Intercloud Communication for Value-Added Smart Home and Smart Grid Services

Philipp Grubitzsch(✉)

Computer Networks Group, Technische Universität Dresden,
01062 Dresden, Germany
philipp.grubitzsch@tu-dresden.de

Abstract. The increasingly decentralized generation of renewable energy enables value-added smart home and smart grid (SHSG) services. The device data on which those services rely are often stored in clouds of different vendors. Usually, the vendors' clouds all offer their own service interfaces. It is increasingly challenging for service providers to access the data from all these clouds. Hence, each cloud forms a data silo, where users' device data are captured. Intercloud computing is one suggested approach to solve this uprising vendor silo problem. Introducing a standardized service interface and simply interconnecting the clouds can easily result in an unnecessary communication overhead. Compared to other domains applying Intercloud computing, the device data in the SHSG domain has special characteristics. These characteristics should be considered for the design of an appropriate communication architecture. Thus, the focus of this research is on an efficient communication for discovering and delivering device data in an SHSG Intercloud scenario. Therefore, we present an architecture introducing an Intercloud Service (ICS) on top of the vendor clouds. An evaluation methodology is proposed to investigate the efficiency of the chosen solution for the ICS.

Keywords: Smart home · Smart grid · Intercloud computing

1 Introduction

Smart homes enable users a new quality of living by automation. In contrast, the smart grid is primarily an emerging technology to enable the shift from centralized energy production, based on fossil and nuclear sources, towards distributed production of renewable energy. Nevertheless, the smartness in both domains relies on *value-added services*. From several influence factors, they calculate new information and then present it to the users or make decisions to automatically control devices. In smart home, benefit of this services is a gain in comfort or to save money. In smart grid, they support the transmission and distribution side operators to keep the grid stable. One important influence factor for generating added value is the households real time *device data* like sensor measurements and device states.

I. Ciuciu et al. (Eds.): OTM 2015 Workshops, LNCS 9416, pp. 10–19, 2015.
DOI: 10.1007/978-3-319-26138-6_3

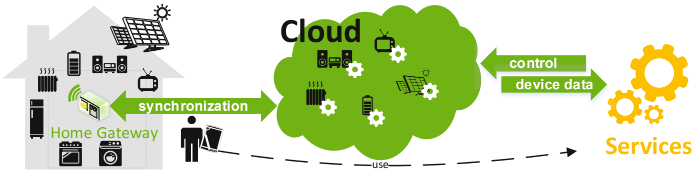
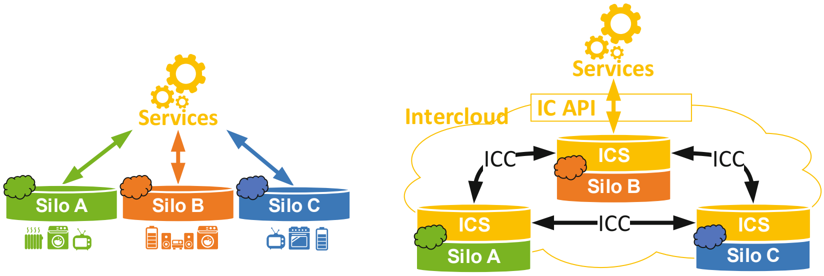


Fig. 1. SHSG cloud architecture

General Architecture: It is a trend in progress to move both, the value-added services [15] and the stored devices data [17] from the home into the cloud. Fig. 1 shows the common way how this is achieved for the device data. We call this approach the Smart Home/ Smart Grid (SHSG) cloud architecture. All devices in a household are connected to a *home gateway* [3,11,20]. Commonly, the home gateway uses an existing internet access (e.g. ADSL or Cable) to establish a bidirectional connection to a cloud. Every local device has a virtual counterpart in the cloud which fully represents its state. Synchronization is automatically triggered when devices send their live data to the home gateway, which then immediately pushes the data to the cloud. This changes the state of the corresponding virtual devices in the cloud. Vice versa, devices can also be remote-controlled by the cloud or a service. In that case, the virtual device representation changes and the cloud sends commands to the home gateway. To control devices and access their data, the cloud offers interfaces to users and services. Home gateways are bound to users, who can grant or revoke access rights to their devices and device data. Services or other users need access rights to request device data of a certain user's virtual devices.

