14-th IEEE International Conference on Peer-to-Peer Computing

A peer-to-peer service architecture for the Smart Grid

Filipe Campos, Miguel Matos, José Pereira
High-Assurance Software Laboratory,
INESC TEC & University of Minho,
Braga, Portugal
Email: {fcampos, miguelmatos, jop}@di.uminho.pt

David Rua
Power Systems Unit
INESC TEC & University of Porto,
Porto, Portugal
Email: drua@inescporto.pt

Abstract—**Important challenges in interoperability, reliability, and scalability need to be addressed before the Smart Grid vision can be fulfilled. The sheer scale of the electric grid and the criticality of the communication among its subsystems for proper management, demands a scalable and reliable communication framework able to work in an heterogeneous and dynamic environment. Moreover, the need to provide full interoperability between diverse current and future energy and non-energy systems, along with seamless discovery and configuration of a large variety of networked devices, ranging from the resource constrained sensing devices to servers in data centers, requires an implementation-agnostic Service Oriented Architecture. In this position paper we propose that this challenge can be addressed with a generic framework that reconciles the reliability and scalability of Peer-to-Peer systems, with the industrial standard interoperability of Web Services. We illustrate the flexibility of the proposed framework by showing how it can be used in two specific scenarios.**

I. Introduction

Smart Grids (SG) have been introducing a paradigm change in electric power system with the objective of enhancing the integration of renewables and promote the generalized participation of different entities. Unprecedented research initiatives have been established with the purpose of addressing the architectural and technological aspects of power, information and communications systems [1], [2]. The SG concept includes different visions and strategies that allow the modernization of the electric industry in order to ensure high levels of adaptability, scalability, security, economy, self-healing, robustness and protection in highly dynamic systems [3].

SG embody the future of the power grid because of the associated benefits, such as the reduction of carbon emissions and fuel costs, transmission losses, increased reliability to power failures, and deferral of investments, among others. Distributed Energy Resources are becoming widespread in power grids, in different segments of the power system, and require monitoring and control schemes to allow their enhanced participation in both market and system services.

The role of Information and Communication Technologies (ICT) in SG is gaining importance since it represents the underlying support infrastructure that allows the necessary information exchange towards the integration of different participants while supporting a diversified set of applications and services. As such, a complex interconnection between different segments, domains, and players requires a suitable ICT. To achieve the complete integration of the various systems that compose the SG, a general ICT solution will need, among others, to: achieve the necessary interoperability between largely disparate devices; be scalable, in order to cope with the continuously increasing number of devices on the grid; and be highly reliable to support the operational requirements introduced by the SG.

Peer-to-peer communication and coordination protocols based on gossiping have been proposed to address scalability and reliability issues in SG, namely for secondary and tertiary control on a microgrid [4], or for disseminating and aggregating important information in the Automated Metering Infrastructure [5]. However, reconciling such protocols with existing systems and paving the way for their long term maintenance and evolution is an open problem. On the other hand, Web Services in general, and the Devices Profile for Web Services (DPWS) [6] in particular, can have an important role in SG [7], as it provides several of the required features for an Energy SOA, by supporting dynamic, adaptive and auto-configurable architectures, and by embracing the heterogeneity on this environment, thus achieving full interoperability with energy systems based on the main standards and models [8] as well as with other systems. However, the DPWS stack has only very limited support for peer-to-peer communication, assuming central coordination components.

We thus propose to use a framework that provides gossip based dissemination and coordination built on top of Web Services [9], more precisely on DPWS, within the SG context. This framework allows taking full advantage of existing standards, including current devices, while paving the way for evolving to a decentralized peer-to-peer service architecture that can be tuned according to each scenario's requirements.

The rest of this paper is organized as follows: Section II provides background on gossip protocols and Web Services standards. Section III describes the components of the proposed approach and some application scenarios. Section IV presents related work and Section V concludes the paper and provides directions for future work.

II. Background

A. Gossip

In peer-to-peer computer networking, gossiping describes the process where a participant that intends to disseminate

978-1-4799-6201-3/14/$31.00 ©2014 IEEE

some information chooses a small random subset of other participants and forwards the information to them. Each of these destinations, upon receiving the information, repeats the same procedure, hence, the gossip moniker. This also mimics how epidemics spread in populations, justifying the alternative denomination of epidemic protocols [10]. Despite simple, gossip protocols are highly reliable, scalable and adjustable to a wide range of performance tradeoffs. A key issue in gossiping is the maintenance of peer lists [11]. Some of the basic properties of gossip protocols are briefly described next.

