

jOSEF: A Java-Based Open-Source Smart Meter Gateway Experimentation Framework

Michael Hoefling, Florian Heimgaertner, Daniel Fuchs, and Michael Menth

University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany
{hoefling,florian.heimgaertner,menth}@uni-tuebingen.de,
daniel.fuchs@student.uni-tuebingen.de

Abstract. Smart meter gateways are the core component of the advanced metering infrastructure in Germany, and provide a unified interface for metering data retrieval to third parties. Different standards and communication protocols exist for smart metering, ranging from transmission protocols to architectural recommendations. This work briefly presents the concept of the German BSI TR-03109 smart metering architecture, reviews implementations of smart metering protocols and architectures, and provides a *Java-based open-source smart meter gateway experimentation framework (jOSEF)*. The proposed framework combines and extends established protocol frameworks to provide a flexible tool for the validation of smart metering communication use cases involving smart meter gateways.

Keywords: smart grid, smart metering, smart meter gateway, advanced metering infrastructure, BSI TR-03109

1 Introduction

Electrical power distribution networks are undergoing major changes in operational procedures and monitoring, thereby evolving from passive to active networks [1, 2]. One consequence of these changes is the introduction of *advanced metering infrastructures (AMIs)* in the distribution grid to provide automatic billing, and acquisition of network status data. *Smart meters (SMs)* are the basic component of AMIs, replacing traditional electricity, gas and heat meters in the long run. SMs measure energy consumption and production in private households, commerce, and the industry, and provide mechanisms for remote meter reading. *Smart meter gateways (SMGWs)* gather metering information from several SMs, and provide a unified interface for meter information retrieval to interested and legitimate *external market participants (EMPs)*.

The *Cyber-secure Data and Control Cloud for power grids* (C-DAX) project [3] is an FP7 project funded by the European Commission which aims to develop a cyber-secure communication middleware for smart grids, applying the publish/subscribe (pub/sub) paradigm to enable scalable, transparent, and secure end-to-end communication [4] between publishers and subscribers. Additional major advantages of the C-DAX architecture include resilient communication [5], inter-domain communication, and support for real-time applications [6]. This work was conducted as part of the C-DAX project to evaluate the suitability of the C-DAX architecture for smart metering.

The main contribution of this paper is a brief presentation of the SMGW-based smart metering architecture as defined in BSI TR-03109 [7], and a description of jOSEF, a Java-based open-source SMGW experimentation framework. While existing implementations allow the isolated simulation and evaluation of certain smart metering communication aspects, a framework providing the minimally necessary building blocks for a BSI-03109-compliant SMGW-based architecture has been missing in both literature and in practice. This work addresses this gap and actually allows to model an SMGW-based smart metering architecture utilizing open-source components. This work mainly focuses on the German AMI approach [7] but considers the Dutch AMI approach [8] for technical details that have not been defined for Germany yet.

This work is structured as follows. We review relevant protocols for smart metering in Section 2, and present the concept of the BSI TR-03109 smart metering architecture in Section 3. In Section 4, we describe existing implementations and discuss their suitability for the development of an BSI TR-03109-compliant SMGW. Section 5 describes jOSEF in detail and Section 6 illustrates its functionality with selected communication scenarios. Section 7 concludes this work.

2 Related Work

The DLMS/COSEM suite is a set of standards for the exchange of energy meter data, comprising of *DLMS (device language message specification)* [9] as an application layer protocol for communication with metering devices, and *COSEM (companion standard for energy metering)* [10] as a system for object-oriented modeling of energy metering equipment. DLMS/COSEM uses the *object identification system (OBIS)* [11] to identify data objects in energy metering systems, and *COSEM services* enable clients to query specific attributes of objects, assign values to attributes of objects, or execute methods of objects.

