

Towards new data management platforms for a DSO as market enabler – UPGRID Portugal demo

Ana Alonso¹ ✉, Rui Couto¹, Hugo Pacheco¹, Ricardo Bessa¹, Clara Gouveia¹, Luis Seca¹, Jorge Moreira², Pedro Nunes², Pedro G. Matos², António Oliveira³

¹INESC TEC, Porto, Portugal

²EDP Distribuição, Lisbon, Portugal

³WITHUS, Aveiro, Portugal

✉ E-mail: ana.n.alonso@inesctec.pt

Abstract: In the framework of the Horizon 2020 project UPGRID, the Portuguese demo is focused on promoting the exchange of smart metering data between the DSO and different stakeholders, guaranteeing neutrality, efficiency and transparency. The platform described in this study, named the Market Hub Platform, has two main objectives: (i) to guarantee neutral data access to all market agents and (ii) to operate as a market hub for the home energy management systems flexibility, in terms of consumption shift under dynamic retailing tariffs and contracted power limitation requests in response to technical problems. The validation results are presented and discussed in terms of scalability, availability and reliability.

1 Introduction

The deployment of smart grid infrastructures motivated a discussion about the current and future roles of the distribution system operator (DSO) in terms of data management and market interrelation, e.g. three data models proposed by the EG3 Smart Grids Task Force (EG3-SGLF) for smart grid data handling [1], consultation work made by ECORYS to the European Commission – DG ENER [2]. Most of these works discuss the advantages and disadvantages of the different data models rather than proposing a conceptual and functional architecture for one model. The main innovation and contribution from the Horizon 2020 UPGRID project is to propose a conceptual and functional data hub architecture that adopts the recommendations and models from the state of the art.

The Market Hub Platform (MHP) is hosted by a DSO and regulates all the interactions between the DSO and the various market agents (retailers, Balance Responsible Parties, aggregators etc.), as depicted in Fig. 1. This solution goes beyond a neutral access market platform (such as the one designed in H2020 Flexiciency project [3]) and encompasses the exchange of information through the MHP, including consumption profiles from the DSO and flexibility profiles from the home energy management system (HEMS). This design situates the MHP as a more flexible mediator but raises natural performance and security concerns: it should be capable of handling requests from the multiple stakeholders but facilitating private client information only to rightful stakeholders. To increase scalability and to avoid collusion, our solution considers an additional retailer platform (RP) for each retailer. The RP is responsible for receiving information/requests from the MHP, and processing and sending this information to its HEMS. Moreover, it allows the retailers to collect other types of data that are not relevant for the DSO, but so for their own business model.

2 Portuguese demonstrator

The Portuguese Demo of the UPGRID project focuses on promoting the exchange of smart metering data between the DSO and different

stakeholders, guaranteeing neutrality, efficiency and transparency. Related to the objective of neutral data access to all market agents, the following use cases were considered: (a) billing request; (b) consumption profile request and (c) broadcast tariffs, regarding new network-use tariffs so that retailers and HEMS can respond accordingly.

The MHP also acts as a market hub for HEMS flexibility, in terms of consumption change due to dynamic retailing tariffs ('normal' operation) and load reduction requests in response to technical problems ('emergency' operation). The following use cases were implemented:

- *Reduce load:* requesting from the retailers the availability of certain clients to perform a load reduction, and then reacting accordingly, by requesting them to disconnect some appliances;
- *Trigger emergency mode:* notifying clients, beforehand, that the network will transition into an emergency mode. Hence, the clients' HEMS have the possibility to undertake some actions to adjust their contracted power before the emergency mode takes effect.

The Portuguese demonstrator was deployed in Lisbon, Portugal, within the parish of Parque das Nações as depicted in Fig. 2. It includes more than 12,000 residential consumers who will have access to the smart grid infrastructure, empowering them to be more active and contributing to the energy transition.

