



# Secure Automated Home Energy Management in Multi-Agent Smart Grid Architecture

Hisain Elshaafi<sup>1</sup> · Meritxell Vinyals<sup>2</sup> · Ivan Grimaldi<sup>3</sup> · Steven Davy<sup>1</sup>

Received: 15 August 2017 / Accepted: 29 March 2018 / Published online: 12 April 2018  
© Springer Nature Singapore Pte Ltd. 2018

## Abstract

This paper describes an approach to decentralised and automated demand response and home energy management that takes into consideration privacy and security of home users implemented using a multi-agent system. The novel approach allows the management of flexibility within the low-voltage part of the electricity distribution networks where decentralised decision making will result in better value for prosumers, efficiency and market competitiveness. The hierarchical agent architecture allows agents that represent different stakeholders to make decisions as a result of changes in energy consumption and generation using mechanisms such as optimisation and access control. The multi-agent system connects to home devices through a uniform software layer that abstracts the heterogeneity of home devices.

**Keywords** Multi-agent system · Demand response · Home energy management · Flexibility · Access control · Device abstraction

## Introduction

### Motivation

Energy demand is globally increasing largely due to the advances in technology and increased use of electronic devices and home appliances [1]. The increasing consumption is accompanied by the progress in the development of the Internet of Things and smart grid technologies as well as the growing awareness of the environmental effect of carbon-based fuels. Smart grid technologies are transforming both the infrastructure and operation of the electric grid. Some of the fundamental aspects of this transformation is the flexibility of consumption, *distributed energy resources (DERs)* and decentralised intelligence [2]. The use of *Multi-Agent Systems (MASs)* in smart grids has been considered in a number of applications due to their autonomy, pro-activity and social ability which facilitate some of the important smart grid characteristics such as flexible communication,

decentralised control, scalability and functional extensibility [3]. Such characteristics are important in developing *Demand Response (DR)* and *home energy management systems (HEMS)* which can consequently enhance the smart grid in terms of efficiency and reliability as well as reducing implications on the environment, customer bill savings and competition between suppliers [4, 5].

### Literature Review

Multi-agent systems and intelligent agents are among the promising fields that can support a smart grid that is decentralised, autonomous, heterogeneous and complex in a wide range of applications [2, 3, 6, 7]. Such applications include for example, DR simulation models proposed by Sarvapali et al. [8], Fazal et al. [9] and Gomes et al. [10]. Wu et al. [11] propose a solution that aims to minimize energy charges under a *Time of Use (TOU)* tariff in a way that takes into consideration home power demand, plug-in electric vehicle (PEV) charging need and the fluctuation of solar power. The authors in their more recent work [12] utilise convex programming to optimize the control decision and parameters of home battery energy storage systems in smart homes with PEVs and photovoltaic (PV) arrays. Karfopoulos et al. [13] develop a MAS-based HEM that aims to allow prosumers to provide DR services for grid support taking into consideration both local energy

✉ Hisain Elshaafi  
hisain@gmail.com

<sup>1</sup> Waterford Institute of Technology, Waterford, Ireland

<sup>2</sup> Commissariat à l'énergie atomique et aux énergies alternatives, Paris, France

<sup>3</sup> Telecom Italia, Rome, Italy

resources and household convenience. Radhakrishnan and Srinivasan [14] propose a MAS based distributed energy management system to perform optimal energy allocation and management of renewables, storage and distributed generation.

Alternative technologies have also been investigated for decentralised power system applications e.g. service-oriented architectures (SOA) and cloud computing [15–17]. However, MAS has shown to be more suitable to the smart grid scenarios involving autonomous behaviours [6].

Researchers have investigated the benefits and challenges of DR and energy management systems including communication aspects [6, 18, 19]. The Universal Smart Energy Framework (USEF) [20] is a model for developing smart grid products and services. It also describes a framework for processes and types of interaction between grid actors. Our paper uses USEF as a reference model to ensure compliance of the solution to the energy industry's vision of the smart grid.

There have been several proposed mechanisms and research activities in the area of home energy management particularly focusing on optimisation algorithms. Mohsenian-Rad et al. [21] propose a game theory based, autonomous and distributed demand-side energy management system using the two-way smart grid communication infrastructure. Molderink et al. [22] describe a control methodology to support a hierarchical non-agent based coordination of multiple home energy management systems to maximize global energy efficiency.

Security and privacy are major challenges in HEMS, DR and the smart grid in general [23, 24]. Researchers e.g. [25–27], propose that access control is one of the principal methods of defending against cyber-attacks and assuring confidentiality, integrity and availability in smart grid communications. Namboodiri et al. [28] describe a security framework that addresses some of the vulnerabilities within the HAN but does not consider external security and privacy issues. Rottondi et al. [29] propose a privacy-friendly framework to coordinate energy consumption using Shamir secret sharing scheme. Such et al. [30] propose a security platform for multi-agent systems that preserves user privacy and anonymity in agent groups.

Existing policy-based access control solutions that follow the XACML policy language specification have the flexibility of the policy language to effectively express advanced policies. They can also take advantage of existing policy analysers and processors to ensure that policy sets are optimal. However, current mechanisms to evaluate XACML policies are slow and do not perform in situations where a policy decision is required in near real-time, or when there is a high volume of policy requests. For example, XEngine, an open source XACML engine [31], is not XACML v3.0 compliant and it is designed to only

handle simplified policies. It also cannot handle policies where there may be many different types of resources and group hierarchies. Ros et al. [32] extend XEngine with support for more XACML policy specifications and provide better performance than Sun XACML PDP [33]. In our solution, we implemented a technology stack with multiple languages and technologies exploiting their optimal features to maximise the efficiency and performance of access control.

## Paper Contribution

The related work discussed in the previous section has shortcomings such as being centralised solutions, the reliance of the optimisation techniques on simulations rather than realistic environments and the lack of or weak privacy considerations. In this paper, we describe a novel MAS communication architecture connected to other components in the grid and collectively providing secure energy flexibility management. The paper describes the architecture and validation in industrial labs with home devices that mirror operational environments. The contributions of this paper include the novel decentralised approach to DR and HEMS using a decentralised multi-agent system, agent communication protocols that organise the interactions between the agents and the scalable attribute based access control enforced by agents. Stakeholders in the architecture include *prosumers*, *aggregators* and *Distribution System Operators (DSOs)*. Prosumers are customers who can in addition to consuming energy, act as provider of DERs. Aggregators provide and manage the flexibility service. Access control is vital in ensuring security and privacy of prosumers in addition to allowing them to have granular control over their home devices. The entry point to the prosumer's home is called the *Energy Box* which runs software including the agent that allows communication with heterogeneous home devices through a uniform abstraction layer. The paper describes the approach of the communication architecture and evaluates its strengths and the difficulty that faces its adoption by the energy industry.

