Project FLEXUM, D2: Development of flexibility engine/services and OneNet integration

Authors:

ODINS. Rafael Marín Pérez, Jesús Sánchez, Alfredo Quesada, Antonio Skarmeta

UMU. Adelaida Parreño, Alfonso Ramallo-González, Juan Sánchez

Distribution Level	PU
Responsible Partner	ODINS
Checked by WP leader **[name surname]**	Date: N/A
Verified by the appointed Reviewers **[I-DE/Beatriz Alonso** **I-DE/ F. David Martin Utrilla]**	Date: 9/12/2022
Approved by Project Coordinator	Date: N/A

Dissemination Level		
PU	Public	x
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Issue Record

Planned delivery date	November 8th, 2022
Actual date of delivery	November 24th, 2022
Status and version	Revised, version 2.0

Version	Date	Author(s)	Notes
1.1	07/11/2022	ODINS & UMU	
1.2		ODINS & UMU	
2.0	24/11/2022	ODINS & UMU	New edition of the document after Iberdrola's feedback
3.0	07/12/2022	ODINS & UMU	Final version with some corrections after Iberdrola's feedback

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

About OneNet

OneNet has the aim of providing a seamless integration of all the actors in the electricity network across Europe to create the conditions for a synergistic operation that optimizes the overall energy system while creating an open and fair market structure.

The project OneNet (One Network for Europe) is funded through the EU's eighth Framework Programme Horizon 2020. It is titled "TSO – DSO Consumer: Large-scale demonstrations of innovative grid services through demand response, storage and small-scale (RES) generation" and responds to the call "Building a low-carbon, climate resilient future (LC)".

While the electrical grid is moving from being a fully centralized to a highly decentralized system, grid operators have to adapt to this changing environment and adjust their current business model to accommodate faster reactions and adaptive flexibility. This is an unprecedented challenge requiring an unprecedented solution. For this reason, the two major associations of grid operators in Europe, ENTSO-E and EDSO, have activated their members to put together a unique consortium.

OneNet will see the participation of a consortium of over 70 partners. Key partners in the consortium include: the already mentioned ENTSO-E and EDSO, Elering, E-REDES, RWTH Aachen University, University of Comillas, VITO, European Dynamics, Ubitech, Engineering, and the EUI's Florence School of Regulation (Energy).

The key elements of the project are:

1. Definition of a common market design for Europe: this means standardized products and key parameters for grid services which aim at the coordination of all actors, from grid operators to customers;
2. Definition of a Common IT Architecture and Common IT Interfaces: this means not trying to create a single IT platform for all the products but enabling an open architecture of interactions among several platforms so that anybody can join any market across Europe; and
3. Large-scale demonstrators to implement and showcase the scalable solutions developed throughout the project. These demonstrators are organized in four clusters coming to include countries in every region of Europe and testing innovative use cases never validated before.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

List of Figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

List of Abbreviations and Acronyms

Acronym	Meaning
AMQP	Advanced Message Queueing Protocol
API	Application Programming Interface
DRE	Demand Response Event
DSO	Distribution System Operator
ENTSO-E	European association for the cooperation of transmission system operators (TSOs) for electricity
EV	Electric Vehicle
FSP	Flexibility Service Provider
IoT	Internet of Things
ISP	Imbalance Settlement Period
JSON	JavaScript Object Notation
kW	kilowatt (power)
kWh	kilowatt-hour (energy)
MW	Megawatt (power)
MWh	Megawatt-hour (energy)
NGSI-LD	Next Generation Service Interface – Linked Data
OMIE	Operador del Mercado Ibérico de Energía (Iberian Peninsula energy market operator)
TSO	Transmission System Operator
UFTP	USEF Flex Trading Protocol
USEF	Universal Smart Energy Framework
XML	Extensible Markup Language

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Executive Summary

The first part of the document is divided on the one hand into the description of the base model used for the development of the FLEXUM platform being the PHOENIX platform and the flexibility services offered through its Flexibility Engine. It details the interaction between PHOENIX and the grid within the Universal Smart Energy Framework (USEF) and the negotiation and execution of flexibility via the USEF Flex Trading Protocol (UFTP) based on a series of XML messages to assure a compatible data language within the platform. On the other hand, the FLEXUM platform, based on PHOENIX technologies, is described being the solution to carry out the real flexibility tests.

The second part of the document describes the work done on the integration of OMIE and FLEXUM platform showing how it works by carrying out in a simulated environmental outside the flexibility tests. The result of this integration is a software module that monitors the requests generated by DSOs and sent to OMIE to react to grid congestions, which trigger the creation of auctions for Flexibility Service Providers (FSPs) to provide flexibility in the periods in which these consumption peaks are expected to happen. The module forwards the flexibility request to the Flexibility Engine offered by the PHOENIX platform and controls all the process to launch a final flexibility order once the offer has been confirmed by OMIE.

Based on the work done around this deliverable, a flexibility solution that can handle requests from DSOs through OMIE is ready to be demonstrated and evaluated in the real scenario of the UMU pilot by performing flexibility tests by using the semi-assisted solution within the FLEXUM platform.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

1 Introduction

After having done a flexibility characterization of the demo scenario and having analysed the scenario as it can be seen on Deliverable 1. This deliverable describes the flexibility solution that has been implemented. The so called FLEXUM platform will be described which is based on the technology used in the PHOENIX [6] project. This solution has a semi-assisted system to perform the required flexibility part and integration of the flexibility capacity available from the pilot to perform the flexibility test actions to meet the agreed upon flexibility requirements.

Another important topic that is covered at this stage of the project is the integration of the FLEXUM platform with OMIE, the energy market operator of the Iberian Peninsula, which is working on a platform inside OneNet as described in *Deliverable 9.1: Specifications and guidelines of Western Demos* of OneNet [5] capable of providing Short-Term and Long-Term flexibility products.

The interaction with a real energy market operator as well as the evaluation and results of the flexibility tests will be used on Deliverable 3. The work done so far proves that the solution not only works at a conceptual level in a specific scenario, but also in other environments as it can be integrated in the official energy market in Spain. This integration makes the solution portable to other locations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2 Flexibility

The work developed at FLEXUM is highly cost effective as it benefits from the developments done within European project PHOENIX. In section 2.1 is detailed the Phoenix platform and its Flexibility Engine whose technologies have served as the basis for what is otherwise described in section 2.2, which is the flexibility services in FLEXUM by detailing the FLEXUM platform being the solution to perform the flexibility tests.

2.1 PHOENIX Flexibility engine

The PHOENIX platform and its Flexibility Engine, offer an interaction channel to external agents of the grid to perform a negotiation for the calculation of the availability and the execution of the flexibility.

2.1.1 Flexibility engine in the PHOENIX platform

One of the objectives of PHOENIX is to offer to the users the possibility of adjusting their demand based on a flexibility request coming from the grid agent or from the aggregator in charge of energy trading. To achieve this, PHOENIX must be able to interact with the grid agent and to offer a series of mechanisms necessary for the trading, evaluation and execution of the demanded flexibility. The integration in the platform of the devices installed in the buildings allowed to analyse their energy behaviour and their control, as well as to modify the demand in each device and even in the whole building.

