



MARKET ENABLING INTERFACE TO UNLOCK FLEXIBILITY SOLUTIONS FOR
COST-EFFECTIVE MANAGEMENT OF SMARTER DISTRIBUTION GRIDS

Deliverable 9.3

PL - Validation report and results of UMEI concept test



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 864334

Document

D9.3 PL - Validation report and results of UMEI concept test

Dissemination level

PU Public

Author(s)	Institution	Contact (<i>e-mail, phone</i>)
Mirosław Matuszewicz	ENERGA	miroslaw.matuszewicz@energa-operator.pl
Michał Konopiński	MIKRONIKA	carlospedro.marques@e-redes.pt
Gesa Milzer	NODES AS	gesa.milzer@nodesmarket.com

Key word	Demonstration
Due Delivery Date	2023/11/30
Date of Delivery	2023/12/11

Document version	Date	Change
1.0	2023/08/30	1st draft
1.1	2023/12/06	Version for revision
2.0	2023/12/08	Final version after revision

Reviewers		email	Validation date
MITNETZ	David Brummund	david.brummund@mitnetz-strom.de	2023/12/07
E-REDES	Pedro Marques	carlospedro.marques@e-redes.pt	2023/12/07

Table of Contents

LIST OF FIGURES.....	4
ABBREVIATIONS.....	5
EXECUTIVE SUMMARY.....	6
1. INTRODUCTION.....	7
1.1 BACKGROUND.....	7
1.2 WP9 OBJECTIVES.....	7
1.3 THE UMEI.....	8
2 IMPLEMENTATION OF THE UMEI	10
3 TECHNICAL DEPLOYMENT	12
3.1 TECHNOLOGICAL CHOICES AND ARCHITECTURE	12
3.2 AUTHENTICATION MECHANISMS.....	13
4 USAGE OF THE UMEI	15
4.1 EXAMPLE API CALLS.....	15
4.1.1 List all markets.....	15
4.1.2 Search for portfolios for specific gridNode	16
4.1.3 Create BaselineInterval	16
4.1.4 Get the BaselineInterval list.....	16
4.1.5 Sending the meter reading.....	17
4.1.6 Retrieving meter readings for a specific portfolio for a given period of time	18
4.1.7 Creating an order	18
4.1.8 Downloading a list of orders for a specified grid node, market, regulation type and specified time range	19
4.1.9 Reading public information about orders for a given market, portfolio, type of regulation in a given period of time.....	20
4.1.10 Changing order details, changing quantity and unit price for a given order.....	21
4.1.11 Deleting the order	21
4.1.12 Reading of transactions for the indicated gridNode, market, portfolio in a given period of time	22
4.1.13 Deleting the specified BaselineInterval	22
4.2 API PERFORMANCE TEST	23
4.3 TRADING RESULTS.....	24
5. CONCLUSIONS	24
REFERENCES	25

List of Figures

FIGURE 1- HIGH-LEVEL DEVELOPMENT METHODOLOGY	8
FIGURE 2 - EUNIVERSAL REFERENCE ARCHITECTURE	9
FIGURE 3 - UMEI GROUPS.....	10
FIGURE 4 - FLEXIBILITY PROCESS STEPS COVERED BY THE UMEI.....	11
FIGURE 5 – DSO ARCHITECTURE	12
FIGURE 6 – BASIC API AUTHENTICAITON MODES (WALLARM, 2023).....	14
FIGURE 7 – RESULTS OF THE RESPONSE TIME	23

List of Tables

TABLE 2.1: UMEI OVERALL STRUCTURE.....	11
TABLE 4.1: TRADING RESULTS FROM NODES MARKET PLATFORM; TRADING PERIOD 2023	24

Abbreviations

AMS	Active Management System
BUC	Business Use Case
D	Deliverable
DER	Distributed Energy Resources
DG	Distributed Generation
DLR	Dynamic Line Rating
DSO	Distributed System Operator
FMO	Flexibility Market Operator
FSP	Flexibility Service Provider
LV	Low Voltage
MV	Medium Voltage
OLTC	Onload Tap Changer
PLC	Power Line Communication
PV	Photovoltaic
RES	Renewable energy source
SUC	System Use Case
UMEI	Universal Market Enabling Interface
WP	Work Package

Executive Summary

Under the European H2020 program, the EUniversal Project has the main objective to foster the universal access of system operators to the available flexibility, mainly provided by Distributed Energy Resources (DER), through the interaction with new Flexibility Markets and innovative services. With the development of solutions and services that allow the massive integration of Distributed Generation (DG), energy storage, and the active participation of consumers, the project aims to tailor the concept of the Universal Market Enabling Interface (UMEI). The UMEI will look to overcome the limitations that Distribution System Operators (DSOs), experience in the use of flexibilities, addressing the interlinking of electricity markets with active system management.

The EUniversal project aims to develop a universal approach to the use of flexibility by DSOs and their interaction with the new flexibility markets, enabled through the development of the UMEI. The UMEI has materialized in the conceptual architecture design and the implementation of a standard, agnostic, adaptable, and modular combination of different APIs to link DSOs and market parties with flexibility market platforms, in coordination with other flexibility users. This approach allows distributed communication without the need for a central hub.

The UMEI consists of publicly available APIs, allowing any stakeholder to adopt them or to develop new APIs concerning new services while complying with the UMEI interface specification.

This deliverable aims to describe the implementation of the UMEI in the demonstration held in Poland.

1. Introduction

1.1 Background

The electrical system historically relied upon a set of “implicit services”, provided by classical generation plants. Assuming the future scenarios and the upcoming perspectives, the availability of resources that provide these classical types of services will be significantly reduced. The current outlook of electrical systems shows a growing trend towards the incorporation of DER in the networks. Consequently, the introduction of new ancillary services and explicit services (replacing the previous implicit ones) turns out to be an essential requirement to assure the safe management of the electrical system. Addressing such new services strictly follows the evolution and the integration of multiple electricity markets, to stress the new services needed in the forthcoming panorama, whilst promoting the participation of new flexibility resources in the markets, finally leading these markets to integration at the European level.