## Paper Outline

This paper is organised as follows. Section “[Use Cases](#)” discusses the use case scenario for the architecture structured into three levels. Section “[Multi-Agent Based Communication Architecture](#)” details the communication architecture, protocols, agents and external components. The external components include the access control system described in Section “[Access Control for Flexibility Services](#)” and the abstraction of home devices to provide a uniform interface to agents which is discussed in Section “[Abstraction of](#)

**Home Devices**". Section "Testing and Results" describes the validation of the architecture followed by discussion of the solution in Section "Discussion". Section "Conclusion" concludes the paper.

## Use Cases

The use cases is structured into three levels; Home, Aggregator and DSO levels. In the use cases, we describe the following types of agents:

1. *Device Agent*: This agent communicates with home devices using ZigBee. This agent also has access to the external services, such as access control, historical data and forecasting services.
2. *HEMS Agent*: This agent communicates with the Device Agent and the Aggregator Agent.
3. *Aggregator Agent (AGR)*: This agent is deployed at the aggregator facility and communicates over the Internet with HEMS and DSO Agents.
4. *DSO Agent*: This agent is deployed at the DSO facilities and communicates with the AGR agents over the Internet.

## Home Level

At the home level, prosumers are assumed to be equipped with smart appliances (e.g. smart washing machines) and energy technologies (e.g. DERs, Storage Systems, etc.) connected to an Energy Box. The main objective at this level is to reduce the prosumer's energy bill (i.e. by performing self-balancing, TOU optimisation and/or controlling the max load) while respecting the prosumer's comfort and preferences. In order to achieve this, the HEMS Agent controls appliances and optimises the use of flexibility according to the in-home optimisation objectives and the signals received from the grid. The basic agent communication structure at the prosumer's premises is shown in Fig. 1.

In order for the HEMS Agent to control and optimise the flexibility of the prosumer, it communicates with all the *Device Agents* that encapsulate the communication with the home devices. The current implementation of the HEMS Agent communicates with five devices types including a PV unit, a storage unit (S), an electric vehicle (EV), a smart appliance (i.e. a washing machine) modelled as a deferrable load (DL) and the external tie (ET) connection that represents the energy tariff that the prosumer contracted with the supplier. The communication with agents in the smart grid always goes through a HEMS Agent, which in addition to performing in-home optimization, serves as the first aggregation level. The HEMS and the Device Agents are hosted in the Energy Box.

At the beginning of the plan phase (i.e. as part of the Day-Ahead initialization), the HEMS Agent subscribes to each Device Agent to receive the available flexibility that the device can offer for the required interval of time (and any update that occurs afterwards). The flexibility that a device can offer depends on the prosumer preferences (e.g. thermostat control of the room heating), the forecasted device consumption/production and the current state of the device. To collect this data, the Device Agent connects to the relevant device via HTTPS to the Device Abstraction Layer using ZigBee communication. Once the HEMS receives the flexibility offered by the home devices, it performs an in-home optimization with the objective of minimizing the prosumer's energy bill. The outcome of the optimization includes the configuration of controllable devices and the expected prosumer consumption profile during the day of delivery (called the *P-Plan*). Finally, since the HEMS Agent is also in charge of realizing the P-Plan, it communicates with the Device Agent which then sends the corresponding control signals to the devices executing the corresponding flexibility requests.

## Aggregator Level

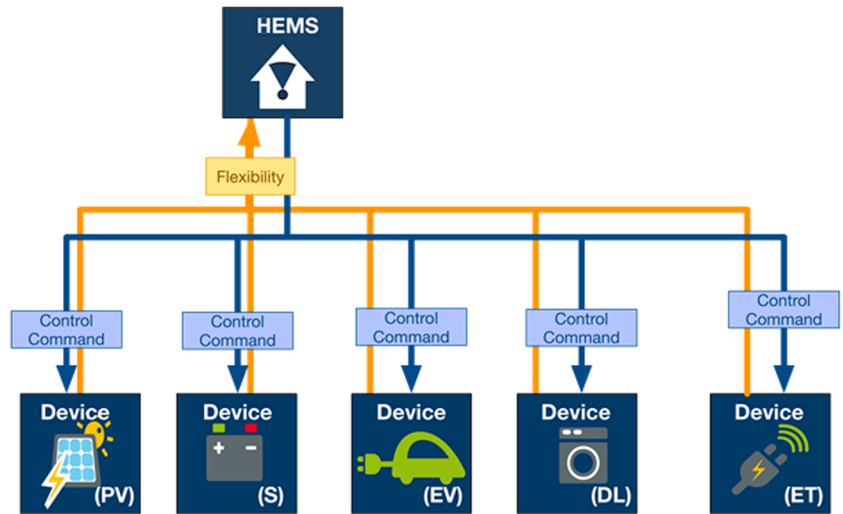
A local community of prosumers is created and managed by an *aggregator (AGR)* to collectively optimise energy flow inside the district without considering grid constraints. The first and second stages of flexibility trading among agents takes place at this level. At the beginning of the plan phase (i.e. as part of the Day-Ahead initialization), the AGR agent subscribes to each HEMS Agent in its portfolio in order to receive its P-Plan for that day (and any update that take place afterwards). From the aggregation of these plans the AGR agent will create its first *A-plan*. An A-plan is the expected consumption profile during the day of delivery for a given Aggregator portfolio of prosumers.

Subsequently, the first stage of optimisation among agents takes place. This stage consists of trading of flexibility among the HEMS that have subscribed to the same AGR (i.e. they are part of the same local prosumer community), with the aim of optimizing the AGR portfolio. Thereafter, the second stage of trading takes place, namely the trading of flexibility among the AGRs which results in the flexibility trading for the further optimisation of their portfolios. Figure 2 presents the interactions between agents at this level.

## District Level

A local community of prosumers is set up by an aggregator to collectively optimise energy flows inside the district in the presence of a grid congestion or capacity constraints. In other words, the third stage of flexibility trading among

**Fig. 1** Home level communication

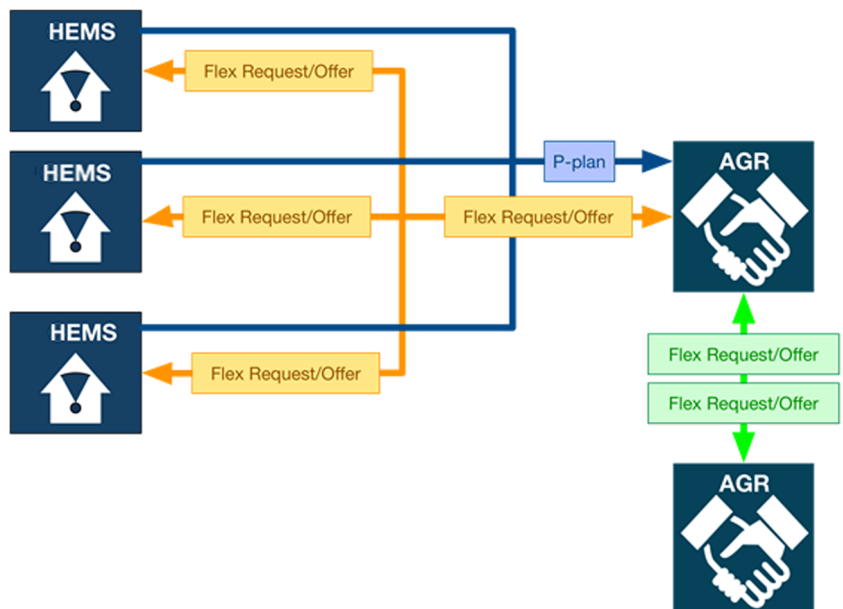


agents takes place between AGRs and DSO. Figure 3 provides a high-level overview of the different agent-to-agent communications that take place at this level.

