT REJECT

Requested permissions viewer
Commercial usage: True

Requested obligations view
Delete after a month: True

Define Global Permissions
Data Category
energy_usage_data ▾

☐ Allow research usage

☐ Allow commercial usage

UPDATE

Define Global Obligations
Data Category
energy_generation_data ▾

☐ Delete after a week

☐ Delete after a month

UPDATE

Figure 7.17: Dashboard of Defining of Permissions and Obligations of Data Owners

The second view focuses on defining of permissions and obligations from data owners. A data owner can link these permissions and obligations to a data category (see second component of Figure 7.17). This allows the consent manager to form policies for data sets based on the permissions and obligations belonging to a specific data category.

The second view also provides an overview of pending custom policy requests from data consumers (see the third component of Figure 7.17). Here a data owner can accept or reject the policy request based on the requested permission and obligations.

Defining permissions and obligations is done by switches that can be seen in the two bottom components of Figure 7.17.

7.2.3 Policy Catalogue Component

The policy catalogue component manages the storage of policies formed by consent managers. Policies stored at the policy catalogue can be deleted by the consent manager. This enables a consent manager to delete a set of policies when permissions or obligations change for a specific data set.

Policy Functionality of the Policy Catalogue Component

POST	<code>/policies/data-provider/{dataProviderId}/data-set/{dataSetId}/data-consumer/{dataConsumerId}</code>	Create data policy
GET	<code>/policies/{policyId}</code>	Retrieve policies
DELETE	<code>/policies/{dataSetId}</code>	Delete all policies belonging to a specific data set

Figure 7.18: REST Endpoints for Policy Related Services at the Policy Catalogue

Figure 7.18 shows the REST endpoints for policy-related services of the policy catalogue. The first API endpoint `/policies/data-provider/dataProviderId/data-set/dataSetId` creates a data policy at the policy catalogue. The second API endpoint `/policies/policyId` can be used to retrieve policies from the policy catalogue. The third API endpoint `/policies/dataSetId` can be used to delete all policies belonging to a specific data set.

Dashboard Implementation of the Policy Catalogue Component

The dashboard view of the policy catalogue component provides an overview of all the stored policies. This is done by tables that sort on data providers, data sets and data consumers.

Stored policies sorted on data sets	
Data Set	Amount of policies
84e51bbf-cdc5-46fb-8ef9-b4895ae44bff	1

Stored policies sorted on data consumer	
Data Consumer	Amount of policies
9272bbdf-89f5-431c-967c-bc3668a338d5	1

Stored policies sorted on data owner	
Data Owner	Amount of policies
b31da16a-4052-4fc1-9ca1-6236103d12b9	1

Figure 7.19: Dashboard Overview of the Policy Catalogue

As can be seen Figure 7.19 a set of tables is used to provide insights on the number of policies that are stored for each actor or resource.

7.2.4 Data Consumer Component

The data consumer component is not representative of a real data space use case. The functionality implemented in the prototype is primarily used for showing the functionality of consent management. Therefore, the data consumer component of this prototype will only have data set and policies functionalities.

Data Set Functionality of the Data Consumer Component

The data consumer component implements the functionality for listing data sets and deleting data sets. With the deletion of a data set, the corresponding policy will also be deleted from the policy catalogue. This means that the listing endpoint will only list the data sets that the data consumer has a valid policy for.

DELETE	/data-sets/{dataSetId}	Delete access to a specific data set
GET	/data-sets	list all data sets the data consumer has access to

Figure 7.20: REST Endpoints for Data Set Related Services at the Data Consumer

Figure 7.20 shows the REST endpoints for data set-related services of the data consumer. The first API endpoint `/data-sets/{dataSetId}` deletes the access to a specific data set. The second API endpoint `/data-sets` can be used list all data sets the data consumer has access to.

Data Policy Functionality of the Data Consumer Component

The data consumer component implements a set of endpoints for requesting and retrieving of data policies. The data policies can then be used to retrieve the data sets at the data provider.

GET	/data-provider/{dataProviderId}/data-set/{dataSetId}	Request a policy for a data set
GET	/policies/{dataSetId}/	Retrieve a policy that the data consumer has obtained

Figure 7.21: REST Endpoints for Data Policy Related Services at the Data Consumer

Figure 7.21 shows the REST endpoints for data policy-related services of the data consumer. The first API endpoint `/data-provider/{dataProviderId}/data-set/{dataSetId}` requests a policy for a data set. The second API endpoint `/policies/{dataSetId}/` retrieves a policy that the data consumer has obtained.

