Co-funded by
the European Union

EDDIE

EUROPEAN DISTRIBUTED
DATA INFRASTRUCTURE
FOR ENERGY

D4.1

Message-Based Communication Implemented

This work has been co-funded by the European Union's Horizon Innovation Actions under grant agreement No. 101069510

DOCUMENT INFORMATION

WP number and title	WP4 – EDDIE Interoperable Communication Layer
Deliverable number	D4.1
Version Number	1.3
Document Reference	Doi: 10.5281/zenodo.11638318
Lead Beneficiary	PON
Deliverable type	DEM – Demonstrator, pilot, prototype
Planned deliverable date	30/06/2024
Date of Issue	29/06/2024
Dissemination level	PU
Author(s)	FHO, PON
Contributor(s)	FHO, AIT, D4G, ENT, PON
Keywords	Interoperable Communication Layer, Region Connector, Message-based Communication

CONSORTIUM PARTNERS

The EDDIE Consortium consists of the following partners:

Participant No	Participant organisation name	Short Name	Country
1	University of Applied Sciences Upper Austria – Campus Hagenberg – Research and Development	**FHO**	**AT**
2	Copenhagen School of Energy Infrastructure, Department of Economics, Copenhagen Business School	**CBS**	**DK**
3	European University Institute	**EUI**	**IT**
4	University of Vienna, Faculty of Computer Science, Cooperative Systems Research Group	**VIE**	**AT**
5	Austrian Institute of Technology, Center for Digital Safety & Security, Competence Unit Cooperative Digital Technologies	**AIT**	**AT**
6	The Lisbon Council for Economic Competitiveness and Social Renewal asbl	**LIC**	**BE**
7	PONTON GmbH	**PON**	**DE**
8	Asociación de Empresas de Energía Eléctrica (aelec)	**AEL**	**ES**
9	DEDA – Public Gas Distribution Networks – Single Member S.A.	**DED**	**GR**
10	EDA Energiewirtschaftlicher Datenaustausch GmbH	**EDA**	**AT**
11	Südtiroler Energieverband	**SEV**	**IT**
12	FlexiDAO	**FLE**	**ES**
13	Digital4Grids	**D4G**	**FR**
14	EASEE Gas	**EAS**	**FR**
15	Entarc.eu	**ENT**	**AT**
16	ETA+ GmbH	**ETP**	**DE**

DOCUMENT HISTORY

Version	Date	Status	Author(s), Reviewer	Description
V0.1	20/02/2024	Draft	Marc Kurz	Document Template
V0.2	27/02/2024	Draft	Marc Kurz	Draft Outline
V0.3	08/03/2024	Draft	Marc Kurz	Table of Contents
V0.4	20/03/2024	Draft	Marc Kurz, Fabian Haas, Henner Carl	Input collected
V1.0	24/05/2024	Review	Marc Kurz	Draft sent to reviewers
V1.1	03/06/2024	Review	Stefan Kienler	Document reviewed
V1.2	14/06/2024	Pre-Final	Marc Kurz	Incorporated reviewer's comments
V1.3	18/06/2024	Final	Marc Kurz	Document finalized

AUTHOR AND REVIEWER ACKNOWLEDGMENTS

Author names	Partner
Marc Kurz	FHO
Henner Carl	PON
Fabian Haas	FHO
Shievam Kashyap	FHO
Stefan Grünberger	FHO

Reviewer names	Partner
Georg Hartner	ENT
Stefan Kienler	EDA

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Acronyms/ Abbreviations	Description
AIIDA	Administrative Interface for In-house Data Access
API	Application Programming Interface
BPMN	Business Process Model and Notation
CAS	Central Access Server
CCM	Customer Consent Management
CCMO	Customer Consent Management Online Process
CIM	Common Information Model
CMS	Central Market System
DAP	Data Access Provider
DEP	Data Exchange Platform
DSO	Distribution System Operator
EDA	Energiewirtschaftlicher Datenaustausch
EDSN	Energie Data Services Nederland
EMIF	Elhub Messaging Interface

EP	Eligible Party
HVD	Historical Validated Data
JSON	JavaScript Object Notation
MDA	Metered Data Administrator
MQTT	Message Queuing Telemetry Transport
OBIS	Object Identification System
PA	Permission Administrator
RC	Region Connector
REST	Representational State Transfer
SMGA	Smart Meter Gateway Administrator
TSO	Transmission System Operator
VHMD	Validated Historical Metering Data
XML	Extensible Markup Language

EXECUTIVE SUMMARY

Deliverable D4.1 of the EDDIE project, "Message-Based Communication Implemented," marks a significant milestone in developing an interoperable communication layer for European energy data exchange. This document, part of Work Package 4, addresses the varied methodologies across EU member states, ensuring interoperability, security, and reliability. The main objectives include developing a communication layer that bridges the gap between varying national implementations of EU-regulated processes, focusing on communication protocols and data formats. The aim is to create a uniform architecture for configuring the communication layer, enabling integration across different business processes and ensuring compliance with current and future standards.

The initial architecture outlined in the grant agreement serves as the foundation for the communication layer. The focus is on creating a standardized, secure, and reliable infrastructure for data exchange, incorporating state-of-the-art analysis and legal framework considerations to ensure compliance and functionality across different regions. The core contribution of this deliverable is the message-based communication, with a particular focus on the Austrian case, utilizing the AS4 messaging protocol for data exchange. Detailed processes for customer consent management and data transmission are illustrated through BPMN diagrams.

The region connector architecture serves as the gateway for data flow into the EDDIE Framework. Middleware systems like Ponton X/P messenger are used for AS4 communication in Austria, ensuring consistent and secure interactions despite regional differences in data-sharing infrastructure. Technological integration involves the incorporation of different communication methods, including REST API and streaming-based communication, to cover a broad spectrum of European solutions.

The EDDIE framework is designed for extensibility, supporting the addition of other messaging solutions. The D4.1 deliverable represents a critical step towards a unified, interoperable communication layer for European energy data exchange. By addressing diverse methodologies and aligning with stakeholder needs, the EDDIE project aims to enhance data exchange efficiency, security, and interoperability across the European energy landscape.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Purpose and Background of the Document

This document represents Deliverable 4.1 (D4.1) – "Message-based communication implemented". Classified as a "DEM – Demonstrator, pilot, prototype" type, it serves as a comprehensive report detailing a tangible aspect of the EDDIE project's progress. According to the grant agreement of EDDIE, Deliverable 4.1 encompasses the following description:

> *"The EDDIE Framework communication layer resulting from T4.1 design efforts, and the implementation of message-based communication resulting from T4.2."*

Work-package 4 ("EDDIE Interoperable Communication Layer") holds significant importance within the broader development scope of the EDDIE Framework. In essence, its primary objective revolves around addressing the diverse array of European energy data exchange methodologies employed by reference processes and national connectors. The focal point is ensuring interoperability and crafting a communication layer that aligns with stakeholder requirements across various data exchange scenarios. The overarching aim involves crafting and tailoring an interoperable infrastructure capable of bridging gaps stemming from differing national implementations of EU-regulated reference processes, particularly concerning communication protocols and data formats. The ultimate goal is to establish an interoperable European communication layer accessible to stakeholders both within and beyond the project. User support will be extended to facilitate seamless business integration activities, while the infrastructure will undergo iterative refinement and expansion to accommodate evolving stakeholder needs during the EDDIE implementation phase.

Although the primary emphasis of this deliverable, as outlined in the grant agreement, is on Tasks T4.1 ("Communication layer design") and T4.2 ("Message-based communication"), this document also provides insights into ongoing implementations across other tasks, aiming to present a comprehensive overview of the current status of the "Interoperable Communication Layer."

1.2 Scope and Intended Audience

The scope of Deliverable D4.1, "Message-Based Communication Implemented," encompasses the design and implementation of a communication layer within the EDDIE Framework, aimed at achieving interoperability across various European energy data exchange systems. It focuses on standardizing communication protocols and data formats to bridge differences between national implementations of EU-regulated processes. The intended audience includes stakeholders involved in the EDDIE project, such as energy providers, regulatory bodies, and technical partners, as well as broader industry participants interested in enhancing the efficiency, security, and interoperability of energy data exchanges across Europe.

1.3 Structure of the Document

The document is structured as follows:

Chapter 2 ("Communication Layer Design") covers the initial architecture as described in EDDIE's grant agreement and recaps the basic idea. The main goal of this chapter is to illustrate the process of designing an interoperable communication layer, incorporating the knowledge of the current state and the necessary technological developments. This is done by a comprehensive analysis of state-of-the-art of data sharing infrastructures in Europe and a discussion of the legal framework and the available solutions. This is followed by an illustration of the architectural design and development progress, where the development team processed the gathered knowledge towards architectural and technological workstreams. Chapter 2 is concluded by a general architecture following a layered approach.

Chapter 3 ("Message-Based Communication") specifically explains the message-based communication as core contribution of the current deliverable. Main part is the Austrian case as this is a typical message-based infrastructure using AS4 messages. The chapter concludes with an exemplary workflow that requests and transmits validated historical data.

Chapter 4 ("Outlook & Roadmap") gives an outlook to communication approaches incorporated within the interoperable communication layer. Results of the REST API-based communication and the streaming-based communication will be presented in the concluding deliverable "D4.2 – Three different approaches for the communication layer are implemented", which is due in month 30 of the project.

Chapter 5 ("Summary & Conclusion") closes the document by summarizing the main aspects.

2 Communication Layer Design

2.1 Introduction

Task 4.1. ("Communication Layer Design") involves standardizing certain aspects of the infrastructure to enable integration across business processes that vary across different member states. This standardization aims to create a cohesive collaboration environment, requiring a uniform architecture for configuring the communication layer. The objective is to achieve interoperability, high levels of security, and reliability for various use cases. Additionally, the data exchange layer must align with current and future standards and best-practices concerning software frameworks and processes. Ultimately, the goal is to establish a coherent, consistent, and configurable foundation for communication protocols primarily usable by "region connector" modules.

In the following section 2.2, the initial architecture and design as envisioned in the proposal and grant agreement is recapitulated. In section 2.3 state of the art is described with a focus on the gaps and necessary technological developments. Section 2.4 describes the development process for realizing the message-based communication and the general interoperable communication layer in general, depicting the steps and the general process within the project in the first 18 months. This chapter is then concluded with section 2.5, describing the final architecture and design for the interoperable communication layer with focus on message-based communication.

2.2 Initial Architecture

The initial EDDIE architectural schema – as defined and describe in the Grant agreement is the following as depicted in Figure 1. The "EDDIE Interoperable Communication Layer" embedded as central component in the EDDIE Framework intends to tackle the diversity of European energy data exchange methodologies to be used by the reference processes and national connectors. Generally, interoperability and the communication layer itself are designed to meet stakeholder requirements regarding the different types of data exchange required. Of course, functionality depends on the procedures needed to access the data families within scope, in the regions within scope. This means to develop and customize an interoperable infrastructure that allows to bridge the gap between varying national implementations of EU-regulated reference processes regarding communication protocols and data formats.

To allow for an integration across business processes that are implemented differently per Member State, a uniform architecture is needed to configure the communication layer by process participants in such a way that interoperability, a high level of security and reliability is reached for use cases. At the same time, the data exchange layer needs to be aligned with current and future standards and best-practices regarding the underlying software frameworks and processes. The goal is a coherent, consistent, and configurable functional basis for communication protocols, configurable and usable primarily by "region connector" modules.

The original EDDIE architectural schema, as outlined and detailed in the grant agreement, can be conceptualized through Figure 1. At its core lies the "EDDIE Interoperable Communication Layer," strategically positioned within the EDDIE Framework. This layer is included to address the wide array of European energy data exchange methodologies that the reference processes and national connectors are expected to utilize. Its primary objective is to ensure interoperability, essentially serving as a bridge between the diverse communication protocols and data formats employed by different European regions.

The overarching aim of this communication layer is to cater to stakeholder requirements pertaining to the various types of data exchange necessary. The functionality of this layer is contingent upon the specific procedures required to access the relevant data families within the designated regions. This necessitates the development and customization of an interoperable infrastructure capable of harmonizing the disparate national implementations of EU-regulated reference processes in terms of communication protocols and data formats.

Achieving seamless integration across business processes, which may be implemented differently across Member States, mandates the establishment of a uniform architecture. This architecture must empower process participants to configure the communication layer in a manner that ensures interoperability, while also upholding high levels of security and reliability across various use cases. Concurrently, the data exchange layer must remain aligned with prevailing and forthcoming standards and best practices concerning the underlying software frameworks and processes. The ultimate objective is to furnish a cohesive, consistent, and configurable foundation for communication protocols, primarily accessible and utilizable by "region connector" modules.

By encapsulating the complexity of European energy data exchange within a structured and adaptable framework, the EDDIE architecture endeavors to streamline cross-border interactions and foster greater synergy among stakeholders across the energy landscape. This comprehensive approach not only facilitates efficient data exchange but also lays the groundwork for future advancements in energy interoperability and collaboration on a pan-European scale.

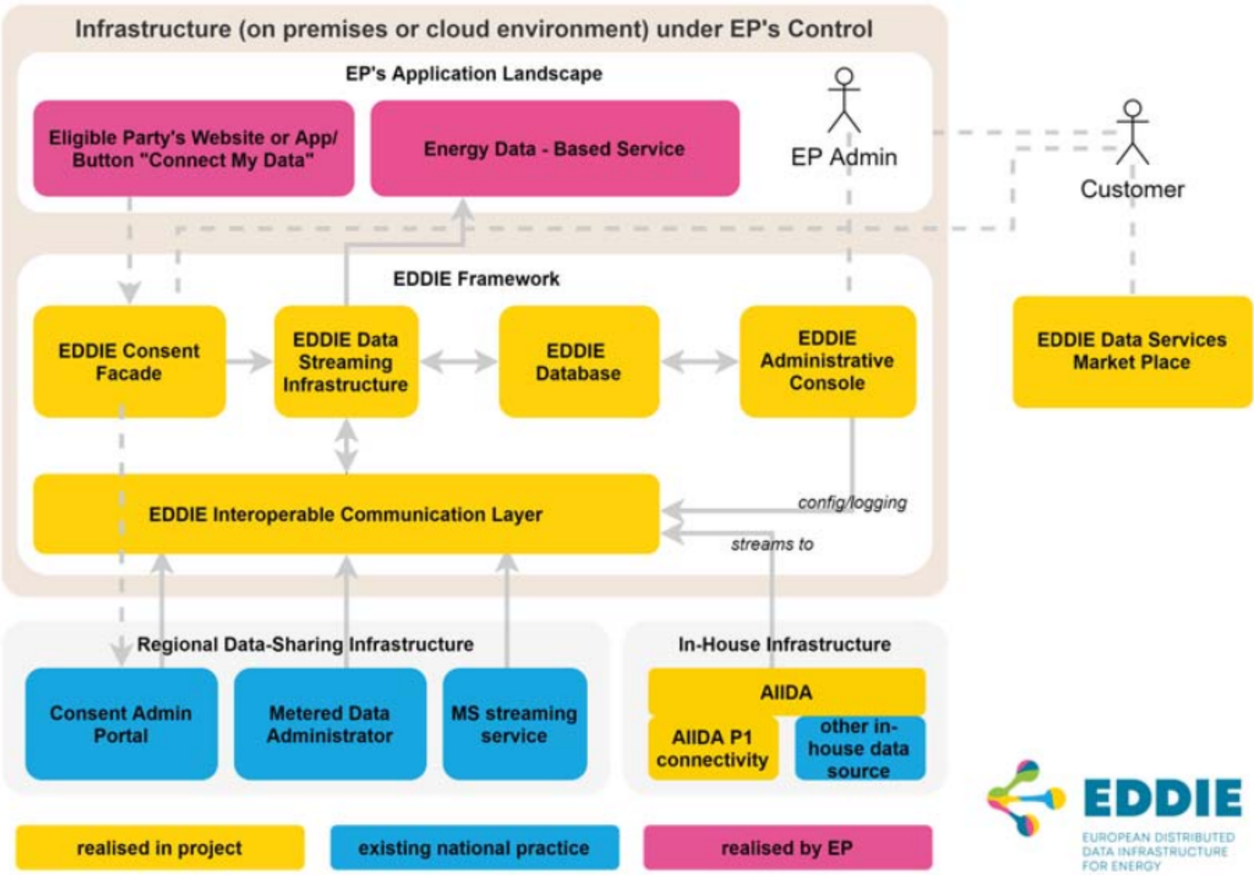


Figure 1: Initial EDDIE Architectural Schema

2.3 State of the Art

This section comprehensively explores the current landscape and historical context of energy data infrastructure and management, particularly focusing on the European Union but also drawing comparisons with the United States.

2.3.1 Legal Framework

The transition to sustainable energy infrastructures is deeply intertwined with the accessibility and transparency of energy data. However, many countries lack the necessary legal frameworks and standards to allow customers easy access to their energy data. This issue is especially pronounced in the findings of a 2022 survey by the University of Vienna, where international students struggled to retrieve their energy data from their home countries, with only 7 out of 42 attempts being successful [1]. This demonstrates the significant barriers customers face in accessing their energy data, which in turn impedes efforts to enhance energy efficiency and provide energy-related services on an international level.

United States

In the United States, there is no overarching federal regulation mandating the nationwide disclosure of energy data. Nevertheless, several states such as California, Columbia, and Illinois have enacted laws requiring metered data administrators to provide customers access to their energy information [2]. A notable initiative in this regard is the Green Button Initiative, introduced in 2012. Although voluntary, this initiative allows customers to download and share their energy usage information in a standardized format [3]. It has seen widespread adoption, with participation from over 50 utilities and electricity providers, thereby providing access to more than 60 million households. The success of the Green Button Initiative highlights the benefits of having a standardized format for energy data exchange across different suppliers and states.

European Union

The European Union has taken a proactive stance in ensuring the transparency and accessibility of energy data through the enactment of the Clean Energy Package in 2019 [4]. This comprehensive framework of laws and regulations aims to accelerate the shift towards a more sustainable energy ecosystem. The package includes directives that liberalize the European energy market and empower citizens to access and exchange their energy data easily [5]. Member states are required to develop transparent guidelines for managing and sharing energy-related data, including detailed metering and consumption data. The directive mandates that data must be made available to eligible parties in a fair and transparent manner, ensuring equal access to crucial information. Additionally, customers should not incur additional charges for accessing or sharing their data with eligible parties.

