Bulletin of the *Transilvania* University of Braşov • Vol. 9 (58) No. 2 - 2016
Series I: Engineering Sciences

PROTOCOLS FOR INTEGRATION OF INTERNET OF THINGS WITH SMART GRIDS

S. ZAMFIR[1] R. CURPEN[1] I. ILIESCU[1] F. SANDU[1]

Abstract: *At the Smart Grid (SG) upper levels - communications and metering - there are emerging technologies like the Internet of Things (IoT) and new business models for Machine-to-Machine (M2M) data flows involving convergent communications (fixed and mobile) operators. In the context of a smart-home use-case, this paper is approaching the interoperability of various intelligent sensors-transducers, actuators, data acquisition and processing instrumentation etc. This paper presents a solution for mediating communication between different actors in a Smart Grid based scenario.*

Key words: *Smart Grid, Internet of Things, CoAP, MQTT, HTTP, REST.*

1. Introduction

Requirements for real-time communication and control in modern electrical Smart Grids (SGs) are demanding and complex. Smart Grids must run in a reliable manner for several years without any discontinuity and with a high availability - close to 100%.

Moreover, the need for a deterministic response time (usually milliseconds, even microseconds) and resilience against increased level of cyber-threats determine that software engineering will play an important role in the design of such high performance and critical software systems.

Building such a system is not something to be improvised, needing a complex "co-design", with sustainability taken into consideration from the early phases [9]. With this high reliability, low latency and high scalability, Smart Grids can be represented as the "central nervous system" of power production, distribution and consumption [1], [7].

The important fulfilment of a Smart Grid is the increasing of energy efficiency, reducing the carbon footprint for power consuming industries, reducing the need of long-distance transport of energy from remote locations and, last but not least, improved safety and reliability.

Regarding the inherent IoT-SG interoperability, recent papers like [2] and [6] are summarizing the existing Internet of Things in several areas: HealthCare, Connected Smart Homes, Connected Cars and Smart Grids.

According to [6] in all the cases studied and presented - see Figure 1 - the common ground is the usage of a physical layer, a transportation layer and an application layer that takes care also about centralized management.

[1] Electronics and Computers Dept., *Transilvania* University of Braşov.