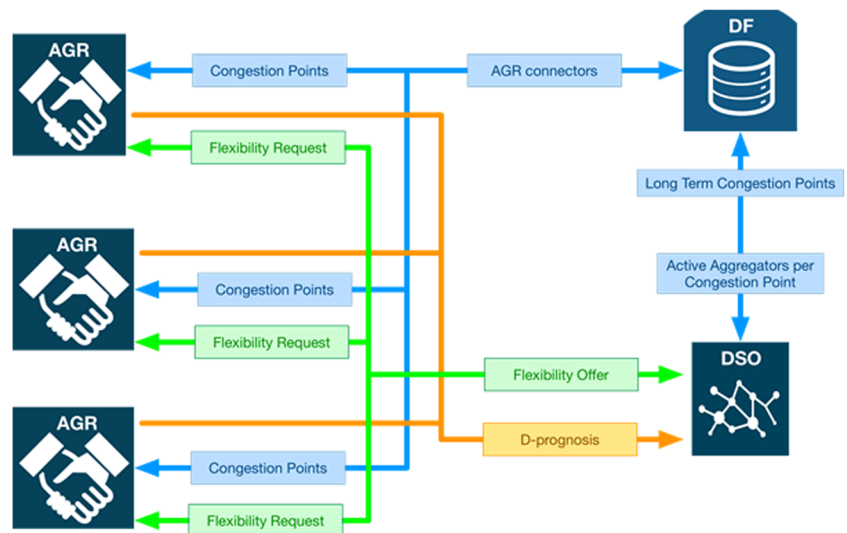
At the beginning of the plan phase (i.e. as part of the Day-Ahead initialization), the DSO Agent sends to the DF (*Directory Facilitator*) agent the long term connection points where congestion may happen. Since the distribution grid has a hierarchical structure, a congestion point can be defined at multiple levels (on feeder level, distribution transformer level, substation level, etc). Similarly, each AGR sends to the DF agent the list of connections in which it has at least one contracted prosumer in order to discover congestion points. Subsequently, each DF agent sends to each AGR a list of all registered connections represented by the AGR, in order to send *D-prognoses* to the responsible DSO. A *D-prognosis* is the expected

consumption profile during the day of delivery for the portfolio of prosumers of an AGR including only prosumers related to a concerned congestion point. Likewise, the DF agent sends to the DSO Agent, for each congestion point, the list of all AGRs that have been declared active in order to know how many *D-prognoses* to expect. Secondly, at the beginning of the validation phase each AGR sends its *D-prognoses* per congestion point to the DSO. The *D-prognoses* only include the forecasted load of prosumers who have connections related to the relevant congestion point. Once the DSO receives all the *D-prognoses* per congestion point (or after a predetermined deadline), the DSO starts a grid safety analysis in which it checks if energy flows exceed calculated safety margins. If the grid safety analysis indicates a congestion will happen, the DSO starts the process of acquiring flexibility from the market

**Fig. 2** Aggregator level communication



**Fig. 3** District level communication



by sending a flexibility request to AGRs (green lines in Fig. 3). Note that these negotiations between DSO and AGRs occur at the congestion point level, turning every congestion point into a local flexibility market. As a result of such negotiations, the DSO will select the flexibility offers that solve the congestion.

## Multi-Agent Based Communication Architecture

Architectures for smart grids are expected to address key communication requirements and capabilities such as those related to reliability, availability, security and interoperability [6, 34, 35]. Therefore, a set of communication requirements has been specified at the start of the research work which the solution must address.

### Architecture Design

The communication consists of two levels:

1. Communication between agents using *JADE* (*Java Agent Development Framework*) protocols [37] and our communication protocols over the Internet.

*JADE* is an open source framework that streamlines the implementation of agents and their communication by supporting their development and debugging. *JADE* is one of the widely used MAS frameworks for research and industrial purposes and it implements *FIPA* (*Foundation for Intelligent Physical Agent*) interaction protocols [38, 39]. The *JADE* framework can be distributed across multiple machines, regardless of the underlying operating system or communication mechanisms. It can be configured and controlled using a remote graphical user interface.

2. *Home Area Network (HAN)* for communication between devices and agents based on *JEMMA* (*Java Energy Management Application framework*) [36]. *JEMMA* is an open source software component running on the Energy Box providing the *Device Abstraction Layer (DAL)*. The DAL is described in Section “[Abstraction of Home Devices](#)”.

The development of the MAS follows the Gaia2JADE process which guides the stages in modelling and implementation of interactions and agents [40]. Our implementation also uses the *JADE-S* security extension [41] which supports agent authentication, authorization and network encryption.

The integration with non-agent software components is essential in order to accommodate other technologies in the smart grid and to support capabilities that are not yet provided by agents. Our solution allows agents to interact with other software services such as access control (Section “[Access Control for Flexibility Services](#)”) and device abstraction (Section “[Abstraction of Home Devices](#)”).

### Communication Protocols and Messaging

*FIPA-ACL* (*Foundation for Intelligent Physical Agent - Agent Communication Language*) [38] is the de facto standard for message exchange and interaction protocols. Interactions among agents in *JADE* are regulated by protocols. These protocols standardise and provide detailed rules for the interactions among agents. *JADE* protocols are generic and abstract. Therefore, they need to be instantiated for a given context. The agent-to-agent protocols in this paper are implemented as instantiations of the generic *JADE* protocols, instantiated with the particular use case messages and using a common OWL-based ontology

**Table 1** Template used to describe agent-to-agent protocols

<i>Protocol</i>	Protocol name
<i>Initiator</i>	Role initiator of the protocol
<i>Receiver</i>	Role receiver of the protocol
<i>Purpose/ Parameters</i>	Description of the protocol
<i>Sequence diagram</i>	Ordered message exchange that takes place in this protocol
<i>Message type</i>	Types of messages exchanged as part of the protocol
<i>Targeted standards</i>	Standards that are targeted for compliance in this protocol
<i>Privacy and security</i>	Analyses the privacy and security issues related to the protocol

[42]. We use a template as in Table 1 to describe each protocol. The JADE platform provides generic interaction protocols that include the *Propose protocol*, the *ContractNet protocol* and the *Subscription protocol*. Table 2 describes the protocols used in our architecture and types of messages exchanged. Figure 4 shows the communication between components of the architecture and where the protocols are used. In order to secure agent communications, encryption and authentication capabilities are supported using JAVA security libraries as add-ons to the JADE platform.