1) Reliability and Scale: Reliability is proactively achieved by the redundancy and randomization of gossip protocols, coping with both process and network link failures. The expected probability for a message being delivered to all destinations can be derived directly from protocol parameters f, the number of targets that are locally selected by each process for gossiping, and r, maximum number of times a message is relayed before being ignored. By adjusting r and f parameters according to the expected system size and fault patterns, gossip can be configured such that messages are received with an arbitrary large probability. The key to scalability is that the value of f is logarithmically proportional to system size. Moreover, the load is evenly spread among everyone because all participants are involved in the dissemination process.

2) Performance: There are two main variants of message exchange patterns in gossip protocols [12], which provide different performance trade-offs. In *push gossip*, a node that becomes aware of new information, conveys it immediately to target nodes, which is adequate for one-to-many dissemination of small messages and events. With *pull gossip*, a node periodically selects a number of peers and asks them for new information. Combining *push* and *pull gossip* results in dissemination being achieved in a lower number of steps [12] and provides a generic framework for gossiping that can be tailored for multiple purposes by parameterizing it with different aggregation functions. In addition, lazily deferring the transmission of payload improves performance in heterogeneous networks, allowing gossip protocols to approximate ideal resource usage efficiency [13].

B. Web Services

WS-Eventing defines the usage of the publish/subscribe pattern by Web Services, and it embodies a flexible filtering mechanism, favoring lightweight implementations and one-to-many communication. It has therefore been the preferred choice for connected devices, namely, within standards like WS-Management and Devices Profile for Web Services (DPWS) [6]. WS-Eventing can be combined with other standards, such as WS-ReliableMessaging, for end-to-end acknowledged message delivery, or WS-AtomicTransaction (WS-AT), for multi-party transactional atomicity guarantees.

DPWS defines a set of protocols that resource constrained devices should implement in order to achieve seamless networking and interoperability through Web Services. It assumes that each device behaves as a standard *hosting service*, providing basal functionality, and exposing one or more *hosted services* that offer device specific functionality. Besides basic SOAP, WSDL, the HTTP binding, WS-Addressing, and WS-Security, that are at the core of Web Services capabilities and interoperability, DPWS also includes WS-Eventing, as previously mentioned, SOAP-over-UDP, enabling UDP as a transport for SOAP messages and network level multicast, which paves the way for dynamic discovery and description, enabled by combining WS-Discovery, WS-MetadataExchange, and WS-Policy.

Although DPWS provides an adequate infrastructure for small scale systems, it is becoming increasingly interesting when managing a large number of components, albeit it has some scale limitations. First, the use of WS-Eventing imposes a burden on the publisher, that has to notify all subscribers. Moreover, when a resource exposed by many devices has to be updated, e.g. to change a configuration variable, the initiator device must contact every destination individually. Finally, as there is no support for transactional coordination mechanisms, such lengthy operations involving large numbers of destinations are susceptible to faults and cannot be restarted or recovered if stopped. This is particularly worrisome as such notifications and configuration updates may correspond to critical alerts and urgent commands. It does not make sense to resort to heavyweight coordination protocols such as WS-Coordination and WS-AT in such a scenario, because, even if devices could support their requirements, they would not scale very well. Thus, a scalable lightweight coordination protocol, that fits the general DPWS assumptions, is necessary.

III. Proposal

For the proposed framework, we consider a simplified architecture of a Smart Grid (SG) focusing on the communications infrastructure depicted in Fig. 1. Briefly, the Information Systems (IS) of the utility are the main point for controlling and monitoring the entire grid, by retrieving data and issuing commands to other devices in the grid, such as Secondary Substation Controllers (SSC), normally connected to a Wide Area Network (WAN). SSC are installed in electric distribution transformers, and are equipped with sensors and actuators for monitoring the grid's conditions while enabling remote control. As previously mentioned, SSC interact with the IS, normally to report metrics or anomalies on the grid, and with Smart Meters (SM), connected to the same Field Area Network (FAN), to notify them on tariff changes or service perturbations. A Smart Meter interfaces with the customer, as well as with their appliances or Intelligent Electronic Devices (IED) through the Home Area Network (HAN), to convey relevant information such as metering and maintenance warnings. Different types of data and scenarios inside a SG have different requirements, namely in terms of maximum allowed communications latency. Protective relaying, status monitoring, and substation SCADA communications endure latency values as high as a few milliseconds to seconds or even minutes, but the loss of messages of these types is not tolerated due to their criticality to the SG operation [14], [15]. Gossip protocols can be of particular importance in such settings, which are stricter in terms of message delivery assurance compared to message latency, as the message delivery assurance of these protocols largely outweighs the overhead of the additional traffic.