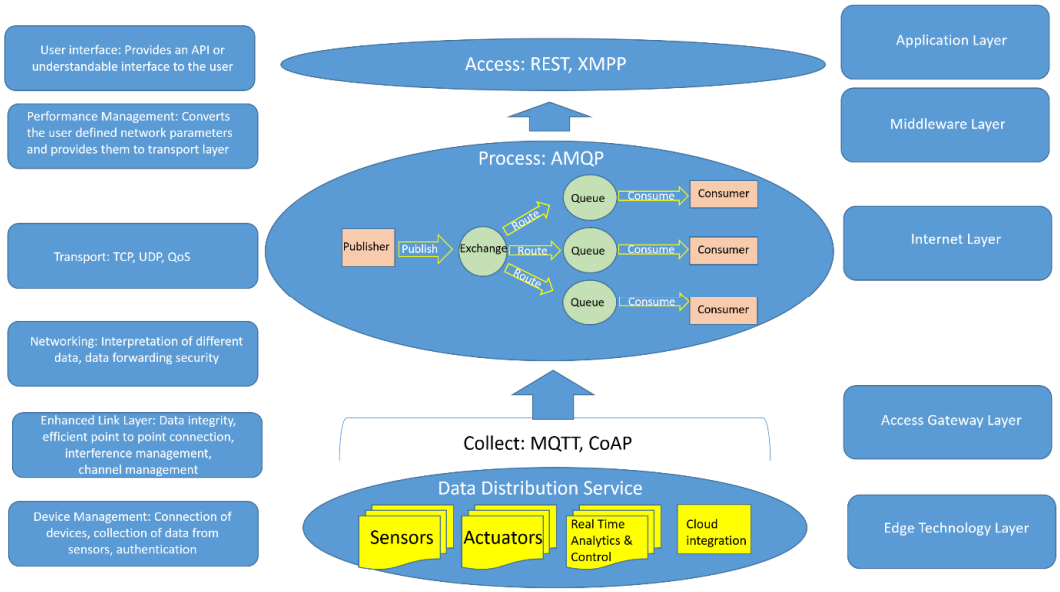

Fig. 1. *Layered model of IoT*

This layered representation follows the principles of the OSI (Open System Interconnect) model and, besides abbreviations like TCP (Transmission Control Protocol), UDP (User Datagram Protocol), QoS (Quality of Services) or AMQP (Advanced Message Queuing Protocol) the others are detailed in the following.

In case of domotic (intelligent residential building) Smart Grids - that are more invoked in this paper - the physical layer is implemented based on short ranged technologies like WiFi, Zigbee, RFID (Radio-Frequency Identification) but also on long range technologies like CIoT (Cellular system support for ultra-low complexity and low throughput Internet of Things [10].

The network layer consists of WSAN - wireless sensor & actuators (sub-) nets, gateway nodes, access networks and core networks. Finally, the application layer is built to satisfy the business needs of Smart Grid. This layer requires simple communications with the central management system in order to assure all the required computation and interpretation of the distributed data, in a service-oriented "Cloud" perspective.

The present paper is addressing one of the main drawbacks of the proposed enhanced architecture of the Smart Grid systems, that is the heterogeneity caused by the multiple combinations that occur in case of utilizing 3-4 technologies at each OSI level. Solutions to this problem can be achieved by IoT devices that can make as much as possible an abstraction of the technology used for the physical layer or for the transportation layer. Such IoT devices would be able to provide information either directly to the application layer, either to the centralized system, or to a peer system (e.g. peered IoT devices or emergency services).

By using an appropriate protocol at the application layer, the IoT devices can make totally abstraction of the underlying physical network (nevertheless, the main concern remains the methodology used to power up the IoT devices, that is beyond the concern of this paper). A disadvantage of this approach will be the increased level

of local computational power required by each IoT node (such a node should become a "micro server").

As shown in Figure 2, an analogy to the classical requirements triangle (Security/ Functionality/Ease of Use) - drawn for IoT devices integrated in a Smart Grid - it is unlikely that one IoT device should fulfil all the conditions that ensure safety and high usability, since it is not possible to implement all these requirements in a small, cheap, but very connectable device.

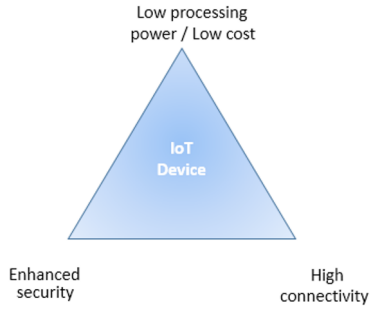

Fig. 2. *Requirements triangle for an IoT device integrated in the Smart Grid*

As a result, the most accessible solution is moving some pressure on the algorithms and the functionalities of a "middleware" layer that should facilitate the integration of the IoT devices into the Smart Grid.

2. Objectives and Motivation

The motivation of our paper is given by the pressure of newer technologies - Cloud, IoT and M2M communication - on Smart Grid models, in the context of well proven security, reliability and efficiency provided by consolidated SCADA (Supervisory Control And Data Acquisition) systems used in electrical domain.

Our research goal was to develop solutions for heterogeneous (multi-protocol) IoT devices interoperability and our objective was to find solutions at higher levels of the OSI stack, by semantic interfacing.

The validation of our solutions was in an "intelligent building" ("domotic") use-case of peak-power compensation in a Smart-Grid that involves the energy storage of electric vehicles in their charging mode.

The research methodology was driven by important international cooperation projects (e.g. Eclipse Ponte) recommendations of the standardization bodies (e.g. IETF or OASIS).

