2013 IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)

An Intragrid implementation embedded in an Internet of Things platform

Elisa Spanò[1,2], Stefano Di Pascoli[1,2], Giuseppe Iannaccone[1,2]
[1]Dipartimento di Ingegneria dell'Informazione, University of Pisa,
Via Girolamo Caruso 16, 56122 Pisa, Italy
[2]SEED Center – PUSL, University of Pisa,
Via dei Pensieri 60, 57128, Livorno, Italy
Email: elisa.spano@iet.unipi.it; stefano.dipascoli@iet.unipi.it; giuseppe.iannaccone@iet.unipi.it

***Abstract*— We present the design and implementation of an Intragrid – the portion of the smart grid on the premises of a single or a small group of customers – built on an Internet of Things platform. Such implementation minimizes the need for additional infrastructure, enables integration with smart home applications, ensure secure and differentiated access to data. In this paper we describe the architecture of the Internet of Things platform and the specific intragrid implementation.**

Keywords—smart grid, internet of things, power meters, telemetering, demand side management

I. INTRODUCTION

The *Customer Domain* defined by the NIST conceptual model of Smart GRIDs [1], is the one most visible to the customer. It enables demand-side energy management [2], which is one of the most disrupting aspects of smart grids.

The customer domain aims at enabling home, commercial, and industrial customers – on the basis of their different energy needs – to optimize energy consumption and production, and to actively participate to demand-response policies. In order to do so, non-technical customers need a simple way to control energy consumption and production, and to exchange power usage data at the proper level of granularity with energy providers or distributors.

Within the E2SG (Energy to Smart Grid) ENIAC project [3], the portion of the smart grid on the premises of the customer or of a small community of customers is defined as *Intragrid*. Conversely, the rest of the smart grid is referred to as the *Intergrid*. By definition, the E2SG Intragrid is mostly concerned with the customer domain.

This paper discusses an Intragrid implementation stemming from few considerations and assumptions on possible attitudes of customers towards a smart grid deployment *in their home.*

First, we believe that the Intragrid should exploit existing infrastructure for in-home connection to smart meters. Therefore, the Intragrid architecture should allow different wireless or wired protocols to be used for communications between meters, users, and other parts of the system.

We also can assume that the typical customer, as early adopter of an Intragrid system, is also a user of smart home applications (dedicated to security, entertainment, home automation, et cetera). Therefore it would be desirable – to avoid duplication and to enable possible synergy – to use a single platform for both smart grid and smart home applications.

Finally, it is important for the Intragrid to provide differentiated and secure access to data. For example, fine-grained data should only be available to the customer, whereas distributors and energy providers should be able to receive coarse-grained and aggregated statistical data.

For these reasons, in this paper we present an Intragrid implementation *embedded* in an Internet of Things platform, that: *i)* seamlessly integrates smart grid with smart home applications *ii)* can gather data from heterogeneous sensor communication protocols *iii)* ensures secure and customized access to data *iv*) allows to univocally map each sensor and actuator to a common abstraction layer for data processing and system behavior control.

The rest of the paper is organized as follows. The related work is presented in Section II. Section III describes the architecture of the proposed platform and its main components. A hardware and software implementation of an intragrid, based on the proposed platform, is described in Section IV. Finally, in Section V, we comment on the use of an Internet of things platform in the context of smart grids.

II. RELATED WORK

In recent years the research community has been extremely active in the neighboring fields of distributed sensor networks, home automation and smart grids. We can loosely classify the large number of related papers in two groups:

i. A set of papers focusing on large-scale systems, in which a home network of power meters, connected to individual loads, is a subsystem in an integrated infrastructure aimed to manage the complete power grid, including power generation plants, transmission and distribution systems, and "smart" consumers, with local generation capabilities, flexible usage and sometimes local energy storage capacity. This large infrastructure is usually headed by a central server/data storage or SCADA system [4], [5], [6].

   The proposed complex systems are typically described only at the architectural level, with an extensive discussion of the goals and objectives but with few details of the implementation. They also require substantial investments in infrastructure, especially for data transmission from the customer site to the last node of power distribution ("last mile"). As [7] points out, the most critical component of the communication infrastructure is this last mile link. In [8] many transport options are proposed, from the use of dedicated lines, to POTS/modem, PLC (power line communication) to wireless links.