The *Propose* protocol matches the description of the *ProposeDevicePowerSchedules* protocol (Table 2) that takes place between the HEMS and the Device Agents as discussed in Section “Use Cases”. Thus, we implemented the *ProposeDevicePowerSchedules* protocol as an instantiation of the JADE *Propose* behaviours.

The *Contract Net* protocol is relevant for designing the optimization protocols that allow agents to be able to negotiate flexibilities at different levels (between HEMS and AGR, among AGRs and between AGRs and DSO). Figure 5 depicts the sequence diagram for this flexibility negotiation protocol (i.e. as an instance of the generic iterated *Contract Net* protocol). In this paper, every optimization level has an optimization protocol in which participants exchange *FlexRequest* (from the initiator of the protocol) and *FlexOffer* messages until convergence (i.e. agreement) or closure of the market. The message exchange ends with *FlexOrder* from the initiator to participants. Therefore, the three negotiation-based optimisation protocols (i.e.

*InPortfolioOptimisation*, *TradeFlexibilityForPortfolioOptimisation* and *TradeFlexibilityForCongestionManagement*) are implemented as instances of the generic *flexibility negotiation* protocol. The *FlexRequest* message can be used by (i) the AGR to procure flexibility from HEMS Agents and other AGRs in order to perform portfolio optimisation; and (ii) the DSO to procure flexibility from AGRs to solve an expected congestion.

The FIPA *Subscription* protocol suits all subscription protocols and related behaviours needed in our implementation (*SubscribePPlans*, *SubscribeFlexibilities*, *SubscribeD-Prognoses*) with moderate modifications.

Our data model defines the common vocabulary and semantics of the messages that are exchanged across the MAS. The JADE platform supports the generic expression of such data exchange model as part of the FIPA-ACL standard by using the FIPA-SL (FIPA Semantic Language) specification. Table 3 describes the structure of the ACL messages.

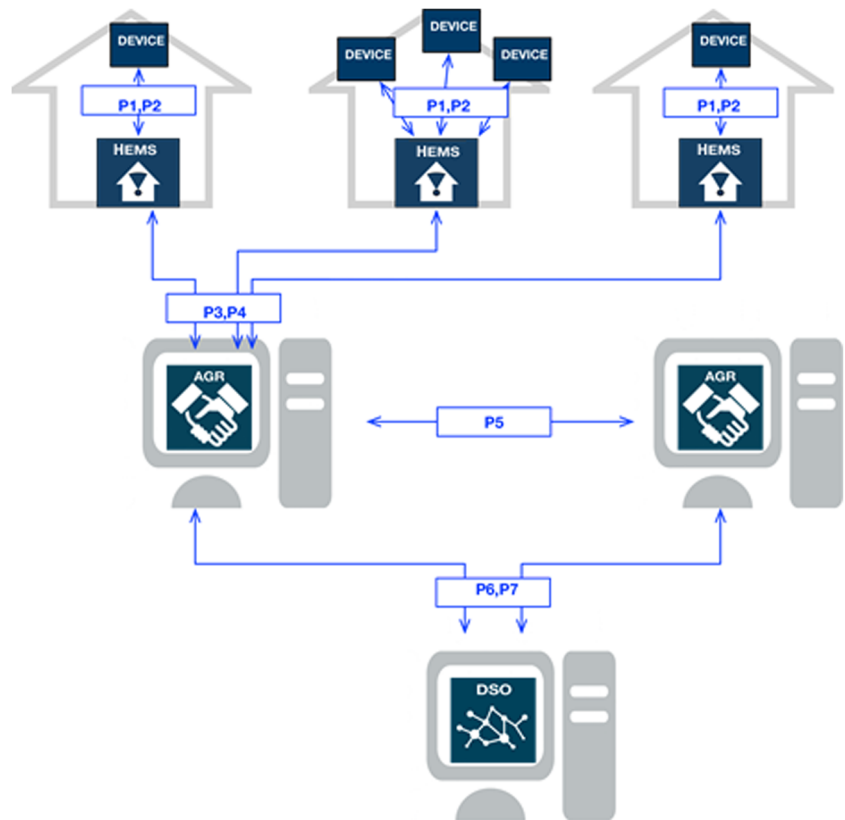
### Access Control for Flexibility Services

Attribute-based access control using XACML [43] is implemented in order to protect privacy and security of prosumers. It also controls the visibility of home devices allowing the prosumer to choose which and when devices can be managed by the HEMS. The access control platform, named *PANDA*, provides policy authoring, analysis and evaluation. Figure 6 illustrates the main components of

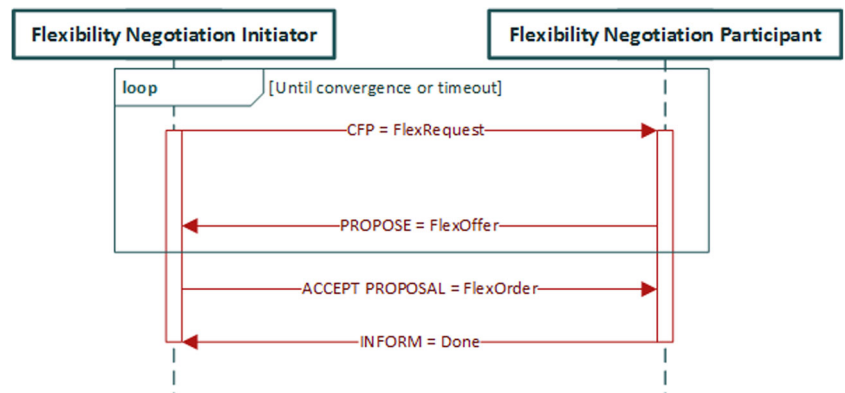
**Table 2** Agent-to-agent protocols and types of messages exchanged

Protocol Name	ID	Initiator	Receiver	Type messages exchanged
SubscribeFlexibility	P1	HEMS	Device	SubscribeToDeviceFlexibility, DeviceFlexibilityNotification
ProposeDevicePowerSchedules	P2	HEMS	Device	DevicePowerSchedules
SubscribePPlan	P3	AGR	HEMS	SubscribeToPlan, PPlanNotification
InPortfolioOptimisation	P4	AGR	HEMS	FlexRequest, FlexOffer, FlexOrder
TradeFlexibilityForPortfolio-Optimisation	P5	AGR	AGR	FlexRequest, FlexOffer, FlexOrder
SubscribeDPrognoses	P6	AGR	DSO	SubscribeToDPrognosis, DPrognosisNotification
TradeFlexibilityForCongestion-Management	P7	DSO	AGR	FlexRequest, FlexOffer, FlexOrder