Dashboard Implementation of the Data Consumer Component

The dashboard implementation for the data consumer component provides an overview of all the available data sets for different data providers (see the first component of Figure 7.22).

Available Data Providers

There is currently 1 data provider available

Data Provider
bfab2bf3-0dbe-4dc9-84e8-fc444173fe24 ▾

Available Data Sets

Data Set ID	Data Category	Amount of Owners
<input checked="" type="checkbox"/> 9d5973b5-5748-4544-837c-e76c24b64dd4	energy_generation_data	1

Default Permissions

Commercial usage: True

Research usage: True

Default Obligations

Delete after week: True

Delete after a month: false

Custom policy request

Define custom permission

☐ Allow research usage

☐ Allow commercial usage

Define custom obligation

☐ Delete after a week

☐ Delete after a month

REQUEST POLICY REQUEST CUSTOM POLICY

Figure 7.22: Dashboard of Data Set Related Services at the Data Consumer Component

From this overview a data set could be selected for a policy request (see second component of Figure 7.22). As can be seen in Figure 7.22, the default permissions and obligations are shown for a selected data set. In this view, also custom permissions and obligations can be requested (see the two bottom components of Figure 7.22).

The dashboard is divided into four main sections:

- Data Sets:** A table with columns 'Data Set ID' and 'Obtained at'. It contains one entry with ID 'b31da16a-4052-4fc1-9ca1-6236103d12b9' and date '03-10-2020'.
- Permissions of data set b31da16a-4052-4fc1-9ca1-6236103d12b9:** Two toggle switches, both turned on: 'Allow research usage' and 'Allow commercial usage'.
- Obligations of data set b31da16a-4052-4fc1-9ca1-6236103d12b9:** Two toggle switches. 'Delete after a week' is turned on, and 'Delete after a month' is turned off.
- Raw policy:** A JSON tree view showing the policy structure. The root has three items: '@context', '@uid', and 'permission'. The 'permission' item has a 'target', 'assignee', 'action', and a 'constraint' with an 'operator' and 'dateTime'.

Figure 7.23: Dashboard of Obtained Data Set Related Services at the Data Consumer

The data consumer dashboard also provides a view of obtained policies and data sets. This can be seen in Figure 7.23. The raw format of the policy can be viewed in the bottom component of this view. In this prototype, the research made use of ODRL [27] for defining the policies.

7.2.5 Policy Broker Component

The PB component takes the responsibility of matching the intent of usage permissions and obligations with data sets that are registered at a CM. It does this by comparing the intent of usage permissions and obligations to that of the permissions and obligations of its registered consent managers.

Registration Functionality

The registration of CMs is done by providing the unique ID of the CM to the PB. The CM reference will then be stored at the PB. The PB can then in a selection request query the CM for its permissions and obligations corresponding to a particular data category.

A green button labeled 'POST' followed by the text '/register Register as a consent manager at the policy broker'.

Figure 7.24: REST Endpoints for Registration Related Services

Selection functionality

The selection of CMs is done by providing a data category and a set of intended permissions and obligations. The registered CMs will then be queried for their permissions and obligations for the given

data category. A matching mechanism is then activated by the PB to select the best matching permissions and obligations. From this selection, the corresponding CM's are returned that provide the data sets belonging to the permissions and obligations.

POST **/selection** Get a selection of consent managers that match the data category and intent of usage permissions and obligations

Figure 7.25: REST Endpoints for Selection Related Services

A selection request is initiated with a selection request body. The selection request body that can be seen in figure 7.26 consists out of a data category and a set of intended permissions and obligations.

```
{
  "dataCategory": "energy_usage",
  "permissions": [
    {
      "attribute_id": "string",
      "attribute_constraint": "string"
    }
  ],
  "obligations": [
    {
      "attribute_id": "string",
      "attribute_constraint": "string"
    }
  ]
}
```

Figure 7.26: Selection Request Body

Dashboard Implementation of the Policy Broker Component

The dashboard of the policy broker is made for the viewpoint of a data consumer as can be seen in figure 7.27. The data consumer can select a data category will define its intent of usage permissions and obligations. It can then click on request to get a list of matching consent managers with corresponding data sets.