3 Architecture

The process chain as depicted in Fig. 3 shows how the MHP interfaces with the other components. In short, the DSO using the new low-voltage operation system will be able to trigger situations wherein the market agents' contribution can solve congestions. In these situations, it will generate requests, which are forwarded by the MHP to the appropriate RPs, allowing the DSO to assume a true role of market facilitator, through this customer response scheme. In this scenario, the 'highly available storage service' (HASS) was implemented using an Apache HBase cluster (version

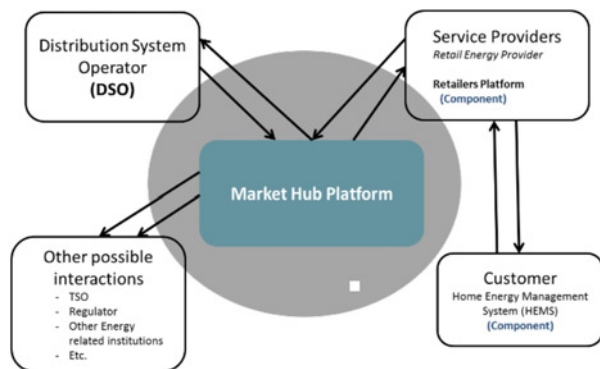


Fig. 1 Market hub platform



Fig. 2 Area of the Portuguese demonstrator

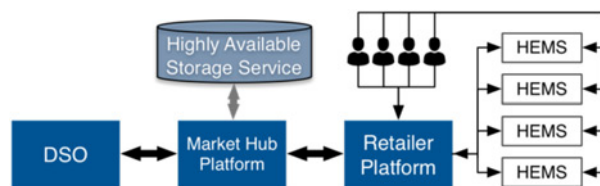


Fig. 3 Market hub platform infrastructure

0.98.6) configured for high availability from the Cloudera distribution version cdh5.3.5, as depicted in Fig. 4.

The RP monitors HEMS flexibility profiles and translates incoming signals from the MHP into appropriate HEMS actuations. The RP also offers a personal area that allows consumers to schedule the flexibility of their home appliances, which will determine how their HEMS react to dynamic tariffs.

The HEMS includes a gateway, installed on the customer premises, which connects to an internet provider router, and contains the communication interfaces to interact with smart devices that provide the ability to measure and actuate on circuits or individual loads.

3.1 Type of information and data flow

Next follows a description of each type of connection.

- Network tariffs calendars: from the DSO to the RP.
- Load information: from the DSO to the corresponding clients.

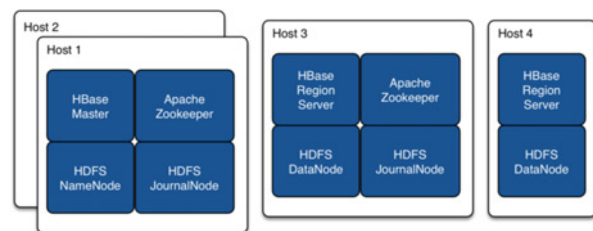


Fig. 4 Highly available storage service

- Emergency request: from the DSO to the corresponding clients.
- Active power measurements: from the RP to the DSO.

4 Test framework

In order to fully test the MHP, a simulation-based test framework that artificially injects requests was developed. Such a scenario is essential in order to stress the system, inject faults, and analyse not only the effect in the MHP, but also the expected side effects (e.g. delays) in the DSO/RP. A 'DSO Simulator' component generates requests for the MHP, according to the provided parameters. This component is configurable, for instance, one can define the number of clients, clients per retailer and the rate of requests (per minute) of each type. A 'DSO Interface' component provides a front end to generate specific requests and monitor progress. Finally, an 'RP Simulator' supports both (a) parameterising the number of requests and (b) simulating responses without overloading the real RPs.

The HBase cluster was deployed in four physical machines, and components were co-located as depicted in Fig. 4. This configuration enables the storage service to remain available even if one of the hosts fails, for example, a failure of Host 1 will cause the backup HBase Master in Host 2 to assume the service, while the standby HDFS NameNode in Host 2 will also become active. Other failure modes comprised of different combinations of component failures are also handled while keeping the service available. However, if the fault-tolerance threshold is crossed, the service becomes unavailable to preserve consistency and it resumes, when the necessary components become available once again.

The current setup is the minimal configuration for high availability using HBase. By adding more hosts with instances of some of the components to the setup, more faults (for those components) can be tolerated. For example, the Apache Zookeeper component requires a majority of nodes to be available: While an ensemble of three instances tolerate the failure of one, an ensemble of five is required to tolerate the failure of two instances. More information regarding HBase cluster configuration for high availability can be found in Fig. 4.

The 'DSO Simulator', 'DSO Interface' and 'RP Simulator' components were co-located with the MHP in a physical host with a dual-core Intel Core i5 running at 2.5 GHz with 8 GB of RAM with an solid-state drive (SSD) disc drive. Hosts 1–4 feature quad-core Intel Core i3 running at 3.4 GHz with 8 GB of RAM with 7200 rpm disc drives.