3. Research Methods

Introduced initially as an architectural concept by Roy Fielding to create scalable and distributed hypermedia systems, REST (Representational State Transfer) evolved as a concept for the models of web services.

Besides being a state-control model - as services abstracting is usually an ASM (Algorithmic State Machine), REST has also the advantage to be based on HTTP (Hypertext Transfer Protocol).

In accordance with REST, the architectural constraints in the design of a web service are [8]:

1. Identification of a web resource is done via an URI (Universal Resource Identifier) - naturally a REST service consists of making available to customers a number of resources they could combine to perform various operations;

2. Uniform interfacing - using standard HTTP protocol action verbs, given their strict sense for manipulation of resources (GET, POST, PUT and DELETE); in order that a web service to be considered a REST service, it is very important that the semantics of verbs coincide with the operations with the resources; as already mentioned, our examples belong to an intelligent house Smart Grid scenario:

Given some BOEVs (battery-only electric vehicles) that have one or more batteries, automatic power-peak compensation is possible using these sources as alternative in a zonal electric network.

The procedure is transparent to all the actors in the Smart-Grid: consumers and suppliers. (E.g. during the night, when the cars are in the charging-mode).

3. Descriptive messages - technically, the REST service messages encapsulation is independent of lower level protocols and this is leading to "semantic interfacing" - similar to SOAP (Simple Object Access Protocol) but less restrictive (no validation scheme or pre-defined formats - all these remain to provider's choice).

4. Stateless interactions through Internet/ Intranet links - every interaction between client and server has no state (to a request it corresponds only a response and this is all).

This simplifies the deployment and ensures that every request from the client to the server contains all the information needed for its processing [5].

Stateless interactions make such a system much easier to maintain because the management system shouldn't monitor the full dialogue but the service can be analysed from the perspective of simple request-response pairs [3].

In addition, the scalability of the system is enhanced, as the server should not keep a history of each client state.

One drawback underlined by Fielding is the additional load on the network caused by redundant information contained in each request [5].

Another specific feature of the REST technology is the approach of using components situated on different levels with the consequence that each component of the architecture is not capable of perceiving entities being beyond the immediate vicinity (but only the interacting components).

Limiting knowledge at one level provides quasi-independent evolution (e.g. there can be encapsulated traditional services that cannot be exposed through simple web protocols).