The constant evolution of the electricity networks associated with electricity markets' structures, follows the advances in promoting renewables, and, through this, new participants are entering the electricity markets, such as aggregators, Flexibility Service Providers (FSP), Balance Responsible Parties (BRP), among many others.

Many projects and conceptual initiatives have been proposed to improve DER integration as active flexibility providers in local (oriented to distribution grid) and system-wide (oriented to transmission grid) services to contribute to a more efficient operation of the system.

The EUniversal project aims to develop a universal approach to the use of flexibility by DSOs and their interaction with the new flexibility markets, enabled through the development of the concept of the Universal Market Enabling Interface (UMEI) – a unique approach to foster interoperability across Europe.

The need for flexibility provision that is fulfilled with distributed resources and the Renewable Energy Source (RES) that are also connected mostly to the distribution grid are clear signs of the decentralization of the energy system. This leads to the need for enhanced coordination schemes to integrate the decentralized actors and the UMEI aims to be part of the enhanced coordination schemes.

Therefore, within the EUniversal project, a Universal Market Enabling Interface (UMEI) has been developed to overcome current barriers between different systems and to facilitate the use of flexibility services and interlink DSO's active system management with flexibility markets. A set of market-oriented flexibility services from Distributed Energy Resources (DER) will be implemented to serve DSO's needs cost-effectively, supporting the energy transition.

1.2 WP9 Objectives

The operative objective of WP9 is to validate the Universal market Enabling Interface (UMEI) and the developed tools in different contexts and scenarios made available in the Polish Demo. It assesses flexibility for distribution grids and the market capacity to provide new services to the DSO (*Figure 1*).

The Polish demo has tested the provision of market-based flexibilities from customers and producers of the MV grid via the UMEI and the integration for an improved and smarter distribution grid operation. As the first step, establishing the estimation and forecast of the grid state from the chosen

MV grid has led to enhanced grid observability. This has helped aggregate and predict the flexibility potential in the MV grid.

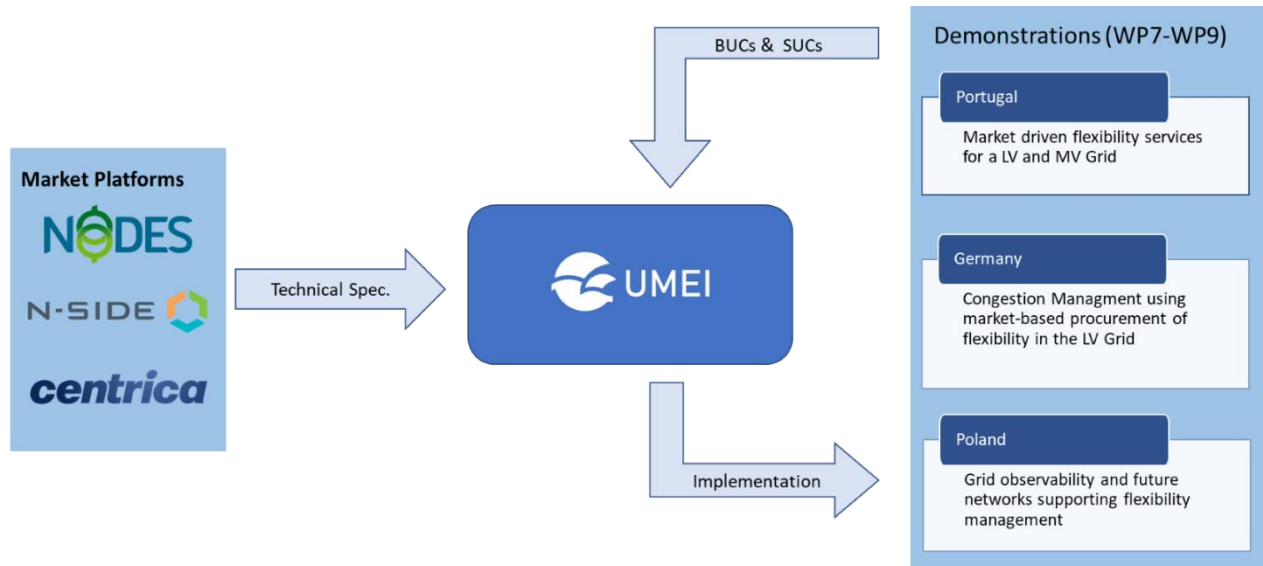


Figure 1- High-level development methodology

1.3 The UMEI

The UMEI is based on the conceptual architecture design and the implementation of a standard, agnostic, adaptable, and modular combination of different APIs to link DSOs and market parties with flexibility market platforms, in coordination with other flexibility users. This approach allows distributed communication without the need for a central hub.

The development of the UMEI was inserted in the workstream of WP2, which has several objectives, namely the definition of flexibility services, the definition of global architecture for the project through the several developed business and system use cases, the and the technical development of the UMEI. Hence, the UMEI is the materialization of the whole process which strengthens the cooperation between the several market parties – DSOs, FMOs and FSPs – to enable the provisioning of the previously defined flexibility services, represented in *Figure 2*.

Through this distributed architecture, in which every party is responsible for performing the necessary setup that is needed for the whole system to operate, several challenges are also posed. The UMEI does not include any central registries, databases, or instances of applications, so all the data which is exchanged resorting to this tool must be kept at the origin and/or the destination, being each party responsible for ensuring the good handling of the data.