In order to achieve these energy services related to demand flexibility and consequently the design and execution of Demand Response Events (DREs), a series of components have been designed and developed as part of the platform architecture. Figure 1 shows the designed architecture with the necessary components for the realisation of these demand flexibility services.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

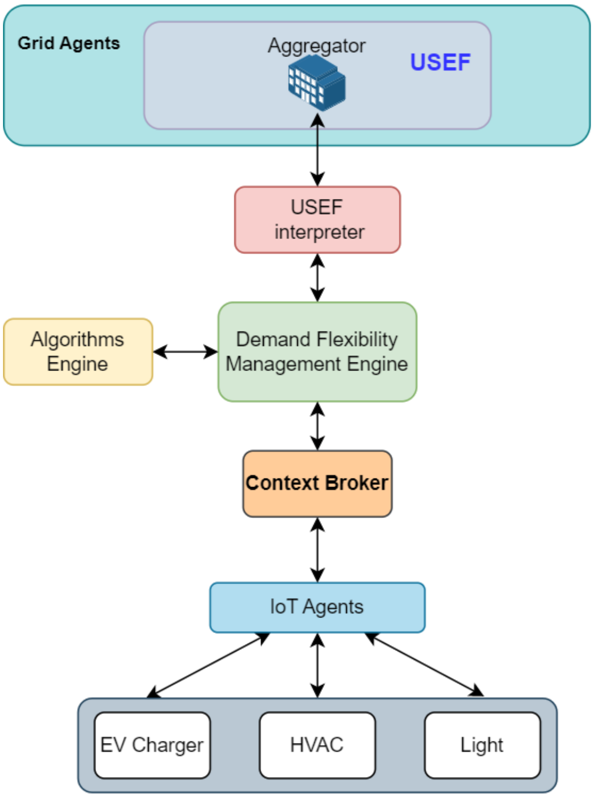


Figure 1 – Architecture Diagram and Flexibility Engine

The PHOENIX platform was designed initially to make use of the USEF communication protocol for the interaction between the platform and an external grid agent. Therefore, for the flexibility negotiation and execution processes between grid agents and PHOENIX, it is necessary to comply with the message format proposed by USEF via UFTP (XML format). This protocol and messages used are detailed in the following section. Finally, for this interaction, the integration of the USEF interpreter component in the PHOENIX platform was required as can be seen in the architecture, which is in charge of translating the messages coming from the grid (in USEF, XML format) into a data language compatible with the PHOENIX platform namely the NGSI-LD (Next Generation Service Interface – Linked Data).

On the other hand, all the context information of the platform and the deployed devices that allow adjusting the energy demand of the building are managed by a dedicated set of components with this purpose that has given the name of Context Broker. This component stores all the real-time and historical information necessary for the analysis of the energy performance of each building, as well as information on a variety of services offered in PHOENIX such as comfort, self-generation, etc. All this information is collected by the Flexibility Engine for its study and evaluation. In addition to the historical information, a series of estimation and forecasting analyses were needed to know the energy performance and the expected demand for the period in which

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

flexibility is required. These analyses are provided by the Algorithm Engine integrated in the architecture in connection with the Flexibility Engine.

These are some of the data managed by the Flexibility Engine for flexibility assessment and negotiation with external grid agents:

- Expected energy consumption of each device.
- Configuration and flexibility parameters of each device.
- External and room temperature.
- Comfort parameters for each device.

The top layer of the architecture shows the different possible grid agents that can interact with the platform for flexibility negotiation and execution. In the case of the aggregator, capable of understanding the USEF protocol, it interacts directly with the platform through the USEF interpreter.

2.1.2 Communication with the grid

As discussed above, we built over the PHOENIX platform complies with the USEF framework [2], which provides a common standard on which to build smart energy services and a set of tools for the development of distributed flexibility mechanisms. To meet this flexibility negotiation and interaction format, the USEF interpreter component has been created to provide the ability to listen to requests from the grid regarding demand flexibility in order to integrate flexibility negotiation and execution into the platform. Figure 2 shows the integration of the developed interpreter into the PHOENIX platform and its connection to the flexibility engine.

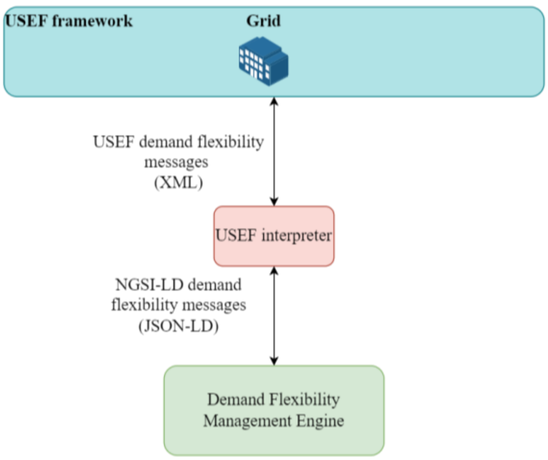


Figure 2 – USEF Interpreter Integration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

FlexRequest, FlexOffer and FlexOrder are the main messages that are part of the negotiation process between the grid agent and the platform. These messages are defined by the UFTP protocol [3]. This protocol can be used in the forecast, bid, offer, order and flex settlement processes.

USEF makes use of the Imbalance Settlement Period (ISP) as a time unit to achieve the necessary time granularity for the realisation of demand flexibility. One ISP is equivalent to a 15-minute interval, which means that one day contains 96 ISPs. Flexibility requests, offers and orders are based on ISPs and therefore this is the unit of time used in the context of demand flexibility and interaction with external grid agents.

Figure 3 shows the content of a FlexRequest message defined by the UFTP protocol (only the first 6 ISPs of the day are shown for simplicity). This request includes the requested ISPs for load reduction and the amount of power to be reduced. In addition, the ISPs that are available and the amount of power that can be shifted to these periods can also be indicated. In this way, the reduction of consumption at the required ISPs can be carried out by designing and executing DREs or through a load shifting to the available ISPs (if present).

In the example shown, a load reduction of 5 kW is required for ISP number 5. In addition, a part of this load reduction can be diverted to the other available ISPs.

Figure 3 – FlexRequest content

Figure 4 shows the content of a FlexOffer message in response to the previous FlexRequest. It shows how the required 5 kW at ISP number 5 are feasible and can be achieved by controlling and modifying the energy behaviour by means of the plataform. In addition, it provides information on the load that can be shifted to other ISPs. The example shows that 3 kW can be shifted to ISP number 6, as the previous FlexRequest indicated that there was 3 kW at ISP number 6 available for use.

As for the FlexOrder message, the content is the same as the FlexOffer as it is a confirmation/acceptance of the flexibility offer.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

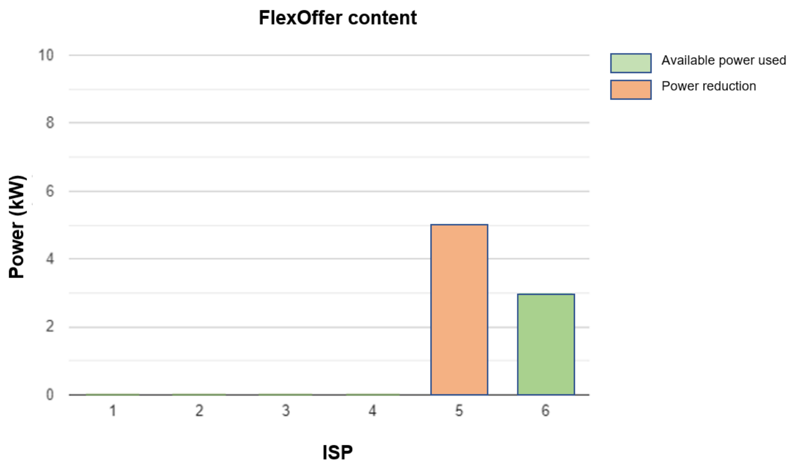


Figure 4 – FlexOffer content

Finally, the USEF interpreter translates these messages allowing the integration in PHOENIX of grid agents established in an external framework such as USEF.

XML is the language chosen by USEF for the serialisation of these messages defined by the UFTP protocol. The USEF interpreter offers a REST API that allows receiving XML messages from the external grid agent and translating them into JSON-LD, a format compatible with the NGSI-LD API used throughout the PHOENIX platform. In the opposite case, it also allows the grid agent to retrieve flexibility messages in USEF compliant format (XML) following the reverse order in the translation. Figure 5 shows an example of a FlexRequest message translation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

