Support System for Rational Use of Electric Energy

Tiago Teixeira[1,2] and Benedita Malheiro[1,2]
[1] School of Engineering, Polytechnic Institute of Porto
[2] INESC TEC – INESC Technology and Science (formerly INESC Porto)
Porto, Portugal
teitiago@gmail.com and mbm@isep.ipp.pt

Abstract—**This paper presents the system developed to promote the rational use of electric energy among consumers and, thus, increase the energy efficiency. The goal is to provide energy consumers with an application that displays the energy consumption/production profiles, sets up consuming ceilings, defines automatic alerts and alarms, compares anonymously consumers with identical energy usage profiles by region and predicts, in the case of non-residential installations, the expected consumption/production values. The resulting distributed system is organized in two main blocks: front-end and back-end. The front-end includes user interface applications for Android mobile devices and Web browsers. The back-end provides data storage and processing functionalities and is installed in a cloud computing platform – the Google App Engine – which provides a standard Web service interface. This option ensures interoperability, scalability and robustness to the system.**

Keywords—Rationalization, Efficiency, Electricity, Android, Cloud computing.

I. INTRODUCTION

The European Union (EU) defines, through the Directive 2006/32/EC, its policy concerning energy usage efficiency and energy services. According to this directive, the member states must improve end-user energy efficiency and, in particular, must create conditions for the development and promotion of new services that attribute the final customer a preponderant role in the efficient and rational use of the energy resources [1]. As a result, the final consumer needs to have access to pertinent information, *e.g.*, energy usage profiles, and tools to adopt a pro-active attitude and make decisions that contribute to reduce his energy consumption.

The goal of this project is to develop a system that allows the final energy consumers to monitor their energy usage based on the user and the energy consumption profiles. Thus, this tool provides the final consumer with the necessary means to monitor and manage efficiently his energy use and, consequently, contribute for the reduction of his energy bill. It contemplates both smart meters and legacy meters, covering residential and commercial consumers.

The system back-end is installed in a cloud computing node and provides data storage (users, installations, meters and energy consumption readings) and processing functionalities through a Web service interface, ensuring interoperability, availability, scalability, robustness and standardization.

In terms of user front-end, the system provides support both for mobile devices and fixed devices with Internet access. The development was performed resorting to free or open source technologies and tools.

The original contribution of this work is to use a cloud computing platform as the system back-end to provide processing and storage capabilities, to host the Web application and to interface with existing applications seamlessly.

This paper, apart from this introductory section, includes sections on energy measurement systems, system development, tests and the conclusions.

II. ENERGY MEASUREMENT SYSTEMS

According to the European Smart Metering Landscape Report [2], all smart measurement systems include the collection of the energy consumption data and the presentation of the resulting load profile.

A. Architectures

The system architecture is dictated by the types of meters contemplated and by the functional requirements, namely, the data collection, processing and storage, accessibility, security, standardization and interoperability requirements.

Stand-alone systems are used when the data collection, storage, processing and presentation is performed locally. These systems are based on In-Home-Display (IHD) equipments that read, process, store and display the data. The advantages of this approach are that the data privacy and accessibility are ensured, since the storage is done locally. On the other hand, it withholds data and suffers from serious storage capacity limitations.

Distributed systems are made of a set of physically distributed modules with well defined roles that require permanent connection to the Internet. As a result, they can only be adopted when Internet access is ensured. The Web applications are examples for this type of solution that resort to distinct platforms for storage, information processing and for the user-interface application (the Web browser) [3]. The main advantages of this architecture are the high storage capacity, remote data access, data distribution and the possibility of adopting distinct user interface equipment. The disadvantages include the user data vulnerability, the additional complexity of the system and the need to maintain data backups.

B. Data Collection

The evolution of the Information and Communication Technologies (ICT) and measurement systems led to the creation of meters with Automatic Meter Reading (AMR) capabilities that allow the remote collection of readings through a data link supported by radio frequency, mobile communication, *e.g.*, the Global System for Mobile

Communications (GSM) and General Packet Radio Service (GPRS), or through Power Line Communication (PLC). This is the case of automatic meters and smart meters. AMR coexists with the traditional energy consumption readings performed at the consumer premises by humans. This is the case of the all installations equipped with electromechanical meters.