The Data Silo Challenge: Today, many companies are trying to establish their own cloud solution. This increases the gap of service interface and data model interoperability between these solutions [19]. As a result, customers' data get captured in individual data silos (Fig. 2a). Nevertheless, the services generate more additional value with more available device data of more users. Hence, it has to be possible to access data from other silos to fully leverage the SHSG technology. In consequence, service providers are forced to support many different interfaces with their service implementation. Furthermore, they need to request device data separately from each cloud provider. Therefore, services have to know about each cloud. A centralized Google-like search index to work around this issue is not possible due data control concerns of the cloud providers. Dependency of a centralized index contradicts their business interests.

One suggested way to solve this issue is to utilize Intercloud computing [17]. As shown in Fig. 2b, our own approach follows this suggestion by extending the data silos with an *Intercloud Service* (ICS). It enables the silos to establish an *Intercloud-communication* (ICC) to access user-shared device data from other silos. Services use an *Intercloud API* (IC API) to transparently access the entire Intercloud by querying just a single arbitrary cloud, acting as *gateway cloud.* Thereby, services do not need to discover and collect the entire distributed data



(a) service accessing vendor silos (b) service using an Intercloud

Fig. 2. From silo challenge to Intercloud approach

from all clouds in the Intercloud, being *peer clouds*. The chosen gateway cloud has to process the query in two main steps: the ***discovery*** of the device data associated with the query, and the ***distribution*** of these data to the requesting services by collecting it from the peers and aggregating it for delivery.

The goal of this work is an efficient communication for discovery and distribution of device data in an SHSG Intercloud.

2 Related Work

SHSG clouds might be deployed on an Infrastructure as a Service (IaaS) solution like Amazon EC2[1] or MS Azure[2]. Beyond that, they themselves are usually provided as Platform as a Service (PaaS), which enables 3rd-party developers to create and run own value-added services on top of them. These services are then offered as Software as a Service (SaaS) to customers [16].

Numerous research has been conducted to clarify and classify SHSG services (SaaS) [5,12,16] and SHSG clouds (PaaS) [9,11,16,17]. Mostly, they have following requirements in common: a *data model* which supports many different devices encapsulated in a provided *service interface*, an *information management system* for virtual device and data management including appropriate privacy and access control, *bidirectional real time communication* with support of *data-centric* and *topic-based group* communication models and the ability to ensure *Quality of Service* (QoS) by a *Service Level Agreement* (SLA) model.

In [8] the authors suggest and evaluate a model for information management based on a classification of devices, user activities and communication. [2] investigates how Message-oriented-Middlewares (MOM) like XMPP, AMQP or DDS can be utilized to fulfill the requirements of the bidirectional real time communication for SHSG, while [3] examined these protocols for their suitability of *QoS*. QoS is important to ensure that a service can work as intended. In [4] the authors investigated QoS stovepipes and emphasizes the importance of the MOM

1 http://aws.amazon.com/de/ec2/
2 http://azure.microsoft.com

to achieve interoperability. [14] propose an SLA framework for monitoring QoS of smart grid services. Results of other research investigated show whole architectures or frameworks trying to address several of the mentioned requirements for smart grid [9] and smart home [20,21] clouds. All of them just show how to establish a single SHSG cloud, which will lead to vendor silos as mentioned in Sec. 1.

With the increasing number of cloud platforms in any domain, interoperability among several clouds moved into focus of research. Intercloud computing is the suggested way to achieve this [6,7,17,19]. Primarily, this relies on an interoperable *data model* and *service interface* and a communication management model between clouds, often called *broker*. The broker is responsible for provisioning of resources (discovery and distribution of services/ data) among all cloud entities in a QoS manner, determined by an SLA. Access control is supposed to be handled across several clouds. Trust between clouds is also an important concern. According to patterns of Intercloud communication, the MOM (and utilized protocols), should support a federated communication model.