The directive also encourages the installation of smart metering systems across the EU, allowing customers to request the installation of such systems to enhance energy efficiency and participation in the energy market. The broader Data Act Regulation (EU) 2023/2854 further supports these rights by stipulating that data holders must provide access to customer data in a standardized machine-readable format, extending this right to eligible third parties as well [6].

2.3.2 State of the art in the EU

The state of energy data infrastructure in the European Union is highly heterogeneous due to the diverse approaches adopted by different member states. While there is no single state-of-the-art solution for energy data access, the EU has provided a reference model through the Implementing Regulation (EU) 2023/1162 [7]. This regulation acts as a guideline for member states to implement data access rights for energy data and should in some shape or form be implemented by the member states.

Data Exchange Models

Energy data exchange models in the EU can be broadly categorized into centralized, decentralized, and hybrid exchanges [8]:

1) Centralized Data Exchanges
   In centralized exchanges, a single entity acts as both Metered Data Administrator (MDA) and Permission Administrator (PA), meaning it collects, stores, and distributes data, ensuring that all data is consistent and up-to-date. This entity is very often a central authority like a Transmission System Operator (TSO) or a Distribution System Operator (DSO). This model simplifies the process for third parties to access data as they only need to connect to one system. However, it can present a single point of failure, making it vulnerable to outages or attacks.
2) Decentralized Data Exchanges
   In decentralized exchanges, data management responsibilities are distributed among multiple entities. Each DSO acts as the MDA for their respective areas, making third-party access more complex as connections with multiple DSOs are required. This model avoids a single point of failure but can lead to inconsistencies and increased complexity in data access.
3) Hybrid Data Exchanges
   Hybrid exchanges combine aspects of both centralized and decentralized models. Typically, data is collected and stored in a decentralized manner but distributed centrally. This requires high standardization and cooperation among DSOs. Hybrid models aim to balance the benefits of centralized data access with the resilience of decentralized systems.

Despite these classifications, what matters most to third parties is whether a country operates a Data Exchange Platform (DEP). A DEP facilitates the exchange of energy data between market participants, acting as a single access point for third parties. Countries without a DEP usually have a decentralized model, posing higher entry barriers for third parties. Figure 2 shows the state of centralization of data exchange models in Europe in 2016 as illustrated in [9]. As it can be seen, the majority of countries in Europe have a decentralized data exchange in place, but there is a trend towards implementing a DEP in the future.

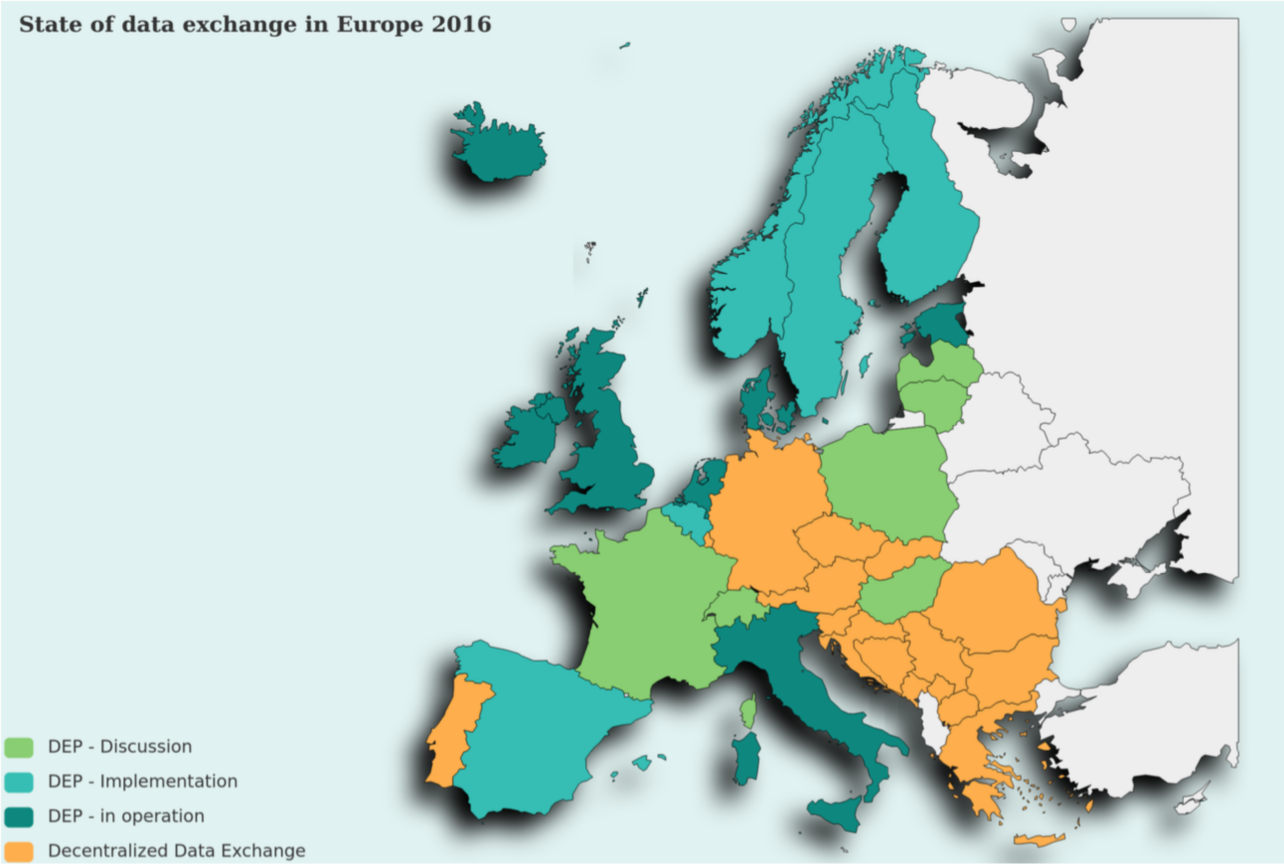


Figure 2: State of centralization of data exchange models in Europe in 2016 [9].

Reference Model

The EU's Implementing Regulation (EU) 2023/1162 [7] provides a reference model divided into six procedures, covering both near real-time data access and access to validated historical metering data (VHMD). Member states are required to map their national practices to this reference model and implement it by January 2025 [7]. The key procedures include:

1) Procedure 1: Access to validated historical metering and consumption data by the customer (process shown in Figure 3)
    - Customers identify their Data Access Provider (DAP), authenticate themselves, and request VHMD.
    - The DAP verifies the customer's identity and facilitates data transfer from the MDA to the customer.

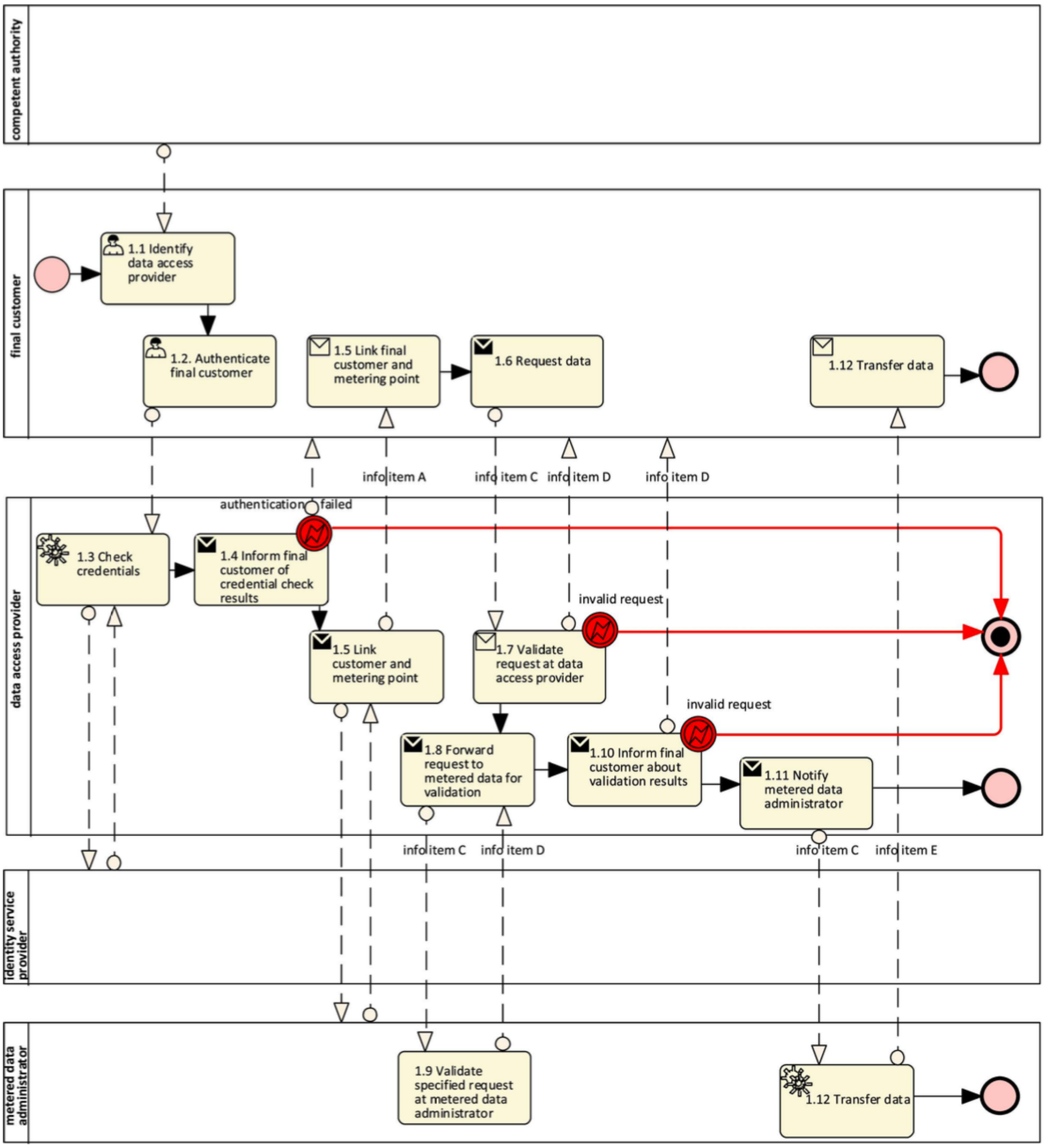


Figure 3: Accessing VHMD by the customer [7].

2) Procedure 2: Access to VHMD and consumption data by an eligible party (process shown in Figure 4)
   - Eligible parties must obtain customer permission through a Permission Administrator (PA) before accessing VHMD.
   - The PA manages permissions and ensures the transfer of data from the MDA to the eligible party.

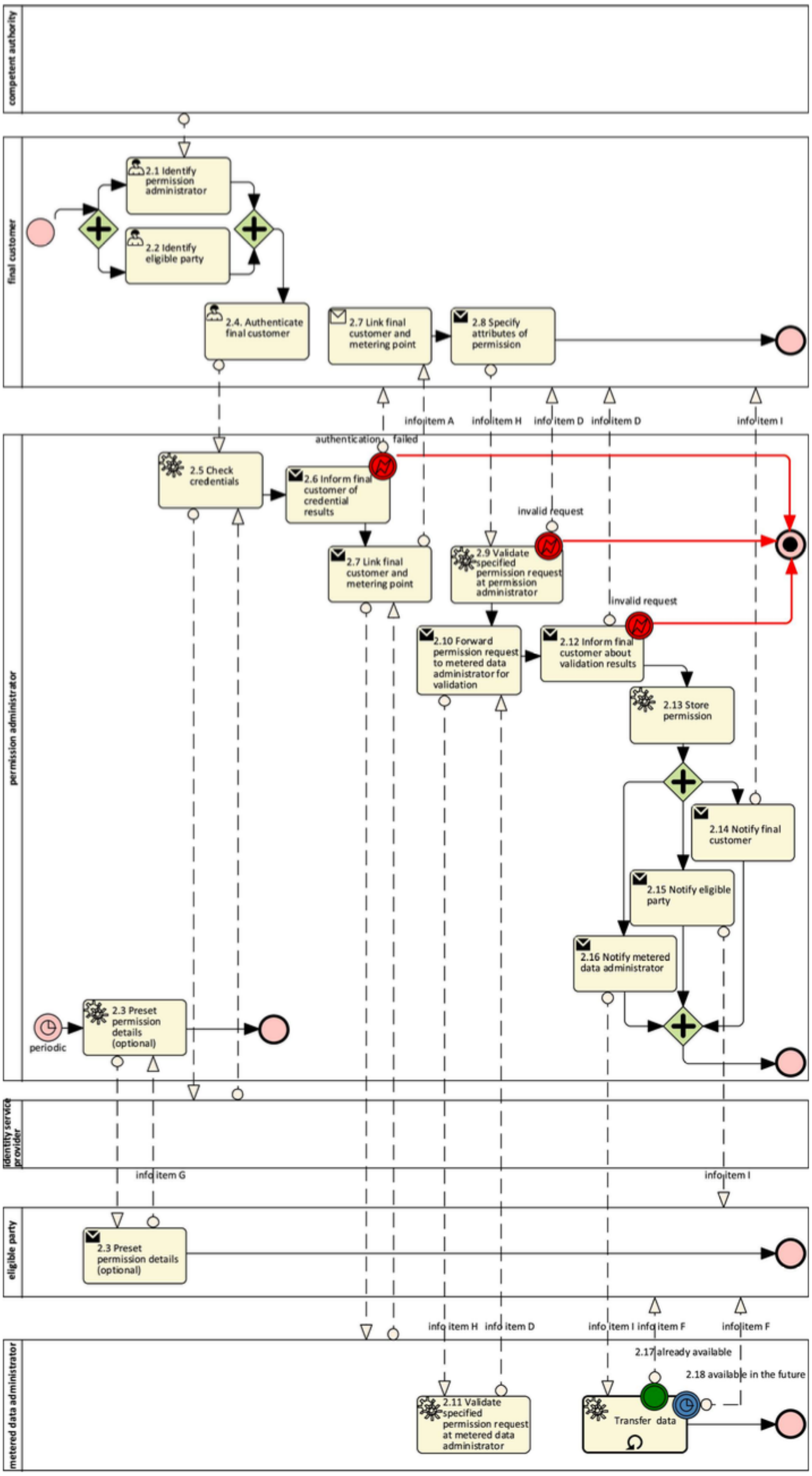


Figure 4: Accessing VHMD by an eligible party [7].

3) Procedure 3: Termination of service by an eligible party (process shown in Figure 5)
    - Eligible parties can terminate permissions, notifying the PA and the customer, who in turn ensures the MDA stops data transfer.

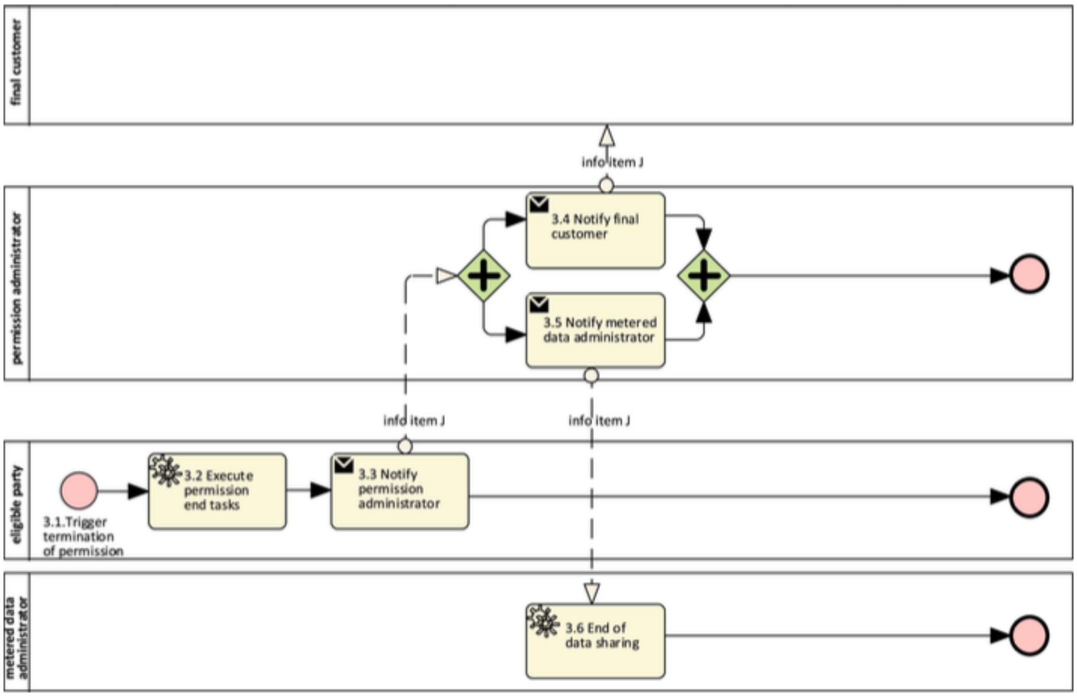


Figure 5: The process of termination of service by an eligible party [7].

4) Procedure 4: Revocation of Permission by the Customer (process shown in Figure 6)
    - Customers can revoke permissions using the PA, which then notifies the MDA and eligible party, stopping further data transfer.