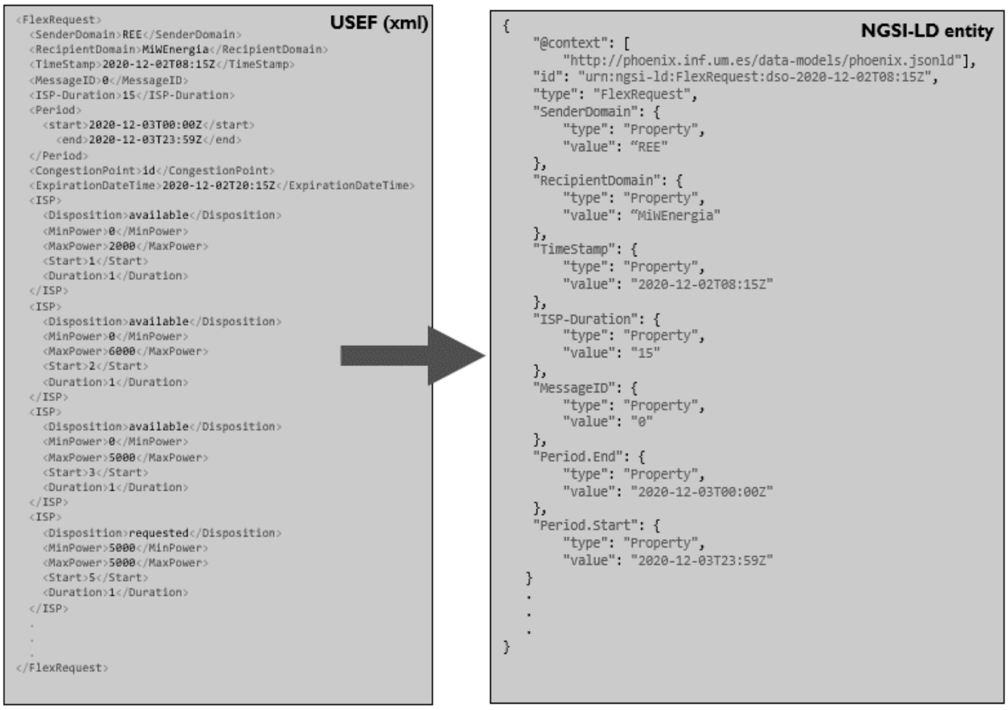


Figure 5 – FlexRequest translation

On the other hand, an example of the reverse translation of a FlexOffer message is shown in Figure 6.

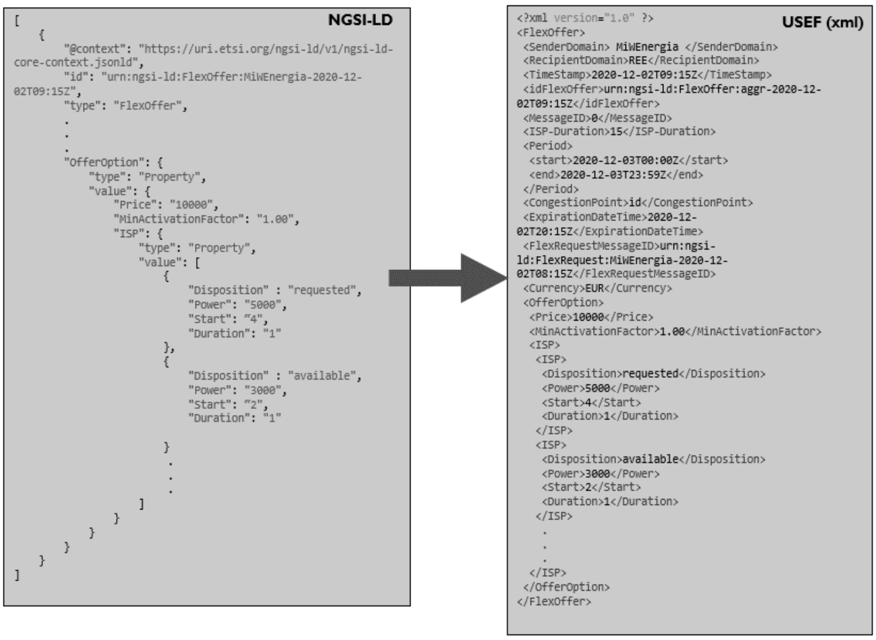


Figure 6 – FlexOffer translation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2.1.3 Demand Flexibility Workflow

The Flexibility Engine's negotiation with the grid is mainly based on three messages. These messages are as follows:

- FlexRequest (Flexibility Request). It is sent by the grid agent to the PHOENIX platform and indicates the magnitude and timing (ISPs) of the requested flexibility. That is, it indicates the amount of power to reduce and the ISPs available to perform time-shift load.
- FlexOffer (Flexibility Offer). This is a response to the FlexRequest. It contains the load profile change that can be made to perform the flexibility.
- FlexOrder (Flexibility Order). After receiving the different FlexOffers, the grid agent chooses the most suitable offer and sends the FlexOrder. FlexOrder has the same content as the previous FlexOffer (it is a confirmation of the Flexibility Offer). Once received by PHOENIX, the flexibility must be performed.

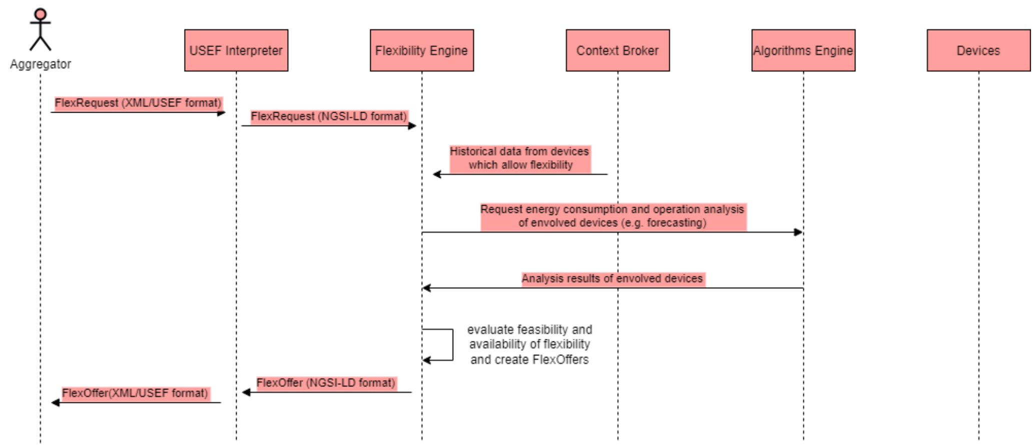


Figure 7 – Flexibility Negotiation Workflow

These three messages guide the operation of the Flexibility Engine, including the negotiation phase with the grid agent and the subsequent execution of DREs.

Figure 7 shows the flexibility negotiation steps between the grid agent and the solution. First, the grid agent makes use of the interpreter to access the platform, which converts the request into NGSI-LD format. Once the flexibility engine has received the flexibility request, it must examine in the Context Broker all the devices that may be involved in the DRE to solve that flexibility. The devices that can be involved in a DRE are Shiftable or Controllable (see the categorisation of devices in section 2.1.4).

After obtaining the devices that can be controlled or time-shift loaded, it is necessary to calculate the load profile of each device in the period indicated in the flexibility request. Up to this point the data analytics involved in the design of DREs has been done using forecasting algorithms. For completing the analysis, a connection to

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

the algorithm engine is required. This engine is in charge of analysing the data coming from the devices involved in the flexibility.

When all the required information is collected, the Flexibility Engine evaluates the feasibility of the flexibility request and calculates whether it is possible to decrease the load on the indicated ISPs. To do so, it first evaluates the Controllable devices and estimates the amount of energy that can be decreased considering their flexibility characteristics, such as the controllability factor. In a second step, it evaluates the Shiftable devices and tries to shift the operation pattern to other ISPs with available power (indicated in the FlexRequest), thus releasing the ISPs where demand adjustment is required. With this estimation, flexibility offers are proposed to the grid agent by means of the FlexOffer messages.

So far there has been a negotiation between the platform and the grid agent in which a decrease of the load on specific ISPs has been required and the platform has offered flexibility offers with the demand adjustment profile that it is able to reach. When the grid agent has received the flexibility offers, he must take the decision and choses the one that best suits its needs.