An orchestration service (broker) to achieve C2C service interoperability with focus on QoS, access control and privacy across multiple cloud instances is described in [10]. [13] proposes a broker-based approach to interconnect arbitrary dynamic service plattforms (similar to PaaS-like clouds) with a simple trust model. An implementation utilizes XMPPs rostergroups for the trust model, presence function and further XMPP extension protocols (XEP) for discovering and availability tracking of the distributed services. [18] built a federated sensor network between three involved universities on top of XMPP Multi-User chat (MUC) rooms. An architecture for a Media Intercloud which takes almost all mentioned Intercloud requirements into account is proposed in [1]. They also utilize XMPP as MOM for the signaling and RDF/SPARQL for the data model/requesting in their API.

Research has rarely been conducted to extend the SHSG requirements to an Intercloud approach, yet. [17] proposed a decentralized model as cloud of clouds for the smart grid domain. Moreover, they demand a uniform and transparent (agnostic to C2C infrastructure) device data access for all stakeholders called GET API.

After thorough review of related work, there is no – to the best of our knowledge – complete architecture for an SHSG Intercloud, yet. None of the related Intercloud work has taken into account the special characteristics of the device data in order to obtain an efficient Intercloud communication within the SHSG domain. Only if this aspect is considered for the design of the *discovery* and *distribution* model, we expect a much more efficient communication.

3 Research Hypotheses

Our proposed architecture for an efficient Intercloud communication relies on the following hypotheses:

Hypothesis 1 - *A Semi-Distributed Directory Service with Partially Replicated Meta-Data Enables Efficient Resource Discovery in SHSG Scenarios, Yielding Data Control Concerns.*

As already mentioned, a centralized discovery solution is not desirable due to data control concerns. A dedicated solution is desired by each cloud provider. A semi-distributed directory service could avoid centralized storing of searchable SHSG device meta-data. Such meta-data could be key-value pairs of devices, location, or access permissions. These may be either replicated to all instances, trusted subsets, or not replicated. Having no directory services, queries would have to be sent broadcast-like to all peer clouds. However, with a rising number of clouds this approach appears to become inefficient. Thus, we want to elaborate possibilities for a partially replicated placement of meta-data. Semi-distributed directory services could be a reasonable possibility. Within that, part of the meta-data is shared among the directory services in an Intercloud compound while part of it remains at the individual clouds. Feasible replication strategies for exchanging and organizing the meta-data are required for an efficient query processing in the Intercloud. This leads to the following research questions:

Q1: What are the important device meta-data to be shared between the directory services of the clouds in order to achieve an efficient discovery communication between the clouds?
Q2: In order for our approach to surpass the efficiency of broadcasting queries, what are the thresholds for numbers of clouds and devices as well as device type distributions in the participating clouds?

Hypothesis 2 - *Dynamic topic grouping for data-centric communication avoids redundant data delivery for continuous device data*

As mentioned before, the requested device data might be distributed among many clouds. It can be delivered in two ways specified by the request itself: one time (PULL) or as subscription for a continuous delivery (PUSH). After successful retrieval of peer clouds from its directory service, the gateway cloud is also responsible for delivering the requested data to the querying service. Therefore, it must aggregate the result streams from the peer clouds and deliver them to the querying service. To address use cases in which two or more services request data sets for PUSH delivery, forming an intersection, we propose a data-centric communication applying topic grouping to avoid redundant transmission of the same data between clouds. We expect that this will significantly increase efficiency of network resource usage. This leads to the following research questions:

Q3: What are the circumstances for (re-)grouping (aggregate/split) the topics and what is the strategy for that?
Q4: When considering the resource costs of the platform infrastructure (processing, network), does the gain in network efficiency justify the additional required processing power for topic grouping?