SML (smart message language) [12] is a message-oriented protocol for communication with SMs. The SML application protocol defines *SML files* consisting of one or multiple *SML messages*. An *SML message* can be either a request or a response. SMs act as servers, receiving SML files from clients, and processing the contained SML messages in order of reception. Starting with version 1.04, SML supports COSEM services, i.e., the COSEM object model can be used with the SML application protocol. Currently, SML is not widely used outside Germany.

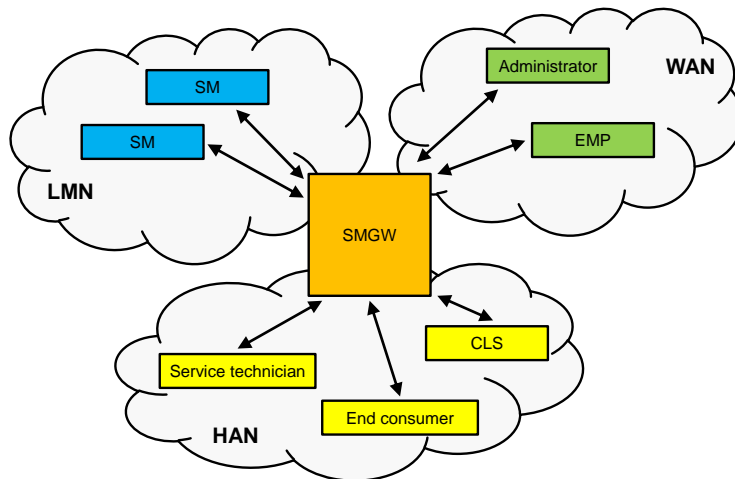


Fig. 1. System boundaries of the SMGW architecture according to [20]. The SMGW mediates between LMN, HAN, and WAN.

However, international use of SML is expected to increase if plans to adopt SML as part of the DLMS/COSEM suite [13] are successful.

M-Bus is a protocol suite for communication with SMs. M-Bus is defined in the European standard EN 13757 which comprises data model [14], application layer [15], and both wired [16] and wireless [17] specifications for the physical layer. The Open Metering System (OMS) [18, 19] is a smart metering communication architecture based on M-Bus. OMS proposes several modifications to the M-Bus protocols, and adds an optional authentication and fragmentation layer to the M-Bus protocol stack.

The *Dutch SM requirements (DSMR)* [8] are a joint specification of the Dutch grid operators. DSMR is based on DLMS/COSEM and M-Bus, and defines a data model for SMs including corresponding OBIS codes. We use selected parts of DSMR to fill the technical gaps of the German BSI TR-03109 for our framework.

3 Smart Meter Gateways: A Communication Topology for Smart Metering

Smart meter gateways (SMGWs) are the central communication components in the future smart metering infrastructure in Germany [7, 20]. The two most important functionalities of SMGWs are (1) gathering of metering data from SMs, and (2) providing a unified interface for metering data retrieval to interested and legitimate EMPs.

In general, the SMGW mediates between three networks, as shown in Figure 1: the *local metrological network (LMN)*, the *home area network (HAN)*, and the *wide area network (WAN)*. The LMN connects SMs to the SMGW only. The HAN connects end consumers, service technicians, and *controllable local systems*

(*CLSes*) to the SMGW, e.g., electric vehicles, photo-voltaic panels, and remote-controllable heating and air conditions. The WAN connects administrators and EMPs to the SMGW, e.g., distribution grid operators, metering point operators, and suppliers of electric energy.

3.1 Functionalities and Communication

The functionalities and the used communication protocols of SMGWs can be differentiated by the networks they mediate between.

In the LMN, SMGWs are responsible for gathering metering data from SMs according to metering profiles, time-stamping the measurements based on an externally synchronized time source, tariffing, and finally storing the time-stamped, tariffed metering data for further dissemination to EMPs. SMGWs support bidirectional and unidirectional communication with SMs. Bidirectional communication involves interactive communication between SMGWs and SMs to poll for metering data or to manage SMs. Unidirectional communication stands for unsolicited metering data dissemination from SMs to SMGWs. Generally, COSEM [10] with OBIS [11] codes are used as data model between SMs and SMGWs. Depending on the underlying physical layer, M-Bus [14, 16, 17, 15] or SML [12] is used as transport protocol.

