Practical Implementations of Cloud Computing Technologies for Smart Metering in Electrical Power Systems

Mitko Petrov Shopov

Abstract **The paper presents a practical implementation of a system for smart metering in electrical power systems that makes use of cloud computing technologies and IoT. The use of cloud computing technologies increase reliability and protection of collected data (it is stored and replicated in multiple secure, commercial-grade storage systems) and eases the software developers in management of remote devices – the actual devices are managed through web services and provided REST-based interfaces. Two M2M cloud platforms are considered in the paper – Digi device cloud and DeviceHive.**

Keywords – **Distributed Embedded Systems, M2M, IoT, Cloud computing, Device cloud, Power systems, DeviceHive**

I. Introduction

With the development of sensor networks, wireless mobile communication, embedded system and cloud computing, the technologies of Internet of Things (IoT) have been widely used in areas such as Smart Meter, public security, Smart homes and so on [1]. There are three essential components of IoT [2]: embedded devices – consisting of both low cost/low power devices and high-end gateways; scalable connectivity – each embedded device should be connected; cloud-based mass device management – centralized management of distributed devices.

II. Background

The joint use of distributed embedded systems and cloud computing technologies is an extension to the efforts of integrating monitoring and control within the business information systems. The aim is to support the binding of monitoring and management of physical processes from the environment with the business logic and business processes [3].

A. Machine-to-machine

Machine to machine (M2M) is a broad term used to describe any technology that enables networked devices – wired or wireless - to exchange information and perform actions without the manual assistance of humans. As such the M2M is an integral part of IoT. The support of legacy devices and devices without proper communication capabilities, will require the use of M2M gateways/aggregation points. These gateways could also be used for some local intelligence and value-added services [2].

A. Smart metering

One of the applications of M2M is smart metering. It integrates communication capabilities with electrical power systems and delivery infrastructure to automate monitoring and control. Dynamically linking utility supply with demand could result in optimization of resource consumption [2].

Smart meter is an advanced power meter that represent advanced metering infrastructure for enabling an automated, two-way communication between the utility meter and the utility provider. The smart meters are equipped with two interfaces: power reading interface and communication gateway interface [4].

A. Cloud-based applications

Some of the advantages of moving the server applications within the cloud are [5], [6], [7]:

- Application development: Development on the cloud will shorten development and prototyping time;
- Device heterogeneity: all the computation and storage is performed on the cloud infrastructure. The end-devices should only implement the communication protocol;
- Flexibility and scalability: Elastic resources allocation. Software updates does not affect end-devices;
- Mobility: Smart meters needs to be connected to Internet. All the computations are on the cloud infrastructure;
- Common user interface: Decoupling the computation infrastructure and the input system, enables multiple user interfaces to exist side by side allowing user-centric customization.

III. Cloud-based Smart Metering Applications

In this section, a practical implementation of a smart metering system will be described. The proposed implementation uses a power meter sensor PST04 [8], an M2M gateway and two different cloud platforms – Digi device cloud [9], and DeviceHive [10].

M. Shopov is with the Department of Computer Systems and Technologies, Faculty of Electronic and Automation, Technical University – Sofia, branch Plovdiv, 25 Tsanko Diustabanov Str., 4000 Plovdiv, Bulgaria, e-mail: mshopov@tu-plovdiv.bg

A. Power meter

The PST04 power meter measure the values of the main parameters of the three phase electric power system – voltage, current, frequency, active power, reactive power, power factor, active energy and reactive energy [8]. These transducers are developed at Development Laboratory for Semiconductor Circuit Engineering at Technical University of Sofia. They have two-wire instrumental serial interface, which enables development of industrial network [11].

B. M2M gateway

The M2M gateway is based on the hardware platform ConnectPort X4 [12]. It offers Ethernet, Zigbee, WiFi communication interfaces and USB/RS232 serial interfaces. It can be configured to route and filter the traffic between different networks. The gateway comes with a custom version of Linux-based operating system and a python-based framework Device Integration Application (DIA) [12].