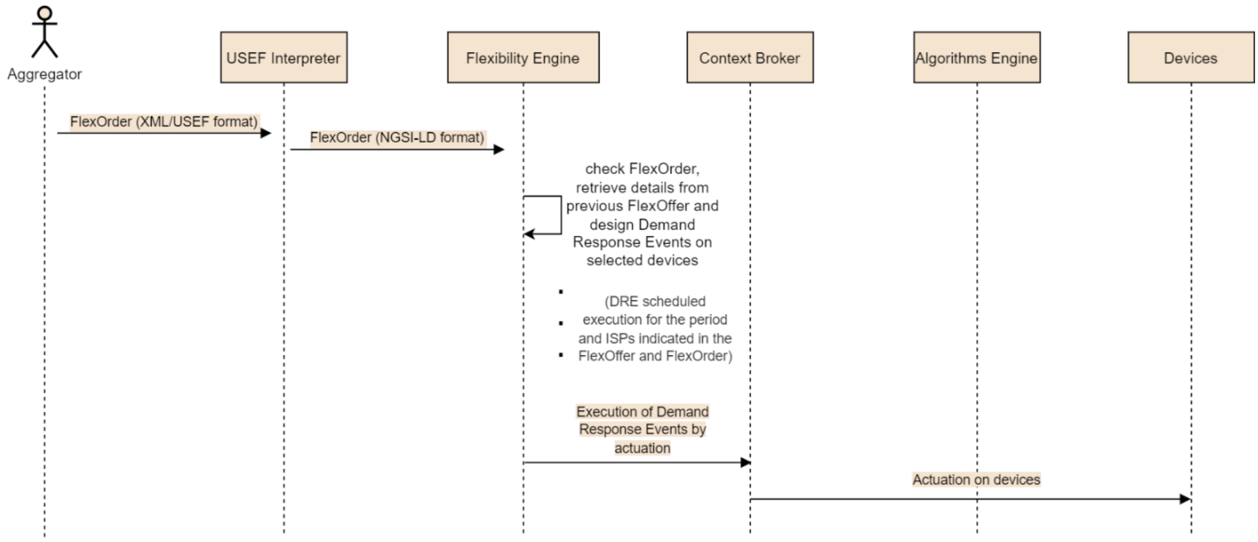


Figure 8 – FlexOrder: flexibility execution workflow

Once the grid agent has chosen the appropriate flexibility offer, it proceeds to command the agreed flexibility by sending a FlexOrder message. The content of this message is the same as the content of the associated FlexOffer as it is a confirmation and acceptance of the offer sent by the platform. Therefore, this message contains the ISPs with the amount of power to be decreased or shifted over time, considering the demand adjustment capability of the building devices. Figure 8 shows the workflow of this flexibility ordering and execution process.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2.1.4 Categorisation and identification of device's flexibility

A number of devices in each building are integrated in the platform and provide context and measurement information. However, there are not only devices that are able to read information periodically and report it to the Context Broker, but also others that are able to receive action commands and change their behaviour. These devices are called "*actuators*" and are key to the demand adjustment required by the Flexibility Engine. A change in the operation of one device can lead to a change in the energy/power demand and can scale up to a demand adjustment of an entire building if a large number of devices are actuated.

After receiving a flexibility signal from the grid, a direct load control strategy can be performed on the actuating devices to alter their behaviour (e.g. reducing consumption switching off an HVAC device or changing the temperature setpoint). Therefore, these actuator devices are the ones involved in the flexibility process and must be properly configured to optimise the flexibility evaluation, negotiation, and execution process.

Actuators can be categorised into the following types of devices according to the flexibility they are able to offer:

DEVICE TYPE	DESCRIPTION
Shiftable	Allow time-shift load. It is possible to deviate the power consumption from one time period to another, e.g. EV (Electric Vehicle) chargers.
Non-shiftable	Do not provide flexibility as consumption cannot be controlled and time-shift load is not allowed.
Controllable	Allow load reduction. It is possible to control the device and perform a reduction of power consumption, e.g. HVAC.

Only Controllable and Shiftable devices offer flexibility. Therefore, the Flexibility Engine performs a scan of all such devices to assess the flexibility they are able to offer by means of a DRE. However, the Flexibility Engine should not have the total freedom to change the operation of the devices but should be limited by the users according to their preferences and needs.

To this end, some parameters have been defined to allow occupants to limit the control of the Flexibility Engine:

- Flexibility Type (type of device in terms of flexibility). Possible values: Controllable, Shiftable and Non-shiftable.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

- Controllability Factor (only for Controllable devices). Controllability factor associated with the device. Integer between 0 and 1. For example: 0.20 means that it can be controlled to 20 percent of the total control. (HVAC whose setpoint can only be increased or decreased by 20 %).
- Operation Window (only for Shiftable devices). Time window available for the operation of the device. This allows sliding the operating period with the operation window as a limit, allowing time-shift load.

When the Flexibility Engine receives a flexibility request, it evaluates Shiftable and Controllable devices and calculates the flexibility offers based on their capabilities (defined by the parameters discussed above) to alter their operating pattern. Therefore, the initial configuration of these devices is necessary to establish their flexibility characteristics.

The definition and configuration of the flexibility parameters are intended to increase the context information available in the platform necessary for the realisation of the power services offered. Therefore, being NGSI-LD the common data format used in the platform for the representation and management of context information, an entity called FlexConfiguration has been defined, which houses all the information and flexibility parameters mentioned above in section 2.1.4.

Each entity is linked to one actuator device. The following box shows an example of an entity related to an HVAC device where the device is of type controllable and the controllability factor (1, i.e. full control) are indicated.


```
{
    "id": "urn:ngsi-ld:FlexConfiguration:UMU-Pleiades-BlockB-B1.1.015-Thermostat-FlexConfiguration",
    "type": "FlexConfiguration",
        "refDevice": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:Device:UMU-Pleiades-BlockB-B1.1.015-Thermostat"
    },
    "flexibilityType": {
        "type": "Property",
        "value": [
            "controllable"
        ]
    },
    "controllabilityFactor": {
        "type": "Property",
        "value": 1
    },
    "operationWindow": {
        "type": "Property",
        "value": {
            "startTime": "",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```
                "endTime": ""
            }
        },
        "@context": "http://phoenix.inf.um.es/data-models/phoenix.jsonld"
    }
```

2.2 Flexibility services in FLEXUM

The previous section has detailed the operation of the Flexibility Engine and the different components developed in the PHOENIX project to offer the flexibility service to the grid. PHOENIX has been used as a model to create the platform developed in FLEXUM to offer a flexibility scenario according to the requirements of FLEXUM. The FLEXUM Platform is therefore the PHOENIX counterpart for the FLEXUM pilot.

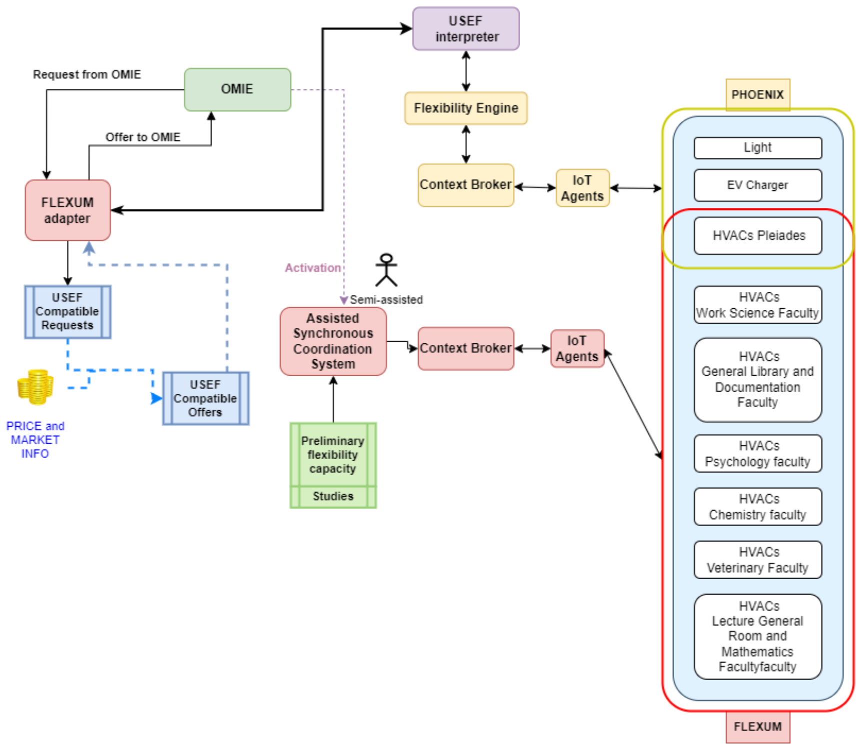


Figure 9 – The co-existing platforms of PHOENIX (yellow components) and FLEXUM (red components).

