Towards a Cloud-based Platform Architecture for a Decentralized Market Agent

Oliver Kopp[1] Michael Falkenthal[2] Niklas Hartmann[3] Frank Leymann[4] Holger Schwarz[5]
Jessica Thomsen[6]

Abstract: Reorganization of power generation, thereby replacing conventional energy sources by innovative renewable energy sources, demands a change in distribution grid structure and operation. The foreseen Decentralized Marked Agent is a new role in the energy market sector accomplishing not only trading on energy and operating reserve markets but also regulating flexibilities at the distribution grid level, such as energy storage and decentralized energy generators, and thereby considering system services and securing system stability. This paper presents requirements on an IT system to support this new role. We design an architecture matching these requirements and show how Cloud computing technology can be used to implement the architecture. This enables data concerning the distribution grid being automatically gathered and processed by dedicated algorithms, aiming to optimize cost efficient operation and the development of the distribution grid.

1 Introduction

To match the objectives of the German energy transition [BM10], excessive changes will be necessary. Shifting towards power plants based on renewable energy sources consequently not only demands predominantly innovative technologies but also a change in the energy supply structure. That means, the current centralized energy supply structure composed of large central power plants must be replaced by a structure consisting of more decentralized power plants. Thus, the distribution grid structure, operation, and expansion have to adapt to new energy generation patterns. Considering these arising or already existing challenges, the new market role "Decentralized Market Agent" (DMA) is envisioned in the energy sector [Th15]. The DMA addresses problems in coordination of the synergy of energy generation, storage systems, and variety of the consumers on distribution grid level. Additionally, in order to ensure a cost efficient operation and expansion, the distribution grid itself is an important factor in future energy systems. To realize this task, the DMA has to be efficiently

[1] Institute for Parallel and Distributed Systems, Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany, kopp@ipvs.uni-stuttgart.de

[2] Institute of Architecture of Application Systems, Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany, falkenthal@iaas.uni-stuttgart.de

[3] Fraunhofer-Institut für Solare Energiesysteme ISE, Energy Systems and Markets, Heidenhofstraße 2, 79110 Freiburg, Germany, niklas.hartmann@ise.fraunhofer.de

[4] Institute of Architecture of Application Systems, Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany, leymann@iaas.uni-stuttgart.de

[5] Institute for Parallel and Distributed Systems, Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany, schwarz@ipvs.uni-stuttgart.de

[6] Fraunhofer-Institut für Solare Energiesysteme ISE, Energy Systems and Markets, Heidenhofstraße 2, 79110 Freiburg, Germany, jessica.thomsen@ise.fraunhofer.de

supported by IT. During the past years, Cloud computing established itself as an IT cost saving alternative and enabled companies to focus on their core business [VHH12]. Thus, Cloud computing seems to be a suitable basis especially due to the associated paradigm of rapid provisioning of IT resources coupled with expenditures for only actually used resources. Based on these facts, we envision a Cloud-based platform enabling the DMA to fulfill its dedicated task. The Cloud-based platform is realized by an IT system that is aligned and optimized to the needs of the DMA.

In this paper, we exemplify the DMA and its environment in Sect. 2. In Sect. 3 we abstract from the DMA environment and identify the core components and concepts, which have to be supported by an IT system. Based on that, we derive requirements on a platform in Sect. 4 and propose a platform fulfilling these requirements in Sect. 5. The Cloud-readiness of that platform is discussed in Sect. 6. Related work is the focus of Sect. 7. Finally, we conclude the paper and provide an outlook on future work in Sect. 8 .