   Most of these projects are integrated or include a "smart meter" used for both for data collection and billing [6], and can be deployed only by the power distributor on in strict coordination with it. Some vendors of power distribution equipment propose this approach [9]. A

pilot project deployed by a power distributor [10] required an investment of 10 M€ for the territory covered by a single primary distribution transformer (about 30 MVA). It is worth noting that deployed smart metering networks are usually based on PLC links [11].

ii. Another group of works present a home system for power metering and analysis, and is therefore close to the present paper. While [12] provides an architectural description, [13], [14], [15], provide implementation details. Most of the proposed implementations connect the home sensor or automation network to the wide area network or a central server by means of a complex gateway, with large computational power (several MB of RAM and FLASH and a complete operating system); a similar approach has been followed also for a pilot project [16]. The installation and configuration of this device makes the deployment of the system out of the reach of many end users. Many communication technologies are used for the local transport of information inside the home network: ZigBee and 6LoWPAN are popular [15], [17] but also dedicated point-to-point radio links are proposed in [13]. A recent IETF document [18] discusses in detail many aspects of the use of IP protocols for communication between the components of a smart grid system.

III. PLATFORM FOR THE INTERNET OF THING

We have developed a platform for the Internet of Things that can seamlessly support an in-home Smart Grid, and can leverage other services and functionalities available on the same platform. Its architecture is that of a scalable distributed system on which different concurrent applications for remote monitoring and control can run. Fig. 1 illustrates the platform architecture in terms of block diagram:

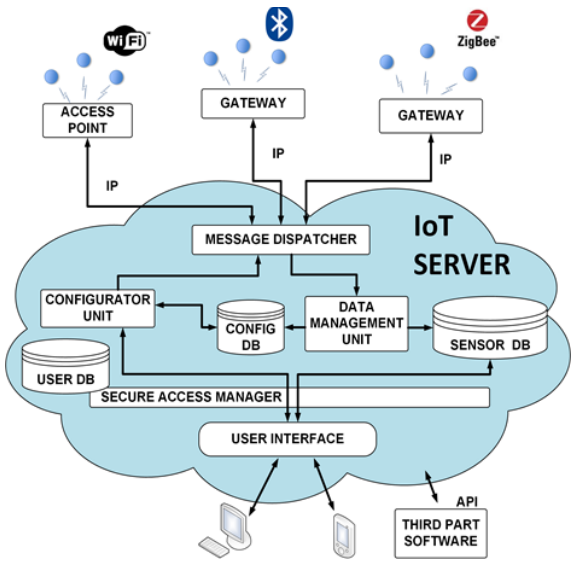

Fig. 1. Block diagram of the Internet of Things platform supporting the in-home Smart Grid.

The Internet of Things platform consists of several modules, each described by its functions and its interfaces with others modules. In this way the architecture is simple to understand, robust, and easily scalable. Any module can be modified, redesigned, and extended with almost no impact on the rest of the system.

The main seven modules are:

- sensor and actuator nodes;
- gateways;
- message dispatcher;
- data management unit and sensor database;
- configurator unit and database;
- secure access manager;
- user interfaces.

We refer to the set of all modules except nodes and gateways as *IoT server*. Modules are described in more detail below:

A. Sensor and actuator nodes

The sensor and actuator nodes can be part of networks implemented with wired (CAN, power line, etc.) or wireless (ZigBee, Wi-Fi, Bluetooth) network protocols. The architecture is designed to accommodate different and heterogeneous sensor and actuator networks. The data management unit (*Sec. IIID*) is responsible for translating information to the format required by the sensor database.

On the other hand, bidirectional links to the nodes enable the IoT server to interrogate, configure, and program them. Configuration messages mainly carry node-specific information (measurement thresholds, alarm settings, etc.) or a new version of the program running on the node.

For global devices accessibility, each node has to be uniquely identified. This can be a potential problem because node addresses in sensor networks may change over time and are often unique only within a single network. To overcome this problem without imposing address constraints on network nodes, our solution gives to the gateway the task of assigning to each node of the network a unique virtual address and to maintain the mapping between such address and the real one.