The dashboard is divided into several sections. At the top, there is a 'Data Category' dropdown menu currently set to 'energy_usage_data'. Below this is a section for 'Intent of Usage Permissions' with two toggle switches: 'Allow research usage' (which is turned on) and 'Allow commercial usage' (which is turned off). The next section is 'Intent of Usage Obligations' with two toggle switches: 'Delete after a week' (turned off) and 'Delete after a month' (turned on). At the bottom, there is a blue 'REQUEST' button and a table titled 'Matching Consent Managers'.

Consent Manager ID	Data Set IDs
12793409-0ac5-4f95-a1d9-ba252e8b90da	752cba86-01b5-48b9-8158-70ce83f81a13, d79c7fc9-5bbe-4436-8e27-daa790ec7d6c
2be92329-3006-4cae-b85b-23287ca0d08c	9f5464b7-14da-4dce-8772-bc9879e8298d
46540c09-84ba-4ae4-baf4-093d4c147184	7e2e28cc-001a-4698-a158-3371556d8fea, e83e7d9d-26e3-4338-80af-4abbe99ce902

Figure 7.27: Dashboard of the Policy Broker

The data consumer can then use the consent manager referenced and data set IDs to request policies of consent.

7.2.6 Permissions and Obligations Mapping Mechanism

As defined by the requirements in 4.2.5 in order to form policies of the permissions and obligations a mapping mechanism must be in place. The mapping mechanism used in this prototype translates a set of allowed attributes directly to an ODRL policy. There are no libraries or frameworks used for this process. This mapping mechanism is primitive in its implementation. As shown in Figure 7.10 permissions attributes (as chosen for this prototype) can be defined in a wide variety of ways. However, the REST API of the consent manager only allows a specific set of *attribute_id*'s. For this prototype this set consisted out of: *delete_after_a_week*, *delete_after_a_month*, *commercial_usage* and *research_usage*. This is a very limited set of attributes and a real use case should implement a larger set of *attribute_id*'s. Also, it can be interpreted that a real use case should provide mechanisms and frameworks for specification permissions and obligations that do not limit data owners and data consumers similar to this prototype.

7.2.7 Context Attributes

As described in the ontology in [28], the context in which consent was given must be stored. However, in this research the attributes that define the context differ from the context implementation of [28]. The context attributes of this research are use case specific. For this prototype the context attributes were:

- **data provider id:** Reference to the data provider
- **data set id:** Reference to the data set
- **policy catalogue id:** Reference to the policy catalogue
- **creation date:** Date of creation of the policy
- **data consumer id:** Reference to the data consumer

Any change to these attributes will revoke the policy that is obtained for this context.

7.3 Applicability to the Use Case

As described in the introduction, this research provided with the use case a set of findings that made implementing consent management valuable. With this prototype an evaluation can be made for each use case finding summarized in section 1.4.2

- **Finding 1:** In the prototype the permissions and obligations are stored at the consent manager. This allows data owners to manage their permissions and obligations at a central repository.
- **Finding 2:** In the prototype, a mapping mechanism is implemented that combines the different permissions and obligations of a multitude of data owners and forms a single policy. This mapping mechanism is implemented by the consent manager.
- **Finding 3:** As stated in finding one the consent manager enables data providers to have access to the permissions and obligations of their data owners. It is however noted that there are no restrictions on the number of consent managers in data spaces. This means that the permissions and obligations of a certain data category can only be found at a specific consent manager. Data providers in this case should be guided by the data owners to right consent manager.
- **Finding 4:** In the prototype, a data consumer does not have trusted contact with that of the data provider or the data owner. This research assumes that a dynamic exchange of data is possible if the data consumer can be identified and obtains consent for the data set through a consent manager. This can also be seen in the prototype, wherein obtaining consent the identity of the data consumer becomes part of the context in which consent has been given.

Chapter 8

Discussion

In this chapter, the results of the proposed reference architecture and prototype are discussed.

8.1 Findings of the Research

Consent management can be applied to a network approach such as with data spaces.

In [28] a reference architecture is introduced that focuses on consent management in a single application infrastructure. A network approach required a translation of the requirements that formed the reference architecture in [28]. The reference architecture proposed in this research implemented all the requirements and is compatible with data spaces as defined in the terminology Chapter in Appendix A.

Context in consent management is different for a network approach then for a single application infrastructure.