**Fig. 4** Communication architecture and protocols



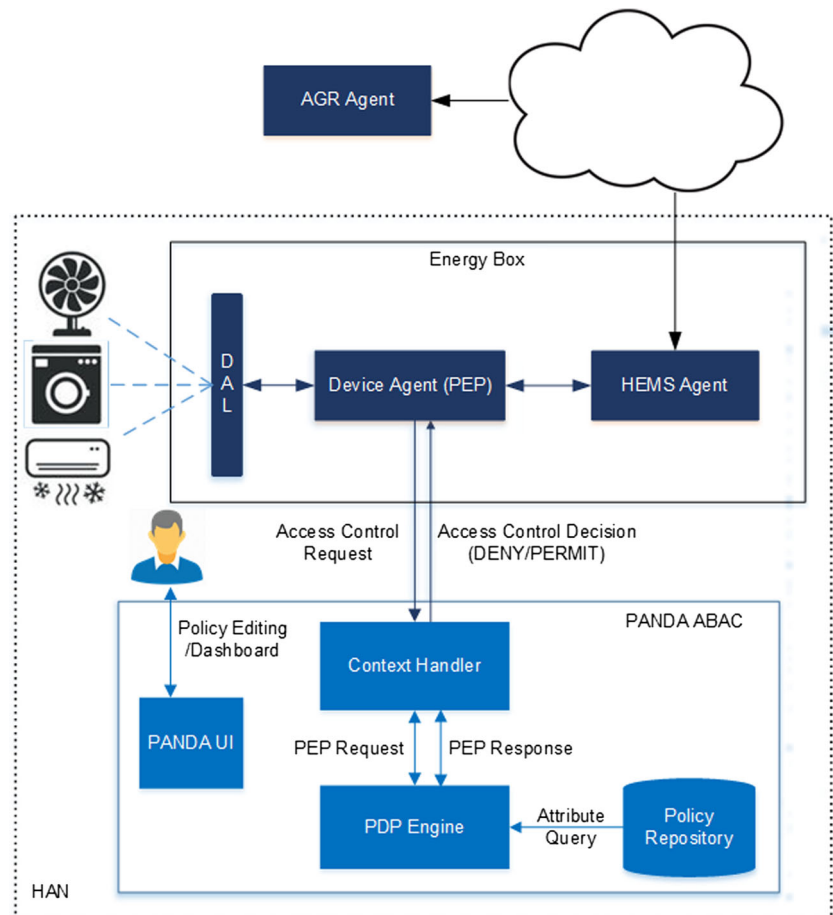
**Fig. 5** Flexibility negotiation protocol



**Table 3** Template used to describe ACL messages

Parameter	Purpose
<i>Protocol</i>	Protocol name
<i>Performative</i>	Characterises the nature of the message to help agents decode message content
<i>Sender</i>	The agent that sends the message
<i>Receiver</i>	The agent that receives the message
<i>Payload name</i>	Descriptive name of the payload
<i>Payload language</i>	The encoding of the content (FIPA-SL)
<i>Ontology</i>	The vocabulary used in the content
<i>Payload content</i>	The sequence of message exchange that takes place in the protocol
<i>Payload description</i>	Lists the different types of messages that are sent as part of the protocol

**Fig. 6** Access control for flexibility services



PANDA and their interactions. The Device Agent acts as PEP (Policy Enforcement Point) intercepting requests for data and action commands sent to devices in the HAN. The Context Handler acts as a proxy between external components (such as components acting as PEPs) and PANDA. The *Policy Decision Point (PDP)* component provides access control decisions based on attributes of requests including the source of the request (subject), relevant device (resource) and the request type (action). The other determinant of the eventual access control decision is the applicable XACML policies. These inputs along with possible extra attributes are used by the PDP engine. The two main outcomes of the process is *'PERMIT'* or *'DENY'*. The Device Agent then uses the decision; a *'PERMIT'* response will allow the data request or command to proceed and a *'DENY'* response will result in a request denial with no data to be sent back and no actions performed on smart appliances.

As discussed in Section “Literature Review”, current mechanisms to evaluate XACML policies are slow and do not perform in situations where a policy decision is required in near real-time and when there is a high volume of policy requests. PANDA is implemented in a

scalable *Micro Services* architecture that includes support for efficient concurrency where multiple policies can be evaluated simultaneously and various tasks can run at the same time using *Golang (also known as Go) goroutines and channels* [44]. This allows the *PDP* to exploit multi-core processing environments in cloud infrastructures. Our approach also supports the pre-processing of XACML policies and storing the policies in an optimised format so that lookup and evaluation is orders of magnitude faster than traditional approaches. We store the policies in the more efficient *JavaScript Object Notation (JSON)* format instead of the verbose and processing-intensive XML format but with equivalent semantics. The use of JSON also enables the use of highly scalable data store for policies in NoSQL databases. In addition, we use distributed caching using *Redis* in-memory data structure store [45] in order to further improve performance of policy evaluations. *Node.js* [46] is used to implement the REST APIs and UI dashboard which helps support large throughputs and scalability. This is because of the fast JavaScript execution and the use of the *Event Loop* mechanism that allows the components to perform non-blocking I/O operations. The PANDA platform is based on components that are



loosely linked and communicate using REST APIs and RabbitMQ (which implements the *Advanced Message Queuing Protocol (AMQP)* [47]). The combination of these methods and tools provides a significant advantage for PANDA over existing solutions in terms of support for near real-time applications due to their efficiency, performance and scalability.

XML-based XACML policies are known for their complexity and verbosity as noted by researchers such as [48]. Therefore, a normal user needs to have a simplified interfaces to the policies. The prosumer has access to a Web interface (Fig. 7) that allows to edit policies in a user friendly fashion using forms. Changes to the policies will be reflected on the level of access that agents have to home devices. The dashboard also allows for the inspection and monitoring of the performance of various components of the PANDA platform.