In the HAN, SMGWs provide read-only access to their internally stored metering data and status messages to end consumers. SMGWs can support several end consumers facilitating multi-client operation, e.g., in an environment involving many SMs and many households. Service technicians must only access status messages of SMGWs. SMGWs relay control messages between CLSes and EMPs as configured by administrators. [20] does not specify protocols between SMGWs and potential HAN communication partners but security mechanisms to be used, e.g., secure transport layer communication, and mandatory authentication of clients against the SMGW. Essentially, any IP-based protocol may be used between SMGWs and HAN entities, e.g., end consumers or service technicians.

In the WAN, SMGWs are responsible for forwarding their internally stored metering data to interested and legitimate EMPs based on communication profiles. SMGWs must not accept connections from the WAN for security reasons but a wake-up service facilitates remote SMGW administration. When SMGWs receive specific control packets from the WAN, they contact an external administrator for maintenance, e.g., for firmware updates, changes in the communication profiles, time synchronization, or access to status messages. WAN communication is based on RESTful web services as defined in [20], and SMGWs act as RESTful web service clients because they must not accept connections from the WAN. EMPs must provide the server side of a RESTful web service according to the interface definitions in [20, 21]. As for LMN communication, COSEM with OBIS codes are used as data model between SMGWs and EMPs but XML and *cryptographic message syntax (CMS)* [22] are used as transport protocol on top of REST. Time synchronization of SMGWs is handled over the *network time protocol (NTP)* instead of web services.

3.2 Security

The BSI *SMGW protection profile (SMGW-PP)* [23] requires all LMN, HAN and WAN communication to be secured by *transport layer security (TLS)* in combination with a public-key infrastructure [24, 25]. WAN communication is further protected by CMS between SMGWs and EMPs. SMGWs are equipped with a security module which provides cryptographic functions, e.g., generation and secure storage of encryption keys, and verification of digital certificates. The security module is realized as a smart card. Further information on the security module and its requirements can be found in [26–28].

These security requirements limit the suitability of the C-DAX middleware for smart metering in Germany because C-DAX provides its own strong security mechanisms [4] but does not support TLS between communication partners without modification. However, if the BSI security regulations would permit replacing TLS by other security mechanisms with the same level of security, C-DAX may be used as communication middleware for HAN communication, e.g., between SMGWs and EMPs. In that case, C-DAX’ pub/sub mechanisms would allow scalable, secure and resilient dissemination of tariff information or firmware updates to all SMGWs, or transparent and secure remote control of a customers CLSes.

4 Existing Implementations

In this section, we review selected open-source implementations and discuss their suitability for the development of a BSI TR-03109 compliant SMGW.

OpenMUC [29] is an open-source implementation of a *multi utility communication controller (MUC)* developed at Fraunhofer ISE. OpenMUC is implemented in Java and licensed under the terms of the GNU General Public License (GPL). The core component of OpenMUC is the data manager which interfaces to optional components like data server, logger, protocol drivers, and custom applications. The OpenMUC framework provides the functionality specified in a previous draft standard for a German AMI. While BSI TR-03109 requires the use of XML for the RESTful web service, OpenMUC uses the JSON format. Additionally, the URI hierarchy used by OpenMUC differs from the BSI specification, and the protocol drivers do not satisfy the minimum requirements. Implementing a BSI TR-03109 compliant SMGW based on OpenMUC would require major changes to the OpenMUC code. However, the OpenMUC framework also includes protocol libraries that can be used independently, e.g., jDLMS and jSML. We discuss those in the following subsections.

jDLMS [30] is a Java implementation of the DLMS/COSEM protocol available under the terms of the GNU Lesser General Public License (LGPL). jDLMS supports the DLMS/COSEM application layer protocol over serial lines using HDLC, or over TCP or UDP. As the current version 0.9.0 only implements the client side of the DLMS/COSEM protocol, and does not include the COSEM object model, jDLMS is not suitable for implementing a SMGW according to BSI TR-03109.