2 Overview on the DMA

With regard to the energy transition the DMA aims to incorporate all elements located on distribution grid level ensuring cost efficient operation. Thus, the DMA represents the aggregated entities at the central markets and will not only trade on energy-only and reserve markets, but will also consider system services and secure system stability. By

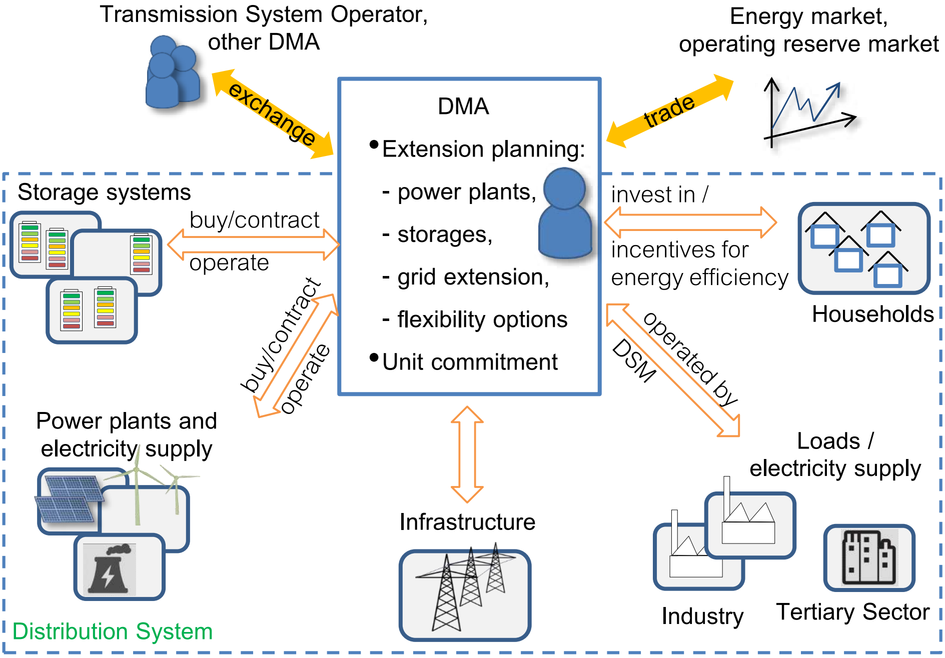

Fig. 1: The DMA in the liberalized electricity system [Th15]

coordinating decentralized elements while simultaneously trading at the central markets, the expansion of energy generation, storage, and other flexibility capacities can be stimulated based on market incentives. At the same time, the DMA is responsible for an efficient system operation. The new actor can benefit from aggregating existing information and, hence, from employing more flexibility options within the annual operation. Furthermore, aggregating information from generation, consumption, and grid operation additionally enables a more system friendly expansion of each distribution grid system. The embedding of the DMA in the liberalized electricity system is presented in Fig. 1. In general it becomes apparent that the DMA focuses on two major activities: First to optimize a distribution grid in short term schedules, i. e., in an interval of minutes, in order to regulate its actual utilization, and second to optimize the constructional development and improvement of the grid's topology and its physical components. The latter optimization scenario envisioned to be performed several times a year.

3 Abstracted Environment of the DMA

Mathematical approaches are the basis for short term as well as for long term optimization. The approaches rely on models of the distribution grid that are optimized based on specifically defined optimization criteria. These criteria are specific for each instance of a DMA, i. e., for each distribution grid, because the models have to take into account the local conditions of the distribution grid, which is managed by the DMA. One optimization criteria could be to achieve the cost optimum for the whole distribution grid while strictly avoiding any disadvantageous situations that could lead to blackouts of the grid. In this scenario especially critical factors such as weather conditions have to be considered as they have strong effects on the power generation by renewable energies. Another optimization criteria would be to minimize the emission of green house gases. In this scenario the major goal consists of avoiding fossil fuels and covering as much power generation as possible using renewable energy sources. In real world scenarios, however, a combination of several optimization criteria being weighted against each other seems to be feasible to leverage the DMA's needs.