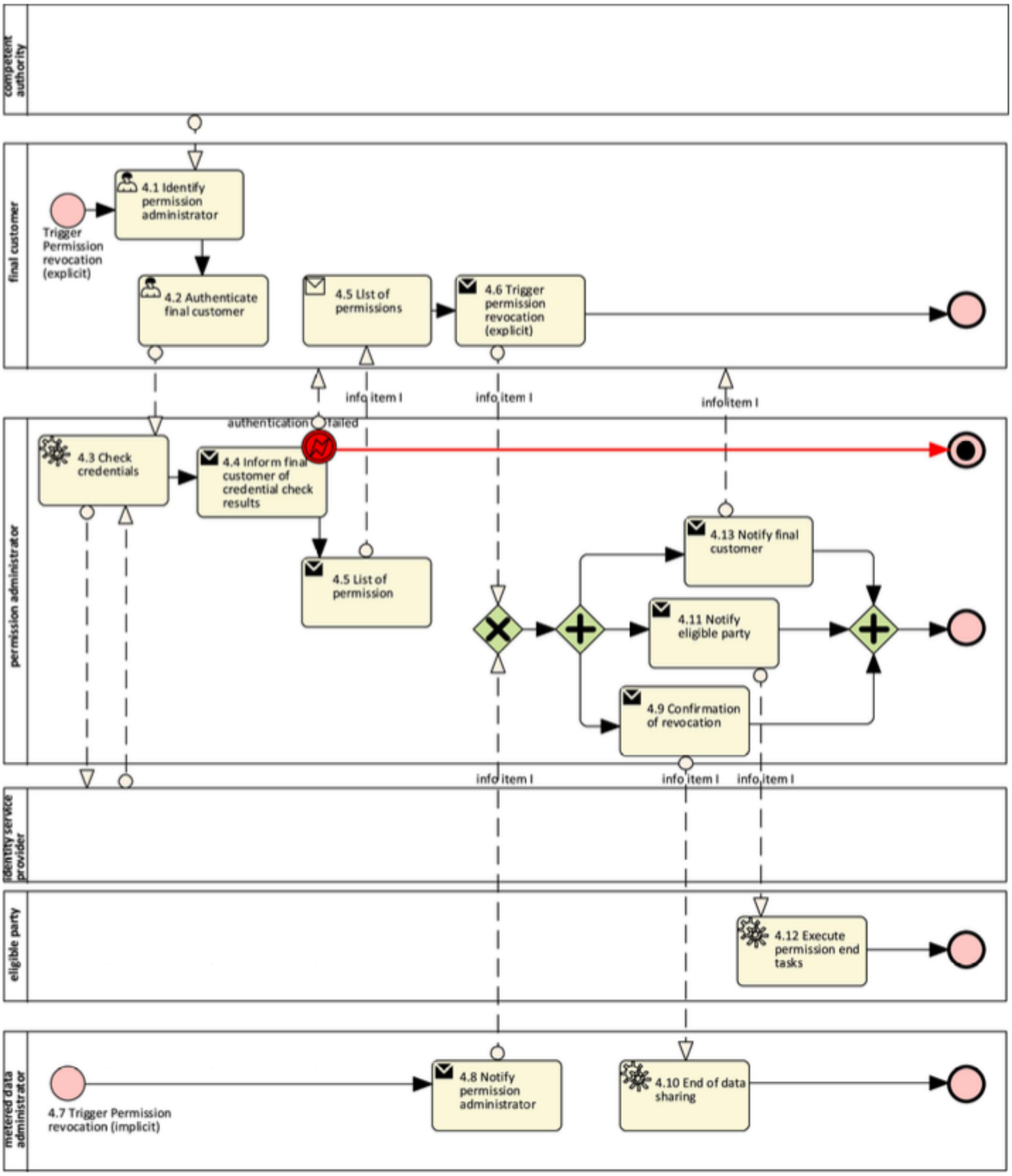


Figure 6: The process of revocation of a permission by the customer [7].

5) Procedure 5 & 6: Near Real-Time Data access (respective processes are shown in Figure 7 and Figure 8)
    - These procedures cover the activation and reading of near real-time data from smart metering systems, ensuring customers and third parties can access real-time data directly.

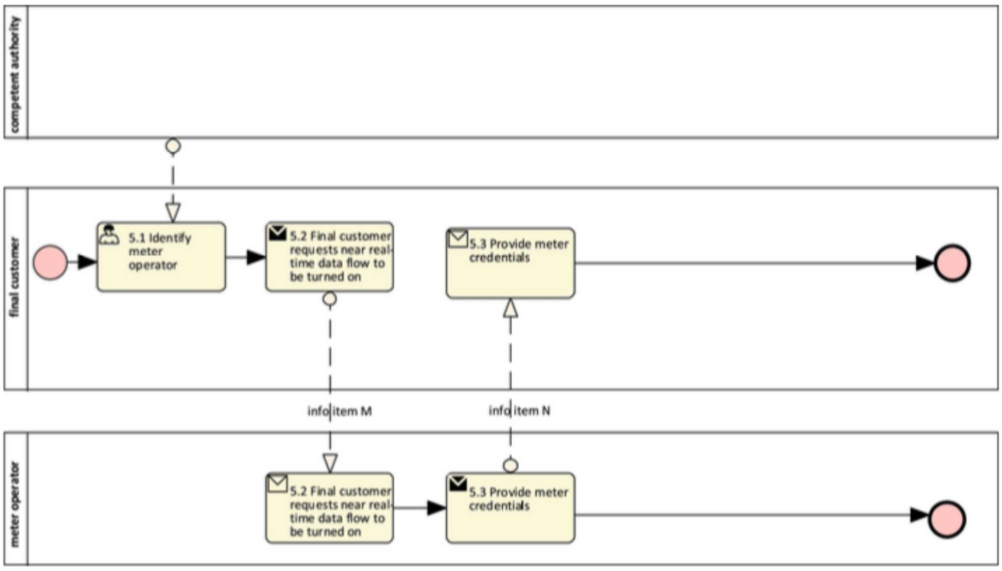


Figure 7: Activate near real-time data flow from smart metering system [7].

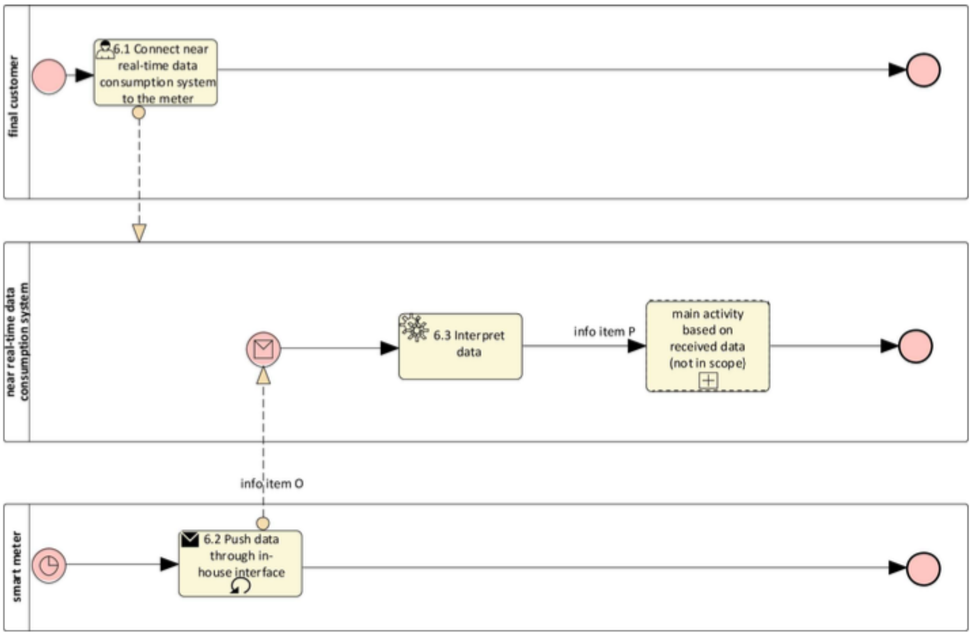


Figure 8: Read near real-time data from smart meter or smart metering system [7].

Progress of Implementation

This section will provide a brief overview of the progress of the Implementing Regulation (EU) 2023/1162 [7] in some European countries to the best of the authors knowledge as of May 2024 as this also influences the current development of the EDDIE framework and the respective region connectors (please be referred to Deliverable 5.1 for further details [29]) and the EDDIE interoperable communication layer.

The overview is based on publicly available information and might not be exhaustive. We also cover some countries that are not part of the European Union but have already implemented systems that are compliant with the Implementing Regulation (EU) 2023/1162 [7].

- **Estonia**: Estonia operates a DEP called estfeed, managed by Elering, which provides a centralized platform for data access by market participants and third parties [10]. Customers can manage their permissions through the e-elering platform, covering all required procedures of the reference model.
- **Italy**: Italy uses the Sistema Informativo Integrato (SII), operated by Acquirente Unico, which facilitates data exchange among market participants. The Portale Consumi platform allows final customers to access their data, but third-party access is not yet fully compliant with the reference model [11].
- **Netherlands**: The Netherlands features a decentralized system where VHMD is stored on smart meters. The Central Access Server (CAS), operated by Energie Data Services Nederland (EDSN), facilitates data access. Customers use the Mijn Energie platform to manage permissions, covering all required procedures [12].
- **Finland**: Finland uses a centralized DataHub operated by Fingrid Datahub, which stores VHMD and manages permissions. The DataHub provides both REST and SOAP interfaces for data access, ensuring compliance with the reference model [13].
- **Norway**: Norway's Elhub, managed by Elhub AS, serves as a centralized DEP, facilitating data exchange and managing permissions through the Elhub Messaging Interface (EMIF). Customers can use the Min side platform to access and manage their data, covering all procedures of the reference model [14, 15, 16].
- **Belgium**: Belgium has a DEP called the Central Market System (CMS), operated by Atrias. While the CMS facilitates data exchange among market participants, it is not yet accessible to third parties. Some DSOs have their own platforms for data access, but national-level compliance with the reference model is pending [17, 18, 19].
- **Germany**: Germany operates a decentralized data exchange using the EDIFACT standard [20] for communication among 890 DSOs [21]. The country is transitioning to smart metering systems, which will store VHMD locally and facilitate secure data exchange through Smart Meter Gateway Administrators (SMGAs). The new system will cover all procedures of the reference model once fully implemented.

- **Austria**: Austria uses a decentralized data exchange with standardized communication facilitated by the Energiewirtschaftlicher Datenaustausch (EDA) [1] platform. Third parties must host their own Ponton X/P Messenger for large-scale data access, which can be a barrier for smaller companies. Compliance with the reference model is partial, with customers needing to contact their DSOs directly for data access [22, 23, 24, 25, 26].
- **Spain**: Spain operates a DEP called Datadis [27], which acts as a central access point for VHMD. The DSOs store the data, while Datadis manages permissions and provides standardized data formats. This system covers all procedures of the reference model.
- **Denmark**: Denmark's DataHub, operated by Energinet, serves as a centralized DEP for all market participants. Customers and third parties interact with the DataHub through the ElOverblik platform, which provides both web and API access for managing permissions and accessing data [28].

In the rest of the deliverable (and especially in chapter 3, where the message-based communication is covered in detail), we will focus on Austria and France, since these two countries were the first region connectors that have been integrated into the interoperable communication layer and internally mainly follow the message-based communication process. The following section 2.4 illustrates and discusses the development process during the first 18 months of the project.

[1] See https://www.eda.at

2.4 Architectural Design and Development Progress

In the beginning of the development phase of project EDDIE, we focused on Austria and France and started to define and discuss potential architectural solutions. Figure 9 illustrates the first attempt of an architectural baseline for connecting Austria via EDA. There we envisioned a simple application accessible via a browser, realized with embedded micro-frontends. The central component "EddieFramework", respectively the "EdaConnector" was meant to handle the connection to EDA in the Austrian case. This very basic structure built the baseline for further developments.

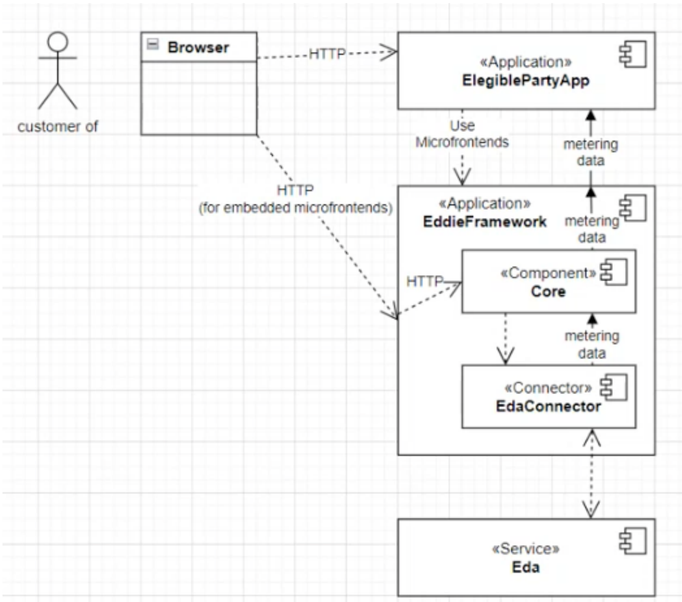


Figure 9: Early definition of a potential architectural baseline for connecting Austria via EDA.

Figure 10 shows a more refined overview of the frameworks core components depicting the differences of the Austrian and French workflow. Following up on the state-of-the-art and especially the EU's Implementing Regulation (EU) 2023/1162 [7] reference model with the six key procedures (i.e. (i) access to validated historical metering and consumption data by the final customer, (ii) access to validated historical metering and consumption data by an eligible party, (iii) termination of service by an eligible party, (iv) revocation of a permission by the final customer, (v) activation of near real-time data access, and (vi) reading near real-time data). There, the user interacts with the permission facade (be referred to deliverable D3.1 – EDDIE Consent Façade [30]) and requests data. The region connectors then care about the regional workflow. For example, in Austria, this is done via CCMO requests in XML format and is sent to the DSO via Ponton X/P messenger, following a message-based communication. The user must grant permission at the DSO website. Details on the Ponton

X/P messenger and the message-format follow in chapter 3 within this deliverable. After the user has granted permission, the EP can access the historical validated data (HVD). The second workflow that is depicted in Figure 10 shows the French case.

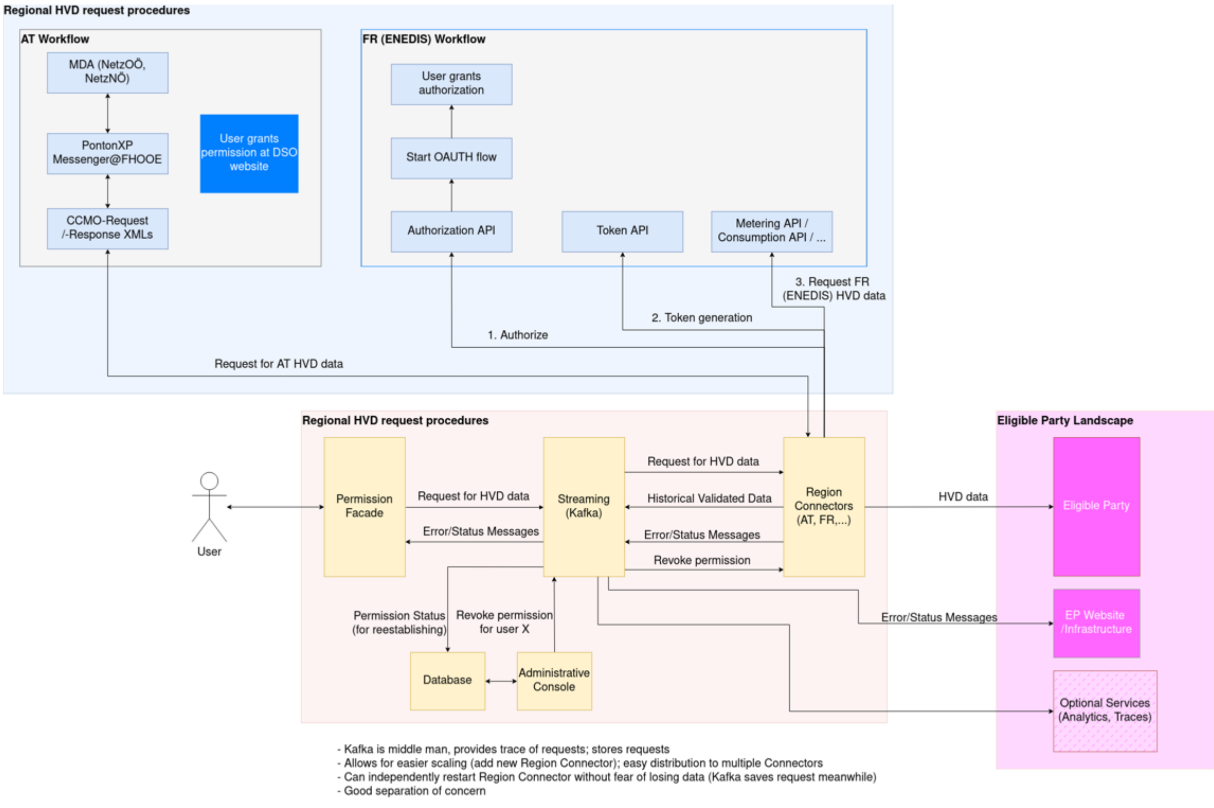


Figure 10: Regional HVD request procedures for Austria and France.

In the following Figure 11, the country-specific workflows are highlighted and shown with more details. For Austria the workflow is the following:

1) The EDDIE Framework send the permission request: this request is transmitted via Ponton X/P messenger in CCMO XML format (see Chapter 3 for further details) to the Distributed System Operator (DSO).
2) The user is redirected to the DSO website where the permission needs to be granted (see Figure 12 for an example of an Austrian DSO).
3) The DSO sends the permission response via Ponton X/P messenger back.
4) With a granted permission, the requested data can be sent.