Even if specific node characteristics depend on the network implementation, the proposed architecture supports the possibility to add or remove any network component in real time. Any node can join the system requiring no modification to network implementation. For this reason a new node that joins a network connected to the platform, it is automatically identified and immediately accessible from the network administration interface for registration and configuration. Similarly, updating or un-joining nodes are automatically referred to the IoT server. The interface between sensor networks and platform, from a functional point of view, consists of the communication protocol between the gateway and the IoT server, which is defined by API specifications.

B. IP Gateway

The gateway is the element connecting a sensor/actuator network (without direct IP capability) to the IoT server via an IP link. The gateway is bidirectional: for uplink communication it collects data received from the network nodes, performs reformatting/encapsulation if required, and sends them over a secure TCP/IP link to the message dispatcher. In downlink, on the other hand, it forwards to the receiver node(s) any commands received from the IoT server.

We propose a different gateway concept with respect to the one commonly used to integrate heterogeneous networks with an external network [19], [20], [21]. These systems use a gateway-based approach [22], where the gateway performs a protocol conversion of data into a *universal* format. In our architecture, such operation is performed by the server, where data will be stored. Therefore, the gateway sends network packets over TCP/IP in the *native* format and both the gateway and the message dispatcher are transparent at the logical communication level between nodes and IoT server.

The choice of giving to the IoT server the possibility to receive packets in the original format has three main advantages: *i)* it avoids the possible loss of information due to gateway limitations, *ii)* it allows the development of different applications and of new functionalities without modifying the gateway, and *iii)* it allows the platform user to communicate at the application level directly with network nodes.

From the gateway development point of view, this choice has the advantage of reducing the hardware requirements and the computational complexity. Our gateway has only to ensure an IP connection, to implement the encapsulation of the nodes' native protocol into TCP/IP packets, and to ensure the security level required by the specific application. As a validation of this concept, the gateway is currently implemented with Cortex-M3/M4 microprocessors.

C. Messages dispatcher

The message dispatcher manages the bidirectional communication between each gateway and the rest of the system. It only deals with low-level communications from nodes (through the gateways) to the data management unit, and from the configurator unit to the nodes.

It has the main task of listening to new connections from IP nodes that want to join the system. For every connection, it decrypts all incoming packets and forwards them to the data management unit, for interpretation and storage. In the other direction (downlink), it encapsulates all messages from the configurator unit into an IP packet, encrypts and forwards them to the destination gateway.

The communication between the gateway and the message dispatcher follows the TCP/IP client-server model. Messages exchanged between sensor networks and IoT server are structured as TCP/IP packets whose payload is the sensor data in the native format.

D. Data management unit and database storage

The Data management unit is a collection of software components, each able to manage the messages of a specific sensor network type. The modules receive node packets in their native format and extract their payload. If the message contains measurement data from a sensor or an event notification by an actuator, it will be stored with a network-type independent data format in a streaming *sensor database*. Specific network messages (configurations, management information, communication channel, node address, etc.) are stored in the original format into the *config database*, containing configurations and information on networks and nodes. The proposed solution does not impose limitations on the data volume a node can send to the system. The distributed nature of the database makes the system easily scalable.

E. Configurator unit

The configurator unit configures networks and nodes according to inputs from users and authorized applications and according to the status of the system stored in the configuration database. Also the configurator unit is a collection of modules, each dedicated to a specific type of sensor/actuator network. This unit must elaborate instructions to nodes and create the correct sequence of messages required to perform the node/network configuration.

F. Secure access manager

The secure access manager provides access to stored information and networks configuration only to authorized users or third part applications, on the basis of a database of users, networks, sensors, keys and permits. By default, networks owners have administrator rights on their networks. They have access to all information, can manage access rights to nodes and data and configure alarms and event triggers.

G. User interface

Through the User Interface module users control and configure nodes, and visualize and use collected data. In order to maximize interoperability, the user interface is implemented as a web application. It is formed of two main parts: a configuration/administration interface and visualization interface.