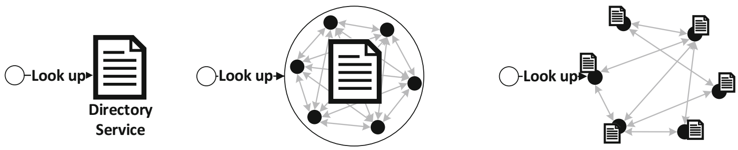


Fig. 3. Directory Services - A) single centralized B) single distributed C) semi-distributed

4 Research Methods and Material

The Intercloud Service (ref. Fig. 2b) is the core component of our research prototype. The IC API will support both queries mentioned before, namely PULL and PUSH. They work with filter mechanisms on top of a device model. The interoperability of the IC API shall be deemed out of research scope; instead, we wish to focus on:

Discovery: Under hypothesis 1, we expect a directory service (DS) to perform better than an approach which broadcasts queries. We have taken several DS architectures into our considerations (Fig. 3); A) a single centralized DS was already excluded in Sec. 1 due to data control concerns; B) a single distributed DS (e.g. based on a distributed hash table or full replication) is also infeasible, because every cloud would also have the full data set; and finally C) an approach of multiple semi-distributed directory services with partially replicated meta-data meets our requirements best. Directory information is partially shared by trusted peering between clouds. This is similar to friends in a contact list of an instant messenger as we have prior described in [13]. Hence, every cloud has its own directory service, storing just the information provided by their trusted peers. In attempts to answer *Q1*, we identified three possible device meta-data for replication: the device model, the location and the access permissions. The information structure within the DS can effect the efficiency of searches on the meta-data. This affects the information which has to be stored in the directory services and therefore also their sizes and required processing time. Further issues to be discussed are privacy aspects of users' device data. This can lead from just sharing coarse grain data (e.g. instead of exact location just the city or region) up to not exchanging certain data. Further details of the strategy and also the structuring of the data in the DS are still being investigated. We plan to experiment with different existing solutions for storing and querying the device meta data. Possible candidates are LDAP-based DS, or index implementations of Apache Lucene[3] like Elasticsearch[4] or SolR[5].

Distribution: To address hypothesis 2, we outlined a data-centric delivery which uses dynamic topic grouping (Fig. 4). To deliver information topic based from the

[3] https://lucene.apache.org/
[4] https://www.elastic.co/products/elasticsearch
[5] http://lucene.apache.org/solr/

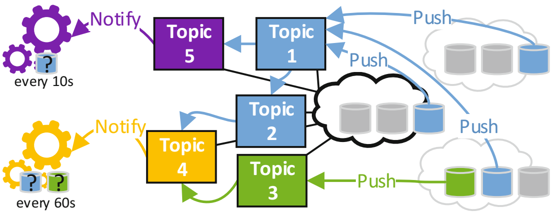


Fig. 4. Dynamic Topic Grouping - two services querying partially the same data

source side, the smallest information are key-value pairs described by the device model. From the service point of view, a topic is an aggregation of all information to serve a specific request. The topic grouping can be found between data source and service. Multiple requests can partially query for the same data but still differ in some details. For instance, the SLA/QoS may influence the outcome: SERVICE A requests DATA to delivered every 10 seconds, while SERVICE B requests DATA every 60 seconds for delivery. Therefore, just one stream per cloud is required to send DATA to TOPIC 1 every 10 seconds. The gateway cloud then also creates a new TOPIC 2 for SERVICE B and copies the value from TOPIC 1 every 60 seconds. Several other cases still have to be investigated.

The Intercloud Service is planned to build on top of *XMPP* as communication middleware. We have been evaluating several other middlewares; one of the outcomes is described in [18]. A major advantage for XMPP is the possibility of federated communication between servers and the extensive support of different communication patterns by extensions called *XEP*.

To prove our hypotheses, we want to compare our approach to a state-of-the-art architecture which serves as *baseline*. The baseline is using a broadcast mechanism for the discovery of device data. Thereby, a query for arbitrary device data results in a forward of the request to all known clouds. For the data distribution it uses a common request-oriented delivery. This means, data is transmitted separately for each individual request to the cloud, which initially has forwarded it. If two or more requests are sent for the same device data, it will result in redundant transmission.

For the suggested research method we are developing a hybrid testbed. It consist of an *emulator*, a *commercial cloud solution* and is controlled by a *testbed Manager*. The emulator is able to emulate thousands of clients on a single computer. A client is either a home gateway with several simulated devices producing real device data, or a generic value-added service requesting device data according to a predetermined pattern. Clients have a real network connection to several clouds. For the clouds we are using a solution from our industry partner KIWIGRID[6], extended by our proposed Intercloud Service. The testbed Manager comes with editors to setup and link devices, home gateways, value-added services and clouds, and a distributed execution environment.