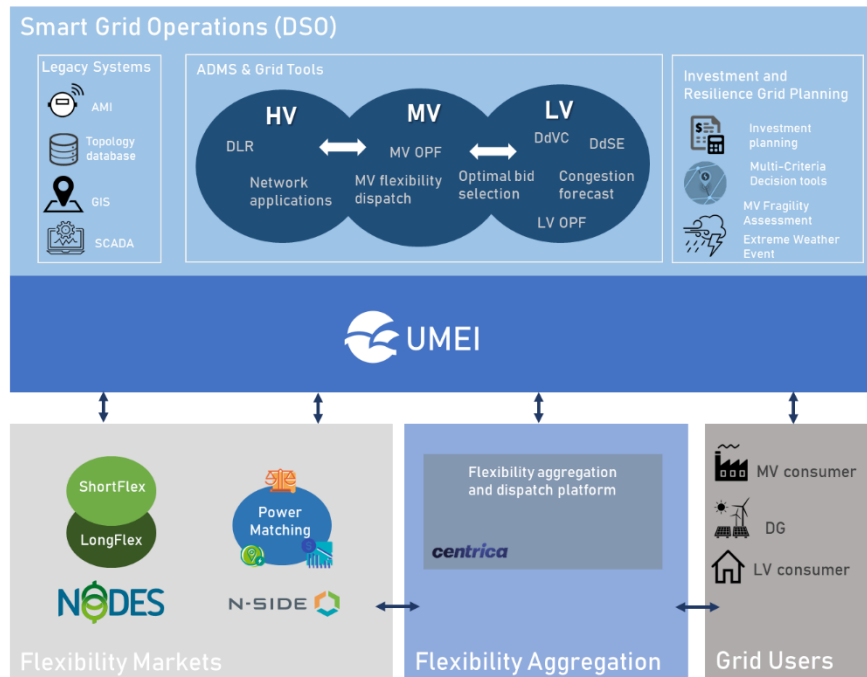


Figure 2 - EUniversal Reference Architecture

The UMEI is composed of a set of APIs, organized in several functional groups, as repress *Figure 3*, which allows market participants to retrieve and send information to the Flexibility Market Operators (FMO).

The API interface is divided into several functional groups. The division follows the natural chronological and functional order of operations:

- Prequalification phase – *not covered by this version of the UMEI. In this phase, basic master data like portfolios, grid nodes, etc are set up*
- Pre-trading phase
 - Defining baselines, the expected power usage, for portfolios
 - Defining flexibility zones – the expected demand for flexibility
 - Extracting this information, along with market and portfolio information, for use in the next phase
- Trading phase
 - Posting orders, reading orders, and trades
- Post-trading phase
 - Providing meter readings for assets used in trades
 - Settlement - *not covered by this version of the UMEI.*

Each group is composed of a set of APIs which allow for CRUD (Create, Read, Update and Delete) operations to be performed on the resources registered on the market platform for flexibility trading. For instance, with the order group it is possible to submit new buy and sell orders, coming from the FSP and DSO side, but also to fetch, eliminate and update orders already submitted before the clearing of the market occurs.

Flexibility Zones	Portfolio	Baseline	Market	Order	Trade	Meter Reading
Flexibility Zones	Manage Portfolios	Managing portfolio baselines	List All Markets	Manage Market Orders	List Market Trades	Manage Meter Readings
Used by DSOs to define specific flexibility areas, composed by a set of portfolios.	Used by FSPs to submit and manage portfolio on the market.	Used by FSPs to manage baselines on the market platform.	Used by market participants to retrieve the available markets.	Used by the DSOs and FSPs to execute orders' related operations in the market platform.	Used by market participants to retrieve the market trades.	Used by the DSOs to manage metering data submission both to the FMO and the FSP.

Figure 3 - UMEI Groups

The UMEI development at this stage focuses mainly on technical and operational requirements on each stakeholder side. The registration and pre-qualification phases were therefore not foreseen to be developed and tested within the scope of the project. The payment processes included in the settlement phase will also not be covered by this specification. Certainly, the successful continuation of the UMEI will require further developments regarding registration and pre-qualification as well as validation and settlement. Anyhow, EUniversal as a research project is not being developed in a specific regulated environment, and without the present definition of flexibility remuneration, this will not allow for any monetary transaction.

The UMEI is publicly available on GitHub, allowing for any stakeholder (DSO, Market Operator, Aggregators, Consumer, and even TSO) to adopt it or to develop new APIs concerning new services while complying with the UMEI interface specification.

2 Implementation of the UMEI

The present version of the UMEI covers the following stages shown in the *Figure 4*:

- Flexibility needs assessment
- Flexibility procurement/trading
- Flexibility activation
- Measurement data retrieval

The functionalities listed above have been implemented in the DSO system. The architecture and method of communicating with the NODES market platform using UMEI is described in Chapter 3.