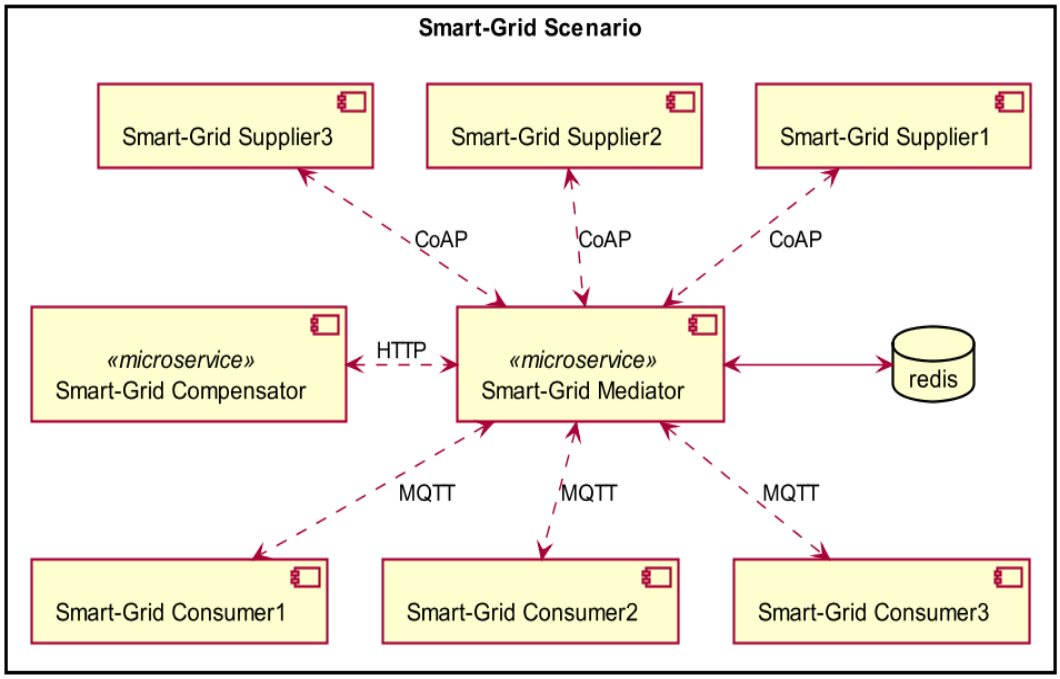

Fig. 3. *The compensator service in our Smart-Grid example*

3.1. The Communication Models

In terms of the interaction model, each operation contains all information required to successfully execute a transaction in the service shop. Generally, the description of REST services is made via a project documentation written in a natural language and listing all resource identifiers (URI) and the effect of standard HTTP verbs against those as well as the dialect used in the servers-generated responses.

With the popularization of JSON (Java Simple Object Notation) syntax, to obtain a service interface documentation became an operation in itself, containing a minimalist indication about the set of resources and unique indicators as well as a description of the possible operations using the Swagger specifications notation (included in Open API Initiative).

Querying the resource that describes the service interface can be done in two ways, depending on the client used (a user interface easily interpretable by humans, when using a browser for example or, respectively, a Swagger format that can be interpreted automatically by a program of machine learning, for example).

Another approach for a client that requires no previous knowledge on the service interface is found in the variability of operations available depending on the current status of the resource explored. In this case, the client can process hypermedia content that varies by the state of resource interrogated.

Depending on resource availability and on the capabilities that are queried, operations are dynamically changed. This approach is known as HATEOAS (Hypermedia as the Engine of Application State) and represents a constraint which has its origins in a uniform interface concept explored also by Fielding [8].

While all of the above is true for the web world applying the same design principles to the IoT world is not so straightforward. This is mainly because interoperability is not a key concern for hardware vendors which tend to impose proprietary communication protocols along with the devices they put on the market.

As it can be seen in Figure 3 the aggregation of multiple protocols in a single point of mediation is the only valid approach when considering a heterogeneous pool of resources.

In our scenario the consumers interact with the Smart-Grid using a “publish-subscribe” mechanism specific to the message brokering world, while the suppliers notify the management system of status changes by periodically posting updates. Because of different communication models and protocols, the need for de-coupled semantic dialogue arises.

3.2. Resources in the Smart-Grid

The purpose of this interfacing comes in mind with the need of the Smart-Grid to react to different outside events making it an intelligent system. If the actors involved can be uniformly addressed, then they can also be uniformly controlled or observed.

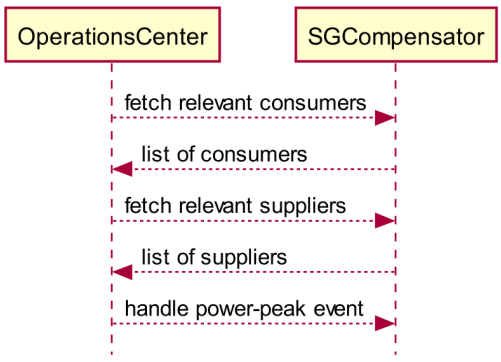

Fig. 4. *Proposed interaction model between the SG and the Operations Center*

As a result, our approach is to expose through a series of RESTful APIs all the entities within the Smart-Grid providing various useful operations on both the

consumers and suppliers of the whole network. In this way the whole process can be automated without the need to directly interact with each actor in the Smart-Grid. As a result all the devices that are within the ecosystem can be seen as resources that can be controlled through unique URIs:

```
GET smartgrid/{resource-type}/
{resource-id}
```

```
POST smartgrid/{resource-type}/
{resource-id}
```

GET is used to obtain (”to read”) information about remote resources - all of them/all about one of them (e.g. capabilities of one battery - capacity etc. - and status - charging level, internal temperature, availability etc.) /or particular data out of the individual information (e.g. type - an important “credential” in the decisional process of such resources aggregation in the Smart Grid).