Figure 8 demonstrates the performance of PANDA PDP. Policy evaluation requests are sent sequentially to the PDP instance gradually increasing the request speed. As it can be seen from Fig. 8, PANDA PDP is able to handle around 100 requests/sec at 0.09 sec response time. Since PANDA can be run as a load balanced cluster using Docker Swarm [49]

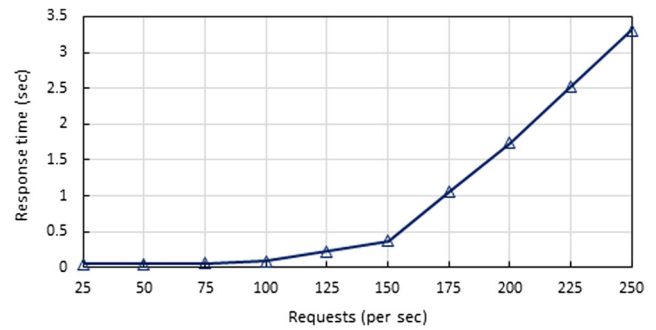


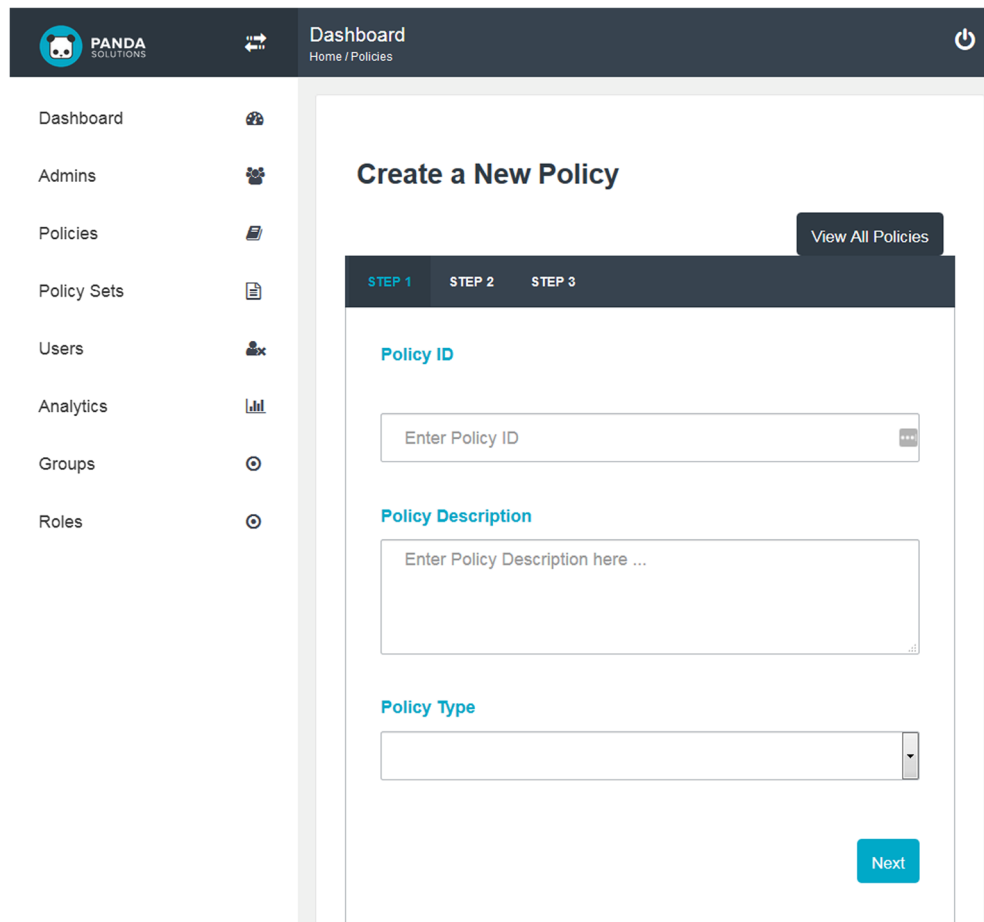
Fig. 8 PANDA PDP performance

on OpenStack [50], further performance enhancement can be achieved through horizontal scaling of all or a subset of the independent components.

### Abstraction of Home Devices

The Telecom Italia Energy Box provides standard APIs to interact with devices through the JEMMA software, which in turn communicates with the ZigBee wireless low power network used by HAN devices (smart appliances and smart

Fig. 7 PANDA dashboard - creating new policy



meters). The Device agent can reference the devices through HTTP REST and WebSocket APIs provided by a uniform *Device Abstraction Layer (DAL)*. The purpose is to provide a unified programming model that the MAS agents can use to interact with devices which are independent from the proprietary device protocols and technologies. The DAL is based on the Device Abstraction Layer specifications from the OSGi Residential Release 6 [51]. DAL APIs offer a set of java interfaces and guidelines that enable device integrators to register devices and their functionalities as OSGi Service-Oriented architecture (SOA) services. Each device is registered as a device service and each device operation is registered as a function service. The implementation extends the functions that are available from the specification as more device function support is needed in our solution. To export DAL functions to third party applications that do not use OSGi or are hosted in different possibly distributed runtime environments, a set of HTTP REST APIs has been implemented. They allow the interaction with devices using a simple HTTP client and JSON syntax for data exchange. The DAL Events can be obtained through a WebSocket protocol interface implementing a custom protocol for subscribing to device events. The data collected by the Energy Boxes is stored inside a cloud historical data server hosted in a Telecom Italia Cloud solution. The historical data server exposes a set of HTTP REST APIs.

The Device Agent is based on the JADE platform but uses the DAL's REST APIs to actuate and receive data from devices. The DAL APIs provide REST and WebSocket access to the devices, functions and events. For example, the DAL's *PowerProfile* function allows to receive information regarding the scheduled energy consumption of devices.

The prosumer UI is a static responsive JavaScript based web application. It interfaces with the API hosted by the historical data server in order to get the relevant data to

the prosumer. The UI allows the customer to be aware of his/her energy consumption and production at home, adding an estimation of the costs or the incentives related to the energy measures. The prosumer behaviour in terms of energy efficiency is also compared with other similar prosumers. This allows to calculate some KPIs on energy consumed, produced energy, auto-consumed energy and the energy sold to the network. According to those KPIs, the users receive a rating (0 to 5 stars) and suggestions that can help improve their behaviour and reduce energy bills.