Finally, for testing purposes, the MHP was enhanced in order to support registering the request flow, delays and timestamps in order to support a later analysis of the results. This type of data is stored in a local MySQL database to minimise the impact on the evaluation of the system.

5 Evaluation

Each feature was tested both with the RP and with the simulated RP. Second, the requests performed by the DSO Simulator were used to fully simulate end-to-end processes. Each feature is automatically generated, using the same application programming interface (API) available to the real DSO and delivering the requests to the RP through the same interface.

5.1 Test monitoring and control

Test execution was controlled in order to monitor the health of the MHP. Storage was measured by the rate at which data is being created and stored in the disc. Memory was measured by directly analysing the impact in the memory of having several clients making requests. CPU was measured by the load in the CPU during normal usage. Network bandwidth was measured by analysing the latency and volume of data being transmitted to the MHP.

5.2 Correctness

5.2.1 Goal: Ensure that the MHP relayed correct tariff information upon request, demand reduction requests to the correct clients, the emergency mode to the requested clients and the correct consumption provided by the DSO. Also, ensure that the MHP correctly updated client data according to what was provided by the RP and never relayed client data to incorrect retailers.

This is achieved by prepending each operation that either handles sensitive data or triggers sensitive actions with checks to ensure that the stakeholder is allowed to do so.

5.3 Performance

5.3.1 Goal: Evaluate the request rate, processing throughput and request latency.

The base setup to test the performance of the MHP mirrors what is expected for the project demonstrator: 100 clients and 1 retailer. As it is expected that the rate of requests to the MHP will be sparse, e.g. 4 requests/hour for flexibility requests, we opted to test the MHP for short time periods, under a heavier load as a stress test. For all performance tests, the request rate for each of the four types is of 30 requests/min, which means a total request rate of 120 requests/min. This far exceeds any expected spikes in utilisation.

Also, in terms of the size of each request, we considered that each emergency, flexibility or load shift requests would affect 10% of the set of clients, and each billing request would affect 30% of the set of clients.

Also, to make the tests more realistic, an artificial delay chosen at random from the interval [100,110] ms was injected in the RP.

5.3.2 Resource usage for the MHP: The data regarding CPU utilisation and disc usage as gathered by the iostat tool is depicted in Fig. 5. The CPU utilisation is obtained by summing system-level usage to user-level usage. The disc throughput is obtained from the 'MB/s' column of iostat, which aggregates disc reads and writes. As previously stated, the reported values are averages per minute.

Again, because the MHP was running co-located with other components, this serves as an upper bound for resource usage. Even for a request rate that is much higher than what is to be expected for nominal utilisation (or for the demonstrator), the CPU utilisation and disc IO remain at acceptable values.

5.3.3 Request latency analysis: The request latency is the most significant metric when evaluating the proposed system, due to its characteristics and the type of integration with the DSO and RP, particularly because of the sparse nature of the expected request rate.

Fig. 6 shows the contribution of the MHP and the HASS to the overall latency of a request, for each type of request. In this configuration, most of the latency is due to processing in the MHP.

In order to evaluate the scalability of the proposed system (MHP+HASS), we ran experiments with varying numbers of clients and varying numbers of retailers. First, we generated sets of 100, 1000 and 10,000 clients and conducted experiments wherein only the number of clients is different from the base setup.

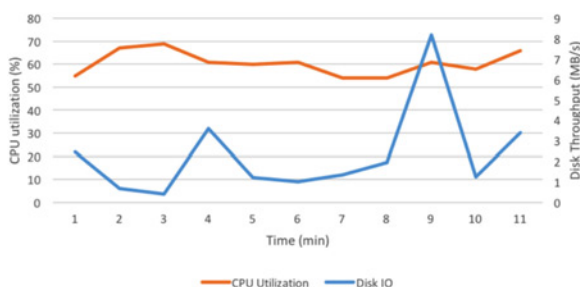


Fig. 5 CPU utilisation and disc usage for the base setup

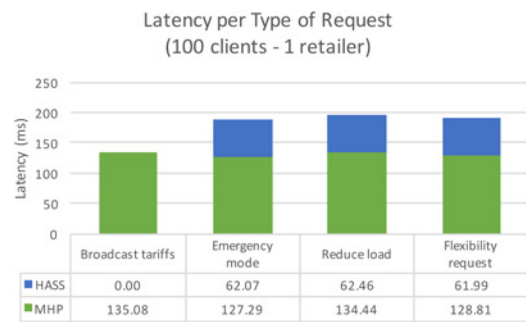


Fig. 6 Breakdown of request latency per component under test per type of request for the base setup