POST is used “to write” information at the remote resource, to change its information set (on capabilities, on labelling etc. - generally speaking a subscribing, a registration procedure) or to change its status (that is the abstracting of remote control in state-driven models).

4. Implementation and Results

The basis of our implementation is a mix of different technologies to emphasize the heterogeneous nature of devices within the ecosystem: the consumers are implemented as Mqtt daemons that subscribe to a specific command queue waiting for different events while the suppliers are implemented as CoAP daemons that periodically publish status updates. The core of our prototype is the Compensator micro-service that is responsible for exposing the devices in a RESTFul way and offering an event driven interface to interact with the Operations Center.

In this way the logic behind the Smart-Grid concept is distributed across multiple entities making the solution scalable and fault tolerant. This is one fundamental principle when adopting micro-services design style.

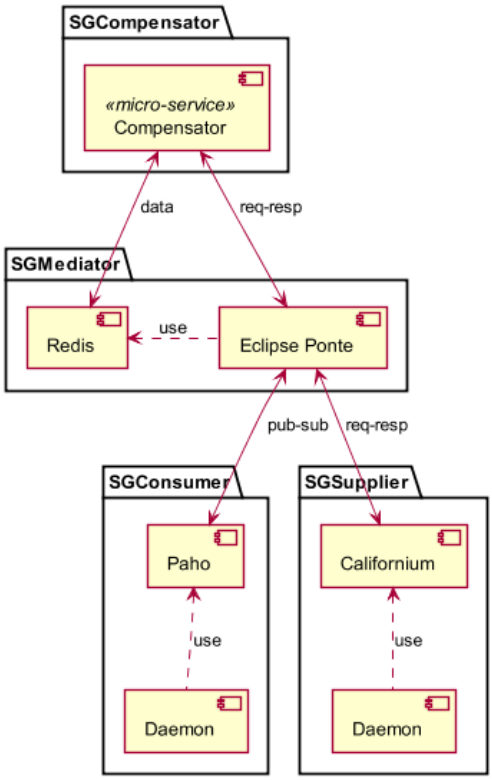

Fig. 5. *Implementation details of the demonstrator*

Interfacing between different components is achieved passing around JSON control objects (as expressed within Figure 6).

```
{
  "id": "1025",
  "latitude": 45,
  "longitude": 25,
  "powerDrain": 54,
  "powerSource": "ac_source"
}
```

Persistency of all exchanged messages is achieved by integrating the NoSQL Redis database that stores all the relevant

information using an internal hash with various key-value pairs (see Figure 7). This is necessary since we cannot rely on the availability of each consumer or supplier and furthermore the compensator service must always retrieve the last known state of each actor in the grid.

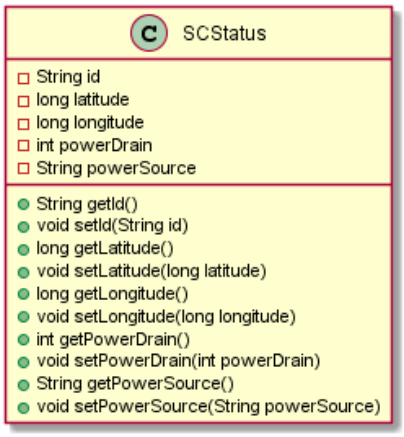

Fig. 6. *Serializable status object of the consumer*

The location coordinates serve as basis for the decision algorithm whether to switch or not to alternative power source (e.g. EV batteries). This factor is of upmost importance since the power suppliers will periodically change their location and this must be transparent to the end user and the grid itself. We observe thus that in such a scenario availability becomes a critical factor for taking decisions.

