D3.1.3
Evaluation of Communication Standards

June 13, 2019

EcoGrid 2.0 is a research and demonstration project funded by EUDP (Energiteknologisk Udviklings- og Demonstrationsprogram). The 9 partners in the project are:

Main Authors:

Name/Partner	Email
Michael Baentsch/IBM	mib@zurich.ibm.com
Peter Buhler/IBM	bup@zurich.ibm.com

TABLE OF CONTENTS

1 Overview

This document aims at providing an overview of potential alternative communication standards and methodologies for use in the EcoGrid 2.0 project. It begins with an overview of the motivation for various high-level communication standard goals to be achieved within EcoGrid 2.0, commences with a review of alternatives with recommendations on which approach is most reasonable within EcoGrid 2.0, taking into consideration business as well as security aspects and elements concerning practical viability within the constraints of the project.

2 EcoGrid 2.0 goals

2.1 Baseline goals

An overarching goal for the technical components developed and deployed during EcoGrid 2.0 always was to enable the most efficient and economic support of its business goal, namely to demonstrate the utility of controlling and successfully marketing energy flexibility for all market participants. The initial focus was on supporting the concrete EcoGrid 2.0 demonstration goals but there always had to be significant consideration placed on a potential deployment of the solution developed beyond the island of Bornholm and ideally also beyond Denmark and Europe.

This baseline was thus translated into the following technical goals:

- Use of standards: Wherever possible, existing standards should be used to avoid the custom development of functionality readily available and thus arguably subject to competitive tenders in any potential follow-on wide-scale productive deployments.
- Security: The solution components should not skimp on security – neither at the operational end of the spectrum nor the other end, the handling of potentially sensitive personal data, nor any element in between, e.g., protection of the various hard- and software assets comprising EcoGrid 2.0.
- Responsiveness: In order to support reasonably quick reaction to changes in the market environment, the basic infrastructure should conceptually be able to quickly react to external events and not only cater to regularly recurring procedures.
- Scalability: The solution should always execute in a way that scaling to significant numbers of participants is not only conceptually possible but in a way that can be regularly validated, e.g., by checking piece-part performance data.
- Extensibility: The pre-existing EcoGrid EU components had to be reused but always with the goal to the inclusion of additional current as well as potential future market participants. This basically demands an open architecture based on standards, thus closing the arch to the first goal mentioned above.

2.2 Additional considerations

Beyond the baseline goals documented above and transformed into concrete requirements in the next chapter, some further elements with relevance to the overall EcoGrid 2.0 communications strategy had to be considered:

There should be space for experimentation with novel communications channels, most notably concerning “the last mile”, i.e., concerning the communications with the actual energy consumers. One of the lessons of EcoGrid EU was that some mitigation may be required to improve upon the sometimes error- and failure-prone communications link between the aggregator backend and the actual sensor and actuators in the single houses. To this end, the EcoGrid 2.0 architecture has to be sufficiently open to permit the addition of mechanisms beyond standard Internet technology. Most notably, tests with LoRaWAN had to be conducted. At the same time, these experiments did not have to have a detrimental effect on the remaining core components in EcoGrid 2.0.

Another consideration for the choice of communications components in EcoGrid 2.0 is the very nature of the project as a lighthouse project for promoting more efficient use of energy resources. In this regard, the communications infrastructure deployed should be as efficient as possible on the use of electricity for its own operation; the situation should be avoided where operation of EcoGrid 2.0 infrastructure requires more energy or costs more money than it can save. Therefore, we posit to consider the notion of “battery-powered devices” for all EcoGrid 2.0 hard- and software components wherever possible to inform the choice of communications technology: This has several benefits: First, by reducing communications “busy-wait loops” the involved computing hardware, even servers, can save energy. Second, the components may be deployed in small-scale deployments on smaller compute devices, thus both saving hardware and energy cost. And finally, by optimizing for energy-frugal infrastructure, any conceptual performance problems would become more readily visible even in a comparably small deployment as the one in EcoGrid 2.0 thus serving the project well in the sense of extrapolating to country- or global-scale deployments of the technology and procedures developed in the project.

3 State of the Art

3.1 Baseline standards

A thorough discussion on communications standards certainly is beyond the scope of this document. The interested reader is referred to [1] as this “education gold standard” book on the topic is used throughout this document for terminology and guidance towards architecture and applicable standards.