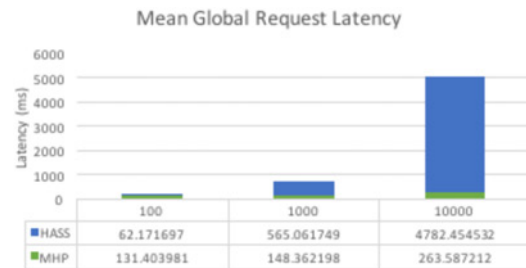


Fig. 7 Distribution of the request latency for sets of 100 to 10,000 clients

Fig. 7 shows how the mean global request latency changes by increasing the number of clients. In particular, it shows that the most affected component is the HASS.

Fig. 8 shows in detail how increasing the number of clients affects each type of request.

To further assess the scalability of the system, we also conducted some experiments wherein the number of retailers is increased regarding the base setup.

Fig. 9 shows how the mean global request latency changes by increasing the number of retailers. In particular, it shows that the most affected component is the MHP. In fact, because increasing the number of retailers while maintaining the total number of clients decreases the size of requests, the effect on the HASS is of improving request latency.

Fig. 10 shows in detail how increasing the number of retailers affects each type of request.

The test results show the MHP's performance. Specifically, we can conclude that the MHP is able to support a large set of clients and large request rates. Indeed, the number of clients and requests used in the test is significantly larger than what is expected for the demonstrator. Increasing the number of clients has the adverse effect of increasing the latency values. However, the MHP

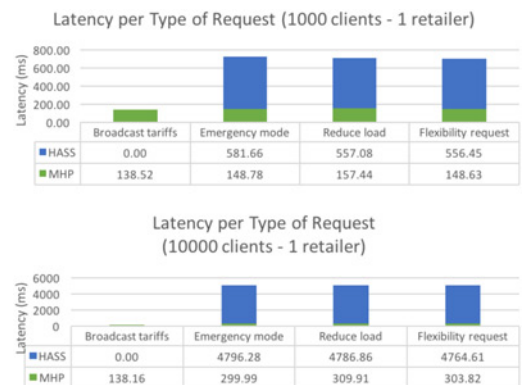


Fig. 8 Distribution of the request latency for 1000 and 10,000 clients

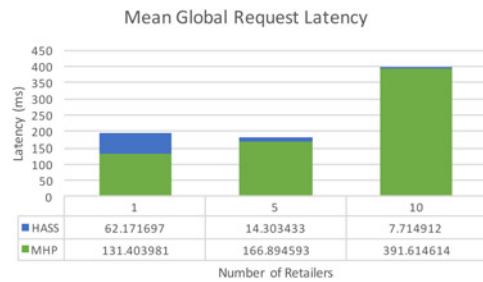


Fig. 9 Distribution of the request latency for 1, 5 and 10 retailers

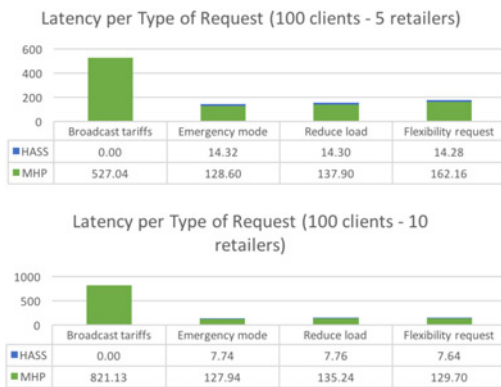


Fig. 10 Distribution of the latency request for 5 and 10 retailers

platform with HASS remains functional and capable of handling all the requests. In the unlikely event of a request peak, the MHP far exceeds the demanded performance requirements for this component.

5.4 Availability

5.4.1 Goal: Use fault injection to assess the behaviour of the MHP in the presence of faults, observing, e.g. how long it takes for the MHP to resume normal operation after a fault or failure.

In order to evaluate the fault tolerance of the proposed system, we inject failures of each of the MHP and HASS components.

Fig. 11 shows how request latency varies over time as measured from the 'DSO Simulator' for the base setup, to serve as a reference.

First, in one of the tests, the MHP was shut down during the execution and booted up some seconds later, in order to understand the impact of this action in the request latency.