## Testing and Results

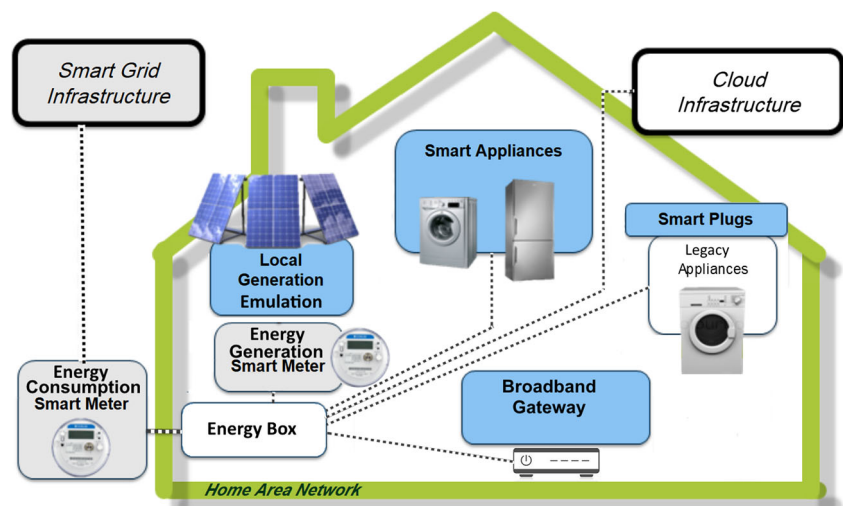
The MAS and software architecture developed in this research are tested in lab facilities at Telecom Italia. The test environment (as illustrated in Fig. 9) consists of the following equipment:

- TI Energy Boxes
- Smart washing machine prototypes
- ZigBee smart meters, adapters and Smart Plugs
- Local generation emulation units (power generators)
- Commercial broadband gateway for internet connectivity

The energy boxes run the DAL described in previous sections which facilitates interaction of the Device Agent with devices. All nodes in the HAN are connected using the ZigBee protocol. The Device Agent communicates with PANDA as an online service through REST APIs and RabbitMQ. The AGR and DSO Agents are deployed in JADE in the test environment.

Table 4 shows the summary outcome of test cases that aim to evaluate the reliability and security of the communication architecture and addressing the requirements. The test cases are mapped to the functional and non-functional

**Fig. 9** Diagram of laboratory testing facility at Telecom Italia



**Table 4** Testing of quality of communication

Test ID	Objective	Threshold	Result
QI1	HAN communication reliability	95%	100%
QI2	Smart Appliance scheduling effectiveness	100%	100%
QI3	HAN & cloud communication reliability	95%	100%
QI4	MAS communication reliability	95%	100%
QI5	MAS and third parties' services communication reliability	95%	100%
QI6	Blocking unauthorised access to Device Agent	100%	100%

requirements of the architecture. For example, for the *HAN Communication Reliability* (Quality Indicator - QI1) the Device Abstraction Layer interfaces are tested in a 6-hour period to verify that they were able to send messages to devices and receive responses from them. A success percentage of 95% or above of the request/response interactions can be tolerated. A java program is used to collect device communication data. During the test, the devices are connected to the Energy Box and Device Agents are used to send the requests to the devices via the DAL APIs once every minute. The devices involved in the test include smart washing machine prototype, smart meter and smart plugs. Every minute the Device Agents invoke *getTotal* operation on *EnergyMeter* DAL function on each device. The log of the software reported that 100% of request/response interactions with devices were successful.

*Smart Appliance scheduling effectiveness* (QI2) is a test to assert that the Device Agent successfully schedules the start time of smart appliances. In the test the Agent schedules the smart washing machine for all possible washing cycles. QI6 tests PANDA using multiple test cases to assert that the correct access control decision and enforcement are made in all cases.

## Discussion

The solution described in this paper is composed of intelligent software agents deployed at different physical locations. Those agents communicate with each other and with non-agent components through message exchange. The autonomous decentralised decision making has advantages in enhancing efficiency and supporting competitiveness. In particular, since each prosumer locally takes its own independent decisions, autonomy and privacy are assured (i.e. decentralisation delegates to the lowest levels all data and control except those which can only be exercised at central points). Moreover, through the introduction of the local flexibility aggregator functionality, the solution stimulates the formation of small cohesive communities. Different distributed optimisation algorithms (e.g. Nedic and Ozdaglar [52]) and optimisation criteria can be selected for the

communities depending on their preferences and objectives. Such objectives may include:

- Reduction of peak load at local energy community and LVG level.
- Reduced grid losses at LVG level.
- Increased renewable hosting capacity of the LVG.
- Increased reliability of the grid.
- Maximization of prosumers revenue.
- Maximization of self-consumption.

In addition, a decentralised organisation facilitates faster decision making. Consequently, it removes some of the burden from a central entity by allowing decisions to be made as close as possible to the data sources.

Nevertheless, the design of the communication in such decentralised architectures is more challenging than the traditional centralised control. In particular, rather than a typical order-based, one-way hierarchical flow it takes the form of a more complex cooperative decision making based on two-way agent communication protocols. In such protocols, agents can share semantics by committing to shared message content ontologies. Additionally, given its application to the electric grid, the compliance and alignment of such agent models and protocols with the smart grid standards are key points for the adoption of the solution as discussed in our previous work [53].

## Conclusion

This paper describes the development of a multi-agent system connected to other components in the smart grid in order to manage flexibility and demand response within the low-voltage part of the electric grid while ensuring security and privacy. The agents can interact with devices through RESTful HTTP and WebSocket APIs provided by a device abstraction layer. The implementation supports providing prosumers full control of their privacy through an attribute based access control. The decentralised hierarchical architecture allows the formation of local communities to collectively optimise energy flow in the district at the aggregator level and facilitate independent

decision making of prosumers. The aim of future work is to exploit the results of this research into the smart grid market with a business model that fits the current and future market needs and opportunities.

**Acknowledgements** The research leading to these results has partly received funding from the EU Seventh Framework Programme under grants no 619682 (Mas2tering). The authors would like to thank all the advice and comments particularly from the reviewers which significantly contributed to the improvement of the paper.