The visualization interface displays relevant information from sensors. In the sensor database, sensor data are all represented by means of a unique abstracted format, and are univocally associated to physical sensors at a lower level of abstraction. In this way, data can then be easily accessed by performing a simple query to the database, processed and visualized independently of the characteristics of the physical source. In addition, the visualization interface allows the user to send commands to actuators, if authorized.

The administration interface provides the user with the possibility to remotely manage and configure her networks. Administration interface pages and fields depend on the type of networks and on the corresponding protocols.

IV. INTRAGRID IMPLEMENTATION

We have implemented an Intragrid prototype on the IoT platform, building dedicated hardware and software. This first prototype only includes a ZigBee network connected to the IoT server though a ZigBee IP gateway. The sensors are smart plugs, placed between home loads (computer, washing machine, TV, etc.) and the wall socket, and able to collect real-time power consumption data from the loads. Customers can have a visual feedback of their energy consumption and can remotely control each load. Let us consider in detail the elements of the Intragrid:

A. Smart plug

As shown in Fig. 2, the smart plug is enclosed in a plastic case with a plug and a socket section and can be easily inserted in a standard wall socket.

Fig. 2. Smart Plug prototype

The smart plug collects load information from the attached electrical equipment. Information includes single-phase active, reactive, and apparent power; power factor; sampled waveforms; RMS current and voltage; and on/off status. Our smart plug has no buttons and can be completely configured and controlled through the user interface.

In the current design, the communication with the ZigBee network is provided by a Freescale MC13224 SoC, equipped with a two-way AES128 encryption engine. The board includes a 32-bit TDMI ARM7™ processor with 128KB of Flash, 96KB of RAM and 80 Kbyte of ROM memory. An Analog Devices ADE7953 is used for energy measurement. It interfaces to the microcontroller through a serial interface (SPI). The current sensing is achieved with a shunt resistor placed on the phase wire, while the voltage sensing is achieved with an attenuation network between phase and neutral (Fig. 3).

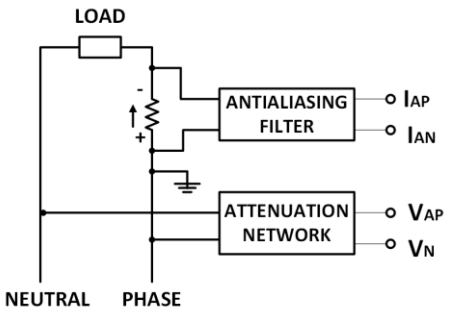

Fig. 3. Current and voltage measurement circuit

Power connection/disconnection to the load is implemented using a Single Pole bistable 12 V relay supporting loads up to 16 A. The board includes a power supply unit, which provides the operational voltages of 12 V for the relay and 3 V for the ADE7953 IC and the MC13224 SoC. The firmware running on the Smart Plug follows the ZigBee Home Automation profile.

The SPI interfacing the ADE7953 to the smart plug microcontroller has been used to calibrate various error components of the meter, including gain, offset, and phase errors. The smart plug has been calibrated using as a reference meter the tabletop power meter PCE-PA 6000 (Fig. 4).

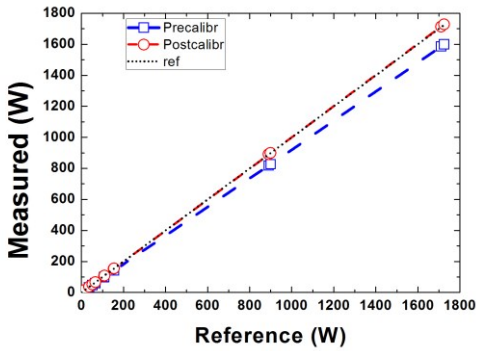

Fig. 4. pre- and post-calibration curves

Calibration coefficients are calculated based on the smart plug measurements, and transferred to the meter ADE7953 IC registers through the ZigBee radio. The ZigBee smart plug has an accuracy of 1.1% (post-calibration).

B. Gateway

As the power meter is a ZigBee device, a ZigBee/IP gateway is needed to allow communication with the IoT server. The gateway is composed by an Ethernet interface, a microcontroller, and a ZigBee RF transceiver.