In principle sensors and actuators form the two major components the DMA has to deal with. Sensors are various meters, brought out in the distribution grid acting as data sources for the DMA. Meters are mainly proprietary devices collecting data in heterogeneous formats, intervals, and details. Actuators on the contrary are flexibility points in the distribution grid that can be regulated by the DMA. They might be points to manipulate the power generation within the distribution grid, such like block-type thermal power stations or wind power plants. Additional points might be flexibilities regulating power consumption and storages in the form of batteries and pumped storage hydro power stations. Finally, agents interacting on energy markets, where energy can be traded and sold to nearby distribution grids, are realizations of the actuator concept. Thus, the concept of an actuator subsumes every logic and control unit of the distribution grid and all other regulation options to influence the distribution grid.

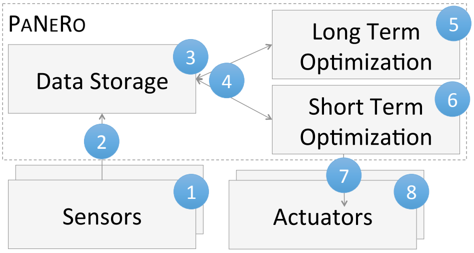

Fig. 2: Components of the DMA environment

The algorithms implementing the short term and the long term optimization operate on historic data about the distribution grid. Therefore, it is important to meter data from sensors and to store them over time. Thus, the platform of the DMA can conceptually be described as a data storage component and two optimization components. The data storage collects metering data from the sensors whereas the two optimization components perform the long term and the short term optimizations. Based on the outcomes of the short term optimization algorithm the actuators are regulated. We call the platform of the DMA PaNeRo: **Pla**tform for the **Ne**w market **Ro**le. It is illustrated in Fig. 2, which additionally shows eight variation points being discussed in the next section where we discuss requirements on PaNeRo.

4 Requirements on the Platform Architecture

The main purpose of a DMA is both to manage energy supply at the level of a distribution grid and to develop and extend a distribution grid based on optimization criteria. To automate data collection and analysis, an IT system is required. In total eight variation points have been identified and depicted as numbered circles in Fig. 2. They emphasize specific requirements to PaNeRo. The requirements have been derived during discussions in the project and are based on the expertise of the involved cooperation partners from the fields of distribution grids, power plant construction, data analysis, database technologies, and Cloud application architecture. This paper presents the current state of the discussion. Currently, concrete requirements on time-constraints are missing. That means, the maximum time, the data store may take to process a query and the maximum time the short term optimization algorithm may take to create commands for the actuators is not fixed yet.

Below, elicited requirements are consecutively stated for all eight variation points. The range of feasible realization options are discussed for each point to show the complex space of possible solutions, which, of course, lead to a vast number of possible concrete realizations of PaNeRo. Nevertheless, the summary of requirements and possible solutions (see Sect. 5) document major design decisions to make in order to architect and develop an IT system for the DMA. The stated requirements and solutions are not only applicable to develop PaNeRo. Since PaNeRo is only one realization of an IT system to support the management of a distribution grid the raised requirements and solutions can conceptually be used as a basis for the development of a whole archetype of IT systems.

Requirement 1: Support heterogeneous sensors

The sensors (variation point 1) may offer their data using a representational state transfer (REST) or Web Service endpoint, or by protocols such as IEC 61850 [IE13] or IEC 62056-21 [IE02], which are both standards in the domain of digital control systems. It must be possible to connect all these sensors to the platform to obtain the values (R1).

Requirement 2: Secure, loosely coupled, and reliable connection to the sensors

The connection of the sensors to the platform (variation point 2) should be secure (R2a). Security is required to ensure privacy and avoid manipulation by third parties [Fe14a].

The connection of the sensors to the platform should be loosely coupled (R2b). Typically, loose coupling means autonomy in the cases of platform, reference, time, and format [Fe14b]. *Platform autonomy* states that differences in platforms, e. g., regarding data representation or processor architecture, are unimportant for the communication. That is required to enable the sensors and PANERO to be implemented on different platforms. *Reference autonomy* states that the concrete network address of an endpoint should not be known. That is required to enable the platform to be scaled when new sensors are deployed during runtime. *Time autonomy* states that the receiver has not to be online when the sender is sending data. Also, communication partners may send and receive information at their own speed, thus, if one communication partner transmits information faster than another one, both remain operational. *Format autonomy* states that the data format used for exchange should be understood by both the sender and receiver. That format may be different from the format used internally.