jSML [31] is a Java implementation of SML available under the terms of the LGPL. jSML supports SML communication over TCP/IP or over serial line. HDLC support is currently unavailable. jSML implements the SML message format and encoding, the SML data types, and the SML transport layer version 1. The current version 1.0.17 is based on SML 1.03, i.e., jSML does not yet support COSEM services. For implementing a SMGW according to BSI TR-03109, jSML needs to be extended to support the current SML version.

Open gateway energy management (OGEMA) [32] is an OpenMUC-based software platform for building automation and load management. OGEMA is implemented in Java and licensed under the terms of the GPL. Since OGEMA is based on OpenMUC and focuses rather on the HAN side than on the SMGW, it is not suitable for implementing a SMGW according to BSI TR-03109.

Gurux [33] is a collection of smart metering software components developed by Gurux Ltd. Gurux code contains implementations in C#, C++, Java, and Delphi and is licensed under the terms of the GPL. Gurux supports DLMS/COSEM, Modbus, and M-Bus. However, the COSEM object model is not separated from the DLMS/COSEM application protocol in the Gurux code. Using the COSEM object model with other application protocols as required by BSI TR-03109, would require major modifications to the code.

5 jOSEF: A Java-Based Open-Source SMGW Experimentation Framework

We now present jOSEF, a Java-based open-source SMGW experimentation framework. jOSEF is licensed under the terms of the GPL version 2 or later. We describe its architecture, specify its operation, and discuss the deviation from BSI TR-03109. The implementation utilizes the jSML library [31] for LMN communication that was extended as part of this work to support SML version 1.04 [12]. Additionally, we used the COSEM implementation of Gurux [33] as a blueprint to implement DSMR's COSEM object model. We used DSMR's COSEM object model because BSI TR-03109 has not defined a companion standard for its COSEM object model yet.

5.1 Components

The framework comprises three main components: a minimal SMGW, an SM simulator, and a simple EMP. The *minimal SMGW* represents an SMGW which provides the minimally necessary functionalities to control SMs and to send meter data to EMPs. It is equipped with a GUI for configuration and operation, and allows several SMs to be connected, as shown in Figure 2. The *SM simulator* represents an SM and can be configured with standard load profiles for energy generation and consumption to simulate SM behavior. It is controlled over a CLI, as shown in Figure 3. The *simple EMP* provides an BSI TR-03109-compliant RESTful web service towards the SMGW and acts as a data sink for meter data. It can be accessed using any HTTP client, e.g., a web browser.