In principle, due to limited requirements, several commercial ZigBee Ethernet boards could be used to implement a gateway able to connect a ZigBee network to the IoT server. Nevertheless, they are usually oversized, being designed to process all received data, or they are closed and use proprietary firmware that cannot be modified.

In the current design, the ZigBee Ethernet gateway is based on an ARM microcontroller with integrated Ethernet capabilities and hardware encryption and is equipped with a coordinator ZigBee node. Among several suitable microcontrollers, we choose the Freescale Kinetis K60 microcontroller. It includes a 32 Bit ARM Cortex M4 processor, hardware encryption and integrated Ethernet controller. The microcontroller exchanges information with the ZigBee chip, through UART communication at the maximum baud rate of 115200 (Fig. 5).

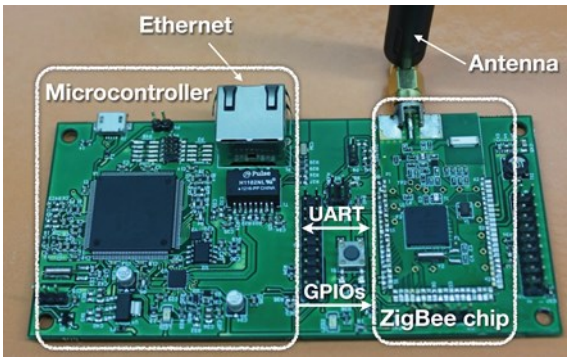

Fig. 5. ZigBee Ip gateway prototype

The gateway firmware is based on the lwIP TCP/IP stack. When connected to a LAN equipped with a DHCP server (like most of home ADSL modem/routers) it can auto-configure its network interface. Since it implements the client side of a TCP connection to the message dispatcher

(*Sec. IIIC*), no NAT configuration in the home ADSL modem router is needed, simplifying deployment and installation. All the messages exchanged between the server and the gateway can be optionally encrypted with AES.

C. Data collection and storage

The data collection unit is implemented using the CoMo platform software [23]. CoMo has been originally developed as a network-monitoring platform and has been used in large testbed deployments, such as PlanetLab [24]. It is therefore scalable to very large systems and very high aggregate data rates.

D. User interface, Configurator unit, Secure access module

Users can connect to the server through a web interface that allows both to configure the smart plug networks and to visualize energy consumption of single loads and in different aggregate ways. Fig. 6 shows an example of smart meter data visualization.

The web interface, the configurator unit, and the secure access module are implemented with Tornado [25], a web framework capable to manage a large number of simultaneous connections. The *user database* and the *config database* are implemented with MySQL. When a user wants to visualize a sensor value, the interface performs a query on the CoMo database through an Ajax request and receives the result in JSON format.

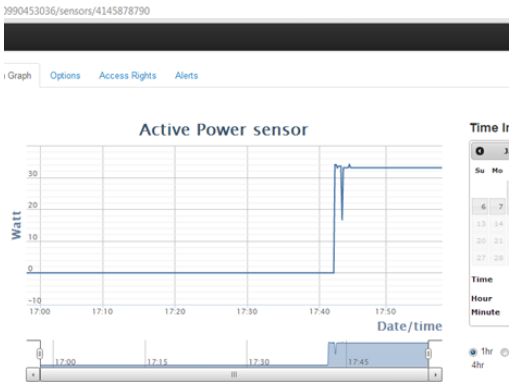

Fig. 6. Example of the visualization of data from a smart meter in the implemented IntraGRID.

V. DISCUSSION

We have successfully implemented the portion of a Smart Grid on the customer premises, – also referred to as the Intragrid [3] - on a recently developed platform for the Internet of Things that can host a broad range of smart home and wellness applications. This solution can be appealing to real customers, since it minimizes the deployment of specific smart grid infrastructure, and leverages other possibly available smart home applications, sensors, and wireless or wired connections. We believe this is key for a widespread acceptance of smart grid applications and equipment to be deployed at home.

REFERENCES

[1] NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0, NIST Special Publication January 2010

[2] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads," IEEE Transactions on Industrial Informatics, vol. 7, no. 3, pp. 381-388, Aug. 2011

[3] ENIAC Joint Undertaking E2SG project. Website www.e2sg-project.eu

[4] A. A. Khan and H. T. Mouftah, "Web services for indoor energy management in a smart grid environment," in 2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, 2011, pp. 1036-1040.