Since some sensors can provide data about critical infrastructures of a distribution grid, those have to be reliably transmitted to the platform. Reliable transfer causes overhead in the communication. Therefore, it is enough to support reliable transfer only for sensors of which the values are critical for the optimization algorithms. As a result, the connection has to be reliable for specific sensors (R2c).

Requirement 3: Data store scaling and triggering capabilities

The data store (variation point 3) has to scale with the connected sensors (R3a). That means, if more sensors are connected or the rate of sensor data is increased, the data store has to be able to handle these increasing loads in terms of storage capacity and the number of input and output operations.

The platform has to be able to trigger the short term optimization in the case exceptional data arrives (R3b). Examples for exceptional data are values differing by a certain threshold from other values or values indicating an outage of a generation unit.

Requirement 4: Suitable data format and support for time series

The data store has to offer (variation point 4) the data in a format required by the optimization algorithms (R4a). This is required as it is assumed that the data store can convert the data more efficiently than the optimization algorithms. Further, this requirement enables to introduce a separate of concerns by separating data preparation logic from the optimization logic of the algorithms. Thus, data preparation logic does not pollute the optimization algorithms so that developers can concentrate on the actual valuable logic of the algorithms.

Further, one sensor date is always related to one specific timestamp, thus, sensor data has in general to be handled as time series. Since data connections to sensors are not guaranteed to be reliable in some cases (see variation point 2) and data sampling rates differ, the data store should provide functionality to interpolate data. As a result the data store should offer time series functionality such as selection of data ranges, aggregation of values, or interpolation of missing data (R4b). Alternatively, the optimization algorithms can do the interpolation by themselves. It is, however, assumed that the data store will perform these operations more efficient than the optimization algorithms. Providing such time series functionality natively by the platform also enables benefits by means of separation of concerns regarding the optimization algorithms as already mentioned above for the preparation of appropriate data formats, accordingly.

Requirement 5: Support for function shipping

Function shipping versus data shipping is a discussion for gaining performance speed-up. Function shipping describes that functions, which in case of PANERO contain logic of optimization algorithms, are shipped to parts of a system where they can optimally run close to the data in order to avoid tedious data transfers [Or00]. By contrast, data shipping describes the process that data is queried by a component and shipped to the component for processing. To achieve performance improvements regarding the optimization components (variation points 5 and 6), it has to be possible to ship the implemented optimization algorithms to the database (R5).

Requirement 6: Secure, reliable, loosely-coupled, and time-constraint connection to the actuators

The commands sent to the actuators should not be viewable or changeable by third parties. Therefore, the connection to the actuators (variation point 7) has to be secure (R6a).

It has to be assured that the commands arrive at the actuators. Therefore, connection and data transfer to the actuators have to be reliable (R6b).

The message format has to be generic to enable loose coupling between the platform and the actuators (R6c). The same arguments as presented in the context of R2b apply.

The commands sent to the actuators have to arrive within a certain time frame. Therefore, the connection to the actuators has to offer delivery guarantees with respect to the maximum delivery time (R6d).

Requirement 7: Plausibility checks

The actuators (variation point 8) may evolve during the runtime of the optimization algorithm. Therefore, the actuators have to be enabled to do a plausibility check of the received commands to avoid implausible regulations (R7). Regarding energy generation and storage systems this means that actuators guarantee operation within predefined thresholds to prevent failures.