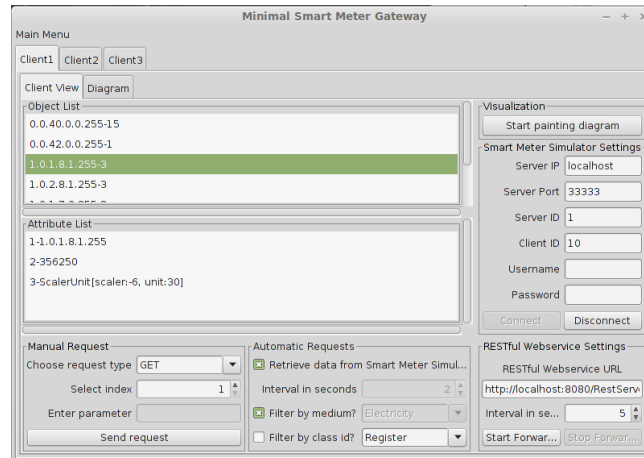


Fig. 2. Screenshot of the SMGW GUI of JOSEF. Client 1, 2, and 3 correspond to three different SMs that are connected to the SMGW, and the attribute list pane shows a detailed view of attribute 1.0.1.8.1.255 (electricity consumption in Wh) of client 1.

```

Welcome to the Smart Meter Simulator user interface!
Menu:
[1] View server settings.
[2] View simulator settings.
[0] Exit.
1
View server settings:
-> Server is listening on port [33333]
-> Server is not using authentication for clients
Menu:
[1] View server settings.
[2] View simulator settings.
[0] Exit.
2
View simulator settings:
-> Simulated annual power consumption in kWh [1000.0]
-> Simulated annual power infeed in kWh [1000.0]
-> Simulation-time of a 'realtime-quarter-hour' in seconds [4]
-> Simulation resolution in milliseconds [1000]
Menu:
[1] View server settings.
[2] View simulator settings.
[0] Exit.
0
Closing program...

```

Fig. 3. Console log of the SM simulator CLI. The CLI allows the user to view its configuration. The configuration may only be changed on SM simulator startup via a configuration file.

5.2 Meter Data Retrieval

When the SMGW wants to retrieve meter data from an SM, it first sends an SML message to the SM requesting all internal COSEM object IDs to discover the SM's internal data model. The returned list of COSEM object IDs is then used by the SMGW to build the actual meter data retrieval request by filtering for metering object IDs based on OBIS codes. The SMGW generates a new SML message containing explicit requests for details on the metering object IDs, and sends the message to the SM. The SM returns the actual metering objects to the SMGW that can perform further processing on the data, e.g., time stamping, tariffing, buffering, or dissemination to EMPs. The SMGW re-

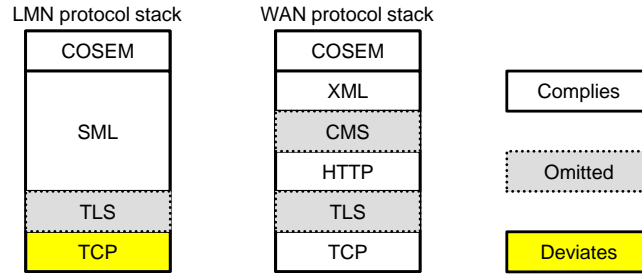


Fig. 4. Deviations of the LMN and WAN protocol stack of the current jOSEF implementation from BSI TR-03109. Currently omitted protocol layers are shown as gray, dashed boxes, and deviating protocol layers are shown as yellow, solid boxes.

sends the meter data request message to the SM to receive new meter data; rediscovery of the SM's internal data model by the SMGW is only necessary when the SM configuration changes.

5.3 Meter Data Dissemination

When the SMGW retrieved new meter data from an SM, data conversion is necessary before actual dissemination to EMPs because LMN communication is based on COSEM over SML, and WAN communication is based on COSEM over XML. The common data model between SM, SMGW and EMP is COSEM so that data conversion works straightforward, i.e., a mapping of COSEM objects to XML is defined in [21]. After conversion to COSEM over XML, the SMGW sends the meter data to the EMP using the appropriate RESTful web service endpoint and HTTP methods. The EMP stores the received meter data and can perform further processing on the data.

5.4 Limitations

Our current framework implementation deviates from BSI TR-03109 in some minor points which we consider not important if the framework is used for laboratory communication experimentations only. These deviations need to be considered when using the framework for experiments involving insecure network connections between framework components. Minor deviations include that we do not support HDLC and serial links between the SMGW and the SM at the moment because the SM simulator uses TCP to communicate with the SMGW, as shown in Figure 4. Further, the SMGW does not perform tariffing on received meter data, it does not support remote administration over the WAN, and it does not perform pseudonymization of meter data before sending them to EMPs. The framework implements only a limited subset security functionalities, e.g., password-based authentication of SMGWs against SMs is available, but no authentication between SMGWs and EMPs. Additionally, we do not use TLS for LMN and WAN communication, and we do not use CMS to further secure WAN communication, as shown in Figure 4.