3.1.1 ISO/OSI Layer 4 – transport layer

The widespread utility and ubiquity of the TCP/IP protocol is undisputed: It is an open and truly world-wide accepted standard for reliable communication and accordingly is to be used wherever possible to fulfill the first goal stated in the previous chapter. Lower layers may be different but shall bear equal global standard properties, e.g, GPRS, LoRaWAN or WLAN-based channels to bridge the communications gap between EcoGrid 2.0 backbone and sensors and actuators in houses in Bornholm.

3.1.2 ISO/OSI Layer 5 – session layer

In order to assure secure communications in pursuit of the second goal, TLS is to be used to protect all transfer of sensitive information between EcoGrid 2.0 components, e.g., authentication (login) credentials or energy consumption data. This may be augmented with VPN as local security requirements mandate.

3.1.3 ISO/OSI Layer 6 – presentation layer

Certain information has to be made accessible to end users, e.g., to facilitate display of current energy consumption or EcoGrid 2.0 participation modes. For this, the use of the standard HTTP(S) protocol(s) via equally standard Web browsers is the norm.

3.1.4 ISO/OSI Layer 7 – application layer

The split in layers 6 and 7 is theoretically debatable but in the context of this document we position only the intra-component interaction between the main EcoGrid 2.0 components as belonging to this stratum. Most notably, this is also the layer at which we aim to fulfill the remaining technical goals posited in the previous chapter: While EcoGrid 2.0 keeps using standard SQL messaging from/to internal and external databases, the actual core marketplace and aggregator components are connected to each other not via a request/response communications pattern but rather via an event-based messaging infrastructure. While there are different implementations available, we again opt for one based on a global standard: OASIS’ MQTT [2].

3.2 MQTT

The core advantages of this protocol and how this is in support of the remaining technical goals posited above are given by the standards organization itself [2]: “*a standard for the Message Queuing Telemetry Transport Protocol compatible with MQTT V3.1, together with requirements for enhancements, documented usage examples, best practices, and guidance for use of MQTT topics with commonly available registry and discovery mechanisms. The standard supports bi-*

directional messaging to uniformly handle both signals and commands, deterministic message delivery, basic QoS levels, always/sometimes-connected scenarios, loose coupling, and scalability to support large numbers of devices. Candidates for enhancements include message priority and expiry, message payload typing, request/reply, and subscription expiry.

As an M2M/Internet of Things (IoT) connectivity protocol, MQTT is designed to support messaging transport from remote locations/devices involving small code footprints (e.g., 8-bit, 256KB ram controllers), low power, low bandwidth, high-cost connections, high latency, variable availability, and negotiated delivery guarantees. For example, MQTT is being used in sensors communicating to a broker via satellite links, SCADA, over occasional dial-up connections with healthcare providers (medical devices), and in a range of home automation and small device scenarios. MQTT is also ideal for mobile applications because of its small size, minimized data packets, and efficient distribution of information to one or many receivers (subscribers)."

Specific product-support material like for example [3] goes into some examples: "*The MQTT protocol is optimized for networks with limited processing capabilities, small memory capacities, or high latency. Compared to HTTP, MQTT imparts several advantages to mobile applications:*

- *Faster response times*
- *Higher throughput*
- *Higher messaging reliability*
- *Lower bandwidth usage*
- *Lower battery usage*

MQTT's efficient use of network and battery power, along with its integration with enterprise messaging middleware, makes it ideal for scenarios where an enterprise application must push data or interact with one or more mobile applications.

HTTP is designed as a request/response protocol for client/server computing, not necessarily optimized for mobile and push capabilities, particularly in terms of battery usage. MQTT helps overcome problems experienced by HTTP, by offering reliable delivery over fragile networks. MQTT will deliver message data with the required quality of service (QoS) even across connection breaks. MQTT includes several useful features:

Last Will and Testament: All applications know immediately if a client disconnects ungracefully.

Retained messages: Any user reconnecting immediately gets the latest business information.

[…]

An actual example where implementing the MQTT protocol rather than HTTP was beneficial is the case of a prominent social networking company. The company had experienced latency problems when sending messages. The message delivery was reliable but slow with their previous method.

A new mechanism was needed that can maintain a persistent connection with the messaging servers without using too much battery power. This capability is critical to users of the company's social networking site, because so many of them use the service on battery-powered mobile devices.

The company's developers solved the problem by using the MQTT protocol. By maintaining an MQTT connection and routing messages through its chat pipeline, the company was able to achieve message delivery at speeds of just a few hundred milliseconds, rather than multiple seconds."

While the use of battery powered components is not a core concern in EcoGrid 2.0, the underlying constraints of such devices are: Energy-efficient and ultra-scalable messaging. For this reason, MQTT is a good choice, particularly considering that many open source and commercial products supporting MQTT are available. Clearly MQTT is not as prevalent as HTTP(s)-based Web services, but the gain in scalability and energy-efficiency in our eyes justifies this decision.