5 Proposed Platform Architecture

Fig. 3 presents an architecture satisfying the aforementioned requirements. The general idea of this architecture is to introduce (i) data gateways serving as an abstraction layer to cover the complexity of sensors and actuators and (ii) to use message queuing to enable specific qualities of service regarding the connections of the sensors and actuators to the platform. The data gateway is responsible to transform the data gathered by the sensors to an intermediate format. It is also used to transform the commands received by actuators to their proprietary format. Furthermore, message queuing is a popular technique to implement loose coupling [HW03]. In the following, we discuss how the proposed architecture satisfies the requirements.

To support heterogeneous sensors (R1), a data gateway is introduced. It is a component forming the gateway between concrete sensors as well as actuators and PANERO. In case the sensors offer their data using a REST endpoint, they can be directly connected to a message queue. In case the sensors offer their data using protocols such as IEC 61850, they can be connected using the OpenMUC platform [Op15], which is an open source software framework for monitoring and control applications in smart grids, and a custom plugin for OpenMUC connecting to the message queue.

To satisfy R2b (loose coupling), a message queuing system (MQS) [HW03] is introduced. The queues managed by the MQS are shown in the figures as pipes. By using a MQS, all the properties of loose coupling can be achieved. Platform autonomy is achieved, since MQS implementations are available for different platforms and can communicate with each other. Reference autonomy is achieved, because the sender puts its message in its local queue and the MQS takes care about the routing. Time autonomy is achieved, because the sender has to put the message in its local queue and the MQS takes care about forwarding it to the queue of the receiver. This way sender and receiver are decoupled since the receiver can process messages from its queue one by one and independently from any sender. Format autonomy is achieved, because MQS allow to convert messages to any appropriate data format during transfer if required. When choosing a concrete MQS, the system has to offer

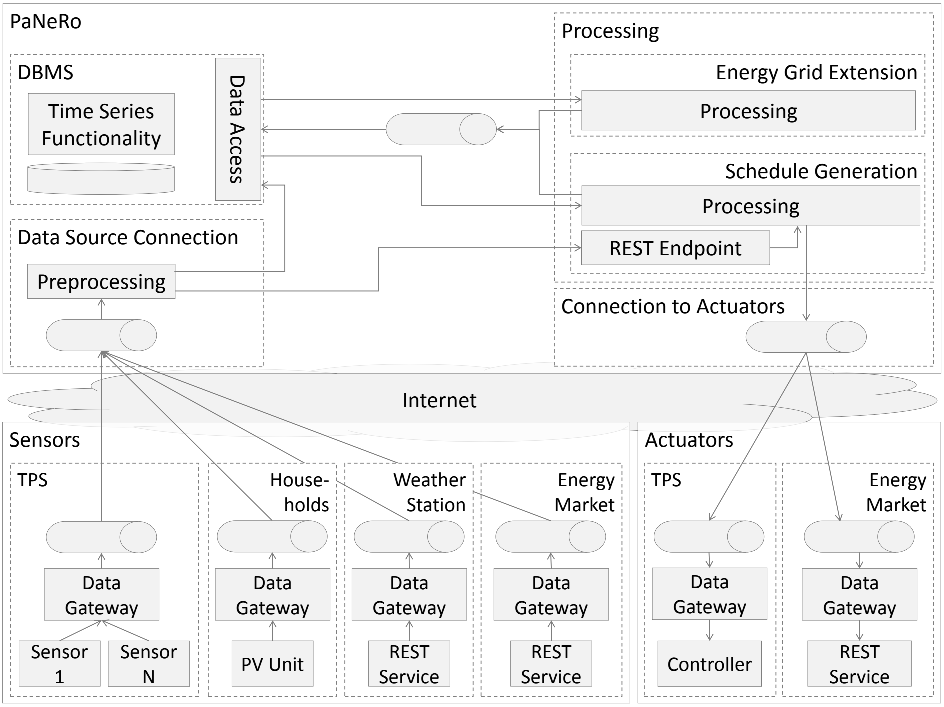

Fig. 3: Proposed Platform Architecture

security (R2a) and optionally also reliability (R2c), which implies that protocols such as the Message Queuing Telemetry Transport (MQTT; [OA14]) are not feasible in such cases.