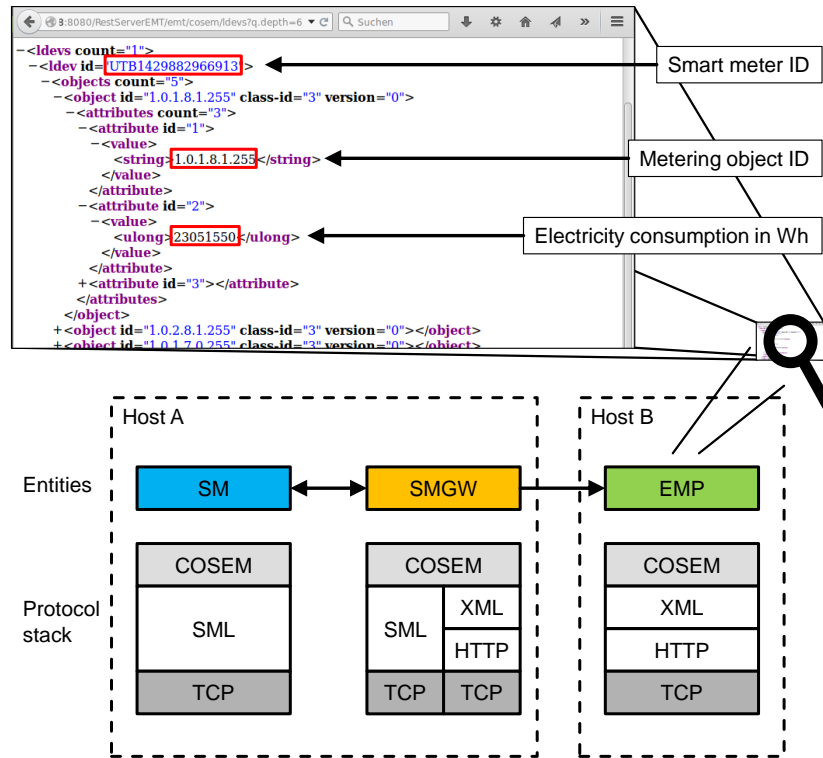


Fig. 5. Illustration of basic LMN and WAN communication including involved entities and protocol layers. A SMGW requests meter data from a SM, translates from COSEM over SML to COSEM over XML, and forwards the meter data to an EMP. The screenshot on the top shows what the XML structure of meter data looks like at the EMP.

6 Illustration

We now illustrate the functionality of the proposed framework by experimentation. The setup of the experiment is described first, followed by experimental results from traffic experiments. Our results show that our framework enables easy modeling of typical smart metering topologies and communication patterns.

6.1 Experiment Setup and Methodology

To illustrate the functionality of the proposed framework, we created a simple dumbbell-like topology with one SM on the left side, one SMGW in the middle, and one EMP on the right side, as shown in Figure 5. The SM was simulated by an SM simulator instance, configured to use a H0 load profile [34] and a EV0 generation profile [35]. The SMGW was configured to actively poll its associated SM every 2 seconds for new meter readings, and to forward all internally buffered

metering data unsolicited to the EMP every 5 seconds. The EMP buffered meter readings from the SMGW, queryable via a RESTful web service interface. We deployed our setup on two end-hosts connected via a 100 Mbit/s Ethernet link.

6.2 Basic LMN and WAN Communication

First, the SM simulator, the SMGW and the EMP are started. The SMGW contacts the SM, queries for the SM’s internal object list, and then subsequently polls for electricity objects only. When the SMGW receives the first meter data from the SM over SML, it translates from COSEM over SML to COSEM over XML, and starts forwarding the meter data to the EMP using HTTP PUT. The EMP stores the received meter data and provides access to it over a RESTful web service. The screenshot on top of Figure 5 shows what meter data looks like at the EMP when the EMP is queried via HTTP GET, e.g., using a web browser. We can see that the SM with device id `UTB1429882966913` consists of five electricity objects. For better illustration, only object `1.0.1.8.1.255` has been expanded in the figure, and the electricity consumption in Watt hours can be read.