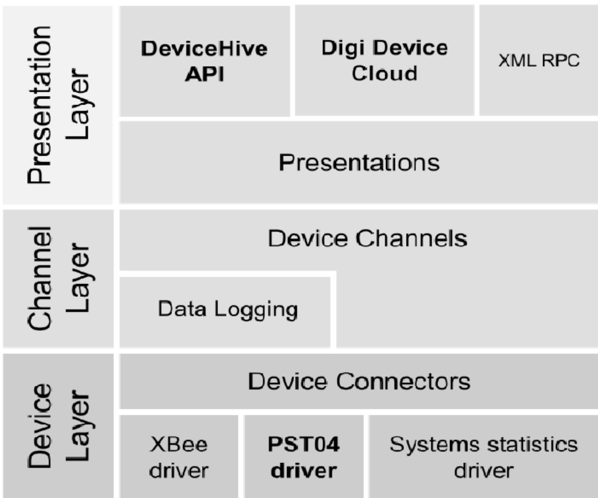

Fig. 1. Device integration architecture [12]

DIA framework provides the core libraries and functions for remote data acquisition, control and presentation between devices and information systems. Its functionality is distributed in three layers as shown on figure 1. The function of device layer is to provide connectors that extract real-world data, represent it as a set of properties and publish them to the appropriate channel on the next layer. Channel layer provides and manages publish-subscribe infrastructure that gives ability to create, remove, and publish to a channel, read from channel and subscribe for channel changes. Presentation layer provides the interface with the outside world. It could be as simple, as a telnet connection or a form of web service interface and even device cloud integration [12]. Using DIA as a basis two components are designed and implemented – (i) PST04 device driver and (ii) presentation driver for communication with DeviceHive cloud. The integration with Digi device cloud is already included.

The PST04 device driver implements the communication protocol of the sensor, extracts its data and loads it into the appropriate data channels. Based on the publish-subscribe mechanism, presentation drivers are automatically notified and uploads the data to its associated cloud services – Digi device cloud and DeviceHive, where it is stored and further processed.

A. Digi device cloud

Digi device cloud is a M2M cloud-based device management platform that includes a variety of APIs. It allows remote management of devices as well as insight into the cloud application health and current state of your entire device network [9].

Devices upload data to data streams on device cloud which are then available for push notifications, alarms, or retrieval by web services clients. Device communication can be achieved using device cloud devices, DIA, or custom cloud adapters (figure 2).

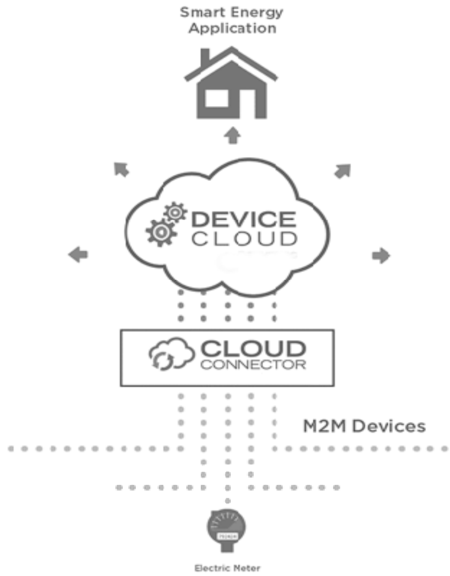

Fig. 2. Digi Device Cloud [9]

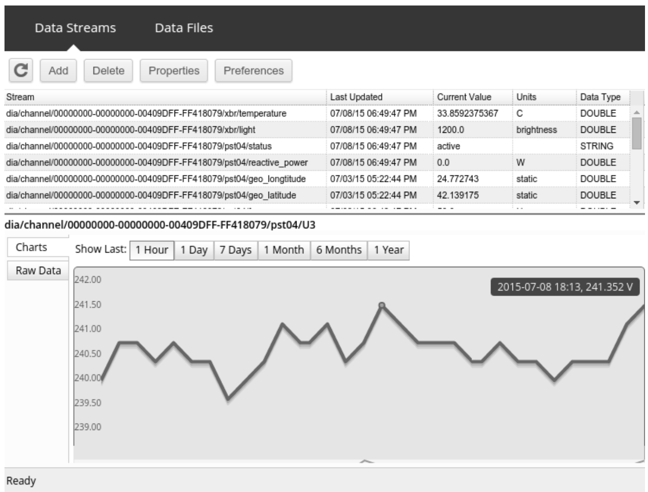

Fig. 3. Digi device cloud data streams for PST04

Time-series data involves two concepts: data points and data streams. Data points are the individual values which are stored at specific times, while data streams are containers of data points. Data streams contain metadata about the data points held within them. Data streams and the data points they hold are addressed using hierarchical paths. Device cloud's data streams service is a REST API to store and access time series data in device cloud [9].

Digi device cloud provide interface for graphical representation of collected data streams (figure 3). The data can be visualized aggregated for different pre-defined time periods. Raw data is also available and can be accessed and managed by external service. Digi device cloud provides a standard HTTP API that allows many ways to access data.

B. DeviceHive

DeviceHive is an open-source M2M framework. It contains a set of services and components that allow establishing a two-way communication with remote devices using cloud technologies as a middleware. The devices can be anything connected: sensor networks, smart meters, telemetry, smart home devices and etc [10].