Another option to accomplish security, loose coupling, and reliability, is to use Web Service technology with WS-Security and WS-ReliableMessaging [We05]. Typically, an enterprise service bus [Jo07] is used to implement loose coupling in the context of Web Services. Although Web Services offer a higher degree of interoperability especially because of the availability of more clients, we claim that message queuing systems offer enough interoperability in the context of the DMA, while keeping the overall architecture more lightweight and less complex.

A database system is chosen to satisfy R3a (scaling data store). The current requirements do not impose a certain type of data base system. Therefore, any type of database system (relational or NoSQL [Gr13]) can be chosen. For triggering an optimization (R3b), an analysis of the inbound data has to be implemented, which is typically addressed by a complex event processing (CEP) system [Lu02]. The current requirements do not impose selection criteria for a concrete CEP system.

The usage of a database system enables the optimization algorithms to receive the data in the required format (R4a), because format transformations can be implemented near the database, while a selection criterion for a proper database system is to provide functions for time series calculations natively (R4b).

Although the requirements leave it open whether the optimization components should be placed near the database or can reside in a separate container, PANERO should offer a plugin possibility to directly deploy optimization algorithms (R5). This might be realized by stored procedures or other ways of tightly coupling implementations and the database system to benefit from a cheap data transfer between the database and the algorithm.

The connection to the actuators is done using a MQS, similar to the connection of sensors. When securing the connections, for example via transport layer security (TLS), this setting ensures security (R6a) and reliability (R6b). The concrete message format is not yet specified. When designing the concrete format, it will be able to cover all commands offered by the actuators and will be extensible to be able to express new commands (R6c). To guarantee that commands are available at actuators within specified timeframes the chosen MQS has to offer real-time capabilities (R6d). This might be the ActiveMQ Real Time sub-project [Ap15].

The data gateway used at the actuators enables a plausibility check of the commands (R7). This functionality has to be implemented for each actuator regarding to their operational constraints.

6 Cloud-readiness of the Platform

Cloud computing is a new innovative paradigm for offering and using IT resources: Resources can be used on demand and users do not need to care about the underlying infrastructure [Le09]. The primary reason for using Cloud technology is the possibility to move from capital to operational expenditure, which promises to save money [Ar10]. Currently, there are four different Cloud offerings available which can be distinguished as follows: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Composite as a Service (CaaS) [Le09, MG11]. In the case of IaaS, hardware is virtualized and the user has to install the software components by himself. On the contrary PaaS, software components such as databases or message queue systems are offered by the Cloud provider and the user does not need to take care about maintenance or scalability. However, if a database is offered in the Cloud, some authors claim that this is not a PaaS offering, but a separate service called Database as a Service—DbaaS [SK11]. Nevertheless, we follow the categorization by den Haan [dH13] and consider a database offering as a PaaS offering. In the case of SaaS, a complete software solution is offered as service. If, however, services are combined aiming to provide a new service in the Cloud, this is called CaaS.

The architecture of PANERO allows for deploying its components in the Cloud and, thus, makes use of the benefits of Cloud computing. The Data Source Connection is provided as a virtual machine image, which can be deployed multiple times depending on the number of incoming messages per second. For the database, a database PaaS offering can be used. Here, the scaling, backup and recovery are ensured by the Cloud provider. The processing itself can be hosted on a virtual machine. The whole system will be offered as Composite as a Service to enable distribution grid providers to setup the platform when required without the need of manual intervention. For the automatic setup, the Topology and Orchestration Specification for Cloud Applications (TOSCA) standard is well suited [Bi12] as it provides a standardized way to package applications and their management plans into so called cloud service archives (CSAR). TOSCA provides a means to describe the components of a Cloud application, i. e., its composite structure, in a Cloud provide agnostic way. Summarized, TOSCA enables to easily deploy applications to different Cloud environments.