C. Customer Equipments

At the customer side, energy measurement systems rely on meter and user interface equipments.

An electricity meter is an intermediate device between the electric distribution grid and the consumption or production (residential, industries, commerce, services or production centrals) premises that measures the electric energy consumed or produced. Nowadays electricity meters include smart meters, automatic meters and legacy meters. Smart meters, which are a key technology of the smart grid, support bi-directional communication between the consumer installation and the utility supplier [4]. Automatic meters provide a one-way communication that is used to report periodically the energy consumption readings and status of the meter to the utility supplier. Legacy meters are stand-alone electromechanical meters that require human reading and reporting.

The user interface equipment depends on the type of meter installed, which influences the data collection methodology, and the type of computing device used by the consumer. There are: (*i*) mobile computing devices with location sensors, high processing power and Internet access; (*ii*) fixed computing devices with high processing power and Internet access; and (*iii*) IHD that are stand-alone devices installed in the consumer premises that can be used to monitor and manage the local electric energy consumption.

D. Examples

The energy measurements systems analysed in this section include both AMR systems, where the energy usage information is automatically collected, and legacy systems, where the readings are done and reported by humans. They adopt diverse architectures, meter and user interface equipment. In the case of the legacy meters without remote access is usual to find stand-alone solutions and, in case of modern meters with remote access it is frequent to adopt distributed solutions. In terms of user interfaces, the energy measurement systems resort to Web browsers, IHD and mobile devices.

1) Visible Energy Trial

The Visible Energy Trial (VET) is a stand-alone system that adopts IHD devices to display the consumer's load profile [2]. The equipment stores up to six months of data for analysis and can be integrated with a Web portal. Beyond the load profile visualization, the IHD devices allow the establishment of daily goals that, when exceeded, generate warnings [5].

2) ACE Vision

ACE Vision is a distributed Automatic Meter Reading (AMR) system developed by Itron for the Commercial and Industrial (C&I) market [6]. This Web application, which is intended for smart meters, provides the load profile, the active and reactive energy and generates warnings based on the consumption ceilings defined by the consumer through the Web page. It also displays the monthly expenditure by tariff and analyses the power factor.

3) EcoreAction

EcoreAction is a distributed AMR system developed by Ecore that provides end users equipped with smart meters with a user-friendly self-service for monitoring and predicting energy consumption. It is a Web application that provides useful comparison data and presents the user's consumption history. Furthermore, the users can create a personal plan for their energy consumption and receive automatic reminders when their actual consumption is about to exceed the budgeted values [7].

4) MeterReading

MeterReading is a stand-alone application developed for Android devices aimed for the residential market equipped with legacy meters. The final consumer introduces, with the desired frequency, the energy consumption readings and, as soon as a minimum dataset is available, the application provides a collection of analysis graphs to help the consumer to make decisions leading to an efficient electricity usage [8].

5) ISMRT

Itron SMaRT (ISMRT) is the distributed system described in this paper. It supports both residential and commercial meters. In the latter case, it obtains the commercial readings by interacting with the ACE Vision system. In the case of legacy residential meters, it relies on the consumer willingness to submit the readings via the mobile or the Web applications. It provides the user with set up functionalities, *e.g.*, consumption ceilings definition, data display functionalities, *e.g.*, the load profile, and data analysis functionalities, *e.g*, forecasting.

III. System Development

The ISMRT functionalities and architecture are presented in this section.

A. Functionalities

The system provides general and market-oriented functionalities:

- General:
  - User registration and authentication;
  - Definition of meters and installations;
  - Presentation of energy consumption/production maps by region and user installation type.
- Residential and C&I markets:
  - Submission of energy consumption readings;
  - Presentation of energy consumption profiles;
  - Presentation of the relative energy tariff weight in the total energy consumption;
  - Definition of maximum energy consumption alarms.
- Micro and mini-energy production markets:
  - Presentation of the detailed or aggregated production in terms of energy, power or capital;
  - Forecast the investment amortization date;
  - Definition of minimum production alarms.

These functionalities can be grouped into two sets: (*i*) information management and storage; and (*ii*) information processing, including the consumption/production analysis and forecast.

Authenticated users can manage (create, edit, recover and eliminate) their stored data (user, installations, meters and readings) and terminate their session. In the mobile application, they can additionally schedule readings and corresponding notifications using the Android's `NotificationManager`.