Context objects contain use case specific attributes. For data spaces the context object will contain attributes related to data owner, data provider and consent manager identifications. These context attributes will differ from use cases that are focused on single application infrastructures. Real use cases need to make sure that agreements are made for the context attributes.

Realization of the requirements that define consent management required the research to introduce new components for data spaces.

In [28] components are introduced such as the user interaction handler, context handler, provenance manager and the consent manager to realize the proposed reference architecture. In this research additional custom components are introduced for the proposed reference architecture. However, these components differ from the ones proposed in [28], which meant that the reference architecture of [28] cannot be used for realization of consent management in data spaces. It is therefore of importance to agree on the definition of consent management and on the requirements that define consent management to make it applicable for different systems.

Abstraction of permissions and obligations to form policies require further research and development.

As stated before the definition of obligations and permissions is very limited in this prototype. Judging from figure 7.10 permissions can be interpreted in a wide variety of ways. However, the REST API of the consent manager only allows a specific set of *attribute_id*'s. For this prototype this set consisted out of: *delete_after_a_week*, *delete_after_a_month*, *commercial_usage* and *research_usage*. Data spaces that will implement consent management should provide a more robust way of defining the permissions and obligations. As this research will discuss in chapter 10, there is a need for a framework to define permissions and obligations that can be translated in policies.

8.2 Evaluation and Limitations of the Prototype

This section critically evaluates the implementation of the prototype and its limitations.

The implementation of the prototype, as stated before, should not be used in a real use case without some adjustments. In order to use this implementation in a real use case, this research advises adding authentication and authorization functionality. This prototype was meant to show the relations and functionalities of the different components, therefore, authentication and authorization was not a priority. However, the functionality that is related to realizing consent management can be used for real use cases.

From the prototype, it can be concluded that the functionality that realizes consent management does not intervene with the required functionality of data spaces. This applies to the context in which it is assumed that the used data space is similar to the definition in the terminology section in the Appendix A. The only requirement that the reference architecture has is the unique identification of actors and resources in the network. As an example, given the reference architecture of IDS [2], actors in the network have a digital identity through certificates and active monitoring (X.509 Certificate). Instead of providing the policies of consent together with the data (sticky policies), the IDS could adopt the reference architecture proposed in this research, if altering some of the components to make them work with the certificates. Also, implementing consent management requires a mechanism for mapping of obligations and permissions and the adopting of data category specification for data sets.

Further, this prototype does not show how such a data space network would behave when using multiple components. In Chapter ?? it is stated that multiple components can be used in a data space network. For this demo, a single instance of each component is used. However, a multitude of consent managers, policy catalogues or policy brokers do not alter processes in the network, nor implement functionality related to multiple instances of the same component. Therefore, the only actors that will benefit from a multitude of the same components are the data owners, data providers and data consumers. Because these actors are part of the base definition of data spaces this research did not simulate multiple reference architecture components.

Chapter 9

Conclusions

9.1 Answers to Research Questions

This chapter answers the research question "What are the requirements that define Consent Management?" and summarizes the answers to the Research Objective: "How can Consent Management be realized in Data Spaces?".

What are the Requirements that Define Consent Management?

In order to give an answer to this research question, a definition has to be given for consent management. In the context of data spaces, this research defines consent management as a system or a set processes to define, exchange and manage data usage permissions and obligations.

In this research a set of requirements were identified based on the ontology of consent management defined in [28], the regulations of GDPR [5] and the characteristics of data spaces. This led to a set of requirements that could be part of the following categories of consent management:

- Defining of permissions and obligations of consent.
- Registration of permissions and obligations and linking with a data set.
- Negotiation of permissions and obligations.
- Forming of policies of consent based of the permissions and obligations given.
- Registration and exchanging of policies of consent
- Revoking of policies of consent.

Each category comes with a set of requirements that formed the basis for consent management in data spaces. The research then evaluated the current data space with the given requirements. From this evaluation further steps were taken to identify components that realize the requirements..

How do Data Spaces Need to be Changed in Order to Realize Consent Management?

From the set of requirements and the evaluation of current data spaces this research came to the conclusion to introduce new components to implement consent management in data spaces. The proposed reference architecture for consent management is based on the requirements that were identified. The proposed reference architecture consists out of a set of components. The three proposed components to facilitate consent management in data spaces and their responsibilities are:

- **Consent Manager:**
 - Storage of permissions and obligations
 - Linking of permissions and obligations with data sets
 - Negotiation of permissions and obligations
 - Forming of policies of consent based on the permissions and obligations given.
- **Policy Catalogue:**
 - Registration and exchanging of policies of consent
 - Revoking of policies of consent.