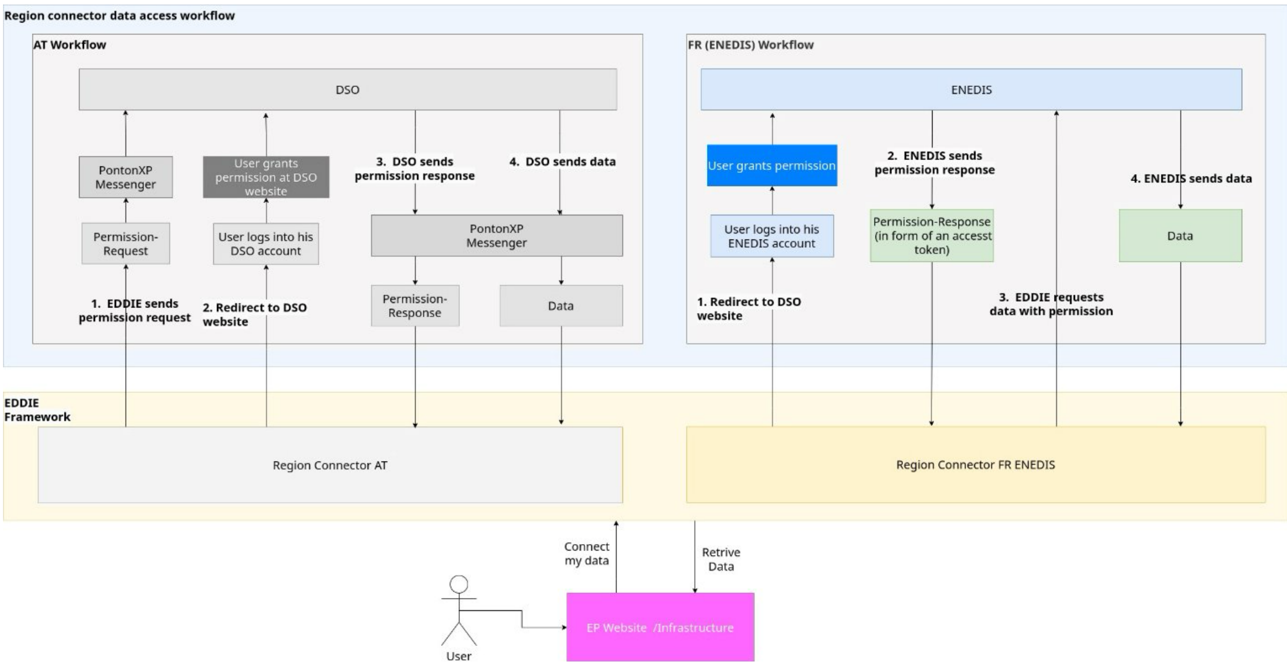


Figure 11: Country-specific workflows for Austria and France.

In France, the workflow is different, as it can be seen in Figure 11:

1) The user is redirected to the DSO (in France this is mainly Enedis[2]) website and needs to log in to grant permission.
2) The DSO/Enedis sends the permission response in form of an access token.
3) EDDIE can request data with this generated token.
4) Finally, the data can be sent safely.

Figure 12: Exemplary permission request on DSO website (personal details are greyed out).

[2] See https://www.enedis.fr

In the following Figure 13 a detailed process description is depicted in Business Process Modelling Notation (BPMN). There, the three different involved instances are highlighted (i.e. (i) eligible party, (ii) the EDDIE Framework, and (iii) the country-specific connection components). By depicting the complex processes in this format and visualizing the necessary steps and country-specific differences, we were able in the design and development phase to specifically work on a message-based communication layer in the general interoperable communication layer and integrate this into the region connector implementations.

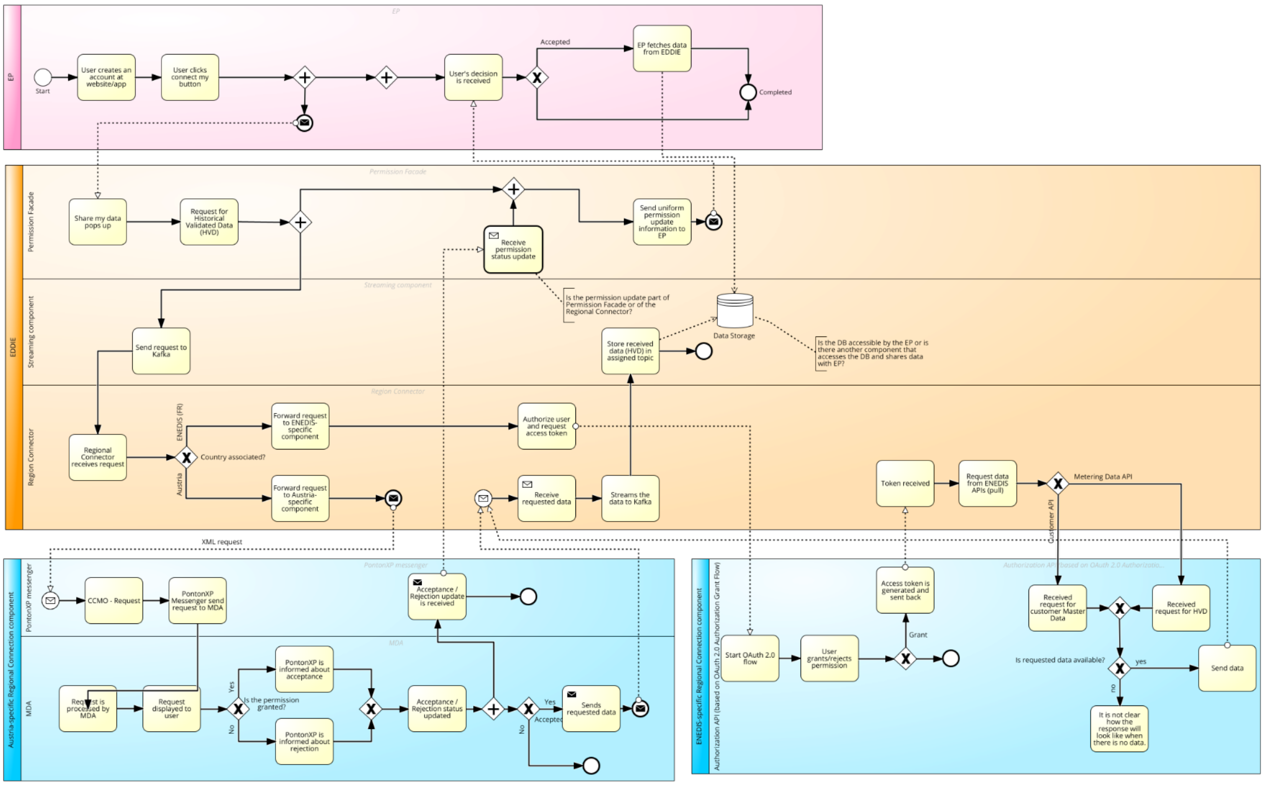


Figure 13: Detailed process view for Austria and France in BPMN notation.

Especially for the Austrian case in Figure 13 – depicted in the lower left blue box – the XML-message request that is transferred to the MDA/DSO via Ponton X/P messenger is notable. Same for the token exchange mechanism for the French case in the lower right box of Figure 13. The following sequence diagram in Figure 14 shows the process for the Austrian case arranged in a time- and message-specific sequence with focus on processes, messages and objects involved as needed to carry out the desired functionality. The upper part – "Permission Workflow" – shows the process that is necessary for retrieving the user's permission for sharing data, the lower part – "Receive Consumption Records" – shows the part of the process where data can be sent.

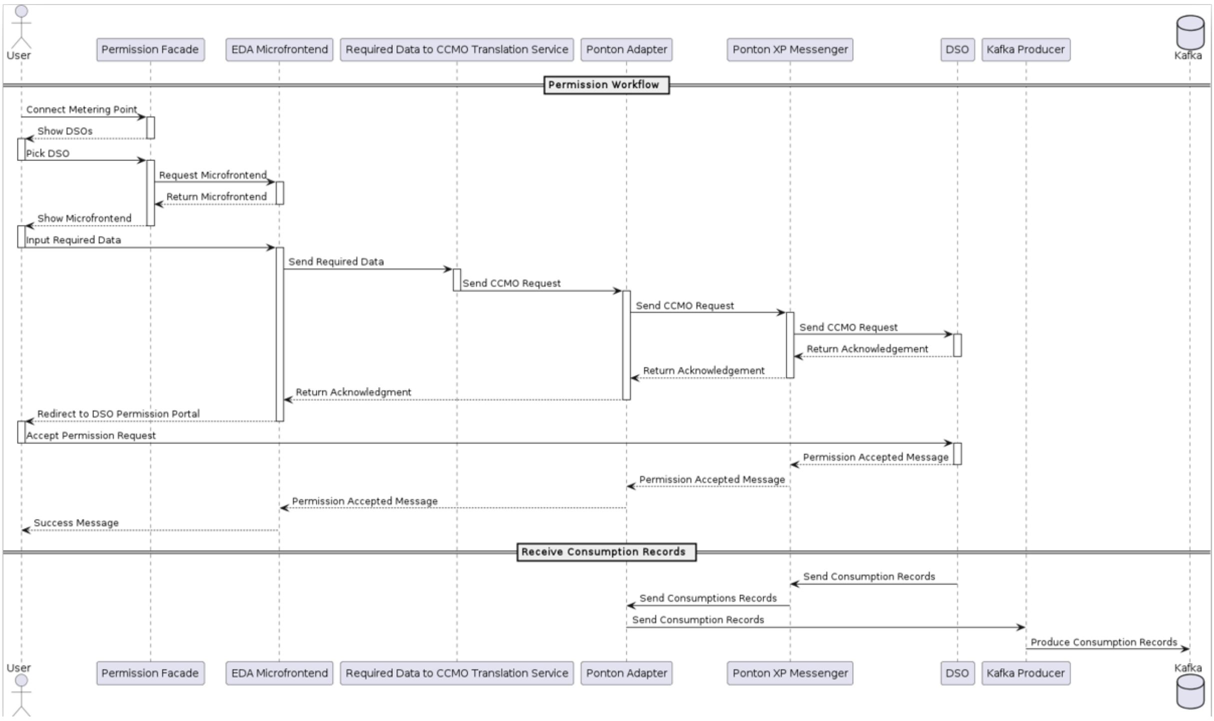


Figure 14: Sequence diagram of the Austrian process.

Another very important part within the design and development process to find a suitable architecture for the message-based communication part in the interoperable communication layer for the Austrian case was the understanding how required data can be translated to respective CCMO XML messages that can be processed within the Ponton X/P messenger. Therefore, we have experimented with a translation service component built in directly in the EDA/Austria region connector. A possible integration of such a service in the respective region connector is depicted in Figure 15. There, the frontend uses the translation service to generate CCMO conform messages that can then be handled within the Ponton X/P messenger and be further transferred. An example for such a message translation is visible in Figure 16. In the sequence diagram in Figure 14 depicted above this translation service is integrated between the frontend and the "Ponton Adapter" as "Required Data to CCMO Translation Service".

In the following chapter 2.5, the general architecture that has been derived from the information of the country-specific workflow as described above is described in detail.

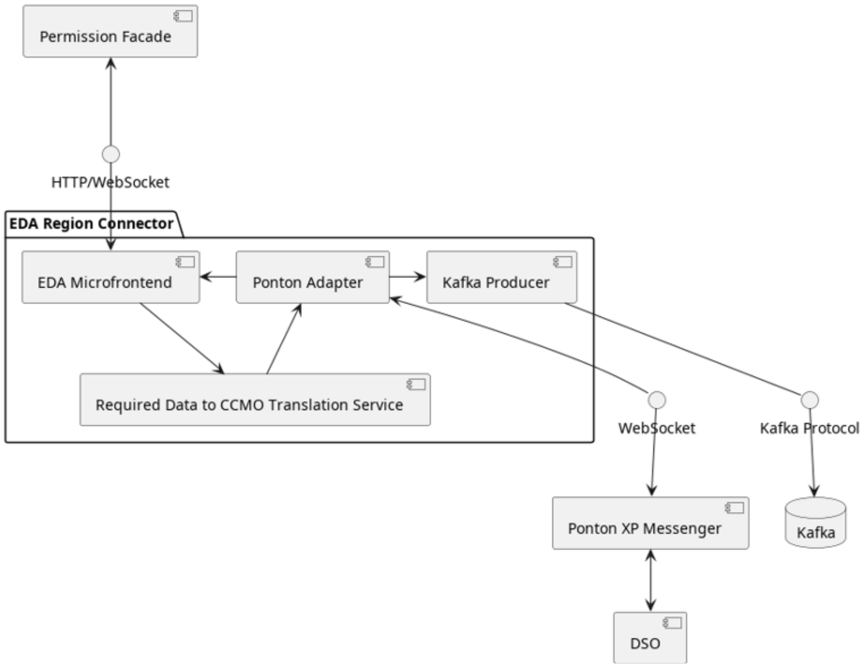


Figure 15: Inside view of the EDA Region Connector.

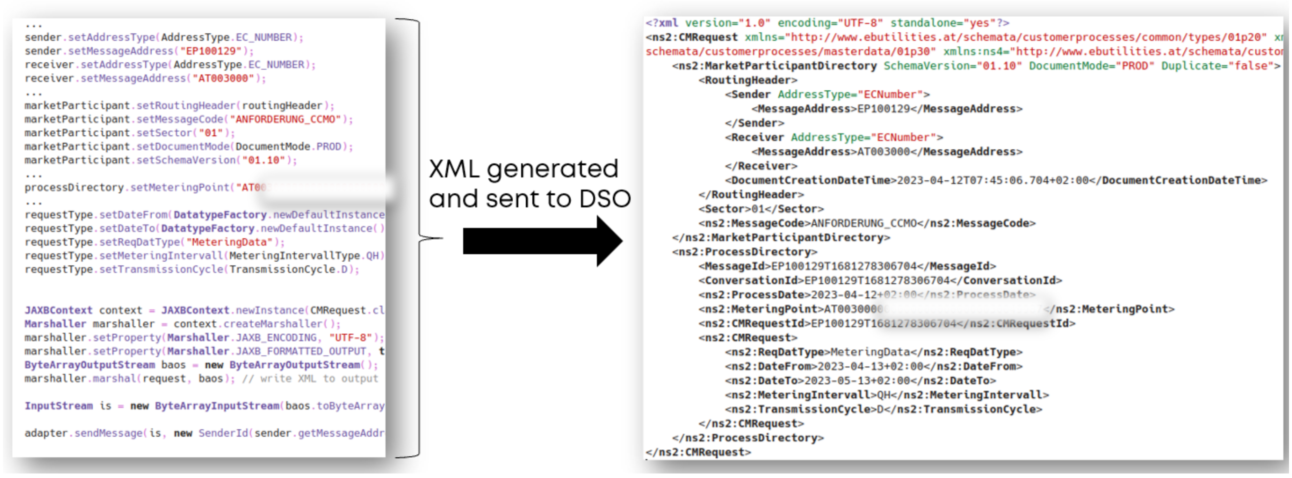


Figure 16: Generation of a CCMO XML message.

2.5 General Architecture

With the knowledge gathered during design and development phase in the first 18 months in the EDDIE project, we have been able to break this down to a composition of three distinct layers ((i) region connector layer, (ii) core layer, and (iii) application connector layer), as depicted in Figure 17.

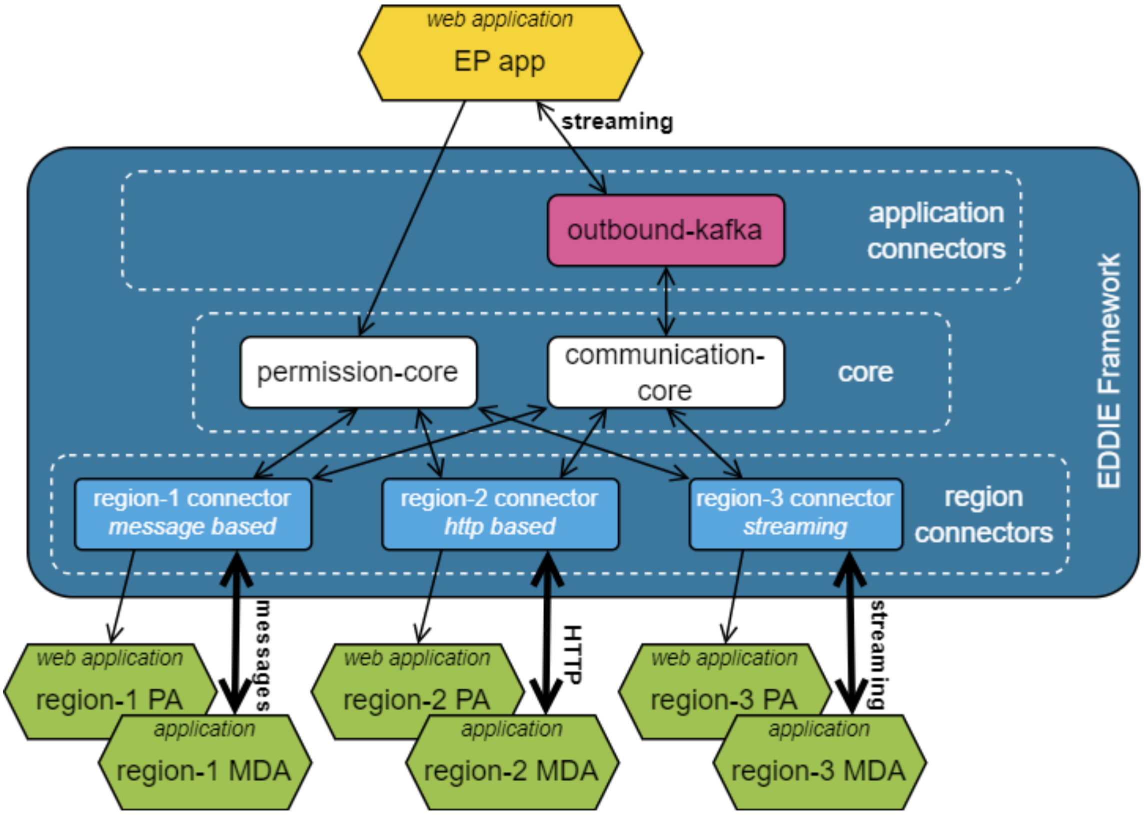


Figure 17: Layered concept of the EDDIE architecture.

We have already foreseen the different messaging and communication aspects that are covered in the project's work-package 4 and the respective tasks 4.2 ("message-based communication"), task 4.3 ("REST API-based communication") and task 4.4 ("streaming-based communication"). The current document is mainly concerned with message-based communication, whereas the baseline for this has already been described in the previous section. In the following section 2.5.1 the three layers and their main purpose are described in detail.

2.5.1 Layered Architecture Concept

1) **Region Connector Layer**: A region connector is responsible for acquiring consumption data from a regional data sharing infrastructure and transforming it into a standardized data format based on the Common Information Model (CIM). To accommodate multiple regions, the region connector layer comprises various individual connectors, each tailored to manage the data transmission and permission processes specific to each region's data sharing infrastructure.
2) **Core Layer**: The core layer realizes the plugin architecture so that region connectors and application connectors can be added to the framework without affecting other plugins. Core components are available for handling permissions as well as the data streams from the region connectors.
3) **Application Connector Layer**: An application connector facilitates the transmission of consumption data to the eligible party's application by implementing a specific protocol. At present, there is one plugin that transmits JavaScript Object Notation (JSON) encoded CIM messages via Kafka. Nevertheless, the framework is designed for extensibility and can readily accommodate more efficient encoding formats, such as Protocol Buffers or Apache Avro.