In Fig. 12, the mean request latency aggregated per minute is shown for a 10 min run as measured by the 'DSO Simulator'. The latency represents the mean time needed for the MHP to process DSO requests, at the rate of 30 requests/min, and four types of requests. During the first minute, the latency is higher than in the rest of the run, since Glassfish is still loading the MHP and the

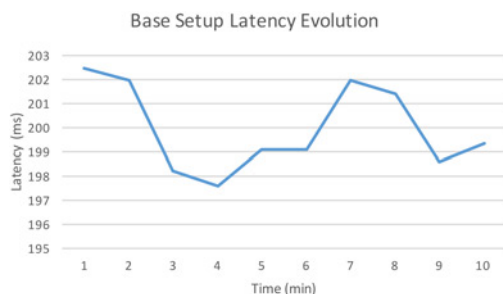


Fig. 11 Latency evolution for the base setup

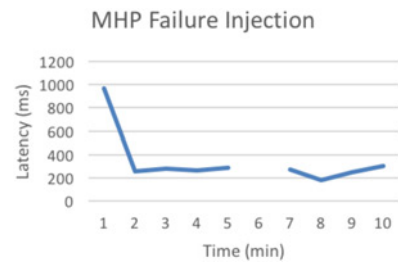


Fig. 12 Latency variation during a MHP failure

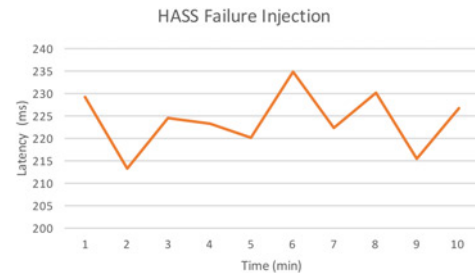


Fig. 13 Latency variation during fault injection in HASS components

DSO simulator, as well as loading into memory information as configuration files. At minute 5, the MHP was shut down, and no requests were processed. During minute 6, the MHP was booted up, taking about 30 s to load. The first request to arrive, during minute 7, had a slightly higher latency, but in the following minutes (8–10) tended to converge to values similar to minutes 3–5. In conclusion, a failure in the MHP causes a loss of requests, and a slightly delay increase in the minutes following its reboot. Since the MHP component itself is stateless and necessary data is persisted in the HASS, boot-up in case of failure is quick and it is sufficient for the DSO (or other agents) to simply repeat a failed request.

The availability test consisted in killing several components of the HASS and verifying the impact in the delay of the requests. In this specific test, the killed components consisted of the HBase Master, the Apache Zookeeper, the HDFS JournalNode and the HDFS NameNode all co-located in Host 1, i.e. the active HBase Master and HDFS NameNode.

Fig. 13 shows how the request latency evolved during this experiment, as measured from the 'DSO Simulator'. These components were killed at minute 5, and booted up again in minute 6. As a consequence, during minute 6, there was a slight increase in the request latency. Even so, the variance is request latency that remains in the order of 20 ms, with the system remaining available throughout.

In short, regarding availability, the performed tests show that the MHP, using HAAS, is able to tolerate or recover from failures without major increases in the latency times. Indeed, the performed tests have shown that the impact is barely noticeable, even when considering fairly larger sets of clients and request rates than expected in the demonstrator.

6 Conclusions

The performed tests focused on the functional and non-functional evaluation of the MHP's behaviour in both real case scenarios and stress scenarios. It was shown that the developed platform fulfils the correctness, performance and availability requirements for its role in the Portuguese Demonstrator in the context of the UPGRID project, where the MHP is being deployed within the EDP Distribuigao domain, and allows the DSO to enable the market agents to actively contribute to the grid management. Additionally,

HEMS units installed in some clients' households create an increased layer of information and control allowing the DSO to evolve towards the integration of new data, thus assuming a central role as an enabler of the future electricity system.

7 Acknowledgments

The research leading to this work was being carried out as a part of the UPGRID project (Real proven solutions to enable active demand and distributed generation flexible integration, through a fully controllable low-voltage and medium-voltage distribution grid),

funded by the European Union's Horizon 2020 Framework Programme under grant agreement No. 646.531.

8 References

- 1 Smart Grid Task Force: 'EG3 first year report: options on handling smart grids data', Expert Group 3, 2013
- 2 van den Oosterkamp, P., Koutstaal, P., van der Welle, A., *et al.*: 'The role of DSOs in a smart grid environment', ECORYS consultation report for the European Commission, DG ENER, 2014
- 3 Stromsather, J., Marretta, L., Soderstrom, P.: 'Opening up for a more competitive energy market with new energy services by making 'real time' metering data accessible to market players'. 23rd Int. Conf. on Electricity Distribution (CIRED 2015), Lyon, France, 2015