- **Policy Broker:**
 - Indexing of consent managers
 - Indexing of permissions and obligations sorted on data categories
 - Matching of intended permissions and obligations with data sets

The components implement the set of requirements and therefore enable consent management in data spaces. From these introduced components it can be concluded that data spaces by the definition of A need to have additional components to implement consent management.

9.2 Research Limitations

This research was limited to be conducted within the time frame of six months and its resources available. Those factors are outlined in the sections below.

9.2.1 Proposed Components

The proposed components are a direct result of the interpretation of the consent management requirements. This research saw it as a necessity to introduce new components for data spaces. This was done after careful consideration of the current functionality that data spaces provide. However, a different interpretation of the requirements could lead to a different set of components or different conclusions. This could mean that future data spaces could implement consent management in a different manner or come up with new solutions that don't need consent management.

9.2.2 Architecture Validation

With the given time constraints and the lack of data this research could not validate the reference architecture in a real implementation. Therefore, the research could not validate how the reference architecture would improve consent management in data spaces or data spaces in general. It is however stated that implementing consent management in data spaces is a new concept judging from the current state of data spaces [2] [8] [12] [13]. Therefore proposing a reference architecture for data spaces could potentially improve the ecosystem of data spaces. This was also the driving factor for a design science research approach, in which new artifacts are introduced and evaluated with a given use case.

Chapter 10

Future Work

In this chapter different topics are addressed that can function as potential future work topics. Each topic was naturally formed from the problems and limitations that were encountered when conducting this research.

10.1 Consent Management in Different Data Spaces

Future work could focus on reaching an agreement on the definition of data spaces. At the moment, there is no universal definition of data spaces. For this research a definition was made in the terminology section. Here it was explicitly stated that the definition of a data spaces used in this research is very similar to the reference architecture of the IDS [2]. In the Results Chapter 7 this research made the statement that the reference architecture can be used in a data space that is inline with the definition. However, because there is no agreement made on the definition of a data space, the proposed reference architecture could not be suitable for any other data space implementation.

10.2 Permissions and Obligations Definitions

Future work could focus on creating a framework for the definition of permissions and obligations in data spaces. As stated in the Results Chapter 7, the definitions of permissions and obligations for the prototype were implemented in a straight forward approach. There was no framework that could have functioned as a basis. Therefore, if different data spaces will implement consent management, all of them could benefit from a universal set of agreements for the definition of permissions and obligations. These can potentially be addressed in a joint effort to create a framework that lets data space actors define the permissions and obligations according to the agreements made.

10.3 Permissions and Obligations Mapping

Future work could focus on defining agreements how certain permissions and obligations will be mapped to policies. Similar to the remarks of defining permissions and obligations, there are currently no agreements made on how permissions and obligations can be mapped to a specific REL. This means that translations of permissions and obligations to REL's will differ based on the used data space. This makes it difficult to guarantee a standardization of the policies. Just as with the defining of permissions and obligations this mapping mechanism can be defined in a joint effort of different organization that will result in a standardized framework.

10.4 Obligations and Permissions Legal Validity

Future work could focus on validating permissions and obligations for legal validity. Laying down the frameworks for defining permissions and obligations and eventually mapping them to REL's as a policy of consent does not mean that the consent will have legal validity. Legal validity of permissions and obligations is important for data owners in their decision to make use of a data space. Thus, this is a

rather non-technical topic that could have a significant influence on adaption of consent management in data spaces.

10.5 Policies Legal Validity

Future research could look into differences between policies formed for single application infrastructure and policies formed for networks. Just as with permissions and obligations, formed policies also need to have legal validity. It can be noted that policies defined in REL's such as XACML [26] or ODRL [27] already have legal validity. However, this does not mean that this will be the case in data spaces. Data spaces must make use of data usage control, that will monitor the actions of actors in the network. In contrast to a closed single application infrastructure, a network approach, such as with data spaces, requires remote monitoring. This could lead to dispute problems where the monitoring will have direct influence on violating the permissions and obligations of the data owner. Further, the monitoring and data usage control must be evaluated in-order to make a claim for legal validity of the policies.