[6] http://www.kiwigrid.com

With the proposed testbed we are able to produce and monitor real world network traffic for certain scenarios for device distribution (number in the clouds, device type heterogeneity), cloud configuration (linking between clouds, overall cloud number in Intercloud) and cloud architectures (baseline vs. our approach).

5 Discussion and Future Work

Each aspects of our conceptual progress, namely communication concept, testbed, and evaluation, needs to be individually reviewed.

Details of the **communication concept** for discovery and distribution remain to be conceived, followed by a determination of qualified data and their structure for the directory service. Further, the numerous special cases of PUSH data distribution need to be identified and investigated. For each of these cases, (re-)grouping strategies need to be designed.

Currently, the **testbed** is the most advanced part of our work. We are able to emulate all the clients and setup several clouds with the Intercloud communication of the baseline architecture. Therefore, we are currently developing a description language and an editor which allows us to create different scenarios of device type distribution, cloud constellation in the Intercloud, and to create query patterns for the emulated services. Next step for the testbed will be the implementation of the distributed execution environment. This will enable deployment of clients and clouds on available computers in our testbed network environment automatically.

For the **evaluation**, we plan to perform experiments in form of several scenarios. Table 1 shows the variables for each scenario and the measurands which serve as comparators to evaluate the performance of each architecture. With the chosen set of parameters we aim to show several possible constellations of clouds, distributed device data and network traffic and how each architecture will perform. Regarding research questions *Q2* and *Q4* we expect our approach to perform best in scenarios with many clouds, each managing a huge number of heterogeneous device types and services often requesting same subsets of data from their gateway clouds to the Intercloud. We assume, the choice for the right

Table 1. Parameters for evaluation experiments

Variables	Measurands

• architecture: own approach, baseline, others
• # of clouds: 1,2,5,10,...,50
• # of homegateways per cloud: up to thousands
• # of overall devices per cloud: up to thousands
• heterogeneity of device type distribution
• # of similar service requests
	– network
	  • average latency in ms
	  • average bandwidth used in MBit/s
	  • overall message count
	– processing
	  • CPU load in %
	  • memory used in MB

architecture depends on the kind of scenario in the future. With our evaluation method it shall also be possible to show which architecture performs better in a certain scenario. For the experiments we still need to design convincing scenarios, including reasonable combinations and values for the variables.

6 Conclusion

In this paper we propose an Intercloud approach for smart home and smart grid clouds to solve the uprising vendor silo problem of user's device data. Our approach focuses on an efficient communication to discover and distribute SHSG device data between participating clouds. For the proposed discovery approach, we address data control concerns of the cloud providers. Our suggested distribution mechanism supports QoS to fulfill certain SLAs while avoiding redundant data transmission for similar service requests. To prove our concept, a testbed currently in development is described. It will enable us to perform experiments for several possible future Intercloud scenarios and thus to answer our research questions.

Acknowledgments. This research is being conducted for my PhD efforts in context of the ZEEBus[7] project (fund number 16KIS0091) and is funded by the Federal Ministry of Education and Research (BMBF) in Germany. Gratitude shall be attributed to my supervisor Daniel Schuster, as well as my colleagues for proofreading. Additional thanks for the free pictographs of household appliances, used in Fig. 1 which where designed by Freepik[8].

References

1. Aazam, M., Huh, E.N.: Inter-cloud media storage and media cloud architecture for inter-cloud communication. In: 2014 IEEE 7th International Conference on Cloud Computing, pp. 982–985 (2014)
2. Albano, M., Ferreira, L.L., Pinho, L.M., Alkhawaja, A.R.: Message-oriented middleware for smart grids. Computer Standards & Interfaces **38**, 133–143 (2015)
3. Alkhawaja, A.R., Ferreira, L.L., Albano, M.: Message Oriented Middleware with QoS Support for Smart Grids, pp. 7–17 (2012)
4. Bakken, D.E., Schantz, R.E., Tucker, R.D.: Smart Grid Communications: QoS Stovepipes or QoS Interoperability? Tech. rep. (2009)
5. Bera, S., Misra, S., Rodrigues, J.J.: Cloud Computing Applications for Smart Grid: A Survey. IEEE Transactions on Parallel and Distributed Systems **PP**(99), 1 (2014)
6. Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., Morrow, M.: Blueprint for the intercloud - protocols and formats for cloud computing interoperability. In: Proceedings of the 2009 4th International Conference on Internet and Web Applications and Services, ICIW 2009, pp. 328–336 (2009)