The data processing includes the generation of different types of charts, *e.g.*, the consumption/production profile, the expenditure by tariff or the relative tariff weight. Furthermore, the system is able to create consumption/production forecast charts, predict the amortization date of an energy production project and generate maps comparing the user's energy consumption/production with an anonymous group of identical users from a specified region. Finally, the user can establish consumption ceilings and schedule consumption/production alarms.

The GAE houses the Web service that provides the described operations, stores the instances of each object in the Datastore and implements a scheduled job that checks daily the defined alarms and updates, in the case of a production installation, the amortization date.

B. Architecture

The ISMRT system is organised in two main blocks: front-end and back-end – see Figure 1.

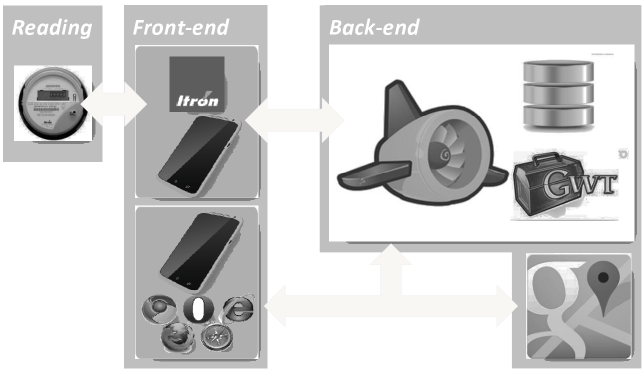

Figure 1. ISMRT architecture.

1) Front-end

The front-end includes two applications: the Web application and the mobile application. The mobile application and the Web application front-end modules are simultaneously user interface applications and back-end system clients.

a) Web Application

The Web application was developed in Java together with JavaScript elements using the Google Web Toolkit (GWT) framework, including the GWT library, the GWT Designer and the Google Visualization library for GWT. The GWT enables writing code both for the client-side (Web browsers) and for the server-side (server application). The communication between client and server is done through a RPC service. The Google Visualization library allows the inclusion of different kinds of charts (pie, linear, bars, maps, *etc.*) as JavaScript classes that are rendered in HyperText Markup Language (HTML) 5, guaranteeing the compatibility between browsers and mobile platforms. Finally, the Google Maps API was used to include personalized maps from Google Maps in Web pages.

The Web application, which is mainly for residential customers, provides access to the following functionalities through a Web browser: user registration and authentication, definition of installations and consuming ceilings, submission of energy consumption readings, analysis and comparison of consumption history. The C&I customers are not included in this application since they use the ACE Vision system.

a) Mobile Application

The Android mobile application was developed using the SQLite, kSOAP. AChartEngine and Google Maps libraries. SQLite was used to provide on device storage. It is an open source library that implements a local Structured Query Language (SQL) compliant relational database. kSOAP, which is an open source Simple Object Access Protocol (SOAP) Web service client library for Java J2ME applications, was used for interaction with the back-end Web service. The graphs were created with the AChartEngine library for Android. The Google Maps API was used to add personalized maps from Google Maps to the mobile application.

The mobile application, which is for both residential and C&I customers, includes the Web application functionalities and, additionally, allows the scheduling of residential readings through notifications and the local storage of user, installations, filters and forecast models data to allow offline work as well as to minimize the data volume exchanged with the server. This information is equally stored at the back-end and can be retrieved by the mobile device when required.

2) Back-end

According to [9], smart grids are estimated to generate dozens of gigabytes of data daily. As a result, the back-end is hosted by the Google App Engine (GAE), which is a cloud computing Platform-as-a-Service (PaaS) platform. The applications hosted by GAE are stored in the Google's cloud computing infrastructure. Depending on the contract agreement, Google guarantees the scalability of the storage and processing needs of the user applications. The GAE can host applications written in Java and provides a group of complementary development libraries, *e.g.*, to dispatch and receive emails or to define scheduled jobs. The GAE data storage service is the Datastore. The Datastore is a storage environment based on entities rather than on schemas. Each entity has a unique key, is identified by its kind and may have different properties. A stored object becomes an entity of a given kind; objects of the same kind can have different properties; a property can be of different data types (text, boolean, numerical or date). Data is accessed through queries specifying the kind and properties of the required entities. Each entity has unique key based in the kind and on the relationship with other kinds. GAE provides also an administration panel to maintain and configure the hosted user applications.