10.6 Context Specification

Future work should evaluate which attributes are needed for certain implementations of consent management. As stated in the evaluation of the results and in the Discussion Chapter 8 the context attributes specified used in this research differ from the ones used in single application infrastructures such as in [28]. This would imply that context attributes are implementation specific and therefore require further research. This research came to the conclusion that the context attributes needed for a network approach are primarily references to the data provider, data owner, and data set.

Acknowledgements

I would like to express my deep and sincere gratitude to my research supervisors of the University of Amsterdam Dr. Paola Grosso and Msc. Lu Zhang and my research supervisors of TNO Dr. Harrie Bastiaansen and Msc. Maarten Kollenstart.

Bibliography

- [1] N. S. Liezenberg C. Lycklama D., *Everything transaction*, 2019.
- [2] B. Otto, S. Steinbuß, A. Teuscher, and S. Lohmann, *International data spaces: Reference architecture model version 3*, <https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf>.
- [3] S. Dalmolen, H. Bastiaansen, M. Kollenstart, and M. Punter, “Infrastructural sovereignty over agreement and transaction data (‘metadata’) in an open network-model for multilateral sharing of sensitive data”, Dec. 2019.
- [4] Cousiel, *A european strategy for data*, Sep. 2020. [Online]. Available: <https://ec.europa.eu/digital-single-market/en/policies/building-european-data-economy>.
- [5] *2018 reform of eu data protection rules*, https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf, European Commission, 2018 (accessed April 11, 2020).
- [6] Gartner_{Inc}, *Definition of consent management - gartner information technology glossary*. [Online]. Available: <https://www.gartner.com/en/information-technology/glossary/consent-management>.
- [7] X. Maroñas, E. Rodriguez, and J. Delgado, “An architecture for the interoperability between rights expression languages based on xacml”, Oct. 2020.
- [8] S. Steinbuss, A. Eitel, C. K. Christian Jung, F. Bruckner, G. Brost, P. Birnstill, R. Nagel, and S. Bader, *Usage control in the international data spaces*. Anna-Louisa-Karsch-Str. 2 10178 Berlin Germany: International Data Spaces Association, Nov. 2019, vol. 2.
- [9] S. Pearson and M. Casassa-Mont, “Sticky policies: An approach for managing privacy across multiple parties”, *Computer*, vol. 44, no. 9, pp. 60–68, 2011.
- [10] S. Dalmolen, H. Bastiaansen, E. Somers, S. Djafari, M. Kollenstart, and M. Punter, “Maintaining control over sensitive data in the physical internet: Towards an open, service oriented, network-model for infrastructural data sovereignty”, Jul. 2019.
- [11] “Vivet, voorstellen om de informatievoorziening energietransitie te verbeteren”, Feb. 2019.
- [12] M. Jarke, B. Otto, and S. Ram, “Data Sovereignty and Data Space Ecosystems”, *Business & Information Systems Engineering: The International Journal of WIRTSCHAFTSINFORMATIK*, vol. 61, no. 5, pp. 549–550, Oct. 2019. DOI: 10.1007/s12599-019-00614-. [Online]. Available: https://ideas.repec.org/a/spr/binfse/v61y2019i5d10.1007_s12599-019-00614-2.html.
- [13] M. M. Sebastian R. Bader, “Towards enforceable usage policies for industry4.0”, *Proceedings of the 1st Workshop on Large Scale RDF Analytics*, 2019.
- [14] J. Gustafsson, *Single case studies vs. multiple case studies: A comparative study*, 2017.
- [15] R. K. Yin, *Case study research and applications: Design and methods*. Sage publications, 2017.
- [16] J. Gerring, “What is a case study and what is it good for?”, *American political science review*, vol. 98, no. 2, pp. 341–354, 2004.
- [17] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research”, *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [18] S. A. Carlsson, S. Henningsson, S. Hrastinski, and C. Keller, “Socio-technical is design science research: Developing design theory for is integration management”, *Information Systems and e-Business Management*, vol. 9, no. 1, pp. 109–131, 2011.