Our proposal to address the scalability and reliability challenges raised by the heterogeneity of the components of the SG, and its complex nature, is to use a Web Services framework for gossip-based dissemination [9]. The inherent

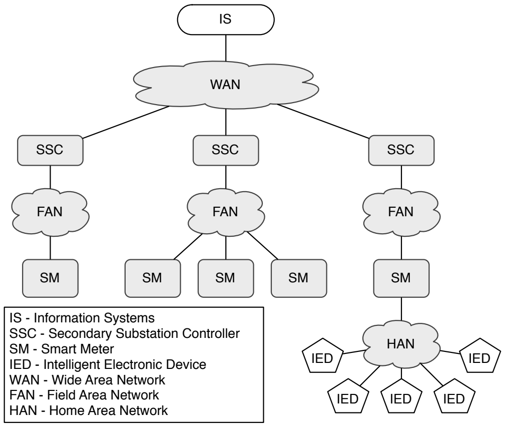

Fig. 1. Overview of a simplified Smart Grid architecture.

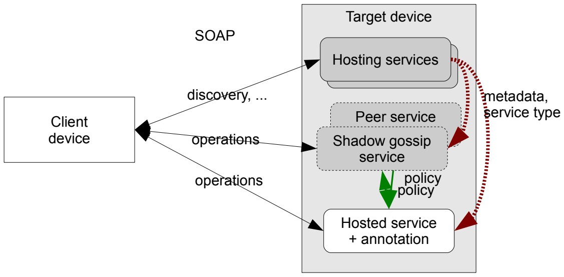

Fig. 2. Overview of the architecture of the framework.

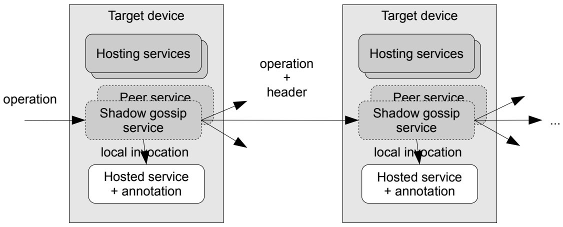

Fig. 3. Initialization of Gossip dissemination.

scalability and reliability of gossip protocols allows the usage of SOAP-over-UDP even if reliable delivery is desired, since it is much less resource consuming than a full-fledged HTTP binding over TCP. Moreover, by assuming the Web Services infrastructure based on the Devices Profile for Web Services (DPWS), we take advantage of each gossiped unit of data being a SOAP envelope, of the self-documenting nature of services through WSDL, and of useful base protocols and standards such as WS-Discovery and WS-Policy. Hence, the proposed framework builds upon DPWS to promote interoperability among largely heterogeneous devices, from top of the range mainframes to small IED running completely different operating systems, and it is composed by two services, gossip and peer, that complement each other. The Gossip Service relies on gossip for disseminating messages, whereas the Peer Service provides information on the services and devices that are currently on the network. This information can then be used to build and enforce logical overlays on top of the SG's components, in order to guarantee communication among all of them. Gossip Service instances rely on this service to obtain the list of targets for disseminating messages.

The usage of the proposed framework is illustrated in two specific scenarios: propagation of information and retrieval of distributed metrics. On the first scenario, assuming a dynamic tariff, where energy overproduction can lead to significant reduction of prices, these variations must be advertised to all the clients in order to adapt energy consumption accordingly thus stabilizing the network by better matching the demand to the supply of energy. On the second scenario, to better plan future power production, each consumer's SM can announce the energy requirements of the connected IED for a specific time frame, and this information will then be aggregated from level to level until reaching the utility's IS. The first scenario focuses on the scalable dissemination capabilities of the framework, whereas the second one demonstrates its data aggregation capabilities.