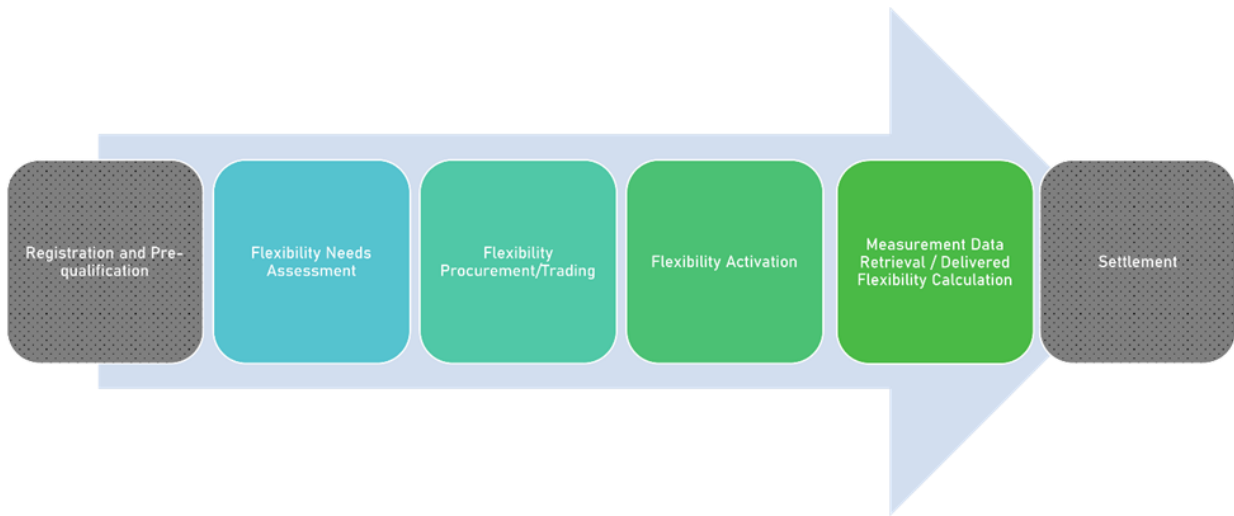


Figure 4 - Flexibility process steps covered by the UMEI

Table 2.1: UMEI overall structure

Group	Name	Usage
Baseline	Managing portfolio baselines	Used by the FSPs to manage baselines into the market platform.
Order	Manage Market Orders	Used by the DSOs and FSPs to view and execute orders' related operations in the market platform. The FMO will perform clearing/matching, either continuously or on a specific schedule, between orders. The result of this process will be the trades.
Meter Reading	Manage Meter Readings	Used by the DSOs, and possibly other market participants, to submit and manage metering data
Market	List All Markets	Used by market participants to get the available markets
Portfolio	Manage Portfolios	Used by FSPs to submit and manage portfolios on the market
Trade	List Market Trades	Used by market participants to retrieve the market trades (the result of the matching process between buy/sell orders)

3 Technical deployment

3.1 Technological choices and architecture

The architecture of the AMS demo system used in the demo is an extended version of the SCADA and EMS systems currently operating at Energa (*Figure 5*).

Energa currently operates separate dispatching systems for high voltage, where issues related to DLR were tested, and for medium voltage, where other issues related to flexibility services were tested. Therefore, the demo environment also consists of two systems, one for HV and one for MV. Within each of the environments, a SCADA system was installed along with the necessary components, such as relational databases. In addition, each environment has been enriched with AMS modules, which are an extension of EMS modules, in which new functions necessary in the EUniversal project have been implemented, in specific functions related to flexibility services and communication with the NODES market platform. Each of the environments was initially supplied with data from production systems, additionally enriched and detailed power grid models (CIM) within the demonstration areas.

Communication with the NODES market via UMEI is carried out by a library made using the OpenAPI generator tool (<https://openapi-generator.tech/>). For this purpose, the "cpp-qt-client" mode was used to generate the base library from the JSON file describing the UMEI interface. This library was then embedded in the AMS module responsible for communication with the flexibility market. For security reasons, it was also decided to use a proxy server in communication with the NODES market platform.

The DLR-related modules have been separated to a dedicated server, and communication with the SCADA/AMS system takes place only via webservice. The DLR server does not require any other communication.

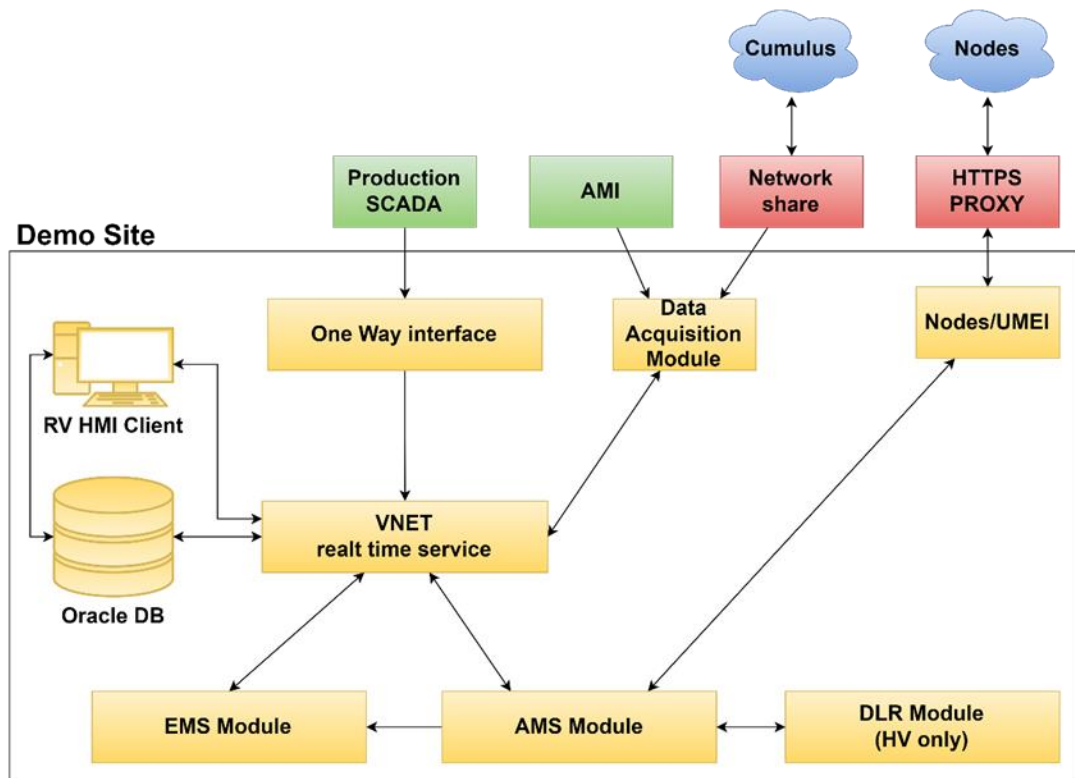


Figure 5 – DSO Architecture

VNET - The basic element of the SCADA system is the real-time services module (called VNET). This module is responsible for the processing and flow of real-time information within the SCADA system. This module has a built-in real-time database, which maintains the current state of the network (analogue measurements, states of switches, etc.) and keeps a log of events. Additionally, this module is responsible for archiving measurement values in a relational database.