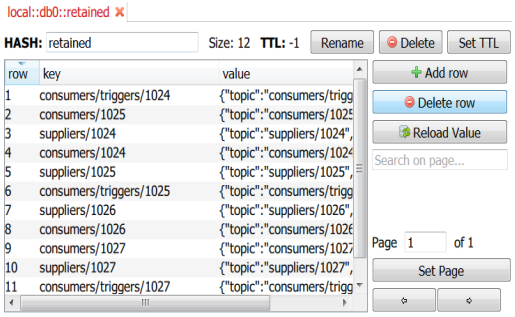

Fig. 7. *Persistence of resources using Redis key-value store*

Direct access to the persistence is needed by the Compensator micro-service since there is no direct way of obtaining the registered consumers and suppliers.

A naïve approach would have been to offer a creation API for inserting different entities within the grid ecosystem but this would reduce the degree of automation. Instead we prefer to apply a concept of resource discovery where each actor controls its own lifecycle.

5. Conclusions

Evolving from the sensor networks concept, the IoT networks didn't not represent per-se something totally new.

Also, the Smart Grid concept is similar with previous works performed to increase efficiency for power energy distribution [4].

An important challenge is why not using the standardized SCADA systems for managing the sensors and actuators network.

In this paper we have justified that IoT is not just an "IIoT" (Industrial Internet of Things) using the Smart Grids to illustrate how they can be integrated heterogeneous end-to-end entities (from production to consumer) via protocol mediation, without impacting on efficiency.

There remain some concerns when IoT is compared with traditional SCADA - the most important trade-offs are security and usability.

The original contributions of this paper reside mainly in the approach of the conglomerate of protocols empowering the usage of heterogeneous IoT devices for Smart Grid.

We aimed to hide the complexity of access technologies in order to increase usability and - considering mostly the usage of HTTP - also improving the security.

We achieved the scope to demonstrate, in a domotic proof-of-concept, a simple set

of messages that are brokered between a control center of the Smart Grid and IoT devices (batteries of modern EV cars, in the charging mode, that can be seamlessly used in the grid overload periods to provide "reverse energy" to the network).

References

1. Al-Ali, A.R., Aburukba, R.: *Role of Internet of Things in the Smart Grid Technology*. In: Journal of Computer and Communications **3** (2015), p. 229-233.
2. Bajpai, G.: *Middleware for Internet of Things*. Rwanda Univ., 2015.
3. Barry, D.: *Service Architecture - Soap*. Available at: http://www.service-architecture.com/articles/web-services/soap.html. Accessed: 06-10-2016.
4. Donitzky, C., et al.: *Digital Energy Network: The Internet of Things and the Smart Grid*. Intel White Paper, June 2015.
5. Fielding, R.: *Architectural Styles and the Design of Network-based Software Architectures*. University of Irving, 2000.
6. Guthikonda, R.T., et al.: *Comparative Analysis of IoT Architectures.* TLEN 5710 Capstone, April 2014.
7. Knab, S., Strunz, K., Lehmann, H.: *Smart Grid: The Central Nervous System for Power Supply*. TU Berlin, 2010.
8. Pautasso, C., et al.: *RESTful Web Services vs „Big" Web Services.* Making the right architectural decision, WWW Conference Bejing.
9. Rahman, A.: *Trends in Automation and Control Systems Needing Efficient Software Engineering*. Asia-Pacific Software Engineering Conference (APSEC), New Delhi, 2015, p. 2-2.
10. 3GPP TR 45.820: *Cellular System Support for Ultra-Low Complexity and Low Throughput Internet of Things (CIoT).* 3GPP TR 45.820, (2015-11).