2.5.2 Connection of other Messaging Solutions

Moreover, the addition of support for other messaging solutions, such as Google Cloud Pub/Sub or Azure Service Bus, is particularly appealing for eligible parties with applications deeply integrated into proprietary cloud ecosystems. This flexibility ensures that the framework can adapt to various technical requirements and preferences.

2.5.3 Region Connector Architecture

The primary requirements for inbound communication are encapsulated within the region connectors, which serve as the gateways for data flow into the EDDIE Framework. This is depicted in Figure 18.

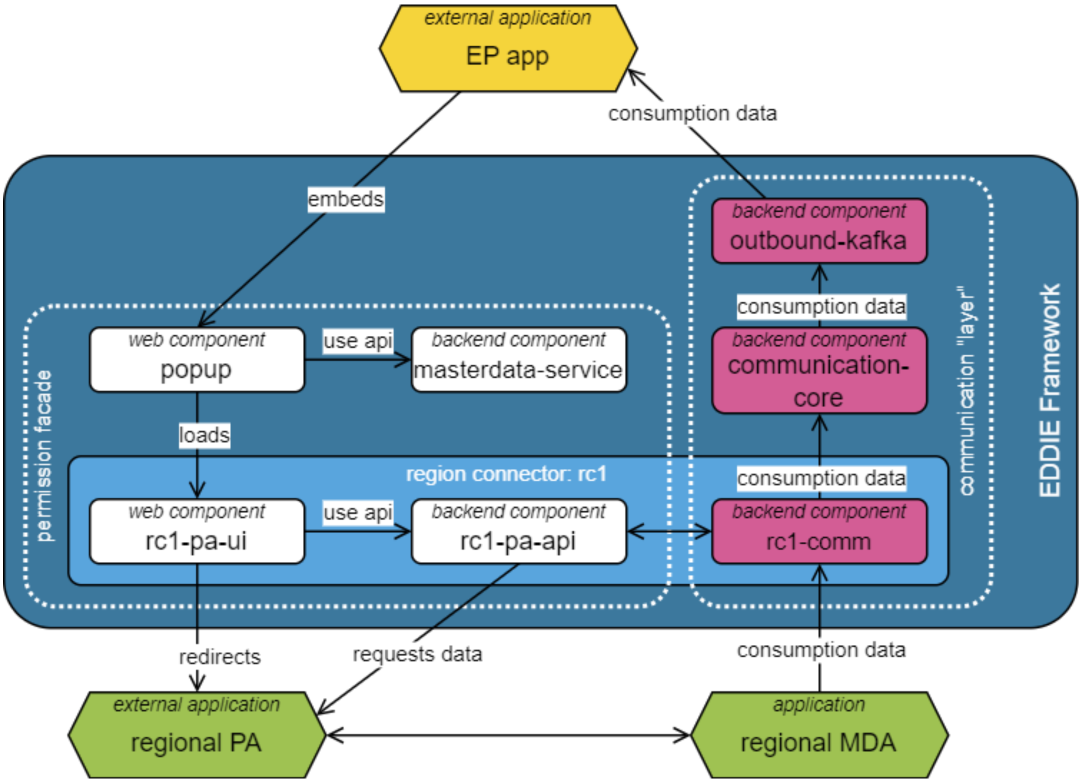


Figure 18: Detailed Region Connector architectural concept.

To facilitate message-based communication, some region connectors make use of middleware systems e.g., the Ponton X/P messenger is used for AS4 communication with EDA in Austria (see chapter 3 for a detailed description of the implementation of the Austrian case) and an MQTT broker is used for streaming communication with Administrative Interface for In-house Data Access (AIIDA) instances.

While the region connectors standardize communication protocols and data formats, they also abstract the disparities among various permission administrators that are inherent to the regional data sharing infrastructure. This encapsulation ensures a consistent and secure interaction layer, despite the underlying heterogeneity.

2.5.4 Technical Implementation

To facilitate ease of use, the EDDIE Framework is deployed as a monolithic application. This architecture not only simplifies operations but also enhances internal data transfer efficiency, thereby minimizing computational demands and reducing the energy footprint of the framework.

Monolithic systems, however, can be challenging to maintain. To address this, the EDDIE Framework's codebase is meticulously organized. The core of the framework realizes a plugin system built on the Spring Framework[3]. This design ensures that each region connector operates independently and interacts with other components exclusively through clearly specified interfaces.

In the following chapter, the message-based communication is described in detail with adhering to the Austrian case as example for this particular communication and data exchange method.

[3] See https://spring.io/projects/spring-framework

3 Message-Based Communication

3.1 Introduction

The primary goal of work package 4 in the EDDIE project is to address the diverse methodologies of energy data exchange across Europe, ensuring interoperability and meeting stakeholder requirements for different data exchange types. This involves developing a customizable infrastructure to bridge the gap between various national implementations of EU-regulated reference processes, particularly in terms of communication protocols and data formats.

Task 4.2 ("Message-based communication") focuses on the traditional method of exchanging structured data sets, crucial for processes like supplier switching, nomination, and invoice data submission. The implementation will reuse existing technologies based on classical protocols like ebMS2.0 and AS4. Nevertheless, tasks 4.3 (REST API-based communication) and 4.4 (streaming-based communication) have already started within the project but are not subject of the current document.

As already mentioned above, the architecture for the country specific regional connectors can have different architectures, due to e.g. legal constraints, energy data hubs, data models etc. Therefore, EDDIE has to be able to handle generic requests for different types of connectors. In general, the regional connector provides the main connection for the customer / EP to interact with the system (via an interface), to forward that information to the EDDIE core and to the database eventually, as well as to handle historic energy data from regional data hubs. For further details on the region connectors, please be referred to Deliverable 5.1 – "Intermediate report on best-practices and EU region connectors created" [29].

In the following the Austrian region connector will be presented in detail, since this incorporates the message-based communication approach and has been implemented as one of the first region connectors.

Generally, the message-based communication is not restricted to Austria, the base functionality regarding message exchange is generic and can be applied within other regional connectors. Nevertheless, to showcase the complexity, the following section 3.2 provides detailed descriptions of the Austrian workflow.

3.2 Region Connector Austria

3.2.1 Introduction

In Austria, where a decentralised, message-based technology (AS4) is used, eligible parties (EP) need to register at the EDA registration website [26], receive an EP identifier and then request a public/private key pair together with some connectivity information to communicate.

Generally, EDA[4] can be thought of as the data access provider (DAP) for historically validated data in Austria. It works as a central messaging service for all the Distribution System Operators (DSOs) in Austria by implementing the AS4 messaging protocol and using the ebMS3 specification. The following Figure 19 (taken from the EDA website[5]) illustrates this.

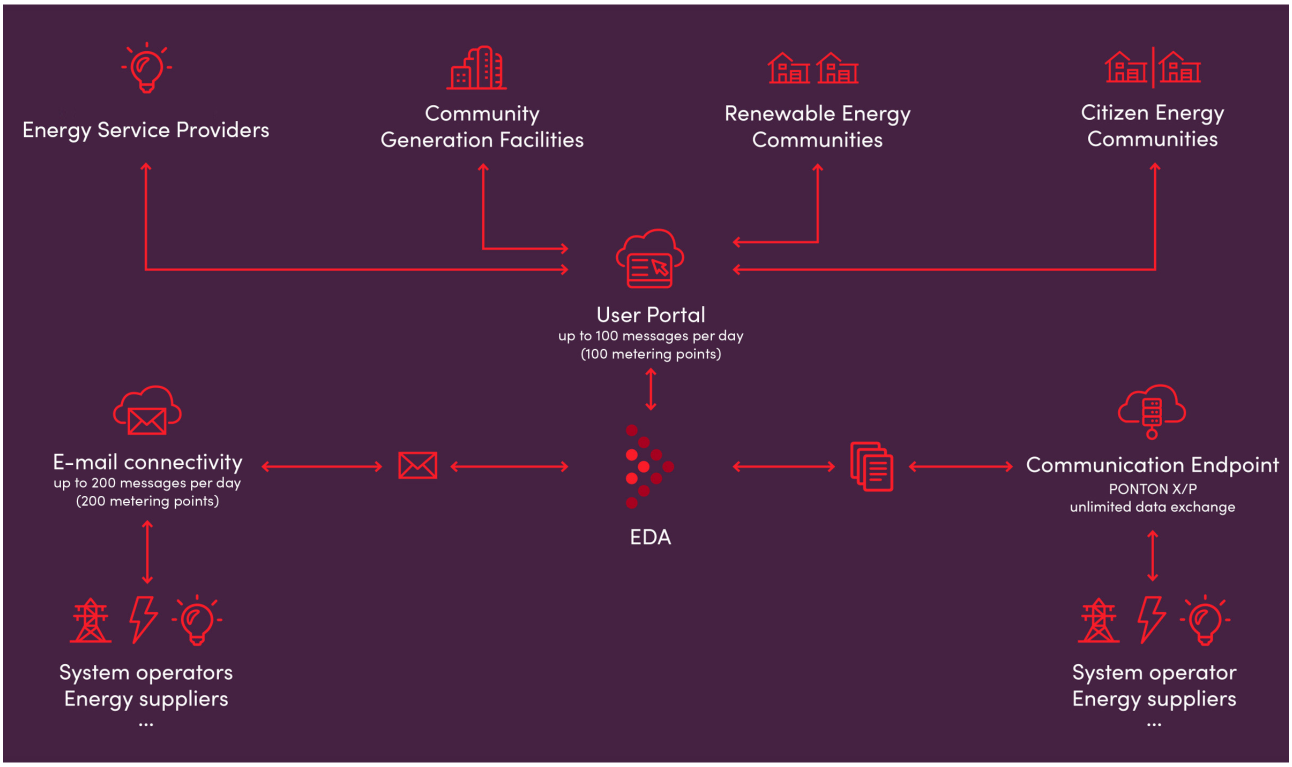


Figure 19: Illustration of data exchange in Austria facilitated by EDA [26].

[4] See https://www.eda.at
[5] See https://www.eda.at/wie-funktioniert-eda?lang=en

As it can be seen, instead of an API based approach, Austria exchanges energy data via encrypted XML files.

How it works and relevant processes can be found at ebUtilities[6]. The relevant processes for EDDIE fall under the "Customer Consent Management" category and are described in the following sections.

3.2.2 EDA Processes

Apart from being a data access provider (DAP), EDA in co-operation with Oesterreichs Energie[7] is also responsible for defining the business processes and market documents for the Austrian energy data exchange. The business processes and market documents are defined, consulted and published on ebUtilities.at. The relevant processes for project EDDIE all fall under the category of Customer Consent Management (CCM). As the name implies, these processes are about managing the permissions between the customers and eligible parties.
The CCM processes can be further subdivided into two categories:

1) requesting customer data
2) terminating and revoking active permissions

The process for requesting customer data is called CM_REQ_ONL. This process is complex but can broadly be simplified into two parts:

1) obtaining permission from the customer
2) the transmission of the requested data

To simplify the explanation, the process has been split into two separate BPNM diagrams. The first diagram (Figure 20) depicts the steps related to obtaining the permission and the second diagram (Figure 21) shows the transmission of the data. Figure 21 also includes the processes that are related to terminating and revoking permissions. In other words, the second diagram shows a combination of the CM_REQ_ONL process with the termination (CM_REV_SP) and revocation (CM_REV_CUS and CM_REV_IMP) processes. This makes it easier to understand the relationship between these processes, as termination and revocation can happen at any point after permission has been given but data transmission has not yet been completed.

---

[6] See https://www.ebutilities.at/prozesse
[7] See https://oesterreichsenergie.at/

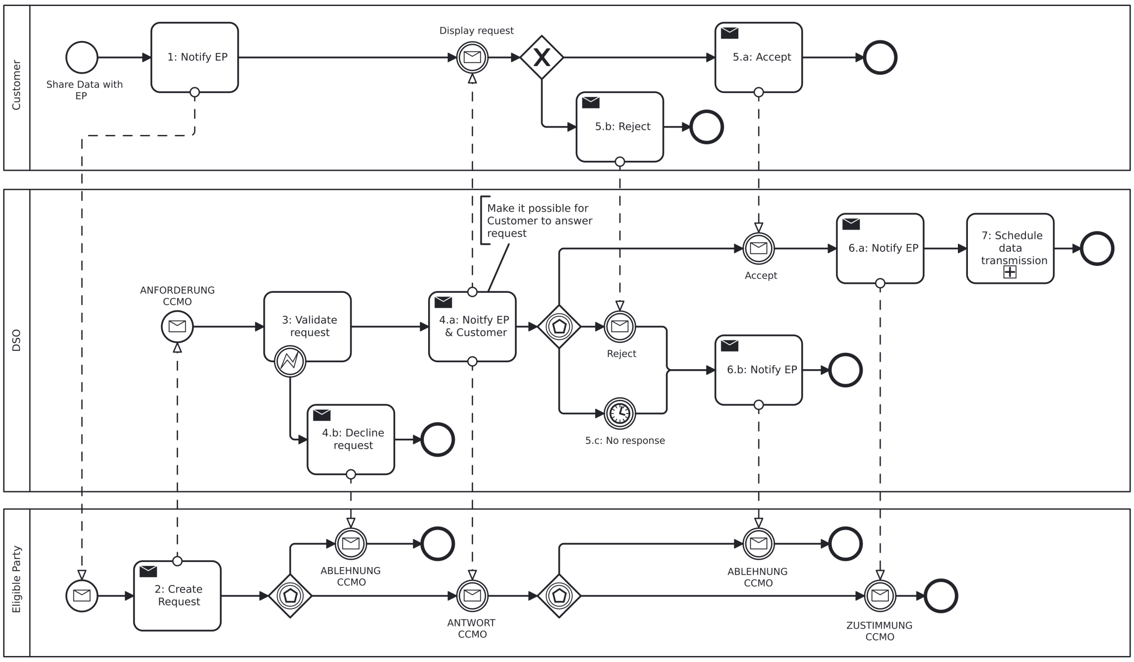


Figure 20: BPMN diagram showing steps for obtaining the permission.

The process starts with the customer informing the eligible party (EP) that they want to share their data (1). The EP then creates and sends a CMRequest message with message code ANFORDERUNG_CCMO (see Figure 22 for an example of such an XML message) for the data the final customer wants to share to the DSO (2). The DSO will then validate the request (3). If the request is invalid, the DSO will send a CMNotification message with message code ABLEHNUNG_CCMO to the EP (4.b) at which point the process ends. If the request is valid, the DSO will send a CMNotification message with message code ANTWORT_CCMO to the EP (see Figure 23 for an exemplary message) and make it possible for the final customer to view and accept the request online (4.a). The process does not specify how the request is presented to the customer. However, it is typically displayed directly in the customer portal of the DSO if the metering point of the final customer was included in the CMRequest message. If the metering point was not included, the final customer has to provide the cmRequestId to the DSO to view and accept the request, this requires that the EP informs the final customer about the cmRequestId. Once the final customer can view the request, they can either accept (5.a) reject (5.b) or ignore (5.c) the request. If the final customer rejects or ignores the request, the DSO will send a CMNotification message with message code ABLEHNUNG_CCMO to the EP (6.b) and the process ends. If the final customer accepts the request, the DSO will send a CMNotification message with message code ZUSTIMMUNG_CCMO to the EP (6.a) and schedule the data transmission (7).

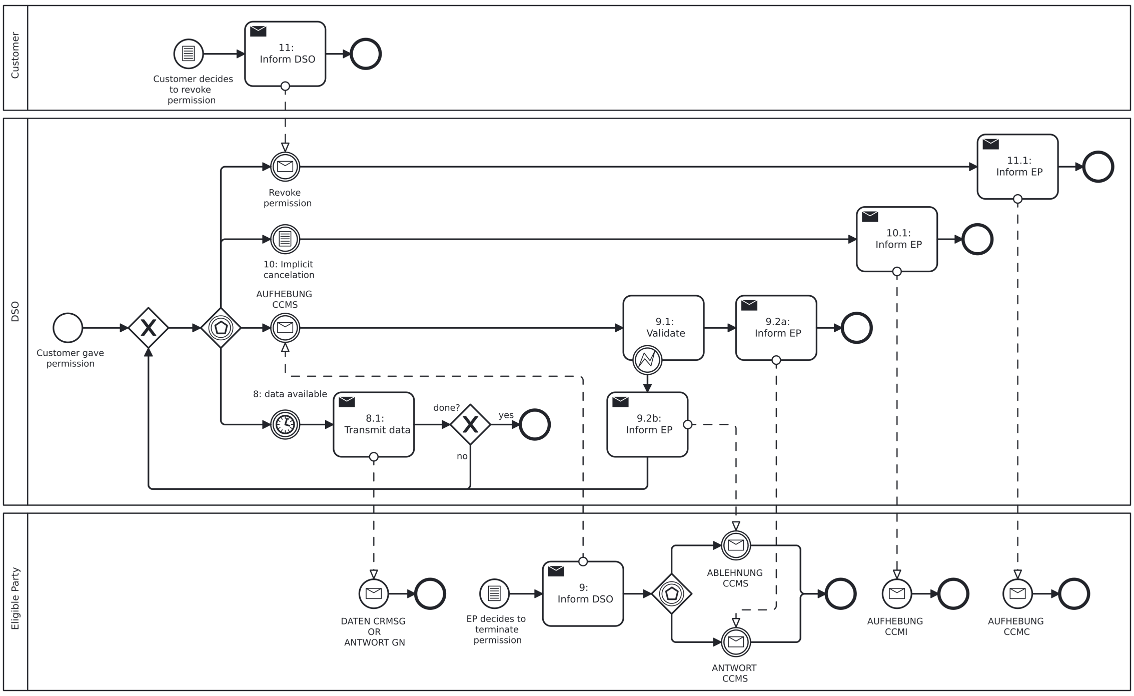


Figure 21: BPMN diagram showing steps data exchange and permission termination.