2.2.1 FLEXUM Platform

To achieve the integration of the large number of devices available for flexibility, several IoT Agents have been deployed to connect the platform with each of the devices that are part of the flexibility asset set. This required gateway components to enable this connectivity to the buildings' BMS systems or directly to the HVAC equipment.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

These flexibility assets have been modelled using the data format established by NGSI-LD and integrated into the platform using NGSI-LD entities. All information related to the flexibility assets are part of the context information of the FLEXUM platform and are stored in the Context Broker component. This component provides access to all the context information through the NGSI-LD API and, in addition, allows the actuation on each flexibility asset by sending commands such as turn off or turn on. As a result of this development, deployment, and integration work in FLEXUM, a platform is obtained that allows access to a large amount of context information from different buildings (in this case, flexibility assets). Being able, in this way, to control each one of them through a common format defined by NGSI-LD.

The development and deployment of this FLEXUM platform enables the integration and development of higher-level components that make use of the provided context information and device control in a homogeneous way. One of these components can be the Flexibility Engine discussed in the previous section that allows the evaluation and execution of flexibility as demonstrated in the PHOENIX project. However, the flexibility tests performed in FLEXUM have been executed by a semi-assisted component called Assisted Synchronous Coordination (See Figure 9) that allows the activation of the flexibility agreed with OMIE for specific periods by scheduling the sending of action commands on the flexibility assets during those specific periods. These action signals have been carried out on the Context Broker, and, consequently, on the lower layers down to the devices using the FLEXUM platform.

In a fully automated platform, the actuations and all the above would be performed by the Flexibility Engine of section 2.1.1 therefore USEF interpreter for a compatible communicate would be required. To achieve this compliance with USEF by the integration of OMIE in the FLEXUM platform, an extra adapter was required to be developed which is defined in Section 3.

2.2.2 Assisted Synchronous Coordination System

In this section will be discussed the process to be followed to fulfil a flexibility request previously agreed with the grid. Considering that the process followed for the flexibility tests has been through the semi-assisted service of Assisted Synchronous Coordination System (ASCS). This system allows combining the semi-supported services (they are partly done manually by introducing data on the OMIE platform) related to the flexibility request, the activation of the flexibility agreement and the integration of the flexibility capability of the assets, with the shutdown actuation by different IoT Agents to carry out the flexibility tests.

For the FLEXUM platform, an adaptor must be included so that the communication between OMIE and the ASCS contains power details and market details, complying in this way with the USEF format and with the OMIE requirements as it is shown in Figure 9. This integration, together with the development of the FLEXUM adapter, is detailed in later sections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

2.2.3 Preliminary Flexibility Capacity

A previous study was carried out to obtain the possible power reduction flexibility of each flexibility asset after several actuations that were previously carried out in the different buildings, being these buildings the ones that will be used for the flexibility tests in the FLEXUM project.

These actuations were carried out in February, March, and June 2022 [1], consisting of shutdowns or modifications of the setpoint temperature corresponding to the different assets with the aim of reducing the power (kW) and analysing the energy performance of the buildings.

As mentioned above, after the completion of these actions, the results obtained of the power consumption response were evaluated and analysed, calculating the power reductions that had been generated for each of these assets [1]. In this way, the flexibility capacity was obtained, being this, the mentioned power reduction that can be offered by each of the assets and that has been integrated in the part of the Assisted Synchronous Coordination indicated in Figure 9.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

3 Integration with OneNet

As part of Task 3, a module has been developed to interconnect the OMIE platform that receives congestion signals coming from DSOs and the FLEXUM platform (see Figure 9), where all the Flexibility components are deployed and have access to the IoT gateways required which control the different devices that are used for providing flexibility.

The new module plays the role of a FSP that connects with OMIE using the certificates and a unique identifier generated beforehand to secure the communications.

The integration has used the architecture proposed in D1 [1] in section *4 Communication between the different actors* as reference. The process is triggered by a request generated by a DSO once a potential (or existing) congestion has been detected, and this is done through the OMIE portal using XML files that are sent encoding the data in a well-known format.

In the current implementation, the auction model is the one used in combination with the hourly market specification, which means a day is divided in 1-hour slots and for each zone, every slot will have its own separate auction.

The control process of the module implemented is described in the following figure.

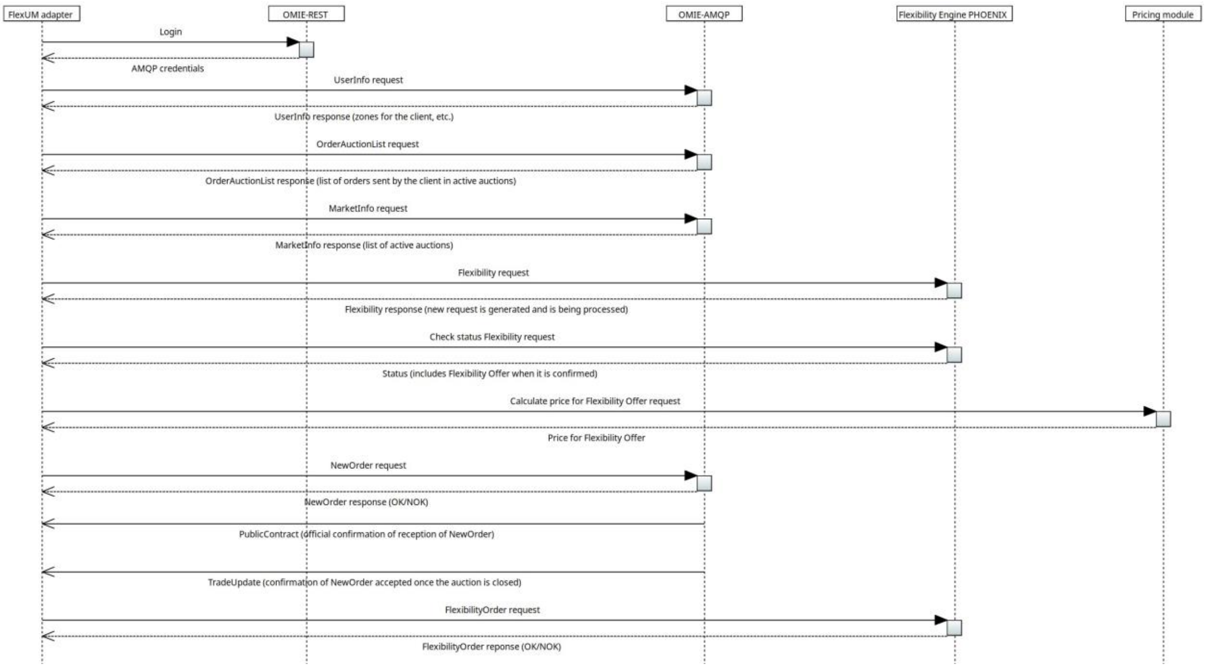


Figure 10 – Integration OMIE ↔ FLEXUM

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

As can be seen in the previous figure, there are 5 logical entities involved in the process:

- FLEXUM adapter. This is the new module. As mentioned earlier, from OMIE's perspective, the adapter is a FSP.
- OMIE-REST. This is the part of the OMIE API that must be used to get the credentials required to access the AMQP server (Advanced Message Queueing Protocol).
- OMIE-AMQP. This server acts as a frontend for integrators to have access to the functionality offered by OMIE.
- Flexibility Engine PHOENIX. This is the entity that provides Flexibility support.
- Pricing module. This virtual entity is in charge of deciding which price must be set in each Flexibility offer.

From a global perspective, the process can be divided in 5 logic blocks:

- Login.
- Wait for an open auction.
- Calculate the amount of Flexibility that can be provided and send an offer.
- Wait for the end of the auction to see if our offer has been accepted.
- Execute the Flexibility actuations.

The format of all the messages exchanged between the module and the OMIE-AMQP entity will be described in a more detailed way in a specific section in the Appendix. In this section only the most relevant aspects of these messages will be introduced to simplify the description of the process.

3.1 Login

In this stage, the module communicates with the OMIE-REST API through the next endpoint:

https://lip.flex.omie.es/jsiom/app/servicesTrading/login/QUc4NS5lOFUzZ0Y5RC52MQ

The identifier used for the login (QUc4NS5lOFUzZ0Y5RC52MQ) is the result of applying a Base64 conversion over the Application Identifier (*AppId*) assigned to FLEXUM, that is, **AG85.e8U3gF9D.v1**.