7 Related Work

For big data analysis, the Lambda Architecture [MW15] is a common pattern distinguishing two layers for the analysis: First, the batch layer to compute results based on the complete set of data. Second, the speed layer to provide results based on new data not yet available for the batch layer. The Lambda Architecture is applied in the approach presented by Strochbach et al. [St15] performing power quality analytics, real-time grid monitoring, and forecasting energy demand. Real world application is shown in the projects Big Data Public Private Forum (BIG) [BI15] and Peer Energy Cloud [PE15]. For the speed layer, they use a CEP system [Lu02]. In our architecture, there is no need for a speed layer as the processing algorithms always query the most recent data from the database. A CEP system may be used at the data source connection, where exceptional events can trigger an analysis at the processing. In the Peer Energy Cloud project, users can access their metered values. Security in that context is discussed by Ebinger et al. [Eb13]. PANERO currently does not yet offer its users a dashboard, but might include one and then will use the concepts presented in that paper.

Busemann et al. [Bu13] present a platform for processing readings of individual devices deployed at households. Although being similar to our architecture, the main difference is our employment of loose coupling between the sensors and the platform, and focusing on data originating from metering devices of a whole distribution grid.

In accordance to our proposal, Hauer et al. [Ha13] also see challenges regarding the shift towards renewable energies. They present an architecture for a future energy distribution system, but differ again in the use of loose coupling technologies.

Gustafsson et al. [Gu14] use the principles of Service-oriented Architecture (SOA) in district heating substations. They use SOA principles and the Web Service technology to connect sensors with a server. Our approach, in contrast, does not offer services using Web Service technology, but establishes loose coupling using message queuing technology.

IEC TR 62357 [IE12] provides an overview on the standards by the IEC technical committee 57 (“Power systems management and associated information exchange”). In our architecture, these standards may be used for the communication of the sensors with the data gateway where required. For the communication with PANERO, we use a MQS and a generic message format to reduce the complexity of heterogeneous data formats of the sensors. Since PaNeRo only has to handle one well defined data format, this method saves additional effort.

8 Conclusion and Outlook

Motivated by the idea of a Decentralized Marked Agent, we derived requirements on an IT platform supporting the DMA. Based on the requirements, we outlined an architecture for the platform. The current requirements, however, do not explicitly point towards a certain type of a database system. As a consequence, we are recently in the process to evaluate whether relational or NoSQL databases are more applicable for the implementation of the data store within the platform. We did not discuss issues on connecting energy markets. A concrete description of the implementation of the respective data gateway will be provided as future work. The next step is to implement the platform and evaluate it in real-world settings of two different distribution grids. Our current research state indicates that the proposed short term optimization can be implemented by algorithms using the capabilities of a time series data base. Other research results indicate that it is also possible to use complex event processing [JZ14]. Thus, we will compare our implemented solution with CEP-based solutions. We mentioned the requirements on the security of smart meters [Fe14a], but did not do an evaluation of the proposed realization. This will be part of our future work after a prototype has been implemented.

Acknowledgments. The research leading to these results has received funding from the German government through the BMWi project NEMAR (03ET4018).

References

[Ap15] Apache Software Foundation: ActiveMQ Real Time, 2015. http://activemq.apache.org/activemq-real-time.html.

[Ar10] Armbrust, Michael et al.: A View of Cloud Computing. Commun. ACM, 53(4):50–58, April 2010.

[Bi12] Binz, Tobias; Breiter, Gerd; Leymann, Frank; Spatzier, Thomas: Portable Cloud Services Using TOSCA. IEEE Internet Computing, 16(03):80–85, May 2012.

[BI15] BIG: BIG Project Website, 2015. http://www.big-project.eu/.

[BM10] BMWi: Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung, 2010.