- [19] S. T. March and G. F. Smith, “Design and natural science research on information technology”, *Decision support systems*, vol. 15, no. 4, pp. 251–266, 1995.
- [20] S. Gregor and A. R. Hevner, “Positioning and presenting design science research for maximum impact”, *MIS quarterly*, pp. 337–355, 2013.
- [21] A. Gunasekaran, T. Papadopoulos, R. Dubey, S. F. Wamba, S. J. Childe, B. Hazen, and S. Akter, “Big data and predictive analytics for supply chain and organizational performance”, *Journal of Business Research*, vol. 70, pp. 308–317, 2017, ISSN: 0148-2963. DOI: <https://doi.org/10.1016/j.jbusres.2016.08.004>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S014829631630491X>.
- [22] C. Marinagi, P. Trivellas, and P. Reklitis, “Information quality and supply chain performance: The mediating role of information sharing”, *Procedia - Social and Behavioral Sciences*, vol. 175, pp. 473–479, 2015, Proceedings of the 3rd International Conference on Strategic Innovative Marketing (IC-SIM 2014), ISSN: 1877-0428. DOI: <https://doi.org/10.1016/j.sbspro.2015.01.1225>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877042815012859>.
- [23] C. Clifton, M. Kantarcioglu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, and D. Suciu, “Privacy-preserving data integration and sharing”, Jan. 2004, pp. 19–26. DOI: 10.1145/1008694.1008698.
- [24] R. Yavatkar, D. Pendarakis, and R. Guerin, “A framework for policy-based admission control”, Jan. 2000.
- [25] M. Sloman, “Policy driven management for distributed systems”, *Journal of Network and Systems Management*, vol. 2, Aug. 1996. DOI: 10.1007/BF02283186.
- [26] D. Ferraiolo, R. Chandramouli, R. Kuhn, and V. Hu, “Extensible access control markup language (xacml) and next generation access control (ngac)”, in *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, ser. ABAC ’16, New Orleans, Louisiana, USA: Association for Computing Machinery, 2016, pp. 13–24, ISBN: 9781450340793. DOI: 10.1145/2875491.2875496. [Online]. Available: <https://doi.org/10.1145/2875491.2875496>.
- [27] R. Ianella, “Open digital rights language (odrl)”, *Open Content Licensing: Cultivating the Creative Commons*, 2007.
- [28] K. Fatema, E. Hadziselimovic, H. J. Pandit, C. Debruyne, D. Lewis, and D. O’Sullivan, “Compliance through informed consent: Semantic based consent permission and data management model.”, in *PrivOn@ ISWC*, 2017.
- [29] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing?”, *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907–928, 1995.
- [30] X. Maroñas, E. Rodriguez, and J. Delgado, “An architecture for the interoperability between rights expression languages based on xacml”, Aug. 2020.
- [31] D. Eastlake, J. Reagle, D. Solo, F. Hirsch, and T. Roessler, “Xml-signature syntax and processing”, *W3C recommendation*, vol. 12, 2002.
- [32] A. Anderson, A. Nadalin, B. Parducci, D. Engovatov, H. Lockhart, M. Kudo, P. Humenn, S. Godik, S. Anderson, S. Crocker, *et al.*, “Extensible access control markup language (xacml) version 1.0”, *OASIS*, 2003.
- [33] J. G. Walls, G. R. Widmeyer, and O. A. El Sawy, “Building an information system design theory for vigilant eis”, *Information systems research*, vol. 3, no. 1, pp. 36–59, 1992.
- [34] J. Eekels and N. F. Roozenburg, “A methodological comparison of the structures of scientific research and engineering design: Their similarities and differences”, *Design studies*, vol. 12, no. 4, pp. 197–203, 1991.
- [35] *Kubernetes, production-grade container orchestration*. [Online]. Available: <https://kubernetes.io/>.
- [36] *Flask web framework*. [Online]. Available: <https://flask.palletsprojects.com/en/1.1.x/>.
- [37] *Postgresql: The world’s most advanced open source relational database*. [Online]. Available: <https://www.postgresql.org/>.
- [38] *Postgres operator*. [Online]. Available: <https://github.com/zalando/postgres-operator>.
- [39] Ishare, *Ishare*, <https://www.ishareworks.org/>, [Online; accessed 24-June-2020], 2002.

Appendix A

Terminology

This chapter presents several terms that are used throughout the research and are essential to the understanding of the reader. Some of these terms are based on GDPR, whereas others refer to technical standards or are explicitly defined for this report. In particular, the following terms are utilised:

A.1 Data Spaces

Data spaces are an abstraction in data management that aims to be a solution for the need for a trusted, controlled, and secure way to exchange data. Data spaces often focus on data integration, data management and privacy persevering data exchange. Commonly data spaces enable widespread integration and sharing of data by defining and implementing commonly shared technical components or services that each participant must use.