After the final customer gave permission, there are four events that can occur: the data is available (8), the EP terminates the permission (9), the DSO implicitly revokes the permission (10), or the final customer revokes the permission (11).

If the requested data is available (8), the DSO will send the data, which is either a ConsumptionRecord with message code DATEN_CRMSG or a MasterData message with message code ANTWORT_GN to the EP (8.1). If this data contained all the requested data, the process ends here, otherwise, this step will be repeated until all the data is sent or another event that terminates or revokes the permission occurs.

The process that allows EPs to terminate a permission is CM_REV_SP. The name is short for "Consent Management - Revocation by Service Provider". To initiate this process, the EP sends a CMRevoke message with the message code AUFHEBUNG_CCMS to the DSO as shown by step 9. This message must contain the consentId and metering point of the permission to be terminated. The DSO then validates the termination request (9.1). If there is an active permission for the given consentId and metering point, the DSO confirms the termination by sending a CMNotification message with message code ANTWORT_CCMS to the EP (9.2a). If the request is invalid, the DSO responds to the termination request by sending a

CMNotification message with message code ABLEHNUNG_CCMS to the EP (9.2b) and the DSO continues the process as if the termination request had never been made.

EbUtilities[8] also specifies an implicit revocation process CM_REV_IMP, short for "Consent Management - Implicit Revocation" by other processes. This is a process that is automatically triggered by other processes the DSO needs to perform. One such process would be the cancellation of a contract between the final customer and the DSO due to the customer moving out of their current residence. If this process is triggered, the DSO will send a CMRevoke message with the message code AUFHEBUNG_CCMI to the EP as shown by steps 10 and 10.1. The message contains the consentId, metering point, the reason for the revocation and when the termination will take effect.

When a final customer wants to revoke a permission, the process is called CM_REV_CUS, short for "Consent Management - Revocation by Customer". This process is triggered by the final customer informing their DSO that they want to revoke a permission as shown by step 11 in Figure 21. The DSO must comply with this and inform the EP of the revocation by sending them a CMRevoke message with the message code AUFHEBUNG_CCMC (11.1). The message will contain the consentId and metering point of the permission that has been revoked.

As this shows, Austria uses an asynchronous message-based system that will push data to the EP when available. Our implementation follows this process and maps messages we receive or send to our own permission state model.

In Figure 22 and Figure 23 two examples of CCMO messages from real data requests generated by EDDIE are displayed.

[8] See https://www.ebutilities.at/

```
▼<ns3:CMRequest xmlns="http://www.ebutilities.at/schemata/customerprocesses/common/types/01p20"
 xmlns:ns2="http://www.ebutilities.at/schemata/customerconsent/cmrevoke/01p00"
 xmlns:ns3="http://www.ebutilities.at/schemata/customerconsent/cmrequest/01p20"
 xmlns:ns4="http://www.ebutilities.at/schemata/customerconsent/cmnotification/01p11"
 xmlns:ns5="http://www.ebutilities.at/schemata/customerprocesses/masterdata/01p32"
 xmlns:ns6="http://www.ebutilities.at/schemata/customerprocesses/consumptionrecord/01p40">
 ▼<ns3:MarketParticipantDirectory SchemaVersion="01.20" DocumentMode="PROD" Duplicate="false">
  ▼<RoutingHeader>
   ▼<Sender AddressType="ECNumber">
     <MessageAddress>EP100022</MessageAddress>
    </Sender>
   ▼<Receiver AddressType="ECNumber">
     <MessageAddress>AT003000</MessageAddress>
    </Receiver>
    <DocumentCreationDateTime>2024-05-23T09:43:29.968059082</DocumentCreationDateTime>
   </RoutingHeader>
   <Sector>01</Sector>
   <ns3:MessageCode>ANFORDERUNG_CCMO</ns3:MessageCode>
  </ns3:MarketParticipantDirectory>
 ▼<ns3:ProcessDirectory>
   <MessageId>EP100022T1716450209939</MessageId>
   <ConversationId>EP100022T1716450209939</ConversationId>
   <ns3:ProcessDate>2024-05-23</ns3:ProcessDate>
   <ns3:MeteringPoint>AT00300000000000000[illegible]</ns3:MeteringPoint>
   <ns3:CMRequestId>T743EXUR</ns3:CMRequestId>
  ▼<ns3:CMRequest>
    <ns3:ReqDatType>HistoricalMeteringData</ns3:ReqDatType>
    <ns3:DateFrom>2024-04-23+02:00</ns3:DateFrom>
    <ns3:DateTo>2024-05-22+02:00</ns3:DateTo>
    <ns3:MeteringIntervall>QH</ns3:MeteringIntervall>
    <ns3:TransmissionCycle>D</ns3:TransmissionCycle>
   </ns3:CMRequest>
  </ns3:ProcessDirectory>
 </ns3:CMRequest>
```


Figure 22: Example for an ANFORDERUNG_CCMO message.


```
▼<ns0:CMNotification xmlns:ns0="http://www.ebutilities.at/schemata/customerconsent/cmnotification/01p11"
 xmlns:ns1="http://www.ebutilities.at/schemata/customerprocesses/common/types/01p20">
 ▼<ns0:MarketParticipantDirectory DocumentMode="PROD" Duplicate="false" SchemaVersion="01.11">
  ▼<ns1:RoutingHeader>
   ▼<ns1:Sender AddressType="ECNumber">
     <ns1:MessageAddress>AT003000</ns1:MessageAddress>
    </ns1:Sender>
   ▼<ns1:Receiver AddressType="ECNumber">
     <ns1:MessageAddress>EP100022</ns1:MessageAddress>
    </ns1:Receiver>
    <ns1:DocumentCreationDateTime>2024-05-24T09:09:40.4917660Z</ns1:DocumentCreationDateTime>
   </ns1:RoutingHeader>
   <ns1:Sector>01</ns1:Sector>
   <ns0:MessageCode>ANTWORT_CCMO</ns0:MessageCode>
  </ns0:MarketParticipantDirectory>
 ▼<ns0:ProcessDirectory>
   <ns1:MessageId>AT003000202405241109396310277129277</ns1:MessageId>
   <ns1:ConversationId>EP100022T1716541777762</ns1:ConversationId>
   <ns0:CMRequestId>ABLJOYGO</ns0:CMRequestId>
  ▼<ns0:ResponseData>
    <ns0:MeteringPoint>AT00300000000000000[illegible]</ns0:MeteringPoint>
    <ns0:ResponseCode>99</ns0:ResponseCode>
   </ns0:ResponseData>
  </ns0:ProcessDirectory>
 </ns0:CMNotification>
```


Figure 23: Example for an ANTWORT_CCMO message.

3.2.3 Exchanged market documents

This section will explain the market documents that are exchanged between the Distribution System Operator (DSO) and the eligible party (EP) during the processes described in the previous section. In order to make the diagrams easier to digest, the diagrams have been simplified. The simplified diagrams only show the most relevant attributes and relationships between the classes. Some parameter and type names have been adjusted to make the diagrams more understandable and enumerations have been replaced with humanreadable values. For the full documents, please be referred to the respective documents on the ebUtilities website[9].

All market documents share the same mark up. They always consist of a header and a body. The header is called MarketParticipantDirectory and is the same for all market documents. It contains information about the sender and receiver of the message, the messageCode of the document and other metadata. From a technical perspective, the header is used by the underlying messaging system to route the message to the correct recipient. The body is called ProcessDirectory and contains the actual content of the message and is different for each type of market document.

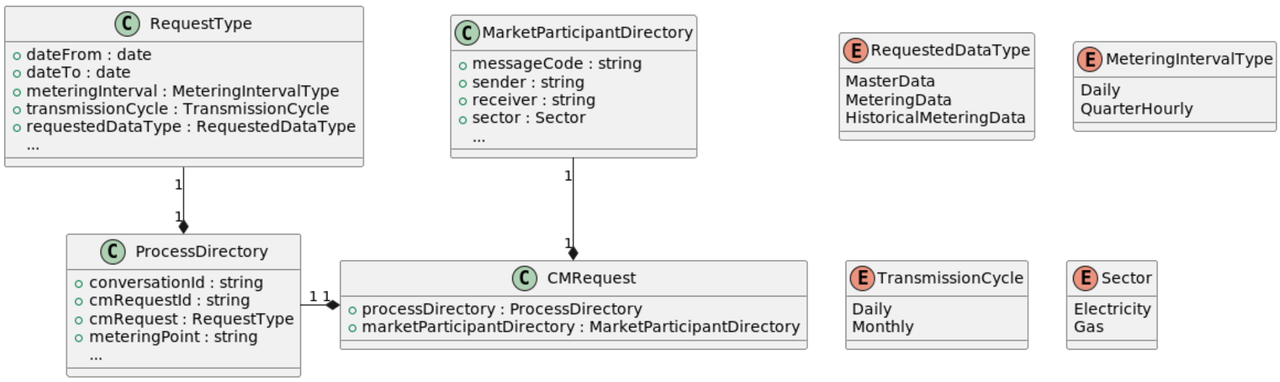


Figure 24: Simplified version of the CMRequest message.

Figure 24 shows a simplified version of the CMRequest message. The message must contain a cmRequestId and specification of whether the request is for validated historical metering data (VHMD) or accounting point master data. The meteringPoint of the customer can also be included in the message if it is known at the time of the request.

In order to request metering point master data, the RequestedDataType needs to be set to MasterData. The other parameters are not required, as the DSOs will always provide the latest data once the final customer has given their permission.

In contrast, requesting VHMD is more complicated, as EDA differentiates between HistoricalMeteringData and MeteringData for the RequestedDataType. HistoricalMeteringData is the VHMD that has already been collected and validated, which is all data prior to the date the request is made. MeteringData is the VHMD that has not yet been collected, which is collected and validated after the date of the request (this includes

[9] See https://www.ebutilities.at/prozesse

the date of the request). This differentiation makes requesting VHMD for time ranges that include both the past and the future more challenging, as it requires creating and sending at least two CMRequests. For example, to request all the data for the current year, one would need to send two CMRequest messages, one for HistoricalMeteringData and one for MeteringData. Both HistoricalMeteringData and MeteringData require specifying the meteringInterval, dataFrom, and dataTo of the data that is requested. The meteringInterval defines the resolution of the requested data, which can be either quarter-hourly or hourly. The ability to provide the requested resolution depends on the smart meter of the final customer. If the smart meter is not capable of providing the resolution, the request will be deemed invalid. The dataFrom and dataTo parameters define the time range of the requested data, subject to the aforementioned constraints regarding the past and future. MeteringData also requires the transmissionCycle parameter to be set. The transmissionCycle tells the DSO if it should send the data once it is available (daily) or accumulate the data and send it once a month (monthly).

Figure 25 depicts the CMNotification message. The CMNotification message is used by the DSO to inform the EP of certain events. The meaning of the received message is determined by the messageCode parameter. In all cases, the message contains the cmRequestId of the request that triggered the notification as well as a list of responseCodes that provide additional information about the event.

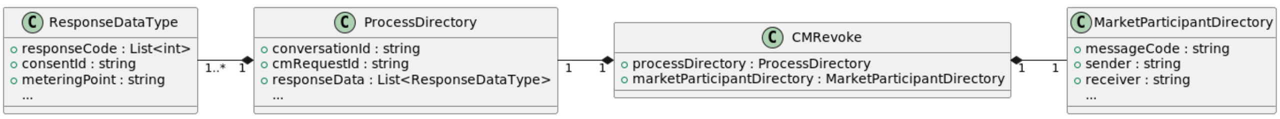


Figure 25: Illustration of the CMNotification message.

For example, receiving a CMNotification message with the message code ABLEHNUNG_CCMO and the responseCode "72" means that the request was automatically rejected by the DSO because the requested data is not available.
Receiving a CMNotification message with the message code ZUSTIMMUNG_CCMO and the responseCode "175" on the other hand means that the customer has accepted the request and the data will be transmitted. In this case the message will also contain the consentId of the permission that was created and the meteringPoint that the final customer has given permission for. This information is required to identify the permission in the termination and revocation processes.

Figure 26: Illustration of a simplified version of the CMRevoke message.

Figure 26 shows a simplified version of the CMRevoke message. As with the CMNotification message, the messageCode parameter determines the meaning of the message as shown by the process diagrams in Figure 20 and Figure 21 in steps 9, 10 and 11. As the name implies, the CMRevoke message is used to terminate or revoke a permission. It must always contain the consentId and metering point of the permission that is to be terminated or revoked. It may also contain a reason for the termination or revocation and the date when the termination will take effect. If no date is provided, the termination will take effect immediately.

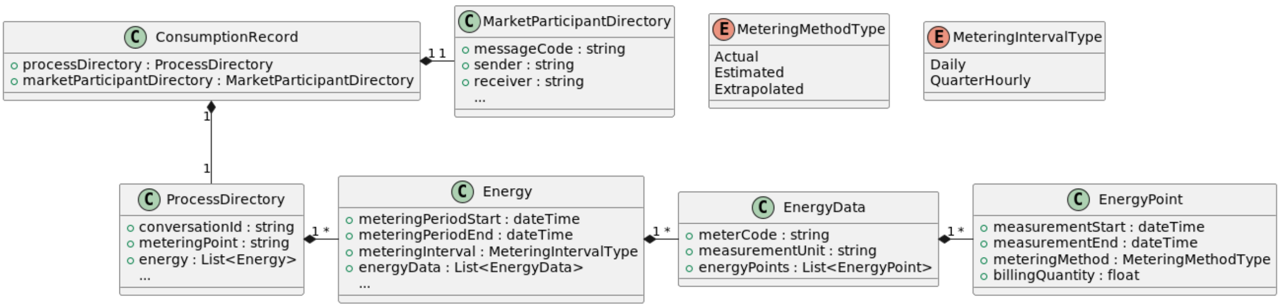


Figure 27: Illustration of a ConsumptionRecord message.

Figure 27 shows a simplified version of the ConsumptionRecord message. A ConsumptionRecord message always contains the metering data of a single metering point represented by the Energy object. The Energy objects contain meta information about the contained data such as but not limited to the time frame, the resolution of the data (meteringInterval), the energy direction and the EnergyData that encapsulates the actual metering data.

The EnergyData object holds information about the metering device such as the meterCode, the measurementUnit and the actual metered values represented as a list of EnergyPoint objects. Each EnergyPoint object contains the actual metered value (billingQuantity), the time information of the measurement and the quality of the measurement (meteringMethod). Although this message is called ConsumptionRecord, the name is misleading, as the message can represent both consumption and production data. The energy direction of the metering data is determined by the meterCode of the energyData objects. The meterCode follows the Object Identification System (OBIS) scheme, which is a schema to express the capabilities of metering devices standardized by IEC 62056-6-1. A metering device could support multiple meterCodes and therefore a ConsumptionRecord message could contain multiple EnergyData objects with different meterCodes for the same metering point.

A typical ConsumptionRecord message contains one Energy object with one EnergyData object and depending on the resolution of the data multiple EnergyPoint objects. The DSOs usually aggregate requests for daily measurements into a single ConsumptionRecord message, while requests for quarter-hourly measurements are sent as one ConsumptionRecord message per requested day in the time range where each message contains 96 EnergyPoint objects.

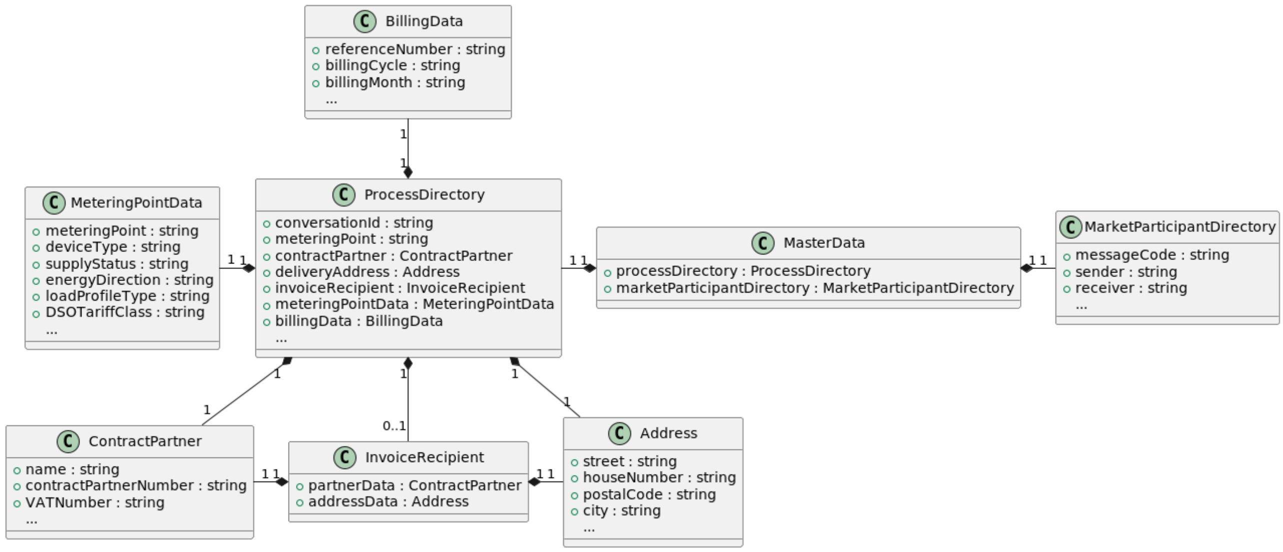


Figure 28: Simplified version of the MasterData message.

Figure 28 shows a simplified version of the MasterData message. The MasterData message contains meta information about the metering point and the final customer. Information about the capabilities of the metering device is represented by a MeteringPointData object. The deliveryAddress stores the physical address of the metering point. The contractPartner identifies the final customer who is associated with the metering point. The billingData refers to information about how the final customer is billed. If the invoice is sent to a different address than the metering point or the contract partner is a company, the invoiceRecipient object will contain the address of the recipient, otherwise, it will not be present.

3.2.4 Setup/Onboarding

For the AS4 communication with EDA, the Ponton X/P Messenger is used. Generally, the region connector for Austria uses the Ponton X/P messenger to communicate with the DSOs. The communication with the Ponton X/P Messenger itself works via their Java API which uses a Websocket connection in the background. As for future solutions, it is considered implementing the AS4 protocol or apply available open-source implementations.