DATA ACQUISITION MODULE - This module deals with communication with remotely-controlled objects. This module is responsible for the cyclical or event-based acquisition of the current state of the monitored objects and the execution of sending control commands ordered by the dispatcher or SCADA system modules (e.g., FDIR). This module is an intermediate layer in communication between remotely controlled objects and the VNET service. In the case of the EUniversal demo installation, this module is responsible for replicating the state of remotely controlled objects from the production SCADA system. For safety reasons, the sending of control commands is not provided for within the DEMO system.

RELATIONAL DATABASE - the database of the ORACLE company. The database stores information about the network model (CIM), information about remotely controlled objects, and archives historical measurement data and forecast measurement data. The information from this database is used by all other SCADA / EMS / AMS modules.

EMS MODULE - a module responsible for cyclical calculations. In a standard SCADA / EMS installation, this module cyclically performs the Network State Estimator and Power Flow calculations. The calculation results are sent back to the VNET module, from where they can be forwarded, e.g., for GUI clients for presentation in the network diagram. Regardless of cyclical calculations, the user has the option to simulate to carry out his calculations for any state of the network. For the needs of EUniversal, this module was adapted to the possibility of forecasting loads and generation based on historical data and weather data and then performing power flow calculations on such prepared data.

AMS MODULE - a module responsible for the implementation of functions specific to the EUniversal project, in particular communication with the NODES flexibility market, communication with the weather service, communication with the DLR module, and management of calculations performed by the EMS module. Based on the collected data, this module identifies the needs for flexibility services.

3.2 Authentication mechanisms

The UMEI API specification allows implementers to choose their preferred authentication methods (*Figure 6*). Rather than dictating a specific approach, flexibility is offered to users to select authentication mechanisms that best fit their corporate security requirements. This practical approach ensures easy integration into various technological environments, allowing prioritisation of security based on the specific needs.

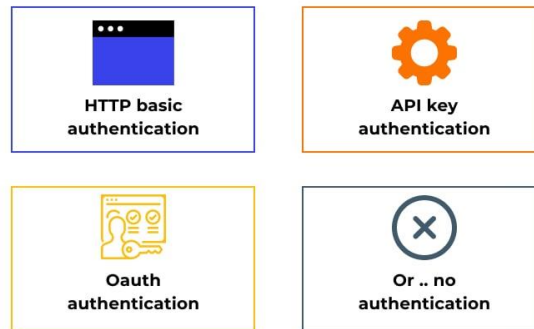


Figure 6 – Basic API Authenticaition modes (Wallarm, 2023)

Ensuring the secure and authorized access of sensitive data through APIs is a fundamental aspect of modern application development. In this context, two prevalent authentication mechanisms come to the forefront: OAuth 2.0 and token-based authentication. Each one of these authentication mechanisms was used by the NODES Flexibility Market Operators in EUniversal.

OAuth 2.0 is a protocol explicitly designed for the secure authorization of third-party applications, allowing them access to user data without exposing sensitive credentials. This process involves four key entities: the resource owner (typically the user), the client application, the authorization server, and the resource server. Within the OAuth 2.0 framework, the client initiates the process by requesting authorization from the resource owner. Upon approval, the client obtains an access token from the authorization server. This token serves as a temporary credential that the client presents to the resource server to gain access to the specified resources. The authorization server plays a crucial role in validating the request and ensuring the proper issuance of access tokens.

On the other hand, token-based authentication is a method where the client is authenticated through a unique token rather than a direct verification of credentials with each request. This approach is particularly common in stateless applications like mobile or single-page applications. In a token-based authentication scenario, the user first provides their credentials for authentication. Upon successful authentication, the server generates a unique token, which is then transmitted to the client and securely stored. When the client needs to access protected resources, it includes this token in the header of its API requests. The server, in turn, validates the authenticity of the token and checks its permissions before granting access to the requested resources.

In the context of the PL pilot, as ENERGA (DSO, FSP) have to exchange data with NODES (FMO) market platform, we chose to implement only OAuth 2.0 authentication method.

4 Usage of the UMEI

4.1 Example API calls

This chapter will present example API calls that have been tested in the Polish demonstration area. Particularly noteworthy are the API calls for the Orders and Trades objects, which are crucial for this project. These specific API interfaces play a key role in managing and executing transactions in the flexibility services market.

Within the Polish demonstration area, these API calls were subjected to intensive testing to ensure their reliability and effectiveness under real market conditions.

It should be noted that a detailed description of this interface goes beyond the scope of this document. For more comprehensive details, please refer to the documents within WP2 of the EUniversal project.