7 Conclusion

BSI TR-03109 defines an SMGW-based smart metering infrastructure as it will be deployed in Germany. In this work, we presented the concept of BSI TR-03109, briefly reviewed implementations of smart metering protocols and architectures, and constituted that they only allow limited evaluation of smart metering communication aspects. Therefore, we proposed jOSEF, a Java-based open-source framework for smart metering communication experimentation, and evaluated its functionality. Our proposed framework combines and extends established protocol frameworks, thus providing a flexible tool for SMGW-based smart metering communication validation, e.g., adapting BSI TR-03109 to the C-DAX communication middleware. Furthermore, our extension of the jSML protocol library allows the implementation of independent programs. The source code of jOSEF and its subcomponents like the SML v.1.04 extension of the jSML library is available online [36].

Future work includes compatibility tests of our SM simulator against commercial SMGWs, and compatibility tests of our SMGW implementation against commercial SMs. We further envision the implementation of additional smart metering use cases such as SM management. Additionally, security mechanisms such as TLS support shall be included in the existing implementation.

Acknowledgement. The research leading to these results has received funding from the European Community’s Seventh Framework Programme FP7-ICT-2011-8 under grant agreement n° 318708 (C-DAX). The authors alone are responsible for the content of this paper.

The authors thank Roelof Klein, and Martijn Kammerling, both working at Alliander, for valuable input and stimulating discussions.

References

1. CIGRE Working Group C6.11: Development and Operation of Active Distribution Networks (Apr 2011)
2. Heydt, G.T.: The Next Generation of Power Distribution Systems. IEEE Transactions on Smart Grid 1(3) (2010)
3. C-DAX Consortium: Cyber-secure Data And Control Cloud for Power Grids (2014), <http://www.cdax.eu/>
4. Heimgaertner, F., Hoefling, M., Vieira, B., Poll, E., Menth, M.: A Security Architecture for the Publish/Subscribe C-DAX Middleware. In: Workshop on Security and Privacy for Internet of Things and Cyber-Physical Systems (IoT/CPS-Security) in conjunction with IEEE International Conference on Communications (ICC). London, UK (Jun 2015)
5. Hoefling, M., Heimgaertner, F., Menth, M., Katsaros, K.V., Romano, P., Zanni, L., Kamel, G.: Enabling Resilient Smart Grid Communication over the Information-Centric C-DAX Middleware. In: ITG/GI International Conference on Networked Systems (NetSys). Cottbus, Germany (Mar 2015)
6. Chai, W.K., Wang, N., Katsaros, K.V., Kamel, G., Melis, S., Hoefling, M., Vieira, B., Romano, P., Sarri, S., Tesfay, T., Yang, B., Heimgaertner, F., Pignati, M., Paolone, M., Menth, M., Pavlou, G., Poll, E., Mampaey, M., Bontius, H., Develder, C.: An Information-Centric Communication Infrastructure for Real-Time State Estimation of Active Distribution Networks. IEEE Transactions on Smart Grid (to appear)
7. Bundesamt für Sicherheit in der Informationstechnik: Technische Richtlinie BSI TR-03109
8. Netbeheer Nederland: Dutch Smart Meter Requirements: P1 Companion Standard. DSMR Version 5.0
9. International Electrotechnical Commission: Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 5-3: DLMS/COSEM Application Layer. IEC 62056-5-3 ed1.0 (2013)
10. International Electrotechnical Commission: Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 6-2: COSEM Interface Classes. IEC 62056-6-2 ed1.0 (2013)
11. International Electrotechnical Commission: Electricity Metering Data Exchange - The DLMS/COSEM Suite - Part 6-1: Object Identification System (OBIS) . IEC 62056-6-1 ed1.0 (2013)
12. Bundesamt für Sicherheit in der Informationstechnik: BSI TR-03109-1 Anlage IV: Feinspezifikation Drahtgebundene LMN-Schnittstelle, Teil b: SML - Smart Message Language. SML Version 1.04
13. International Electrotechnical Commission: Electricity Metering Data Exchange - Part 5-3-8 Smart Message Language SML. IEC 62056-5-3-8 (future standard)
14. European Committee for Standardization: Communication Systems for and Remote Reading of Meters - Part 1: Data Exchange. EN 13757-1:2015-01 (2015)
15. European Committee for Standardization: Communication Systems for and Remote Reading of Meters - Part 4: Wireless Meter Readout. EN 13757-4:2014-02 (2014)
16. European Committee for Standardization: Communication Systems for and Remote Reading of Meters - Part 2: Physical and Link Layer. EN 13757-2:2004 (2004)
17. European Committee for Standardization: Communication Systems for and Remote Reading of Meters - Part 3: Dedicated Application Layer. EN 13757-3:2013-08 (2013)