3.2.5 PONTON X/P Messenger

The PONTON X/P Messenger is an essential building block for connecting the EDDIE Framework to the EDA partners. As the usage of Ponton X/P and AS4 protocols is specific to the Austrian data sharing infrastructure, the connection to the EDDIE Framework is implemented as part of the EDA region connector for Austria.

AS4 offers significant advantages for B2B integration including interoperability, secure data transmission, reliability and non-repudiation of messages. These benefits are realized through the exchange of standardized, encrypted, and signed messages. Non-repudiation guarantees that once a message is sent, the sender cannot deny its submission or authenticity, which is crucial for many business processes in the energy sector.

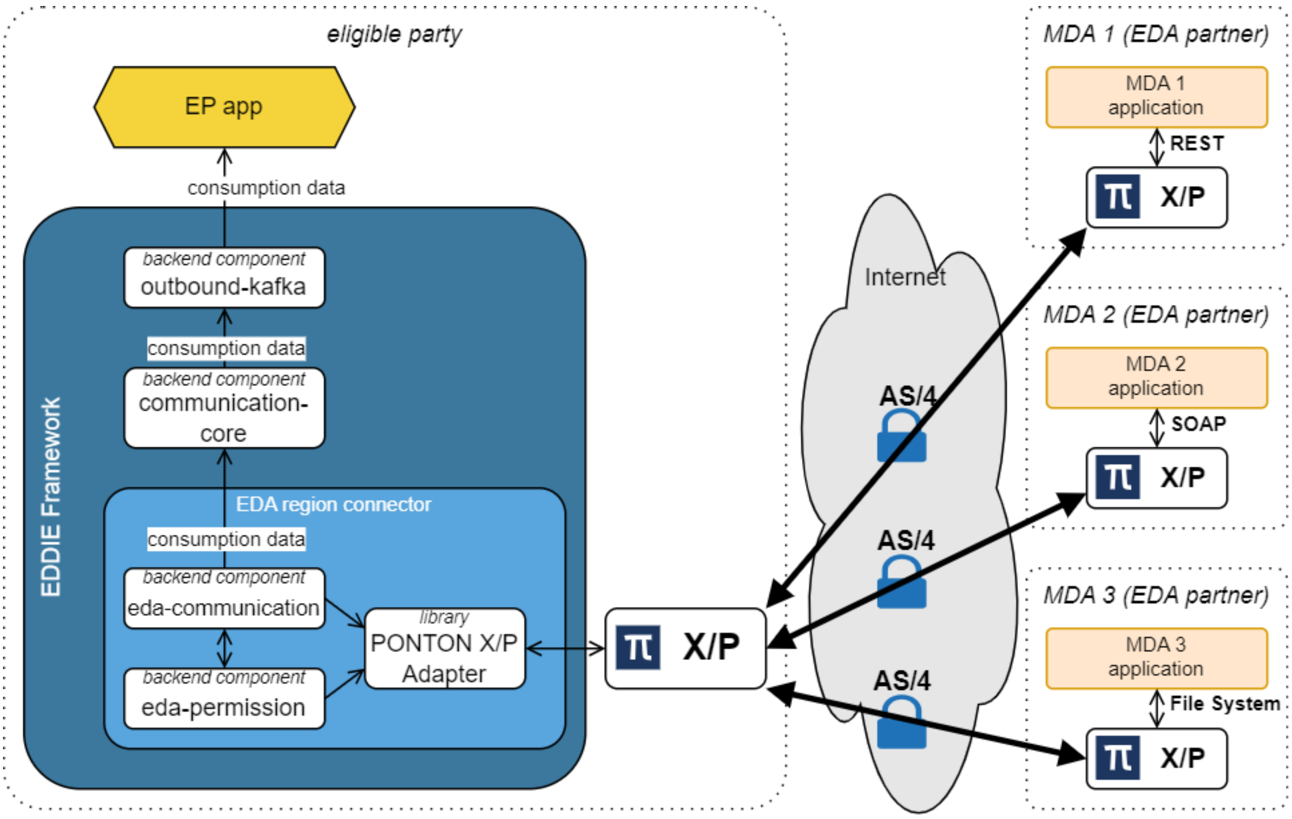


Figure 29: Illustration of the message-based communication via Ponton X/P messenger in AT.

The communication and permission components of the EDA region connector connect to the PONTON X/P messenger using the Adapter library. This handles communication with the Ponton X/P messenger which is deployed as a separate software system on the eligible party's infrastructure.

PONTON X/P fully implements the AS4 standard, incorporating all its benefits. Additionally, PONTON X/P enriches the functionality with certificate management and partner management features.

The Ponton X/P messenger serves as an intermediary by translating AS4-based market communications into formats compatible with the internal software systems used by market participants. These organizations may connect to the Ponton X/P messenger using a variety of protocols, such as REST, SOAP, direct File System access, database access or by employing the Ponton X/P Adapter library.

3.3 Exemplary Message-based Request

As this deliverable is classified as being of type "DEM – Demonstrator, pilot, prototype", this section presents an actual workflow of a message-based data request. For details on the permission façade, the information schema or the details on the region connectors, please be referred to the respective deliverables "D2.2 – Information schema defined" [31], "D3.1 – EDDIE Consent façade" [30] and "D5.1 – Intermediate report on best practices and EU region connectors created" [29].

First, the final customer uses the permission façade and the respective frontend on the EPs website to select the correct country, permission administrator and enters his/her metering point id, and clicks on the button "Connect" as shown in Figure 30.

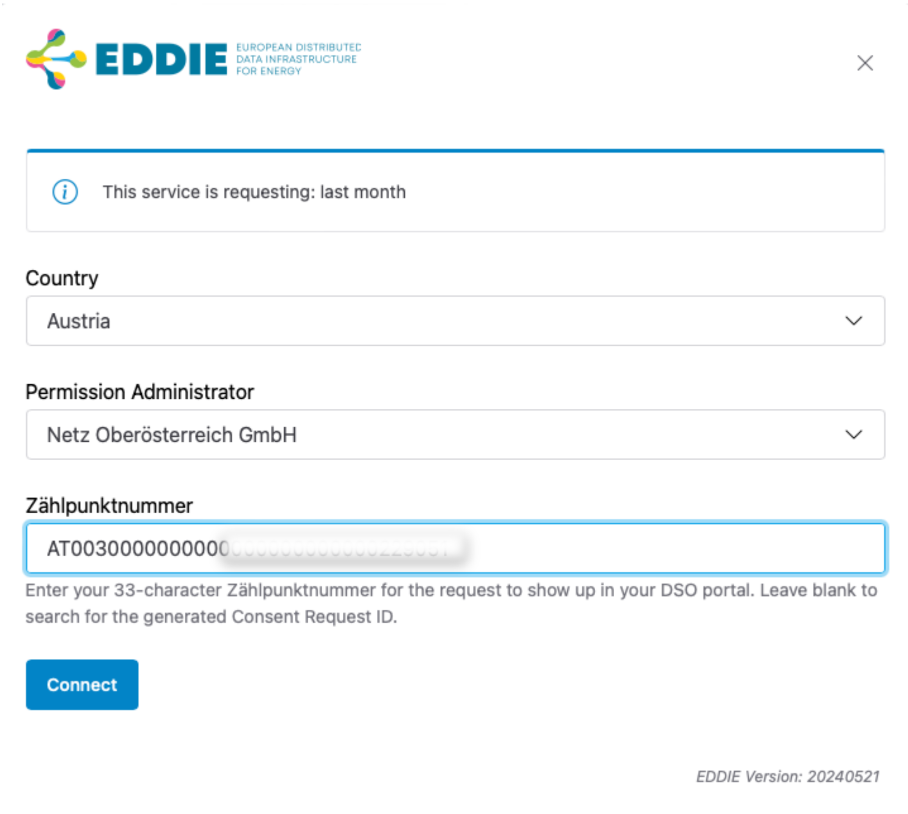


Figure 30: Entering the customer related data.

After the click on the button, the consent request ID is generated and displayed on order to be able to identify this request internally. A corresponding ANFORDERUNG_CCMO message is created and transmitted and the status set to SENT_TO_PERMISSION_ADMINISTRATOR. The information is shown on the EDDIE frontend, as it is depicted in Figure 31.

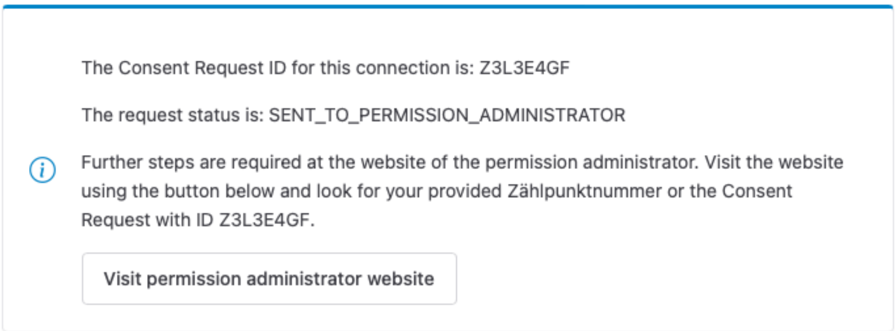
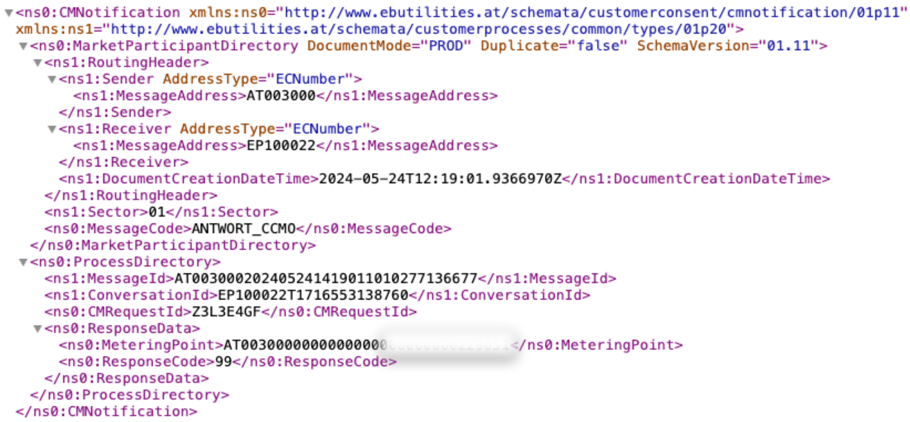


Figure 31: Information of the request on the EDDIE frontend

The corresponding ANTWORT_CCMO message is shown in Figure 32 – please note the matching request ID.



```
<ns0:CMNotification xmlns:ns0="http://www.ebutilities.at/schemata/customerconsent/cmnotification/01p11" xmlns:ns1="http://www.ebutilities.at/schemata/customerprocesses/common/types/01p20">
  <ns0:MarketParticipantDirectory DocumentMode="PROD" Duplicate="false" SchemaVersion="01.11">
    <ns1:RoutingHeader>
      <ns1:Sender AddressType="ECNumber">
        <ns1:MessageAddress>AT003000</ns1:MessageAddress>
      </ns1:Sender>
      <ns1:Receiver AddressType="ECNumber">
        <ns1:MessageAddress>EP100022</ns1:MessageAddress>
      </ns1:Receiver>
      <ns1:DocumentCreationDateTime>2024-05-24T12:19:01.9366970Z</ns1:DocumentCreationDateTime>
    </ns1:RoutingHeader>
    <ns1:Sector>01</ns1:Sector>
    <ns0:MessageCode>ANTWORT_CCMO</ns0:MessageCode>
  </ns0:MarketParticipantDirectory>
  <ns0:ProcessDirectory>
    <ns1:MessageId>AT0030002024052414190110102771366677</ns1:MessageId>
    <ns1:ConversationId>EP100022T1716553138760</ns1:ConversationId>
    <ns0:CMRequestId>Z3L3E4GF</ns0:CMRequestId>
    <ns0:ResponseData>
      <ns0:MeteringPoint>AT00300000000000000</ns0:MeteringPoint>
      <ns0:ResponseCode>99</ns0:ResponseCode>
    </ns0:ResponseData>
  </ns0:ProcessDirectory>
</ns0:CMNotification>
```


Figure 32: Generated ANTORT_CCMO message.

The final customer is now able to accept or decline the request by logging into his/her DSO portal. In our exemplary case, this is "Netz OÖ"[10] – this request is shown in Figure 33.

[10] See https://www.netzooe.at

Offene Anfragen (1)

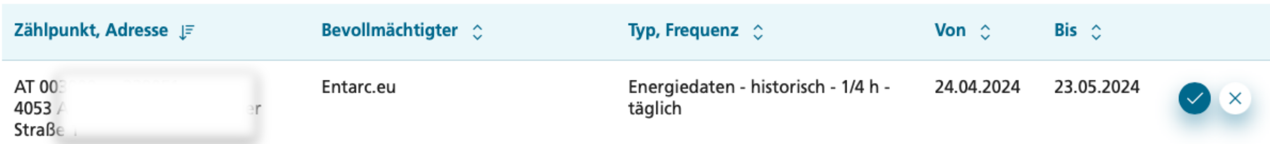


Figure 33: Permission request in the customers DSO portal.

After the final customer clicks on "accept", the final customer must accept the terms (see Figure 34) and after that, the permission is granted.

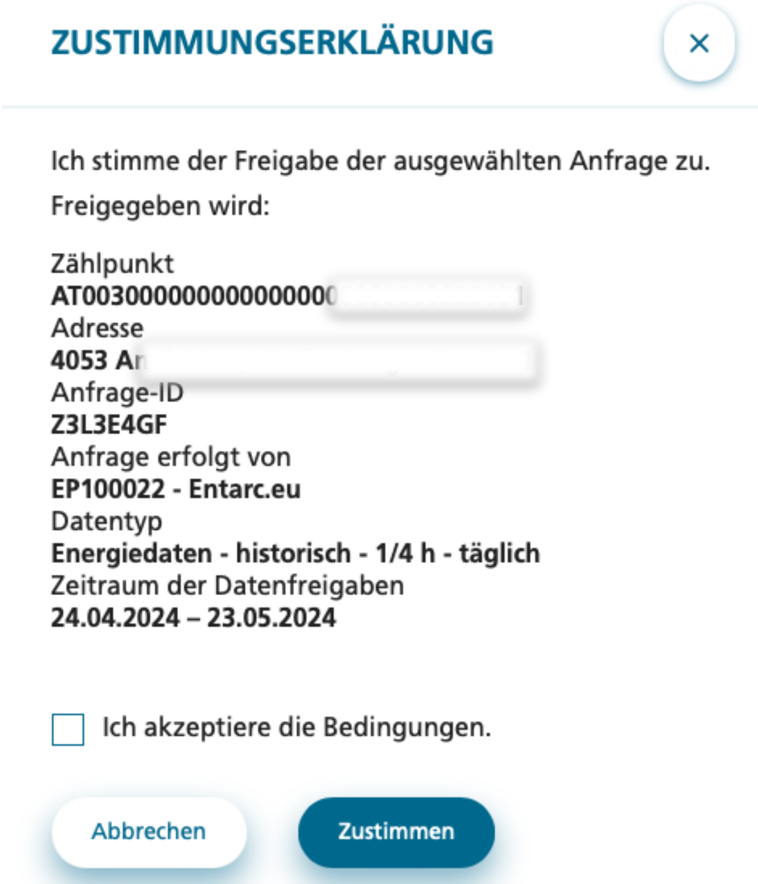


Figure 34: Declaration of consent in the user's DSO portal.

After granting the permission, the status is changed to fulfilled and the request is completed. This information is shown on the EDDIE frontend as depicted in Figure 35.

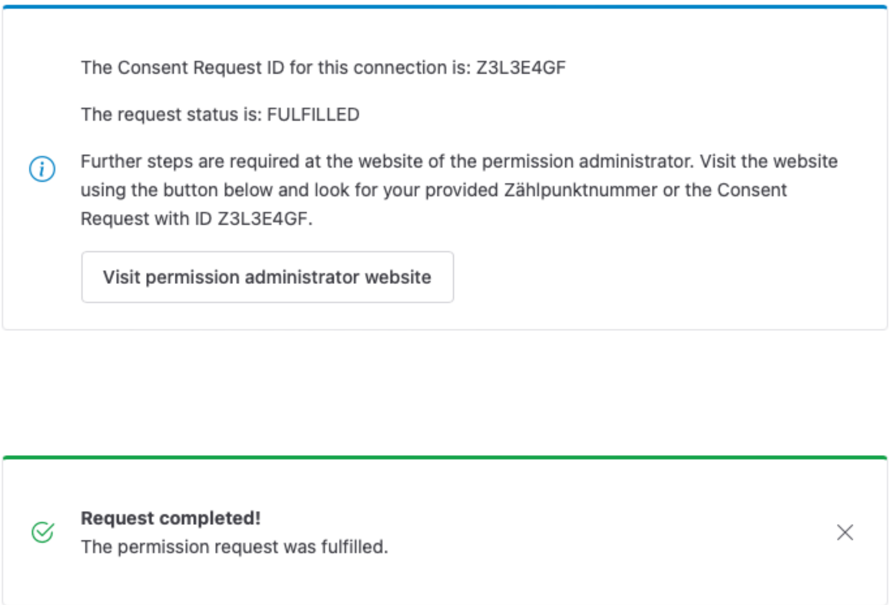


Figure 35: Illustration of the information of a fulfilled request.