4.1.1 List all markets

The "GET Markets" call in the UMEI API interface allows for retrieving information about available flexibility markets. This query is used to obtain an overview of various markets, along with their basic characteristics. It is primarily used for informational purposes.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//Markets
Response	<pre>{ "numberOfHits": 44, "items": [{ "id": "3efc8c07-c610-49d1-8605-830c8d199d6a", "name": "HV/DLR Active Power Market", "description": null, "quantityType": "ActivePower" }, { "id": "71abe36a-a322-4337-927c-db37ae5616fb", "name": "MV Active Power Market", "description": null, "quantityType": "ActivePower" }, { "id": "9cca65b8-a338-4c71-832d-0fa311490c7b", "name": "MV Reactive Power Market", "description": null, "quantityType": "ReactivePower" }], "links": null }</pre> <p>Remarks: Response was truncated to represent only markets from PL Demo. The original answer contained 44 objects.</p>

4.1.2 Search for portfolios for specific gridNode

The GET Portfolios call is used to retrieve basic information about portfolios and their statuses. This example searches for the portfolio associated with the specified GridNodeId.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//Portfolios?gridNodeId=c37d42fe-671b-440e-9735-c15bd3296363
Response	<pre>{ "numberOfHits": 1, "items": [{ "id": "72be17e6-4751-4341-9768-4bd38881e1fd", "status": "Active", "name": "PF Łebcz Windfarm", "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "ownerOrganizationId": null }], "links": null }</pre>

4.1.3 Create BaselineInterval

The POST BaselineInterval call is used to create a BaseLineInterval for the specified PortfolioId

Method	POST
Request	https://umei-extern-test.nodesmarket.com//BaselineIntervals <pre>{ "periodFrom": "2023-11-25T22:00:00Z", "periodTo": "2023-11-25T23:00:00Z", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "quantity": 10, "quantityType": "ActivePower", "status": "Active" }</pre>
Response	OK

4.1.4 Get the BaselineInterval list

The GET BaselineInterval call is used to search for a BaseLineInterval for the specified Portfolio within a specified time period.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//BaselineIntervals?portfolioId=72be17e6-4751-4341-9768-4bd38881e1fd&periodFrom.gte=2023-11-25T21%3A00%3A00Z&periodTo.lte=2023-11-29T21%3A00%3A00Z

Response	<pre> { "numberOfHits": 3, "items": [{ "id": "f2d37f53-efa4-4dbe-8ba5-eb0e4e671166", "status": "Active", "links": [], "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "periodFrom": "2023-11-26T06:00:00Z", "periodTo": "2023-11-26T06:59:00Z", "quantity": 20, "quantityType": "ActivePower" }, { "id": "376c9b1d-b9ed-42af-8b60-f00481af2a9f", "status": "Active", "links": [], "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "periodFrom": "2023-11-25T23:00:00Z", "periodTo": "2023-11-26T00:00:00Z", "quantity": 1, "quantityType": "ActivePower" }, { "id": "545b5a2a-7d38-472e-8296-933cda9f0dd7", "status": "Active", "links": [], "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "periodFrom": "2023-11-25T22:00:00Z", "periodTo": "2023-11-25T23:00:00Z", "quantity": 10, "quantityType": "ActivePower" }], "links": null } </pre>
----------	--

4.1.5 Sending the meter reading

The POST MeterReadings call is used to send the meter reading values for the indicated PortfolioId in a specified time interval.

Method	POST
Request	https://umei-extern-test.nodesmarket.com//MeterReadings <pre> { "periodFrom": "2023-11-25T22:00:00Z", "periodTo": "2023-11-25T23:00:00Z", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "quantity": 10, "quantityType": "ActivePower", "status": "Active" } </pre>
Response	<pre> { "id": "cbfcd21a-5783-4be9-bc05-e0fafb0b8d9f", "status": "Active", "links": null, "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", } </pre>

	<pre> "periodFrom": "2023-11-25T22:00:00Z", "periodTo": "2023-11-25T23:00:00Z", "quantity": 10, "quantityType": "ActivePower" } </pre>
--	--

4.1.6 Retrieving meter readings for a specific portfolio for a given period of time

The GET MeterReadings call is used to retrieve the meter reading values for the indicated PortfolioId in a specified time period.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//MeterReadings?portfolioId=72be17e6-4751-4341-9768-4bd38881e1fd&periodFrom.gte=2023-11-25T21%3A00%3A00Z&periodTo.lte=2023-11-29T21%3A00%3A00Z
Response	<pre> { "numberOfHits": 1, "items": [{ "id": "cbfcd21a-5783-4be9-bc05-e0fafb0b8d9f", "status": "Active", "links": null, "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "periodFrom": "2023-11-25T22:00:00Z", "periodTo": "2023-11-25T23:00:00Z", "quantity": 10, "quantityType": "ActivePower" }], "links": null } </pre>

4.1.7 Creating an order

The POST Orders call is used to place an order or offer for the flexibility service. The call parameters provide necessary information such as the portfolio, owner, type of regulation, order duration and the quantity and price of the flexibility service.

Method	POST
Request	https://umei-extern-test.nodesmarket.com//Orders <pre> { "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "minimumAcceptanceQuantity": 5, "ownerOrganizationId": "3e77ce74-e8c7-4227-bf9a-7b30e4d818c5", "periodFrom": "2023-11-25T23:00:00Z", "periodTo": "2023-11-26T00:00:00Z", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "pricePoints": [{ "quantity": 10, "unitPrice": 400 }] } </pre>

	<pre> }], "regulationType": "Down", "side": "Sell" } } </pre>
Response	<pre> { "id": "4fa10856-5cf3-44c4-ac7b-1793cbd65eb8", "ownerOrganizationId": "3e77ce74-e8c7-4227-bf9a-7b30e4d818c5", "status": "Pending", "completionType": null, "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "flexibilityZoneId": null, "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "regulationType": "Down", "side": "Sell", "pricePoints": [{ "quantity": 10, "unitPrice": 400 }], "minimumAcceptanceQuantity": 5, "periodFrom": "2023-11-25T23:00:00Z", "periodTo": "2023-11-26T00:00:00Z", "longflexContractId": null } </pre>