[5] J. Byun, I. Hong, B. Kang, and S. Park, "A smart energy distribution and management system for renewable energy distribution and context-aware services based on user patterns and load forecasting," IEEE Transactions on Consumer Electronics, vol. 57, no. 2, pp. 436-444, May 2011.

[6] A. Zaballos, A. Vallejo, and J. Selga, "Heterogeneous communication architecture for the smart grid," IEEE Network, vol. 25, no. 5, pp. 30-37, Sep. 2011.

[7] A. Carrasco, "How the "last mile" can make or break the smart grid", Siemens, 2011, http://www.emeter.com/smart-grid-watch/2011/how-the-last-mile-can-make-or-break-the-smart-grid/

[8] T. Sauter and M. Lobashov, "End-to-end communication architecture for smart grids," IEEE Trans. Industrial Electronics, vol. 58, no. 4, pp. 1218-1228, Apr. 2011.

[9] C. Liedtke, "Smart Grid on the Lower Rhine" Siemens AG, Infrastructures and Cities sector, http://goo.gl/A7Q2c

[10] Enel Press Release: 4 Nov. 2011, http://goo.gl/RsY8F

[11] ENEL Telemetering http://goo.gl/74YqY and Smart Homes http://goo.gl/jH9lN

[12] Y. Yang, Z. Wei, D. Jia, Y. Cong, and R. Shan, "A Cloud Architecture Based on Smart Home," 2010 Second International Workshop on Education Technology and Computer Science, no. 60970130, pp. 440-443, 2010.

[13] Q. Liu, G. Cooper, N. Linge, H. Takruri, and R. Sowden, "DEHEMS: creating a digital environment for large-scale energy management at homes," IEEE Transactions on Consumer Electronics, vol. 59, no. 1, pp. 62-69, Feb. 2013.

[14] J. Park, I. Han, J. Kwon, J. Hwang, and H. Kim, "Development of a residential gateway and a service server for home automation," Advanced Internet Services and Applications", pp. 137-149, 2002.

[15] I. Choi, J. Lee, and S.-H. Hong, "Implementation and evaluation of the apparatus for intelligent energy management to apply to the smart grid at home," in 2011 IEEE International Instrumentation and Measurement Technology Conference, 2011, pp. 1-5.

[16] Energy@home, http://goo.gl/ee8GU

[17] B. Becker, A. Kellerer, and H. Schmeck, "User interaction interface for Energy Management in Smart Homes," in 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), 2012, pp. 1-8.

[18] F. Baker and D. Meyer, "Internet protocols for the smart grid," Request for comments RFC, 2011. http://www.rfc-editor.org/rfc/rfc6272.txt

[19] N. Meratnia, P. Havinga, J. Muller, P. Spiess, S. Haller, T. Riedel, C. Decker, and G. Stromberg, "Decentralized enterprise systems: a multiplatform wireless sensor network approach," IEEE Wireless Communications, vol. 14, no. 6, pp. 57-66, Dec. 2007.

[20] C. Pastrone, M. A. Spirito, and P. Torino, "A Jabber-Based Management Framework for Heterogeneous Sensor Network Applications Riccardo Tomasi Federico Rizzo," International Journal of Software Engineering and Its Applications, vol. 2, no. 3, pp. 9-24, 2008.

[21] A. Kansal, S. Nath, J. Liu, and F. Zhao, "SenseWeb: An Infrastructure for Shared Sensing," IEEE Multimedia, vol. 14, no. 4, pp. 8-13, Oct. 2007.

[22] S. Lei, W. Xiaoling, X. Hui, Y. Jie, J. Cho, and S. Lee, "Connecting Heterogeneous Sensor Networks with IP Based Wire/Wireless Networks," in SEUS-WCCIA'06, 2006, pp. 127-132.

[23] G. Iannaccone, "Fast prototyping of network data mining applications," Proc. of Passive and Active Measurement Conf, 2006.

[24] http://www.planet-lab.org

[25] http://www.tornadoweb.org