There are a variety of references architectures, technical implementations and commercial online services that function as data spaces, such as Ishare [39] a set of uniform agreements for identification, authentication and authorization for data sharing, or International Data Spaces (IDS) [2] a set of technical components for the implementation of a data space. All these entities provide a wide variety of functionalities related to data spaces. Therefore it is not really clear what is meant with the term data spaces. Also, from a literature standpoint, the term data space is vague.

For this research, a data space is defined as a peer-to-peer network of shared technical components that facilitate a trusted, controlled, and secure way to exchange data. The reference architecture of International Data Spaces [2] is the closest realization of this definition. The actors and components that this research shall use are closely tied to the actors and components defined by the definitions given in International Data Spaces.

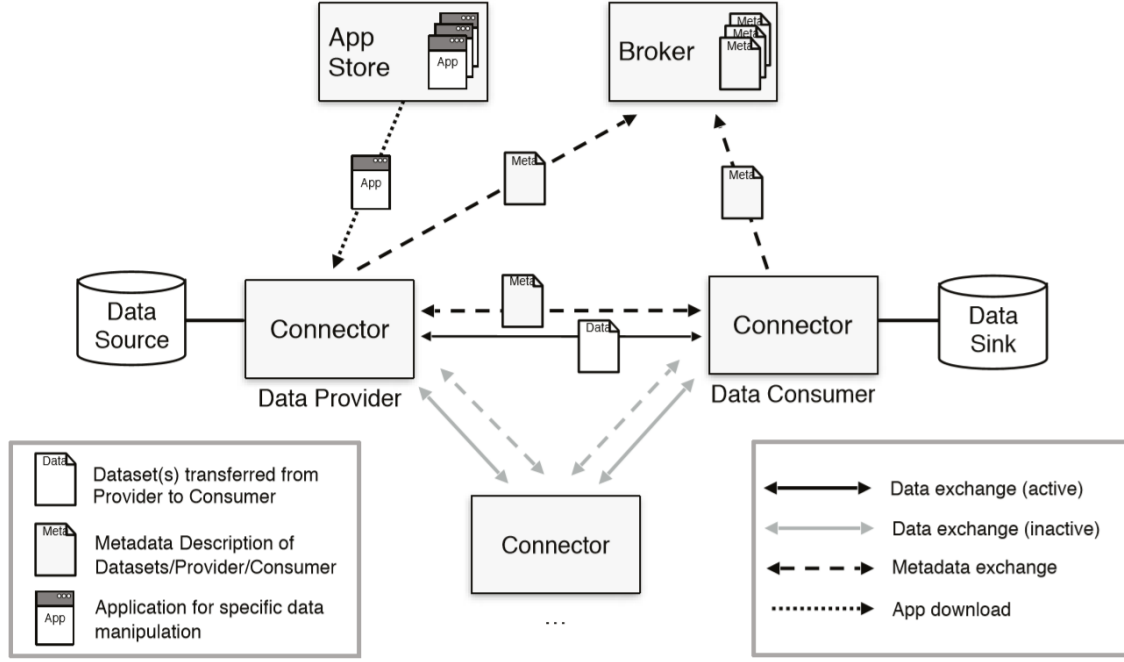


Figure A.1: Reference Architecture of International Data Spaces [2].

In Figure A.1 the base components and their interactions of the IDS reference architecture can be seen. The central component in this diagram is the connector. The connector is the technical component that connects the data provider with the data consumer. Connectors and their data objects are catalogued at a broker. This allows for searching of different data providers. Connectors can make use of app store applications that will add extra functionality to the data exchanges such as anonymization and pseudonymization. IDS makes use of access and usage policy enforcement. These policies are exchanged between connectors with the use of the sticky policy mechanism [9], where a data object is accompanied with the defined policies of the data owner. All these components and their interactions can provide the foundation for privacy-preserving data integration and exchange by supporting access and usage policy enforcement and enables widespread integration and sharing of data by defining and implementing commonly shared technical components

A.2 Consent Management

Consent management is the process of orchestrating who will have access to protected resources, for what purpose, and under what circumstances[6] . Often a system, process, or set of policies are used to implement mechanics of consent management. Consent management can enable the dynamic creation, management, and enforcement of consumer, organizational, and jurisdictional privacy policies [6].