4.1.8 Downloading a list of orders for a specified grid node, market, regulation type and specified time range

The GET Orders call is used to search for submitted offers or orders for flexibility services. The call parameters include the necessary information such as GridNodeId, market, regulation type, and order duration . In response, we receive detailed information about individual orders.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//Orders?gridNodeId=c37d42fe-671b-440e-9735-c15bd3296363&marketId=71abe36a-a322-4337-927c-db37ae5616fb&regulationType=Down&periodFrom.gte=2023-11-25T21%3A00%3A00Z&periodTo.lte=2023-11-26T21%3A00%3A00Z
Response	<pre> { "numberOfHits": 1, "items": [{ "id": "4fa10856-5cf3-44c4-ac7b-1793cbd65eb8", "ownerOrganizationId": "3e77ce74-e8c7-4227-bf9a-7b30e4d818c5", "status": "Completed", "completionType": "Filled", "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "flexibilityZoneId": null, "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "regulationType": "Down", "side": "Sell", "pricePoints": [{ </pre>

	<pre> "quantity": 0, "unitPrice": 400 }], "minimumAcceptanceQuantity": 5, "periodFrom": "2023-11-25T23:00:00Z", "periodTo": "2023-11-26T00:00:00Z", "longflexContractId": null }], "links": null } </pre>
--	---

4.1.9 Reading public information about orders for a given market, portfolio, type of regulation in a given period of time

The GET PublicOrders call is used to search for submitted offers or orders for flexibility services. The call parameters provide the necessary information such as GridNodeId, market, portfolio, type of regulation, period for which we want to search for orders. In response, we receive limited (public) information about individual orders.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//PublicOrders?gridNodeId=c37d42fe-671b-440e-9735-c15bd3296363&marketId=71abe36a-a322-4337-927c-db37ae5616fb&portfolioId=72be17e6-4751-4341-9768-4bd38881e1fd&regulationType=Down&periodFrom.gte=2023-11-25T21%3A00%3A00Z&periodTo.lte=2023-11-26T21%3A00%3A00Z
Response	<pre> { "numberOfHits": 1, "items": [{ "id": null, "ownerOrganizationId": null, "status": "Active", "completionType": null, "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "flexibilityZoneId": null, "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "portfolioId": null, "regulationType": "Down", "side": "Sell", "pricePoints": [{ "quantity": 5, "unitPrice": 400 }], "minimumAcceptanceQuantity": 1, "periodFrom": "2023-11-26T01:00:00Z", "periodTo": "2023-11-26T02:00:00Z", "longflexContractId": null }], "links": null } </pre>

4.1.10 Changing order details, changing quantity and unit price for a given order

The PATCH PublicOrders call is used to modify submitted offers or orders for flexibility services. The parameters of the interface call contain all the information as in the case of placing an order

Method	PATCH
Request	https://umei-extern-test.nodesmarket.com//Orders/548b2ec0-f0e5-4eb9-a1ed-87c6ad246ca7 <pre> { "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "minimumAcceptanceQuantity": 1, "ownerOrganizationId": "3e77ce74-e8c7-4227-bf9a-7b30e4d818c5", "periodFrom": "2023-11-26T01:00:00Z", "periodTo": "2023-11-26T02:00:00Z", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "pricePoints": [{ "quantity": 6, "unitPrice": 450 }], "regulationType": "Down", "side": "Sell" }</pre>
Response	<pre> { "id": "548b2ec0-f0e5-4eb9-a1ed-87c6ad246ca7", "ownerOrganizationId": "3e77ce74-e8c7-4227-bf9a-7b30e4d818c5", "status": "Active", "completionType": null, "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "flexibilityZoneId": null, "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "regulationType": "Down", "side": "Sell", "pricePoints": [{ "quantity": 6, "unitPrice": 450 }], "minimumAcceptanceQuantity": 1, "periodFrom": "2023-11-26T01:00:00Z", "periodTo": "2023-11-26T02:00:00Z", "longflexContractId": null }</pre>

4.1.11 Deleting the order

The DELETE Orders call is used to delete submitted offers or orders for flexibility services. The order identifier is passed in the parameters of the interface call.

Method	DELETE
--------	--------

Request	https://umei-extern-test.nodesmarket.com//Orders/548b2ec0-f0e5-4eb9-a1ed-87c6ad246ca7
Response	OK

4.1.12 Reading of transactions for the indicated gridNode, market, portfolio in a given period of time

The GET Trades call is used to retrieve the list of concluded trades. In the parameters of the interface call, the identifiers of GridNode, market, portfolio and the time period being searched are passed.

Method	GET
Request	https://umei-extern-test.nodesmarket.com//Trades?gridNodeId=c37d42fe-671b-440e-9735-c15bd3296363&marketId=71abe36a-a322-4337-927c-db37ae5616fb&portfolioId=72be17e6-4751-4341-9768-4bd38881e1fd&periodFrom.gte=2023-11-25T21%3A00%3A00Z&periodTo.lte=2023-11-26T21%3A00%3A00Z
Response	<pre>{ "numberOfHits": 1, "items": [{ "id": "62c06692-49a8-4695-bebd-b9b511261191", "ownerOrganizationId": "3e77ce74-e8c7-4227-bf9a-7b30e4d818c5", "status": "Completed", "marketId": "71abe36a-a322-4337-927c-db37ae5616fb", "orderId": "4fa10856-5cf3-44c4-ac7b-1793cbd65eb8", "quantity": 10, "side": "Sell", "unitPrice": 400, "periodFrom": "2023-11-25T23:00:00Z", "periodTo": "2023-11-26T00:00:00Z", "gridNodeId": "c37d42fe-671b-440e-9735-c15bd3296363", "flexibilityZoneId": null, "portfolioId": "72be17e6-4751-4341-9768-4bd38881e1fd", "regulationType": "Down", "links": null }], "links": null }</pre>

4.1.13 Deleting the specified BaselineInterval

The DELETE BaselineInterval call is used to delete the specified BaselineInterval.