In the payload of the response, the credentials required to connect to the AMQP server are returned in JSON format:

{"**agentID**":"AG85","**userRabbit**":"AGE85_FLEX","**passwordRabbit**":"xxxxx"}

Starting at this point, all the references to the module related to the AMQP exchanges will refer to it as *client* instead of *module* so as to follow the naming conventions used in the documentation received from OMIE.

However, both terms are exchangeable as they represent the same concept from an implementation perspective.

3.2 Wait for an open auction

Once the connection to the AMQP server has been established, a **UserInfo** request is sent in order to get some configuration information of the client, especially the set of zones it has assigned. In this case, only one zone is available named **ESPINARDO** in reference to the location of the supply point.

Instead of going directly to check which auctions are open, there is a previous step in which an **OrderAuctionList** request is sent. The response includes the list of offers sent by the client to open auctions. This list will be stored locally to prevent sending duplicate offers in case the module is restarted after having sent an offer.

Finally, the module checks whether there are active auctions in any of the zones assigned to it. This is done by sending a **MarketInfo** request. This request includes as a parameter the code of the area to be used for the search and in this case the code is **10YES-REE------0** representing Spain.

Each of the entries received in the response to the previous message contains, among others, certain relevant attributes:

- Type of operation. Not only auctions are supported, so non-auctions are directly ignored.
- Area of the operation.
- Zone of the operation.
- Contract identifier.
- State of the operation. The module ignores the auctions that are not in TRADING state.
- Start of the period and number of periods linked to the request. Knowing that the market works in hourly mode, each operation will have a timestamp for the beginning of the involved day and the number of period (number of 1-hour slot) inside the day when the energy/power reduction must be provided. In this case, as we are talking about Flexibility, this pair defines the slot when the reduction of consumption must be delivered.
- Type of request (MW for power reduction and MWh for energy reduction). In our case, the solution has been designed to reduce the power in the selected slot.
- Amount of energy/power requested.

The module will cyclically search for open auctions until one is found.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

3.3 Calculate Flexibility

Once an open auction has been detected, the module contacts the framework asking for a provision of Flexibility compatible with the parameters defined in the auction. In our case, there is not automatic calculation of Flexibility, instead, we manually insert the value that is needed, and then operate the ASCS to provide that reduction of power during the requested time.

As described in section 2.1.2, and following the USEF format, a request is sent will the following content:

POST /interpreter/usef/flex_request


```
<FlexRequest>
    <SenderDomain>Iberdrola</SenderDomain>
    <RecipientDomain>FLEXUM-PHOENIX</RecipientDomain>
    <TimeStamp>2022-10-13T20:00Z</TimeStamp>
    <MessageID>1</MessageID>
    <ConversationID>1</ConversationID>
    <Revision>0</Revision>
    <ISP-Duration>PT15M</ISP-Duration>
    <Period>
        <start>2022-10-14T19:00Z</start>
        <end>2022-10-14T19:59Z</end>
    </Period>
    <CongestionPoint>id</CongestionPoint>
    <ExpirationDateTime>2022-10-13T21:00</ExpirationDateTime>
    <ISP>
        <Disposition>requested</Disposition>
        <MinPower>20</MinPower>
        <MaxPower>20</MaxPower>
        <Start>1</Start>
        <Duration>1</Duration>
    </ISP>
    <ISP>
        <Disposition>requested</Disposition>
        <MinPower>20</MinPower>
        <MaxPower>20</MaxPower>
        <Start>2</Start>
        <Duration>1</Duration>
    </ISP>
    <ISP>
        <Disposition>requested</Disposition>
        <MinPower>20</MinPower>
        <MaxPower>20</MaxPower>
        <Start>3</Start>
        <Duration>1</Duration>
    </ISP>
    <ISP>
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```xml
            <Disposition>requested</Disposition>
            <MinPower>20</MinPower>
            <MaxPower>20</MaxPower>
            <Start>4</Start>
            <Duration>1</Duration>
        </ISP>
    </FlexRequest>
```

This request includes certain important parameters:

- The start and end of the 1-hour slot, stored in the **Period** section.
- The ISP duration. As the Flexibility engine of PHOENIX is designed to use the standardized 15-minute slot format, this value will always be 15 minutes also in FLEXUM and therefore a conversion must be made to map the hourly responses received from OMIE to this format.
- For each one of the four 15-minute ISPs, a pair of minimum and maximum power must be included. In our case both parameters have the same value for all the ISPs and this value is the amount of power requested in the auction.

As a result, a Flexibility request is generated with a unique identifier. If the FLEXUM platform were to be fully automated by using the flexibility engine, the flexibility engine would calculate whether the requirements can really be met.

Knowing that this process may take some time, the client uses the Flexibility request identifier received in the response of the previous request to query the status of the request periodically:

GET /interpreter/usef/flexOffer?FlexRequestMessageID=**{ID}**

Once the information is ready, in case the amount of requested Flexibility can be provided, a Flex offer will be included in the response with the following format:

```xml
    <FlexOffer>
        <SenderDomain>FlexUM-PHOENIX</SenderDomain>
        <RecipientDomain>Iberdrola</RecipientDomain>
        <TimeStamp>2022-10-13T20:15Z</TimeStamp>
        <idFlexOffer>id:ngsi-ld:FlexOffer:Iberdrola-2022-10-13T20:15Z</idFlexOffer>
        <MessageID>1</MessageID>
        <ISP-Duration>PT15M</ISP-Duration>
        <Period>
            <start>2022-10-14T19:00Z</start>
            <end>2022-10-14T19:59Z</end>
        </Period>
        <CongestionPoint>id</CongestionPoint>
        <ExpirationDateTime>2022-10-13T23:15</ExpirationDateTime>
        <FlexRequestMessageId>id:ngsi-ld:FlexRequest:Iberdrola-2022-10-13T20:00Z</FlexRequestMessageId>
```

```xml
        <Currency>EUR</Currency>
        <OfferOption>
            <Price></Price>
            <MinActivationFactor>1.00</MinActivationFactor>
            <ISP>
                <ISP>
                    <Power>20</Power>
                    <Start>1</Start>
                    <Duration>1</Duration>
                </ISP>
                <ISP>
                    <Power>20</Power>
                    <Start>2</Start>
                    <Duration>1</Duration>
                </ISP>
                <ISP>
                    <Power>20</Power>
                    <Start>3</Start>
                    <Duration>1</Duration>
                </ISP>
                <ISP>
                    <Power>20</Power>
                    <Start>4</Start>
                    <Duration>1</Duration>
                </ISP>
            </ISP>
        </OfferOption>
    </FlexOffer>
```

After that, the module sends a request to the pricing-calculation module that is in charge of deciding the price-related part of the offer. This module is an abstraction done to differentiate the roles in the interaction, including the flexibility calculations (power/energy) and the pricing calculations. For a basic integration, this module has a simple implementation (it always returns the same value) but in the future it could rely on historic pricing data, real-time energy generation data if there is any or day-ahead prices obtained from different sources such as the ENTSO-E portal (European association for the cooperation of transmission system operators (TSOs) for electricity), etc.

With this, the client will finish this stage by sending a **NewOrder** request. This request includes, among other parameters, the price and amount of power offered by the FSP for the selected auction.

In addition to the response sent back to the client to the previous request, the OMIE-AMQP entity also sends a **PublicContract** message indicating that a new transaction has been registered and accepted, although this offer has yet to be confirmed and might not be accepted.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

3.4 Wait for the end of the auction

Once the auction is over, OMIE analyses all the offers received for the auction and chooses between them those that match the requirements ordered by price (cheaper offers take priority). As a result, a **TradeUpdate** message is sent to the selected clients to let them know they have been selected. If the client has been selected and one of these messages is received, a final Flexibility order is sent to the framework with the same content in the payload as the one received in the Flexibility offer but having adjusted the required power to the value that has been assigned to the offer by OMIE. In the end an offer can be accepted only partially and the amount of power accepted is included in the **TradeUpdate** message. The involved DSO will be informed through the web portal (this is done directly by OMIE and the client has no control over this operation).