[7] http://www.vdivde-it.de/KIS/kmu-innovativ/zeebus-1
[8] http://de.freepik.com

7. Bernstein, D., Vij, D.: Intercloud directory and exchange protocol detail using XMPP and RDF. In: Proceedings - 2010 6th World Congress on Services, Services-1, pp. 431–438, July 2010
8. Capitanelli, A., Papetti, A., Peruzzini, M., Germani, M.: A Smart Home Information Management Model for Device Interoperability Simulation. Procedia CIRP **21**, 64–69 (2014)
9. de Diego, R., Martnez, J.F., Rodrguez-Molina, J., Cuerva, A.: A Semantic Middleware Architecture Focused on Data and Heterogeneity Management within the Smart Grid. Energies **7**(9), 5953–5994 (2014)
10. Dowell, S., Barreto, A., Michael, J.B., Shing, M.T.: Cloud to cloud interoperability. In: Proceedings of 2011 6th International Conference on System of Systems Engineering: SoSE in Cloud Computing, Smart Grid, and Cyber Security, SoSE 2011, pp. 258–263 (2011)
11. Fan, Z., Kulkarni, P., Gormus, S., Efthymiou, C., Kalogridis, G., Sooriyabandara, M., Zhu, Z., Lambotharan, S., Chin, W.H.: Smart grid communications: Overview of research challenges, solutions, and standardization activities. IEEE Communications Surveys and Tutorials **15**(1), 21–38 (2013)
12. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid - The new and improved power grid: A survey. IEEE Communications Surveys and Tutorials **14**(4), 944–980 (2012)
13. Grubitzsch, P., Schuster, D.: Hosting and discovery of distributed mobile services in an XMPP cloud. In: 2014 IEEE International Conference on Mobile Services, pp. 47–54 (2014)
14. Hussain, S., Gustavsson, R., Saleem, A., Nordstrom, L.: SLA conceptual framework for coordinating and monitoring information flow in Smart Grid. In: 2014 IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2014, pp. 1–5, February 2014
15. Kovatsch, M., Mayer, S., Ostermaier, B.: Moving application logic from the firmware to the cloud: towards the thin server architecture for the internet of things. In: Proceedings - 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012, pp. 751–756 (2012)
16. Markovic, D.S., Zivkovic, D., Branovic, I., Popovic, R., Cvetkovic, D.: Smart power grid and cloud computing. Renewable and Sustainable Energy Reviews **24**, 566–577 (2013)
17. Rusitschka, S., Eger, K., Gerdes, C.: Smart grid data cloud: a model for utilizing cloud computing in the smart grid domain. In: 2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 483–488 (2010)
18. Schuster, D., Grubitzsch, P., Renzel, D., Koren, I., Klauck, R., Kirsche, M.: Global-scale federated access to smart objects using XMPP. In: 2014 IEEE International Conference on Internet of Things(iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) (iThings), pp. 185–192 (2014)
19. Toosi, A.N., Calheiros, R.N., Buyya, R.: Interconnected Cloud Computing Environments. ACM Computing Surveys **47**(1), 1–47 (2014)
20. Yang, Y., Wei, Z., Jia, D., Cong, Y., Shan, R.: A cloud architecture based on smart home. In: 2010 Second International Workshop on Education Technology and Computer Science (ETCS), vol. 2, no. 60970130, pp. 440–443 (2010)
21. Ye, X., Huang, J.: A framework for cloud-based smart home. In: Proceedings of 2011 International Conference on Computer Science and Network Technology, ICCSNT 2011, vol. 2, pp. 894–897 (2011)