DeviceHive framework consists of devices, organized in networks, server application working in the cloud and client applications (figure 4). Devices that are not capable of connecting directly to the cloud, uses an intermediate node called gateway, which is responsible for communication with DeviceHive cloud and for managing devices (through some custom or binary protocol). Each device has one or more equipments, i.e. sensors or actuators. Clients can manage only those devices that are part of a network, the client has permission for.

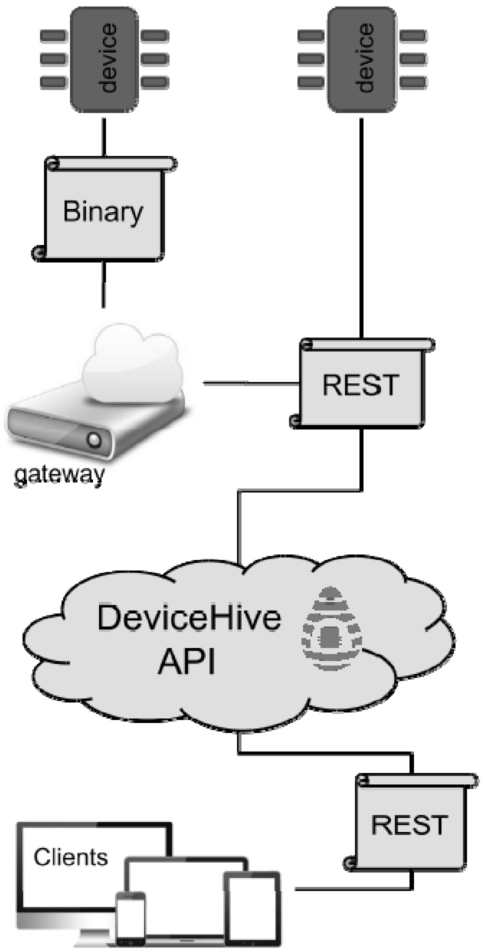

Fig. 4. DeviceHive Framework [10]

- In the presented implementation the PST04 sensor is a binary device, and its integration with the DeviceHive uses the M2M gateway discussed above. The M2M gateway operation flowchart is shown on figure 5. Blocks marked with gray are related to DeviceHive framework support. All the other blocks are implemented in the presentation driver (figure 1). The presentation driver first initialize DIA channels. In the next step, the DeviceHive framework is initialized, which includes connection to the cloud application, authorize the application with its credentials and obtain list of available networks. Then the application register the PST04 device and its equipment to DeviceHive.

After all the initialization and registrations are done, the presentations driver waits for: (i) commands received from DeviceHive, execute the command and responde with the result; (ii) notifications from the DIA channel layer, extract data and push it to the DeviceHive.

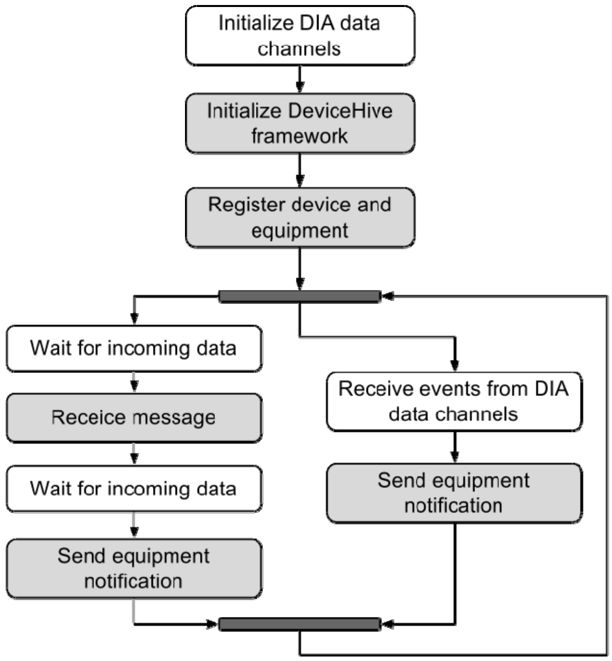

Fig. 5. Flowchart of device-to-cloud operations

For the management of DeviceHive devices a simple Android application is developed. It uses the provided REST API and its functionality includes authentication to DeviceHive, select a network from authorized, view devices and equipment, send commands and receive notifications (figure 6). The application can be configured to receive both synchronous and asynchronous notifications.

DeviceHive does not provide graphical representations of collected data series as Digi device cloud do. This could be provided by external services, through the use of the provided RESTful API interface.