A. Framework Overview

The Gossip Service builds on propagating messages that can include simple data or more complex information that was aggregated throughout a network, being able to compose different useful patterns. The general architecture of the proposed service is outlined in Fig. 2 and works as follows. To provide gossip dissemination in its devices, a manufacturer can use a DPWS stack with gossiping support and annotate every service supporting gossip using WS-Policy assertions. As a consequence, a shadow gossip service is created for each service where gossip is enabled. Moreover a Peer Service can be setup to provide an entry point to the set of target peers.[1] Both the original hosted service and its shadow gossip service are advertised to clients that can use each of them independently. A gossip-aware client can examine policy annotations in both these services and determine their relationship. A client may still address the original hosted service, thus maintaining compatibility with legacy clients.

Assume for now a one-way notification operation (*i.e.*, input or output only) and push-style gossip [12]. Gossiping is started when a client sends a SOAP message to a port in the shadow gossip service, which, is then inspected to determine if it contains a gossip header. If not, default gossiping parameters are obtained, including gossip variant, fanout f, rounds r, peer scope or type (according to WS-Discovery), and target binding (HTTP or UDP). Gossiping is then initiated by adding these parameters to the message header and relaying it to a number of peers and to the local hosted service. Upon reception of the gossip message, the dissemination proceeds by each recipient decrementing the message's r counter and forwarding it to its selected peers. Note that a gossip message can be generated by a target device, as depicted in Fig. 3, or directly by a gossiping-aware client, allowing it to set the gossiping parameters to achieve customized reliability and scalability trade-offs.

The framework also supports output-only operations (*i.e.*, notification), and call-back operations (*i.e.*, solicit-response), besides the more typical client-server interaction (*i.e.*, request-response). These operation styles are combined in different gossip variants, such as lazy pull, in addition to the previously described eager push-style. In request-reply and solicit-response operations, the message is propagated and then all the received replies are propagated back to the initiator.

[1] Further detail on the Peer Service can be found in [9].

Consider the following example: A request-response to query the last measured voltage value throughout a segment of a SG. The client invokes the operation on the shadow service, which is forwarded to its known peers and eventually reaches all targets, *i.e.*, all the IED in that SG's segment. Along the way, each of these peers will decrement the message's r value and forward it to their own set of peers, previously obtained from the Peer Service, or retrieve this information in the event that the device still does not possess such information or if it is obsolete according to the configuration parameters. These operations are repeated by recipients, normally until the message's r value reaches zero. Each response travels back along the path implicitly created by the corresponding request message, eventually reaching the initiator.

An alternative is to use a filter, which can omit or aggregate replies according to some rule specified when gossip is initiated, for example to determine the maximum voltage phase angle difference in a given segment of a SG. Assuming an IED on the grid, it can start this process by gossiping a message containing an XSLT definition of the aggregation function to be applied by each node to combine its own data with the aggregated one in the received message. Assuming a request-response aggregation invocation, after reaching all targets, responses will then travel back along the same tree implicitly created by request messages, but they are buffered and filtered using the conveyed aggregation function, such that only one aggregated value is returned by each peer. The response messages are sent by each peer as soon as a configured minimum of targets have replied, conveying a value, or a fault, for instance, when the timeout expires or in the occurrence of other errors.

B. Case Study

The proposed framework can be used to replace the existing mechanisms of alert and event propagation in the Automated Metering Infrastructure (AMI) for scenarios with a high rate of messages and a large number of targets. In this scenario, publishers might be overwhelmed with subscription storage and maintenance, which could lead to the loss of critical messages. The usage of dynamic price tariff schemes through AMI systems allows utilities to take advantage of operational scenarios to shape the participation of customers, by setting more, or less, attractive tariffs to them, while guaranteeing a stable and normal operation of the power system [16]. The AMI comprises two-way communication between the utility's systems and SM, allowing the conveyance of information in both directions. These tariff modifications will then flow from the utility's IS to all the customers through their own SM or even through some other IED. We will focus on how these communications can occur using our framework in such a scenario.