Method	DELETE
Request	url:https://umei-extern-test.nodesmarket.com//BaselineIntervals/545b5a2a-7d38-472e-8296-933cda9f0dd7
Response	OK

4.2 API Performance test

Out of curiosity and also to confirm the correctness of the interface implementation, a performance test of the interface was also conducted. This test involved sending approximately 100 queries in a single batch to create orders ('POST Orders'). The processing time of individual queries was measured. Response times turned out to be very consistent, the average response time to place an order was 279 ms. Results of the response time measurement have been placed in the chart below (Figure 1).

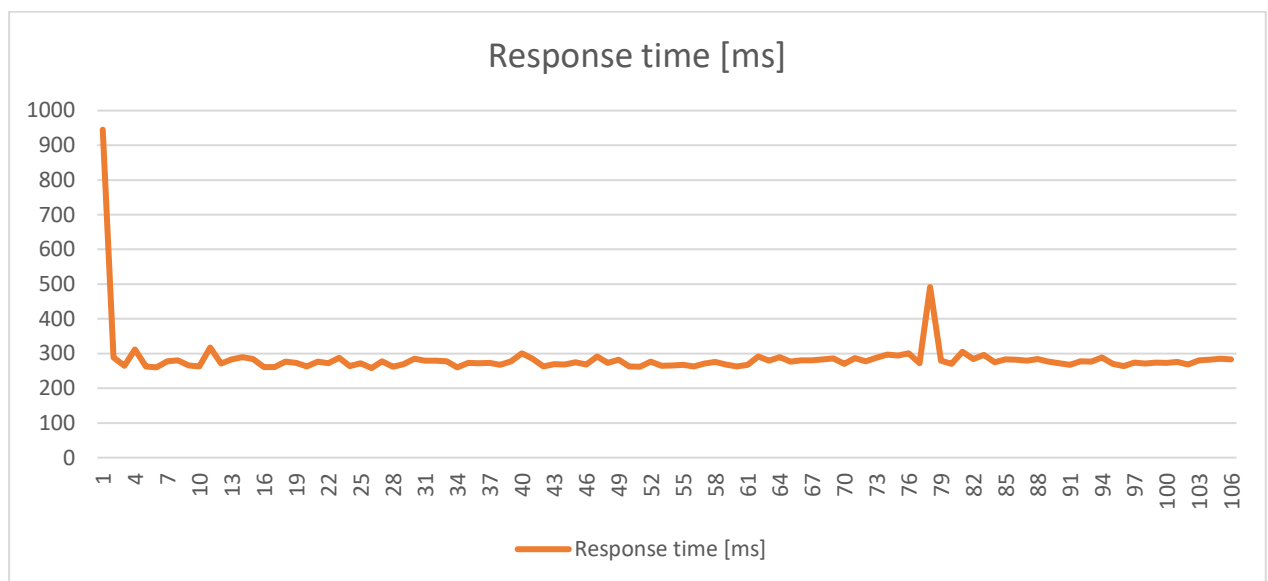


Figure 1 – Results of the response time

From the chart, we can also observe that the first call takes significantly more time – 945 ms, due to including the OAuth2 authentication initiation time.

While using the interface, we encountered virtually no errors. The only errors we encountered were related to trying to operate the interface outside the established parameter limits, e.g. placing an order outside the allowed gate opening time.

4.3 Trading results

Table 4.1: Trading results from NODES market platform; Trading period 2023

	MV AP - down	MV AP - up	HV/DLR AP
Number trades	27	33	152
Net traded volume (MWh)	35,3	18,5	580,5
Trade value (€/MWh)	4.325,15	902,22	79.400,37
VWAP (€/MWh)	122,53	48,77	136,78
Min price (€/MWh)	28,29	25,53	53,82
Max price (€/MWh)	283,82	230	283,82
Min Qt (MW)	0,1	0,1	0,1
Max Qt (MW)	10	10	10

Applying the approach described in the previous chapters, in total of 212 trades have been concluded in 2023 with the maximum number of trades in BUC HV/DLR AP (Table 4.1). Although the flexibility market has been simulated, trading operations have been performed based on energy prices on the day-ahead market of the Energy Exchange. The minimum bid size in every market was 0,1 MW. The highest net volume was traded in the BUC HV/DLR AP amounting to 580,5 MWh and a corresponding trade value of 79.400,37 €/MWh. Downregulation of assets connected to the MV grid, 35,3 MWh in total, resulted in a trade value of 4.325,15€.

Considering the limited amount of FSPs the trading results presented above, prove that flexibility markets are a useful and reliable tool for DSOs to solve local and regional grid constraints. The results prove that using the flexibility of the existing resources efficiently improves the use of existing grid capacity and may ultimately reduce the need for expensive grid expansion investments.

5. Conclusions

The API design tools used - Swagger/OpenAPI made it much easier to understand the interface as well as its implementation. No problems were encountered during implementation on both the DSO and FMO sides. It is worth noting that one of the scenarios involved reversing roles, where the DSO, using the DLR mechanism, assumed the role of FSP on HV Active Power Market. This allowed the DSO to test the interface functionalities associated with the role of FSP.

Data exchange between the DSO and FMO systems using the UMEI interface ran smoothly. Response times were completely satisfactory.

Flexibility markets are a valuable tool to provide solutions to local and regional grid limitations. Furthermore, providing access to distributed assets flexibility markets offers more granular solutions to the new load patterns resulting from increasing electrification and digitalisation.

References

L. Fonseca et al., “EUniversal D2.4: UMEI API functional specification of DSO Interface for standardized market platforms Interaction,” 2021.

C. Silva et al., “EUniversal D2.5: UMEI prototype,” 2021.