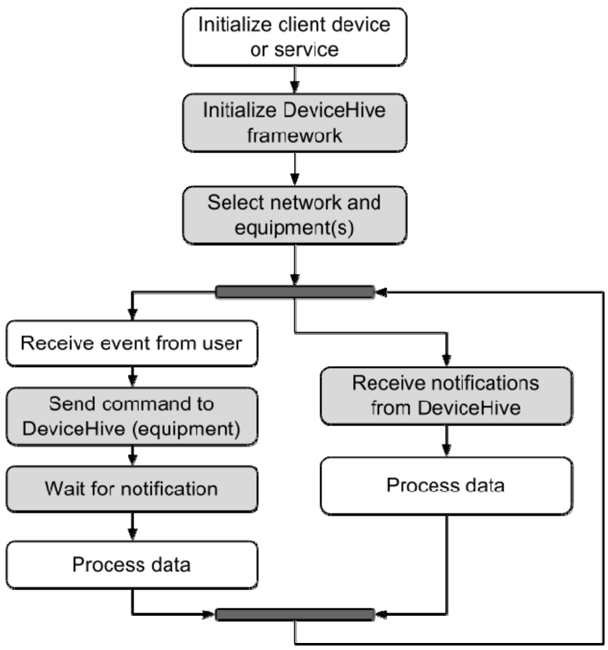

Fig. 6. Flowchart of client-to-cloud operations

IV. CONCLUSIONS

The paper presents a practical implementation of a system for smart metering in electrical power systems that makes use of cloud technologies and IoT. The use of cloud computing technologies increase reliability and protection of collected data (it is stored and replicated in multiple secure, commercial-grade storage systems) and eases the software developers in management of remote devices – the actual devices are managed through web services and provided REST-based interfaces. Two cloud platforms are considered in the paper – Digi device cloud and DeviceHive.

The Digi device cloud has the benefit of providing ready to use set of devices, that requires minimum to no programming for connecting it to the cloud. It also provides some extra services like graphical representation of collected data and device health monitoring. One of the disadvantages of Digi device cloud, however is that it is closed for development and cannot be hosted on your own private environment. On the other hand, DeviceHive is an open-source platform and can be further developed, extended, and hosted in private environment. It still has lack of some functionality, particularly in client-side services. The future work includes developing some new client-side services and extending the functionality of both the Android client application and M2M gateway.

ACKNOWLEDGMENTS

The paper preparation is supported by Project ДФНИ E02/12 “Investigation of methods and tools for application of cloud technologies in the measurement and control in the power system” (http://dsnet.tu-plovdiv.bg/energy/).

REFERENCES

[1] Beng, L., “Sensor cloud: Towards sensor-enabled cloud services,” Intelligent Systems Center Nanyang Technological University, 2009.
[2] Wu, Geng, Shilpa Talwar, Kerstin Johnsson, Nageen Himayat, and Kevin D. Johnson. "M2M: From mobile to embedded internet." Communications Magazine, IEEE 49, no. 4 (2011): 36-43.
[3] N. Kakanakov, S. Stoyanov. *Using Cloud Computing and Sensor Networks for Sensing and Control in Power Systems*, Journal Automation and Informatics, vol., no., pp., 2015.
[4] Zubair Md. Fadlullah, Mostafa M. Fouda, Nei Kato, Akira Takeuchi, Noboru Iwasaki, and Yousuke Nozaki, “Toward Intelligent Machine-to-Machine Communications in Smart Grid,” IEEE Communications Magazine, vol. 49, no. 4, pp. 60 - 65, April 2011.
[5] Alamri, A. et al. "A survey on sensor-cloud: architecture, applications, and approaches." International Journal of Distributed Sensor Networks, vol. 2013, Article ID 917923, 18 pages, doi:10.1155/2013/917923, 2013.
[6] Botta, Alessio, et al. “On the Integration of Cloud Computing and Internet of Things,” In Proc. 2nd International Conference on Future Internet of Things and Cloud (FiCloud-2014), pp. 27-29, 2014.
[7] Nabi, Zubair, and Atif Alvi. "Clome: The practical implications of a cloud-based smart home." *arXiv preprint arXiv:1405.0047* (2014).
[8] Nikolov G., P. Yakimov, Virtual Three Phase Power Transducer, Annual Journal of Electronics, pp. 184, ISSN 1313-1842, 2009.
[9] Digi Device Cloud: https://devicecloud-uk.digi.com
[10] DeviceHive official site: http://devicehive.com/
[11] Yakimov, P. I., S. J. Ovcharov, N. T. Tuliev, E. G. Balkanska, R. V. Ivanov. Three Phase Power Transducer for Remote Energy Management System Application, Annual journal of electronics, vol. 4, No. 2, 2010, pp. 31-34, ISSN 1313-1842.
[12] Digi Embedded gateway ConnectPort X4 – http://www.digi.com/connectportx4