Internally, this is represented by an AS4 message with message code ZUSTIMMUNG_CCMO as shown in Figure 36.


```
▼<ns0:CMNotification xmlns:ns0="http://www.ebutilities.at/schemata/customerconsent/cmnotification/01p11"
 xmlns:ns1="http://www.ebutilities.at/schemata/customerprocesses/common/types/01p20">
  ▼<ns0:MarketParticipantDirectory DocumentMode="PROD" Duplicate="false" SchemaVersion="01.11">
    ▼<ns1:RoutingHeader>
      ▼<ns1:Sender AddressType="ECNumber">
          <ns1:MessageAddress>AT003000</ns1:MessageAddress>
        </ns1:Sender>
      ▼<ns1:Receiver AddressType="ECNumber">
          <ns1:MessageAddress>EP100022</ns1:MessageAddress>
        </ns1:Receiver>
        <ns1:DocumentCreationDateTime>2024-05-24T12:32:35.3651740Z</ns1:DocumentCreationDateTime>
      </ns1:RoutingHeader>
      <ns1:Sector>01</ns1:Sector>
      <ns0:MessageCode>ZUSTIMMUNG_CCMO</ns0:MessageCode>
    </ns0:MarketParticipantDirectory>
  ▼<ns0:ProcessDirectory>
      <ns1:MessageId>AT003000202405241419011010277136783</ns1:MessageId>
      <ns1:ConversationId>EP100022T1716553138760</ns1:ConversationId>
      <ns0:CMRequestId>Z3L3E4GF</ns0:CMRequestId>
    ▼<ns0:ResponseData>
        <ns0:ConsentId>AT00300020240524143234717Z3L3E4GF</ns0:ConsentId>
        <ns0:MeteringPoint>AT0030000000[illegible]s0:MeteringPoint>
        <ns0:ResponseCode>175</ns0:ResponseCode>
      </ns0:ResponseData>
    </ns0:ProcessDirectory>
</ns0:CMNotification>
```


Figure 36: Illustration of a ZUSTIMMUNG_CCMO message.

Since this exemplary request was about validated historical data, the corresponding AS4 message can be immediately sent. A clipping of this message is shown in Figure 37.

```xml
▼<ns0:ConsumptionRecord
 xmlns:ns0="http://www.ebutilities.at/schemata/customerprocesses/consumptionrecord/01p40"
 xmlns:ns1="http://www.ebutilities.at/schemata/customerprocesses/common/types/01p20">
  ▼<ns0:MarketParticipantDirectory DocumentMode="PROD" Duplicate="false" SchemaVersion="01.40">
    ▼<ns1:RoutingHeader>
      ▼<ns1:Sender AddressType="ECNumber">
          <ns1:MessageAddress>AT003000</ns1:MessageAddress>
        </ns1:Sender>
      ▼<ns1:Receiver AddressType="ECNumber">
          <ns1:MessageAddress>EP100022</ns1:MessageAddress>
        </ns1:Receiver>
        <ns1:DocumentCreationDateTime>2024-05-24T12:32:35.8600290Z</ns1:DocumentCreationDateTime>
      </ns1:RoutingHeader>
      <ns1:Sector>01</ns1:Sector>
      <ns0:MessageCode>DATEN_CRMSG</ns0:MessageCode>
    </ns0:MarketParticipantDirectory>
  ▼<ns0:ProcessDirectory>
      <ns1:MessageId>AT003000202405241432354150277136784</ns1:MessageId>
      <ns1:ConversationId>EP100022T1716553138760</ns1:ConversationId>
      <ns1:ProcessDate>2024-05-24</ns1:ProcessDate>
      <ns1:MeteringPoint>AT0030000000[redacted]/ns1:MeteringPoint>
    ▼<ns0:Energy>
        <ns0:MeteringReason>00</ns0:MeteringReason>
        <ns0:MeteringPeriodStart>2024-04-24T00:00:00+02:00</ns0:MeteringPeriodStart>
        <ns0:MeteringPeriodEnd>2024-05-01T00:00:00+02:00</ns0:MeteringPeriodEnd>
        <ns0:MeteringIntervall>QH</ns0:MeteringIntervall>
        <ns0:NumberOfMeteringIntervall>672</ns0:NumberOfMeteringIntervall>
      ▼<ns0:EnergyData MeterCode="1-1:1.9.0 P.01" UOM="KWH">
        ▼<ns0:EP>
            <ns0:DTF>2024-04-24T00:00:00+02:00</ns0:DTF>
            <ns0:DTT>2024-04-24T00:15:00+02:00</ns0:DTT>
            <ns0:MM>L1</ns0:MM>
            <ns0:BQ>0.002500</ns0:BQ>
          </ns0:EP>
        ▼<ns0:EP>
            <ns0:DTF>2024-04-24T00:15:00+02:00</ns0:DTF>
            <ns0:DTT>2024-04-24T00:30:00+02:00</ns0:DTT>
            <ns0:MM>L1</ns0:MM>
            <ns0:BQ>0.014750</ns0:BQ>
          </ns0:EP>
        ▼<ns0:EP>
            <ns0:DTF>2024-04-24T00:30:00+02:00</ns0:DTF>
            <ns0:DTT>2024-04-24T00:45:00+02:00</ns0:DTT>
            <ns0:MM>L1</ns0:MM>
            <ns0:BQ>0.001000</ns0:BQ>
          </ns0:EP>
        ▼<ns0:EP>
            <ns0:DTF>2024-04-24T00:45:00+02:00</ns0:DTF>
            <ns0:DTT>2024-04-24T01:00:00+02:00</ns0:DTT>
```


Figure 37: Clipping of a DATEN_CRMSG message.

Additionally, the complete message protocol for this request can be seen in the Ponton X/P message backend, as shown in Figure 38. Starting with the first message 16407 – ANFORDERUNG_CCMO – to message 16415 – DATEN_CRMSG – the whole bunch of AS4 messages can be tracked.

OK	16415	24.05.2024 14:32:37.672	AT003000202405241432354150277136785@energieag.at	DATEN_CRMSG	AT003000 ENERGIE-AG_NETZ	EP100022	Inbound
OK	16413	24.05.2024 14:32:37.540	AT003000202405241432354150277136784@energieag.at	DATEN_CRMSG	AT003000 ENERGIE-AG_NETZ	EP100022	Inbound
OK	16411	24.05.2024 14:32:37.320	AT003000202405241419011010277136783@energieag.at	ZUSTIMMUNG_CCMO	AT003000 ENERGIE-AG_NETZ	EP100022	Inbound
OK	16409	24.05.2024 14:19:02.685	AT003000202405241419011010277136677@energieag.at	ANTWORT_CCMO	AT003000 ENERGIE-AG_NETZ	EP100022	Inbound
OK	16407	24.05.2024 14:18:58.791	MID-1712861825529@ponton.xp	ANFORDERUNG_CCMO	EP100022	AT003000 ENERGIE-AG_NETZ	Outbound

Figure 38: Message log in the Ponton X/P backend.

4 Outlook & Roadmap

This deliverable mainly dealt with tasks 4.1 ("Communication layer design") and 4.2 ("Message-based communication") of the EDDIE grant agreement. Nevertheless, other ways of communication need to be incorporated to cover the broad spectrum of different European solutions. As foreseen in the grant agreement, two major communication methods are covered, namely REST API-based communication and streaming-based communication. At the time of writing this current document, both tasks have already started and these communication methods have already been implemented to some extent for country-specific region connectors. For example, Figure 39 and Figure 40 show REST clients and REST API implementations for the French region connector incorporating Enedis.

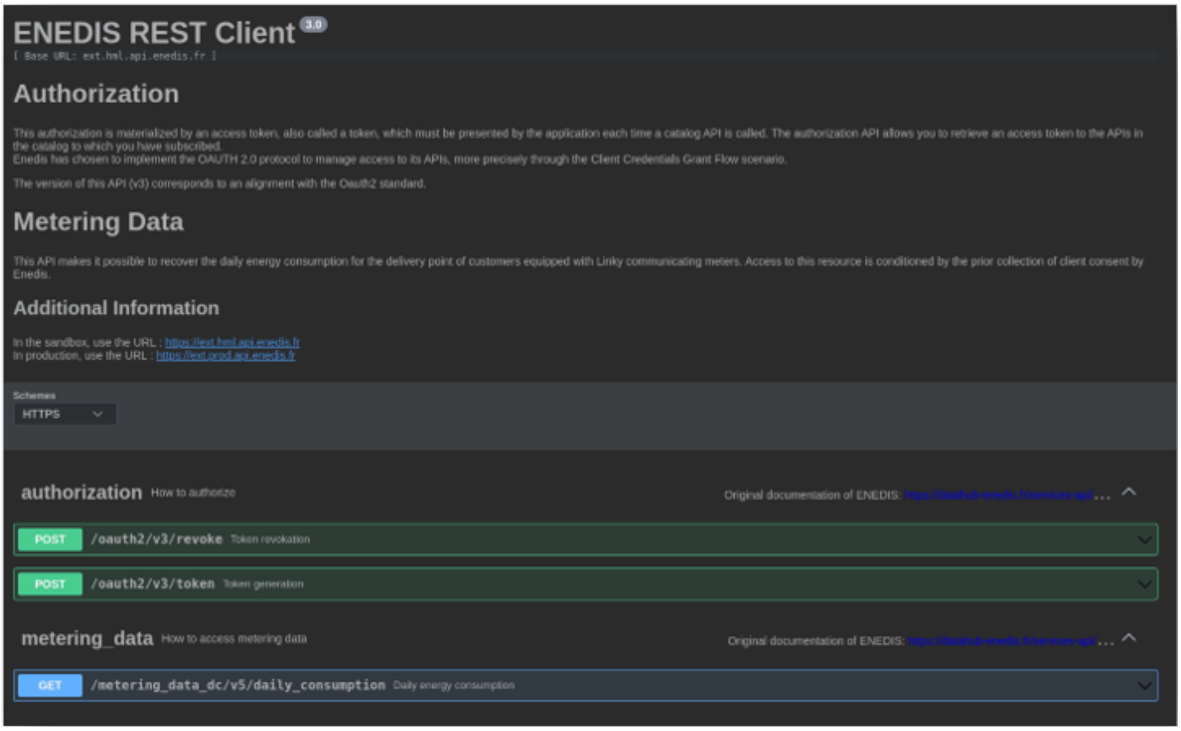


Figure 39: Illustration of the French Enedis REST Client.

Also, streaming-based communication utilizing Apache Kafka[11] as streaming platform is already built in the EDDIE framework. Granted request and exchanged data can be

[11] See https://kafka.apache.org

consumed via Kafka. Additionally, we are also considering other protocols (like MQTT) and generally different communication messages.

```java
public static void main(String[] args) {
    final String filename = "region-connectors/region-connector-fr-enedis/bearer.txt";
    final File file = new File(filename);
    final Main main = new Main();

    try {
        if (!file.exists()) {
            InlineResponse200 inlineResponse200 = main.postToken(file);
            System.out.println(inlineResponse200);
        }

        InlineResponse2001 inlineResponse2001 = main.getMeteringData(file);
        System.out.println(inlineResponse2001);
    } catch (ApiException e) {
        if (e.getCode() == HttpURLConnection.HTTP_UNAUTHORIZED) {
            try {
                main.postToken(file);
                InlineResponse2001 inlineResponse2001 = main.getMeteringData(file);
                System.out.println(inlineResponse2001);
            } catch (ApiException e2) {
                showApiExceptionError(e2);
            } catch (IOException ex) {
                System.out.println("An error occurred.");
                e.printStackTrace();
            }
        } else {
            showApiExceptionError(e);
        }
    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
}
```

```
class InlineResponse2001 {
    meterReading: class InlineResponse2001MeterReading {
        usagePointId: 11453290002823
        start: 2023-05-08
        end: 2023-05-11
        quality: null
        readingType: class InlineResponse2001MeterReadingReadingType {
            measurementKind: energy
            measuringPeriod: P1D
            unit: Wh
            aggregate: sum
        }
        intervalReading: [class InlineResponse2001MeterReadingIntervalReading {
            value: 17060
            date: 2023-05-08
        }, class InlineResponse2001MeterReadingIntervalReading {
            value: 16708
            date: 2023-05-09
        }, class InlineResponse2001MeterReadingIntervalReading {
            value: 28080
            date: 2023-05-10
        }]
    }
}
```

Figure 40: First implementation attempts against the Enedis REST API.

Details on further approaches for the interoperable communication layer within EDDIE's work-package 4 will be covered in deliverable "D4.2 – Three different approaches for the communication layer are implemented". This deliverable is due in month 30 (which would be June 2025).

5 Summary & Conclusion

The EDDIE project's deliverable D4.1, "Message-Based Communication Implemented," signifies a pivotal development in establishing an interoperable communication layer for energy data exchange across Europe. The deliverable, part of Work Package 4, addresses the complex and varied methodologies of data exchange employed by different EU member states, aiming to ensure interoperability, security, and reliability.

The primary objectives of D4.1 include developing a communication layer that bridges the gap between different national implementations of EU-regulated processes, with a focus on harmonizing communication protocols and data formats. This involves creating a uniform architecture to enable seamless integration across diverse business processes while ensuring compliance with both current and future standards. Additionally, aligning the communication layer with stakeholder requirements is crucial to facilitate smooth business integration and iterative refinement throughout the implementation phase.

The deliverable's core contribution centers around the message-based communication system, particularly exemplified through the Austrian case. This system utilizes the AS4 messaging protocol for data exchange. Detailed processes for managing customer consent and data transmission are illustrated through Business Process Model and Notation (BPMN) diagrams, demonstrating the practical application of the communication layer.

A key component of the EDDIE framework is the region connector architecture, which serves as a gateway for data flow into the broader framework. Middleware systems like the Ponton X/P messenger are employed for AS4 communication in Austria, ensuring consistent and secure interactions despite regional differences in data-sharing infrastructure.

Significant achievements have been made, including the successful implementation of message-based communication. The technological integration within the EDDIE framework involves various communication methods, such as REST API and streaming-based communication, to address the wide range of European data exchange solutions. The framework is designed with extensibility in mind, allowing for the addition and support of other messaging solutions.

Looking forward, the deliverable outlines a comprehensive outlook and roadmap. This includes the incorporation of other communication methods to cover the broad spectrum of different European solutions. Future work will be detailed in Deliverable D4.2, which is set to be completed by June 2025. This forthcoming deliverable will present three different approaches to the communication layer, including further developments in REST API-based communication and streaming-based communication using platforms like Apache Kafka.

In conclusion, Deliverable D4.1 represents a critical step towards a unified, interoperable communication layer for European energy data exchange. By addressing the diverse methodologies and aligning with stakeholder needs, the EDDIE project aims to enhance the efficiency, security, and interoperability of energy data exchange across Europe. The work done in this deliverable lays a robust foundation for future developments, ensuring the EDDIE framework remains adaptable and resilient to evolving technological and regulatory landscapes.

6 REFERENCES

[1] Vasileios Karagiannis et al. "A Framework for Enabling Cloud Services to Leverage Energy Data". In: 2023 IEEE International Conference on Cloud Engineering (IC2E). 2023 IEEE International Conference on Cloud Engineering (IC2E). Sept. 2023, pp. 43–50.

[2] Data Access Summary | ACEEE. url: https://database.aceee.org/state/data-access.

[3] Green Button. Energy.gov. url: https://www.energy.gov/data/green-button.

[4] European University Institute et al. The EU Clean Energy Package: (2020 Ed.) Publications Office of the European Union, 2020. url: https://data.europa.eu/doi/10.2870/58299.

[5] Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on Common Rules for the Internal Market for Electricity and Amending Directive 2012/27/EU (Recast) (Text with EEA Relevance.) June 5, 2019. url: http://data.europa.eu/eli/dir/2019/944/oj/eng.

[6] Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on Harmonised Rules on Fair Access to and Use of Data and Amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act). Dec. 13, 2023. url: http://data.europa.eu/eli/reg/2023/2854/oj/eng.

[7] Commission Implementing Regulation (EU) 2023/1162 of 6 June 2023 on Interoperability Requirements and Non-Discriminatory and Transparent Procedures for Access to Metering and Consumption Data (Text with EEA Relevance). June 6, 2023. url: http://data.europa.eu/eli/reg_impl/2023/1162/oj/eng.

[8] A Review of Current and Future Data Management Models, CEER Report

[9] "DATA EXCHANGE IN ELECTRIC POWER SYSTEMS: European State of Play and Perspectives".

[10] Smart Grid Development / Elering. url: https://elering.ee/en/smart-grid-development.

[11] Portale Consumi / Homepage. url: https://www.consumienergia.it/portaleConsumi/.

[12] Pol Van Aubel and Erik Poll. "Smart Metering in the Netherlands: What, How, and Why". International Journal of Electrical Power & Energy Systems 109 (July 2019), pp. 719–725. url: https://linkinghub.elsevier.com/retrieve/pii/S0142061518335142.

[13] Datahub. Fingrid. url: https://www.fingrid.fi/en/electricity-market/datahub/.

[14] Om Elhub. Elhub. url: https://elhub.no/om-elhub/.

[15] The Future Is Electric. Statnett. Mar. 21, 2024. url: https://www.statnett.no/en/.

[16] Elhub Messaging Interface (EMIF). url: https://dok.elhub.no/Elhubstandard/elhub-messaging-interface-emif.

[17] Home / Atrias. url: https://www.atrias.be/.

[18] My Fluvius / Fluvius. url: https://www.fluvius.be/nl/veelgestelde-vragen/mijn-fluvius.

[19] Energiedienstverleners / Fluvius. url: https://partner.fluvius.be/nl/energiedienstverleners.
[20] ISO 9735-11 (EDIFACT).
[21] Marktakteure / MaStR. url: https://www.marktstammdatenregister.de/MaStR/Akteur/Marktakteur/IndexOeffentlich.
[22] Netz und Netzbetreiber in Österreich. E-Control. url: https://www.e-control.at/konsumenten/netz-und-netzbetreiber-in-oesterreich.
[23] Michael Merz. Ponton X/P - Secure Data Communication and B2B Integration. PONTON. Sept. 28, 2022. url: https://www.ponton.de/ponton-xp.
[24] Registrierung - EbUtilities. url: https://www.ebutilities.at/.
[25] "AS4 Profile of ebMS 3.0 Version 1.0" (2013).
[26] EDA / Anbindungsarten. EDA | Energiewirtschaftlicher Datenaustausch. url: https://www.eda.at/anbindungsarten.
[27] Datadis. url: https://datadis.es/about-us.
[28] Energinet. "THE DANISH ELECTRICITY RETAIL MARKET". url: https://en.energinet.dk/media/qouleeb5/danish-electricity-retail-market.pdf.
[29] EDDIE Deliverable 5.1 – Intermediate report on best-practices and EU region connectors created, doi: 10.5281/zenodo.11638341.
[30] EDDIE Deliverable 3.1 - EDDIE Consent Façade, doi: 10.5281/zenodo.11638268
[31] EDDIE Deliverable 2.2 - Information schema defined, doi: 10.5281/zenodo.11565603