3.5 Execute the Flexibility actuations

In this final stage is when the FLEXUM framework would execute the actions required to provide the Flexibility accepted during the selected 1-hour slot, as said before this is done by the ASCS that is manually set depending on the request. As indicated above, this integration of FLEXUM with OMIE has been developed and tested in a simulated environment with requests and offers via the OMIE portal.

4 Conclusions

In terms of the flexibility services performed and the designed IoT architecture deployed, the feasibility of this solution for adjusting demand in certain periods required by the grid through the operational control of devices in a planned manner has been proven. The large number of devices integrated in the platform (both actuators and meters) allows the execution of a flexibility request in a common data format and a common action format, regardless of the manufacturer or low-level protocols used.

The semi-automated system of the FLEXUM platform to handle the final flexibility agreement, the activations and the integration of the preliminary flexibility capacity obtained from the assets in the Assisted Synchronous Coordination towards the Context Broker, to finally with IoT agents perform the flexibility tests allows FLEXUM to prioritize the flexibility reduction, which means the reduction of the agreed power. Using the FLEXUM adapter developed for USEF format we allowed communication compatibility between OMIE and the FLEXUM platform. The process follows a set of steps that matches the requirements of both APIs and combined with an external pricing simulated calculation module, covers all the cycle, from the request sent by the DSO to the provision of the Flexibility.

The next efforts will be focused on evaluating the solution of the FLEXUM platform and on the exploitation of solutions based on Flexibility.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

5 References

[1] D1: Flexibility characterization, baseline calculation and demo definition

[2] USEF Foundation, USEF The Framework Explained," 20 06 2021. [Online]. Available: www.usef.energy.

[3] USEF Foundation, USEF Flexibility Trading Protocol Specifications," 28 01 2020. [Online]. Available: www.usef.energy

[4] Real Decreto 486/1997, de 14 de abril, por el que se establecen las disposiciones mínimas de seguridad y salud en los lugares de trabajo. BOE-A-1997-8669. https://www.boe.es/eli/es/rd/1997/04/14/486/con

[5] OneNet D9.1: Specifications and guidelines for Western Demos

[6] PHOENIX project: https://eu-phoenix.eu/

6 KPIs evaluation

The following KPIs are related to the current deliverable:

KPI2.1: *Being able to communicate with local market platform developed by OMIE in a bi-directional way.*

This integration has been described during this deliverable and additional information is included as an appendix.

KPI2.2: *Intelligent actuation over UMU infrastructure (i.e. HVACs and eVehicles chargers) to deliver flexibility services.*

The actuation over the devices was made in the preliminary tests to quantify the flexibility that can be offered by each building.

KPI2.3: *Demonstrate more than 0.1MW of flexibility by the control of the UMU infrastructure.*

This KPI will be described in detail in Deliverable 3 as part of the evaluation of the solution.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

7 Appendix

7.1 AMQP extended message format

7.1.1 **UserInfo** response


```
{
    "agentID": "AG85",
    "agentDesc": "AG85 UMU",
    "agentClass": "N",
    "certificateID": "AGE85_FLEX",
    "certificateProfile": "A",
    "certificateName": "AGENTE85",
    "certificateSurname": "INNOVACION",
    "certificateEmail": "BPELAGENTPRB@OMEL.ES",
    "validityStart": "2018-07-24",
    "validityEnd": "2024-07-23",
    "serverStatus": "OK",
    "areas": [
        {
            "area": "10YES-REE------0",
            "areaDesc": "Spain Delivery Area",
            "zones": [
                {
                    "zone": "ESPINARDO",
                    "zoneDesc": "IDE-MURCIA",
                    "negotiationType": "S",
                    "units": [
                        {
                            "unit": "A85U01",
                            "unitDesc": "AGFLEX85-UMU ESPINARDO",
                            "unitType": "V"
                        }
                    ]
                }
            ]
        }
    ]
}
```


7.1.2 **OrderAuctionList** response


```
{
    "orders": {
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```
            "37454": [
                {
                    "zone": "ESPINARDO",
                    "orderID": 2370,
                    "orderIDOrig": 2370,
                    "contractID": 37454,
                    "sesionID": " PLSESPINARDOP17_14102022",
                    "unit": "A85UO1",
                    "user": "AGE85_FLEX",
                    "agentID": "AG85",
                    "timestamp": "2022-10-14T11:39:59.490+02:00",
                    "msgResp": "Order added correctly. OrderId: 2370",
                    "msgUser": " New Order 2022-10-14T09:39:59.470Z",
                    "type": "V",
                    "complexCondition": "None",
                    "prc": 900,
                    "initialQty": 0.5,
                    "qty": 0.5
                }
            ]
        }
    }
```

7.1.3 **MarketInfo** response

```
    {
        "contractList": [
            {
                "contractID": 37643,
                "sesionId": "PLSESPINARDOP17_14102022",
                "name": "ESPINARDO 20221014 H17",
                "zone": "ESPINARDO",
                "area": "10YES-REE------0",
                "product": "ML_HORARIO",
                "period": "2022-10-17T00:00:00.000+02:00",
                "numPeriod": "19",
                "state": "TRADING",
                "tradingPhaseStart": "2022-10-17T12:40:00.000+02:00",
                "tradingPhaseEnd": "2022-10-17T14:30:00.000+02:00",
                "productDetails": {
                    "product": "ML_HORARIO",
                    "productDesc": "Mercado Local Horario",
                    "unitsQty": "MW",
                    "decimalsQty": 2,
                    "minQty": 0.01,
                    "tickQty": 0.01,
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```
                "unitsPrc": "Euros",
                "decimalsPrc": 2,
                "minPrc": 0.0,
                "tickPrc": 0.01
            },
            "buyOrdrList": [],
            "sellOrdrList": [],
            "tradeList": [],
            "contractRevisionNo": 4,
            "contractTypeNeg": "S",
            "auctionType": "C",
            "auctionQty": "0.5",
            "auctionReqId": "1124",
            "auctionOrdersC": false,
            "auctionOrdersV": true
        }
    ]
}
```

Now more in detail, the previous message includes the following information (this is a sample response including only one operation although the current implementation returns all the operations in the selected area for all the types of operation, not only auctions):

- Identification of the supply point (**area** + **zone**).
- Time range of the request (**period** + **numPeriod** + **product = ML_HORARIO**, indicating that it is using 1-hour slots).
- Type of operation (**contractTypeNeg = S** indicates it is an *auction*).
- Type of request (**MW = power**) and amount of power requested (**auctionQty**).
- Status of the operation (**state = TRADING**, indicating the auction is open).

7.1.4 **NewOrder** request

```
{
    "orders": [
        {
            "agentID": "AG85",
            "message": "New Order 2022-10-17T11:28:07.470Z",
            "zone": "ESPINARDO",
            "area": "10YES-REE------0",
            "unit": "A85U01",
            "sesionID": " PLSESPINARDOP17_14102022",
            "contractID": 37643,
            "type": "V",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