18. OMS Group: Open Metering System Specification, Volume 1: General Part. OMS Spec Vol1 1.4.0 (2011)
19. OMS Group: Open Metering System Specification, Volume 2: Primary Communication, Version 4.0.2. OMS Spec Vol2 4.0.2 (2014)
20. Bundesamt für Sicherheit in der Informationstechnik: Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems, Technische Richtlinie BSI TR-03109-1, Version 1.0
21. Bundesamt für Sicherheit in der Informationstechnik: BSI TR-03109-1 Anlage II: COSEM/HTTP Webservices
22. Bundesamt für Sicherheit in der Informationstechnik: BSI TR-03109-1 Anlage I: CMS-Datenformat für die Inhaltsdatenverschlüsselung und -signatur
23. Bundesamt für Sicherheit in der Informationstechnik: Schutzprofil für die Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen, Version 1.3. BSI SMGW-PP 1.3
24. Bundesamt für Sicherheit in der Informationstechnik: Kryptographische Vorgaben für die Infrastruktur von intelligenten Messsystemen, Technische Richtlinie BSI TR-03109-3, Version 1.1
25. Bundesamt für Sicherheit in der Informationstechnik: Public Key Infrastruktur für Smart Meter Gateways, Technische Richtlinie BSI TR-03109-4, Version 1.0
26. Bundesamt für Sicherheit in der Informationstechnik: Smart Meter Gateway - Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls, Technische Richtlinie BSI TR-03109-2, Version 1.1
27. Bundesamt für Sicherheit in der Informationstechnik: Kryptographische Vorgaben für Projekte der Bundesregierung, Teil 3 - Intelligente Messsysteme, Technische Richtlinie BSI TR-03116-3
28. Bundesamt für Sicherheit in der Informationstechnik: Schutzprofil für das Sicherheitsmodul der Kommunikationseinheit eines intelligenten Messsystems für Stoff- und Energiemengen, Version 1.02. BSI SecMod-PP 1.02
29. Feuerhahn, S., Zillgith, M., Becker, R., Wittwer, C.: Implementation of an Open Smart Metering Reference Platform - OpenMUC. In: ETG-Kongress (2009)
30. Mueller-Bier, K.: jDLMS. <http://www.openmuc.org/index.php?id=42> (Jul 2013)
31. Feuerhahn, S., Buehrer, M.: jSML. <http://www.openmuc.org/index.php?id=63> (Jun 2014)
32. Fraunhofer IWES: Open Gateway Energy Management - OGEMA. <http://www.ogema.org/> (Apr 2014)
33. Gurux Ltd.: Gurux Open Source Device Communication. <http://www.gurux.fi/>
34. KommEnergie: Lastprofile von KommEnergie. <http://www.kommenergie.de/?id=140>, last visited 14.08.2015
35. Stadtwerke Emmendingen: Lastprofile der Stadtwerke Emmendingen. <https://swemendingen.de/netz/strom-netz/lastprofile/>, last visited 14.08.2015
36. jOSEF: A Java-Based Open-Source Smart Meter Gateway Experimentation Framework. <http://kn.inf.uni-tuebingen.de/software/josef>, last visited 14.08.2015