When a utility decides to set lower price tariff through its IS, this information is then encapsulated in a push gossip message which is disseminated to the target SSC retrieved from the Peer Service deployed at the IS node. The Gossip Service instance of the targets, upon reception of the message, decrements the value of rounds r and retransmits the message to the target nodes that its Peer Service instance proposes, which could be other SSC, reachable through the WAN, or SM, reachable through the FAN to which the sending SSC is connected. When a SM receives the message, the Gossip Service instance behaves in a similar fashion to the one in the SSC, *i.e.*, it retransmits it to targets provided by the Peer Service instance. This instance can be located at the Smart Meter or at any other reachable node. The targets can vary from other SM, connected to the same FAN, to IED connected to the same HAN, that can range from controllable appliances to Renewable Energy Sources (RES). IED can then adapt their operating mode according to the received information of the tariff scheme. For instance, by analyzing the price reduction and the corresponding period, HVAC can increase its consumption to better suit the consumer's temperature preferences, while dishwashers and washing machines can anticipate their washing cycles, among other possibilities. In parallel with the retransmission of the message to the designated targets, the Gossip Service instance of the SM can present the notification on tariffs reduction in a local display or forward it to some other device, as configured by the customer, like a smartphone or a tablet.

In the second scenario, the proposed framework is used to collect metrics from different points of the SG in order to plan power production according to the announced energy requirements. For instance, charging of electric vehicles, and the usage of high power consumption appliances, such as dishwashers, are configured to occur during nighttime, when tariffs are usually lower. Periodically, the central IS invokes the Pull Aggregation operation on SSC, which, in their turn, invoke the same operation on the target SM designated by their Peer Service. A SM, upon reception of such a request, propagates the same request to the IED pointed by the Peer Service in the HAN. Each of these IED, if configured to perform some scheduled task, will respond with the energy requirements to execute these tasks, their duration, and the time by which they should be finished. The SM will then aggregate this information for the entire household, after waiting for responses from IED until a certain number of responses arrives or a certain timeout elapses, according to configured preferences. The aggregated information will combine all the power requirements pointed by the IED for the three 8 hour time periods which divide the day. For simplification purposes, we will consider that all the energy needs for each of these periods will be simply added in order to produce the aggregate information at the SM. Each SSC will then receive the aggregate responses from the previously contacted SM, and again, having waited according to configured preferences, will aggregate those responses in a single message sent back to the IS. The IS will then process this message and assess what are the announced energy requirements and plan the energy generation according to the demand for the next periods.

IV. Related Work

WS-SCADA [17] addresses integration needs of clients, applications, utilities and market participants, by accommodating all their information needs and adapting to dynamic changes at both system and business level. The proposed open, flexible and scalable infrastructure includes two Web Services protocol stacks, for a control center and a substation, which share some protocols with DPWS, such as WS-Discovery, allowing substations to locate Intelligent Electronic Devices (IED) and their services, or WS-Eventing, that allows substations to receive notifications from IED, and control centers to notify substations on control messages or to be notified on status information and real-time operation data of substations. It has also been shown that DPWS is suitable for smart meters communication [18], but for a large amount of devices, in the region of some thousands, an hierarchically structured communication does not cope well with the generated traffic [19].

Gossip protocols can be used in SG to disseminate important information, for instance, load shedding notifications, or metering data aggregation [5]. However, limiting epidemic dissemination to a single pairwise interaction per node in each cycle leads to large dissemination times. The more traditional approach, where each gossiping node contacts various of its neighbors in parallel would prove to be useful in the majority of the SG scenarios. Secondary and tertiary controls in microgrids can be implemented with gossip protocols with the aim of improving power quality and optimizing generation costs, respectively [4]. A gossip aggregation protocol can be used to calculate the average of voltage and frequency deviations measured at Distributed Energy Resources (DER) units, which can then be added to the reference active and reactive power in order to stabilize the network, by decreasing the deviations. To optimize distributed generation costs, each DER unit periodically contacts a random neighbor in order to harmonize their marginal cost functions. Such scenarios could benefit from more advanced aggregation strategies, in order to decrease the number of communication interactions between the DER units.

V. Conclusion

The implementation of Smart Grids is highly supported by Information and Communication Technologies, integrating different computing and networking elements which are present in the multitude of systems composing current and future power grids.

In this paper we show how this challenge can be addressed by instantiating a flexible peer-to-peer service framework [9]. In short, it builds first on gossip based communication variants providing probabilistic message delivery guarantees as well as proactive reliability. Second, we leverage the Devices Profile for Web Services, which allows communications based on Web Services between resource constrained devices and mainframes, automatic detection of the devices present on the network and easy integration of new devices.

The main contribution of our work is showing how simple peer-to-peer primitives, for gossiping and membership management, when properly integrated in a service framework, are a powerful foundation for different communication and coordination applications. This flexibility is key in infrastructures such as Smart Grids, whose current deployments are expected to last for a long period of time and to evolve as new technologies are integrated and new requirements are addressed.