The final decision to be taken concerns the actual messaging format running across the EcoGrid 2.0 MQTT core network. For this, the choice fell on JSON as it is both a standard and still rather human readable: The latter property we consider an important property to foster an efficient and quick exchange of ideas regarding the application-level communications patterns between the different research partner sites: Eliminating potential sources of errors by mis-interpreting more efficient, but only machine-readable on-wire data formats we consider to be more important for project progress and simple extensibility for inter-component communication than the theoretically highest communication performance. This decision certainly needs revisiting when all research challenges have been overcome and the system enters the stage of productive use.

4 EcoGrid communications infrastructure

4.1 Deployment decisions for EcoGrid 2.0

The core EcoGrid 2.0 components are implemented following the goals and implementation options introduced above by following an event-based communications pattern utilizing the open MQTT standard combined with secure messaging based on the TLS standard and the equally open JSON data exchange format; see [4] for details. Some components utilize different communications approaches for reasons explained below and giving scope for potential future work, mostly constrained by economic reasons.

4.2 Future work: Potential enhancements

This section makes a proposal for enhancement of the communications methodologies used within the EcoGrid 2.0 project in the light of the experiences made during the project. The recommendations are delineated along the main components of EcoGrid 2.0.

4.2.1 Data warehouse

Classic warehouses typically do not need an event-based communications mechanism as is underlying the EcoGrid 2.0 infrastructure. The main goal here is to achieve efficient data storage and retrieval, incl. backup. Also, the economics of changing the interface of data warehouses away from standard SQL is prohibitively expensive. Therefore, this component is exempt from any change recommendations.

4.2.2 GWR and Siemens HEMS management components

These components presently do not use MQTT interfaces and it would be advisable to change this to achieve more automation and standardization in implementing market participants following the EcoGrid 2.0 concepts: Presently, explicit software componentry Flexibility Interoperability Platform (FIP) is required to integrate these components into the overall project fabric.

4.2.3 Internal data importers

Mostly due to legacy software interface constraints, these components generally follow a batch-oriented, provider-specific API request-retrieve pattern. As with the external data sources, a gradual replacement of this with event-triggered communications patterns would be advisable if a more reactive overall system would need to be the goal of potential follow-on projects or real-world deployments of EcoGrid 2.0.

4.2.4 External data importers

These are implemented to utilize the available interfaces by the external data sources, e.g., meteorological data and hence cannot easily follow the event-based architecture proposed but rather realize a batch-oriented communications pattern. The latter is arguably most efficient, but if a more reactive overall system would be desirable in the future, a change to encourage external data providers to deliver data in an event-triggered fashion would be a reasonable way to improve the overall system.

4.2.5 Flexibility Interoperability Platform (FIP)

The Flexibility Interoperability Platform (FIP) provides the infrastructure necessary to test interoperability in EcoGrid 2.0. It enables aggregators to talk to DERs using the EcoGrid 2.0 interoperability protocol. FIP uses GWR and Siemens HEMS control interfaces to provide this functionality. This layer is required as todays systems from different manufactures do not offer standard-based communication interfaces and probably there will be a need for such a layer in the future to support legacy systems. As mentioned above this component presently provides a bridge between the messaging concepts of the core EcoGrid 2.0 components which would be unnecessary if the suggestion in 4.2.2 would be implemented and standardisation process would progress. The use of an additional layer is often a source of problems in operation, integration, and performance. This could be eliminated through relevant standards development and support by the manufactures.

4.2.6 Aggregators

These follow the recommendations made. No change is required.

4.2.7 Market clearing house

These follow the recommendations made. No change is required.

4.2.8 Messaging format

As discussed in the previous chapter, the use of JSON as the application level messaging format should be revisited in favour of more efficient approaches such as Protocol Buffers.

5 References

[1]: A. Tanenbaum: Computer Networks, 5th edition, 2010

[2]: OASIS: MQTT Version 3.1.1 Plus Errata 01, Dec 2015; available online at https://www.oasis-open.org/committees/mqtt

[3]: Chen at.al.: Responsive Mobile User Experience Using MQTT and IBM MessageSight, 2014; available online at http://www.redbooks.ibm.com/redbooks/pdfs/sg248183.pdf

[4]: Jansen et.al.: Marketplace Interface Specification, Oct. 2018; EcoGrid 2.0 deliverable

Read more at www.ecogrid.dk