## References

- International Energy Outlook 2016 U.S. Energy Information Administration, May 2016, [Online]. Available: [https://www.eia.gov/outlooks/ieo/exec\\_summ.cfm](https://www.eia.gov/outlooks/ieo/exec_summ.cfm). Accessed Jul 2017
- Farhangi H (2010) The path of the smart grid. *IEEE Power Energy Mag* 8(1):18–28
- McArthur S et al (2007) Multi-agent systems for power engineering applications - part i: concepts, approaches, and technical challenges. *IEEE Trans Power Syst* 22(4):1743–1752
- Barbato A, Capone A, Carello G, Delfanti M, Merlo M, Zaminga A (2011) House energy demand optimization in single and multi-user scenarios. *IEEE SmartGridComm*, 345–350
- Safdarian A, Fotuhi-Firuzabad M, Lehtonen M (2016) Benefits of demand response on operation of distribution networks: a case study. *IEEE Syst J* 10(1):189–197
- Ancillotti E, Bruno R, Conti M (2013) The role of communication systems in smart grids: architectures, technical solutions and research challenges. *Comput Commun* 36(17–18):1665–1697
- McArthur SDJ et al (2007) Multi-agent systems for power engineering applications—part II: technologies, standards, and tools for building multi-agent systems. *IEEE Trans Power Syst* 22(4):1753–1759
- Ramchurn S, Vytelingum P, Rogers A, Jennings N (2011) Agent-based control for decentralised demand side management in the smart grid. In: 10th International conference on autonomous agents and multiagent systems, vol 1, pp 5–12
- Fazal R, Solanki J, Solanki SK (2012) Demand response using multi-agent system. In: 2012 North American power symposium (NAPS), pp 1–6
- Gomes L, Faria P, Morais H, Vale Z, Ramos C (2014) Distributed agent-based intelligent system for demand response program simulation in the scope of smart grids. *IEEE Intell Syst* 29(1):55–65
- Wu X, Hu X, Moura S, Yin X, Pickert V (2016) Stochastic control of smart home energy management with plug-in electric vehicle battery energy storage and photovoltaic array. *J Power Sources* 333:203–212
- Wu X, Hu X, Teng Y, Qian S, Cheng R (2017) Optimal integration of a hybrid solar-battery power source into smart home nanogrid with plug-in electric vehicle. *J Power Sources* 363:277–283
- Karapoulos E, Tena L, Torres A, Salas P, Jorda JG, Dimeas A, Hatzigiorgiou N (2015) A multi-agent system providing demand response services from residential consumers. *Electr Power Syst Res* 120:163–176
- Radhakrishnan BM, Srinivasan D (2016) A multi-agent based distributed energy management scheme for smart grid applications. *Energy* 103:192–204
- Pagani G, Aiello M (2012) Service orientation and the smart grid state and trends. *SOCA* 6(3):267–282
- Grijalva S, Tariq MU (2011) Prosumer-based smart grid architecture enables a flat sustainable electricity industry. *IEEE Innovative Smart Grid Technologies (ISGT)*, 1–6
- Rusitschka S, Eger K, Gerdes C (2010) Smart grid data cloud: a model for utilizing cloud computing in the smart grid domain. *IEEE SmartGridComm*, 483–488
- Siano P (2014) Demand response and smart grids - a survey. *Renew Sustain Energy Rev* 30:461–478
- Kuzlu M, Pipattanasomporn M, Rahman S (2014) Communication network requirements for major smart grid applications in HAN, NAN and WAN. *Comput Netw* 67:74–88
- Universal Smart Energy Framework (USEF) Nov. 2015, [Online]. Available: <https://www.usef.energy>. Accessed Jul 2017
- Mohsenian-Rad A, Wong V, Jatskevich J, Schober R, Leon-Garcia A (2010) Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE Trans Smart Grid* 1(3):320–331
- Molderink A, Bakker V, Bosman M, Hurink J, Smit G (2010) Management and control of domestic smart grid technology. *IEEE Trans Smart Grid* 1(2):109–119
- Aman S, Simmhan Y, Prasanna VK (2013) Energy management systems: state of the art and emerging trends. *IEEE Commun Mag* 51(1):114–119
- Fan Z et al (2013) Smart grid communications: overview of research challenges, solutions, and standardization activities. *IEEE Commun Surv Tutor* 15(1):21–38
- Wang W, Lu Z (2013) Cyber security in the smart grid: survey and challenges. *Comput Netw* 57(5):1344–1371
- Li X, Liang X, Lu R, Shen X, Lin X, Zhu H (2012) Securing smart grid: cyber attacks, countermeasures, and challenges. *IEEE Commun Mag* 50(8):38–45
- Cavalcante RC, Bittencourt II, da Silva AP, Silva M, Costa E, Santos R (2012) A survey of security in multi-agent systems. *Expert Syst Appl* 39(5):4835–4846
- Namboodiri V, Aravinthan V, Mohapatra SN, Karimi B, Jewell W (2014) Toward a secure wireless-based home area network for metering in smart grids. *IEEE Syst J* 8(2):509–520
- Rottondi C, Verticale G (2015) Privacy-friendly load scheduling of deferrable and interruptible domestic appliances in smart grids. *Comput Commun* 58:29–39
- Such JM, Alberola JM, Espinosa A, Garcia-Fornes A (2011) A group-oriented secure multiagent platform. *Softw Pract Exper* 41:1289–1302
- Liu A, Chen F, Hwang J, Xie T (2011) Designing fast and scalable XACML policy evaluation engines. *IEEE Trans Comput* 60(12):1802–1817
- Ros SP, Lischka M, Mármol FG (2012) Graph-based XACML evaluation. In: 17th ACM Symposium on access control models and technologies (SACMAT '12). ACM, pp 83–92
- Sun Reference XACML PDP, <http://sunxacml.sourceforge.net/javadoc/com/sun/xacml/PDP.html>, last accessed: May 2017
- Sauter T, Lobashov M (2011) End-to-end communication architecture for smart grids. *IEEE Trans Ind Electron* 58(4):1218–1228
- Gungor VC et al (2011) Smart grid technologies: communication technologies and standards. *IEEE Trans Indust Inf* 7(4):529–539
- Java Energy Management Application framework (JEMMA), [Online]. Available: <http://ismb.github.io/jemma/>. Accessed Jul 2017
- Java Agent Development Framework (JADE), [Online]. Available: <http://jade.tilab.com>. Accessed Jul 2017
- FIPA ACL message structure specification, Foundation for Intelligent Physical Agents (FIPA), Dec 2002
- Kravari K, Bassiliades N (2015) A survey of agent platforms. *J Artif Societ Soc Simul* 18(1):11

40. Moraitis P, Spanoudakis N (2006) The Gaia2Jade process for multi-agent systems development. *Appl Artif Intell* 20(2–4):251–273
41. Bellifemine F, Poggi A, Rimassa G (2001) Developing multi-agent systems with a FIPA-compliant agent framework. *Softw Pract Exper* 31:103–128
42. Web Ontology Language (OWL), [Online]. Available: <https://www.w3.org/OWL>. Accessed Jul 2017
43. Rissanen E (editor) eXtensible Access Control Markup Language (XACML) Version 3.0, Jan 2013, [Online] Available: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>. Accessed Jul 2017
44. The Go Programming Language, <https://golang.org>, last accessed May 2017
45. Redis, <https://redis.io>, last accessed May 2017
46. Node.js, <https://nodejs.org>, last accessed May 2017
47. RabbitMQ, [Online]. Available: <https://www.rabbitmq.com/> Accessed Jul 2017
48. Stepien B, Felty A, Matwin S (2014) A non-technical XACML target editor for dynamic access control systems. In: 2014 International conference on collaboration technologies and systems (CTS), pp 150–157
49. Docker, <https://www.docker.com>, last Accessed May 2017
50. OpenStack Open Source Cloud Computing Software, <https://www.openstack.org>, last accessed May 2017
51. OSGi Residential Release 6 Specification Aug 2015, [Online] Available: <https://www.osgi.org/developer/downloads/release-6/>. Accessed Jul 2017
52. Nedić A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Control* 54(1):48–61
53. Elshaafi H, Vinyals M, Dibley M, Grimaldi I, Sisinni M (2016) Combination of standards to support flexibility management in the smart grid, challenges and opportunities, smart grid inspired future technologies. *Springer LNICST* 175:143–151