Acknowledgment

This work was made in the framework of the BEST CASE project (NORTE-07-0124-FEDER-000056) financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Foundation for Science and Technology (FCT).

References

[1] National Institute of Standards and Technology, “NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0,” 2010, http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf.

[2] “IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads,” *IEEE Std 2030-2011*, pp. 1–126, Sept 2011.

[3] A. Ipakchi and F. Albuyeh, “Grid of the future,” *Power and Energy Magazine, IEEE*, vol. 7, no. 2, pp. 52–62, 2009.

[4] K. De Brabandere, K. Vanthournout, J. Driesen, G. Deconinck, and R. Belmans, “Control of microgrids,” in *Power Engineering Society General Meeting, 2007. IEEE.* IEEE, 2007, pp. 1–7.

[5] A. Krkoleva, V. Borozan, A. Dimeas, and N. Hatziargyriou, “Requirements for implementing gossip based schemes for information dissemination in future power systems,” in *Innovative Smart Grid Technologies (ISGT Europe), 2nd IEEE PES Int. Conf. and Exhibition on*. IEEE, 2011, pp. 1–7.

[6] “Devices Profile for Web Services (DPWS) 1.1 OASIS Standard,” http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.html, pp. 1–43, 2009.

[7] S. Karnouskos, “Asset monitoring in the service-oriented internet of things empowered smartgrid,” *Service Oriented Computing and Applications*, vol. 6, no. 3, pp. 207–214, 2012.

[8] J. Schmutzler, S. Groning, and C. Wietfeld, “Management of distributed energy resources in IEC 61850 using web services on devices,” in *Smart Grid Communications (SmartGridComm), 2011 IEEE Int. Conf. on*. IEEE, 2011, pp. 315–320.

[9] F. Campos and J. Pereira, “Improving the Scalability of DPWS-Based Networked Infrastructures,” Tech. Rep., Jul. 2014. [Online]. Available: http://arxiv.org/abs/1407.8546

[10] P. Eugster, R. Guerraoui, A. Kermarrec, and L. Massoulie, “Epidemic information dissemination in distributed systems,” *Computer*, vol. 37, no. 5, pp. 60 – 67, 2004.

[11] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, “The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations,” in *Middleware ’04: Proc. 5th ACM/IFIP/USENIX Int. Conf. on Middleware*, 2004, pp. 79–98.

[12] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, “Randomized rumor spreading,” *Foundations of Computer Science. Proc. 41st Annual Symp. on*, pp. 565–574, 2000.

[13] J. Pereira, R. Oliveira, and L. Rodrigues, “Efficient Epidemic Multicast in Heterogeneous Networks,” in *On the Move to Meaningful Internet Systems: OTM 2006 Workshops*, vol. 4278/2006, 2006, pp. 1520–1529.

[14] D. Li, Z. Aung, J. R. Williams, and A. Sanchez, “Efficient authentication scheme for data aggregation in smart grid with fault tolerance and fault diagnosis,” in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*. IEEE, 2012, pp. 1–8.

[15] D. Rua, L. M. Pereira, N. Gil, and J. P. Lopes, “Impact of multi-Microgrid Communication systems in islanded operation,” in *Innovative Smart Grid Technologies (ISGT Europe), 2nd IEEE PES Int. Conf. and Exhibition on*. IEEE, 2011, pp. 1–6.

[16] M. Morgan, J. Apt, L. Lave, M. Ilic, M. Sirbu, and J. Peha, “The many meanings of ”Smart Grid”,” *repository.cmu.edu*, Jan 2009.

[17] Q. Chen, H. Ghenniwa, and W. Shen, “Web-services infrastructure for information integration in power systems,” in *Power Engineering Society General Meeting*, 2006.

[18] S. Karnouskos and A. Izmaylova, “Simulation of web service enabled smart meters in an event-based infrastructure,” *Industrial Informatics (INDIN) 2009. 7th IEEE Int. Conf. on*, pp. 125 – 130, 2009.

[19] S. Karnouskos, P. G. da Silva, and D. Ilic, “Assessment of high-performance smart metering for the web service enabled smart grid era,” *Proc. 2nd joint WOSP/SIPEW Int. Conf. on Performance engineering*, pp. 133–144, 2011.