[Bu13] Busemann, C. et al.: Integrating Sensor Networks for Energy Monitoring with Service-Oriented Architectures. International Journal of Distributed Sensor Networks, 2013:1–12, 2013.

[dH13] den Haan, Johan: The cloud landscape described, categorized, and compared, October 2013. http://www.theenterprisearchitect.eu/blog/2013/10/12/the-cloud-landscape-described-categorized-and-compared.

[Eb13] Ebinger, Peter et al.: Privacy in Smart Metering Ecosystems. In: Smart Grid Security. Springer, 2013.

[Fe14a] Federal Office for Information Security: Protection Profile for the Security Module of a Smart Meter Gateway (Security Module PP), December 2014.

[Fe14b] Fehling, Christoph et al.: Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications. Springer, 2014.

[Gr13] Grolinger, Katarina et al.: Data management in cloud environments: NoSQL and NewSQL data stores. J Cloud Comput Adv Syst Appl, 2(1):22, 2013.

[Gu14] Gustafsson, Jonas et al.: Application of Service Oriented Architecture for Sensors and Actuators in District Heating Substations. Sensors, 14(8):15553–15572, Aug 2014.

[Ha13] Hauer, I. et al.: Concept, architecture and components of a smart distribution control center. In: PowerTech. IEEE, 2013.

[HW03] Hohpe, G.; Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, 2003.

[IE02] IEC: IEC 62056-21:2002 – Electricity metering – Data exchange for meter reading, tariff and load control - Part 21: Direct local data exchange, 2002.

[IE12] IEC: IEC TR 62357-1:2012 Power systems management and associated information exchange - Part 1: Reference architecture, 2012.

[IE13] IEC: IEC TR 61850-1:2013 Communication networks and systems for power utility automation - Part 1: Introduction and overview, 2013.

[Jo07] Josuttis, Nicolai M.: SOA in Practice: The Art of Distributed System Design. 2007.

[JZ14] Jerzak, Zbigniew; Ziekow, Holger: The DEBS 2014 grand challenge. In: DEBS. ACM Press, 2014.

[Le09] Leymann, Frank: Cloud Computing: The Next Revolution in IT. In: Proc. 52th Photogrammetric Week. Wichmann Verlag, pp. 3–12, September 2009.

[Lu02] Luckham, David: The Power of Events. Addison-Wesley, 2002.

[MG11] Mell, Peter; Grance, Timothy: The NIST Definition of Cloud Computing. online, October 2011. SP 800-145.

[MW15] Marz, Nathan; Warren, James: Big Data: Principles and best practices of scalable realtime data systems. Manning Publications, 2015.

[OA14] OASIS: MQTT Version 3.1.1, OASIS Standard, 2014.

[Op15] OpenMUC: OpenMUC, 2015. `http://www.openmuc.org`.

[Or00] Oracle Corporation: Function Shipping: Separating Logical and Physical Tiers, 2000. `http://docs.oracle.com/cd/A87860_01/doc/appdev.817/a86030/adx16nt4.htm`.

[PE15] PEC: Peer Energy Cloud project website, 2015. `http://www.peerenergycloud.de/`.

[SK11] Seibold, Michael; Kemper, Alfons: Database as a Service. Datenbank-Spektrum, 12(1):59–62, November 2011.

[St15] Strohbach, Martin et al.: Towards a Big Data Analytics Framework for IoT and Smart City Applications. Modeling and Processing for Next-Generation Big-Data Technologies, pp. 257–282, 2015.

[Th15] Thomson et al.: Darstellung des Konzepts – DMA Decentralised Market Agent – zur Bewältigung zukünftiger Herausforderungen in Verteilnetzen. In: Informatik. 2015.

[VHH12] Vossen, Gottfried; Haselmann, Till; Hoeren, Thomas: Cloud Computing für Unternehmen. dpunkt.verlag, 2012.

[We05] Weerawarana, Sanjiva et al.: Web Services Platform Architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall PTR, 2005.

All links were last followed on April 15, 2015.