```
            "prc": 900,
            "qty": 0.5
        }
    ],
    "complexCondition": "None"
}
```


The previous example shows the raw content of a **NewOrder** request but the message must be sent ciphered.

On the other hand, this is a sell order (type = V) from the client's point of view (FSP).

7.1.5 PublicContract

The format of this message can be seen in the response to a **MarketInfo** request as it contains an array con **PublicContract** inside the **contractList** attribute.

7.1.6 TradeUpdate

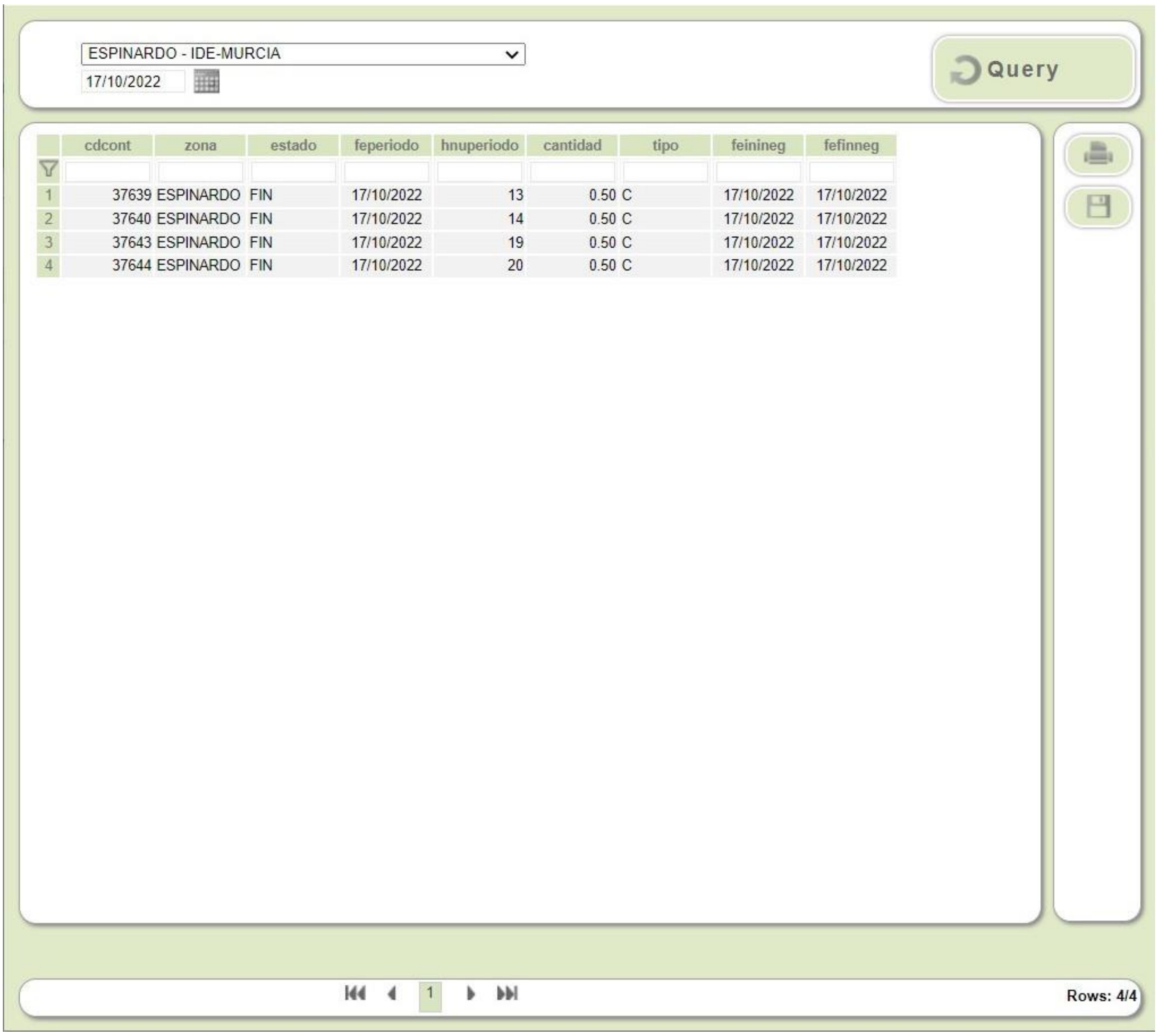

```
{
    "contractID": 37643,
    "revisionNo": 1,
    "tradeList": [
        {
            "tradeID": 1137,
            "sellOrderID": 2370,
            "prc": 900,
            "qty": 0.5,
            "timestamp": "2022-10-17T14:30:15.000+02:00"
        }
    ]
}
```


The **qty** attribute indicates the amount of power that has been accepted for the offer sent and this is the value that will be set in the Flexibility order sent to the framework.

7.2 Overview of the results of the actions executed on the OMIE portal

7.2.1 List of auctions from one day

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739



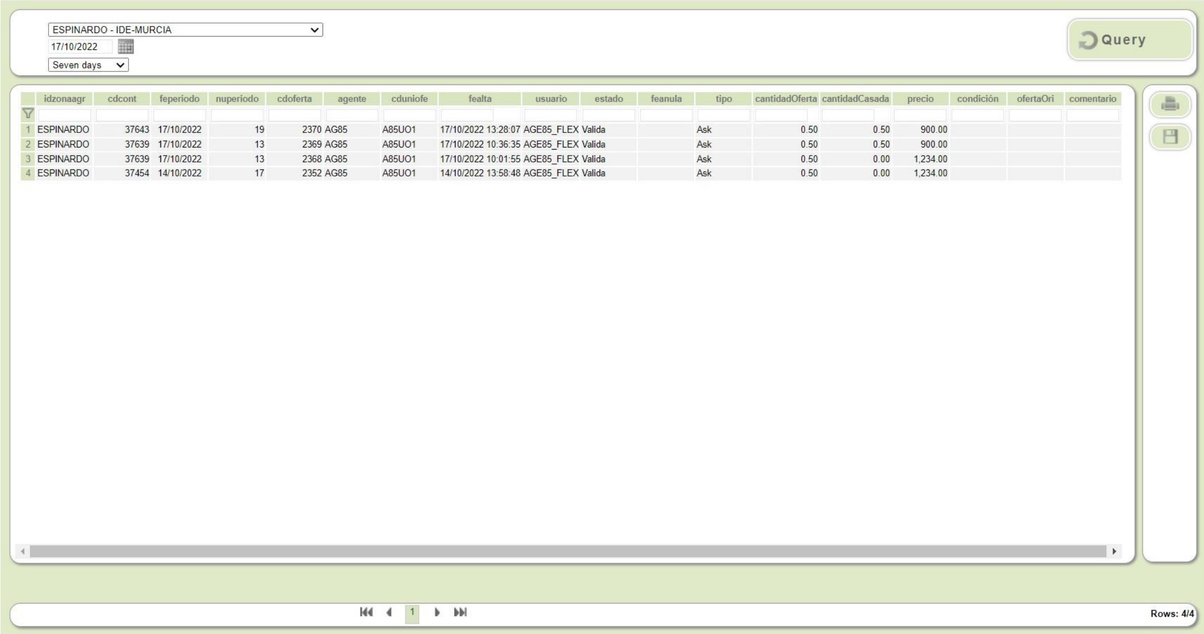
	cdcont	zona	estado	feperiodo	hnuperiodo	cantidad	tipo	feinineg	fefinneg
1	37639	ESPINARDO	FIN	17/10/2022	13	0.50	C	17/10/2022	17/10/2022
2	37640	ESPINARDO	FIN	17/10/2022	14	0.50	C	17/10/2022	17/10/2022
3	37643	ESPINARDO	FIN	17/10/2022	19	0.50	C	17/10/2022	17/10/2022
4	37644	ESPINARDO	FIN	17/10/2022	20	0.50	C	17/10/2022	17/10/2022

Figure 11 – Auctions of the 17th of October

The previous figure shows a list of the auctions created for a specific day of tests.

7.2.2 Result of auctions for one week

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739



	idzonaagr	cdcont	feperiodo	nuperiodo	cdoferta	agente	cduniofe	fealta	usuario	estado	feanula	tipo	cantidadOferta	cantidadCasada	precio	condición	ofertaOri	comentario
1	ESPINARDO	37643	17/10/2022	19	2370	AG85	A85UO1	17/10/2022 13:28:07	AGE85_FLEX	Valida		Ask	0.50	0.50	900.00			
2	ESPINARDO	37639	17/10/2022	13	2369	AG85	A85UO1	17/10/2022 10:36:35	AGE85_FLEX	Valida		Ask	0.50	0.50	900.00			
3	ESPINARDO	37639	17/10/2022	13	2368	AG85	A85UO1	17/10/2022 10:01:55	AGE85_FLEX	Valida		Ask	0.50	0.00	1,234.00			
4	ESPINARDO	37454	14/10/2022	17	2352	AG85	A85UO1	14/10/2022 13:58:48	AGE85_FLEX	Valida		Ask	0.50	0.00	1,234.00			

Figure 12 – Results of auctions

In the previous figure there are 4 auctions from a one-week period. The offers sent for auctions 3 and 4 were not selected (column *cantidadCasada* = 0) because the price sent by the module (1234) was bigger than the maximum established for the auction (1000).

However, the offers sent for auctions 1 and 2 were selected because the new price (900) was lower than the maximum.

This paper reflects only the author's view and the Innovation and Networks Executive Agency (INEA) is not responsible for any use that may be made of the information it contains.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739