The back-end supports three modules: the App Engine service, the GWT server, which holds the Web application, and the Datastore. The App Engine service offers data processing, data storage (Datastore) and seamless interaction functionalities with other applications (ACE Vision) through a Web service. The list of the exposed data management operations is presented in TABLE I.

TABLE I. WEB SERVICE DATA MANAGEMENT OPERATIONS

Entity	Operation
User	CreateUser DeleteUser UpdateUser EditPassword EditProfile
Installation	CreateInstallation EditInstallation DeleteInstallation UpdateInstallation
Meter	CreateResidentialMeter CreateCommeIndusMeter CreateProductionMeter EditCommeIndusMeter EditProductionMeter DeleteMeter UpdateCommeIndusMeter
Reading	CreateReading GetReadings CreateFilter CreateLoadProfileFilter CreateEndOfBillFilter DeleteFilter DeleterFilers UpdateFilters GetMaxPower

TABLE II. displays the list of the Web service data processing operations involving tariffs, ceilings, alarms, comparison filters and forecasting.

TABLE II. WEB SERVICE DATA PROCESSING OPERATIONS

Entity/Process	Operation
Tariff	CreateTariff UpdateTariff UpdateTariffs
Ceiling	CreateCeiling DeleteCeiling UpdateCeilings
Alarm	CreateAlarm SendAlarm DeleteAlarm DeleteAlarms
MapFilter	CreateRegionFilter DeleteRegionFilter DeleteRegionFilters UpdateRegionFilters GetResidentialResults GetCommIndResults ListCountries ListRegions ListTowns ListZipCodes
Forecast	CreateModel

The Datastore module stores user, installation, meter, location, tariff, residential reading, filter and task entities. TABLE III. provides a complete listing of the entities represented.

TABLE III. DATASTORE ENTITIES

Entity	Description
User	User data
Installation	User premises data
Meter	Residential, C&I and production meter data
Reading	Residential meter reading data
Tariff	Meter tariff data
Statistic	Residential statistics data
Ceiling	Consumption ceiling data
SchedUpdateCI	Scheduled C&I meter data updates
Filter	Ace Vision search filter data
Alarm	Alarm data
MapFilter	Map filter data
LocationFilter	Location filter data

IV. SIMULATION RESULTS

The goal was to test the data processing operations and the developed interfaces. The group of tests was organized in: (*i*) user data management; (*ii*) installations and meters data management; (*iii*) reading submission and scheduling; (*iv*) consumption and production analysis; (*v*) definition of consumption ceilings; (*vi*) comparison of user profiles; (*vii*) forecast models; (*viii*) and system communication analysis.

The adopted test environment included the Google App Engine, a 32-bit laptop with Windows XP Operating System (OS) and an htc Desire phone with Android 2.2 OS.

First, two distinct users were created: one with the Web application and another with the mobile application. Deliberate data input errors were introduced in the input forms to test the implemented data input test functions. They were detected and the user was warned. After the creation of the new user accounts, the service successfully sent an account activation email to each user to complete the creation process. The next step was the definition of an installation with one meter per user. The definition of an installation involves the specification of the location and the meter features, including the applicable energy tariffs. In the mobile application the location is obtained by the device using the `LocationManager` class and the `getFromLocation` method. In the Web application the geodesic coordinates of the location are obtained through the `GeocoderRequest` class. Both users defined residential meters.

The tests proceeded with the submission of four readings per tariff and per meter. While the C&I meters submit automatically their readings to the Ace Vision system, the residential meters readings must be submitted manually. First, the mobile application user scheduled a notification task (to notify the user that a reading should be submitted every 5 min) and, then, inserted the four readings per tariff. The Web application user submitted the four readings per tariff using the reading submission menu, selecting the reading date from a calendar and introducing the reading value. At the end of this operation, the readings from both meters and corresponding statistics were stored in the Datastore. Figure 2. displays the consumption and expenditure (pie charts) and the energy consumption (linear charts) by tariff. The left and right charts are from the mobile application and Web application, respectively.

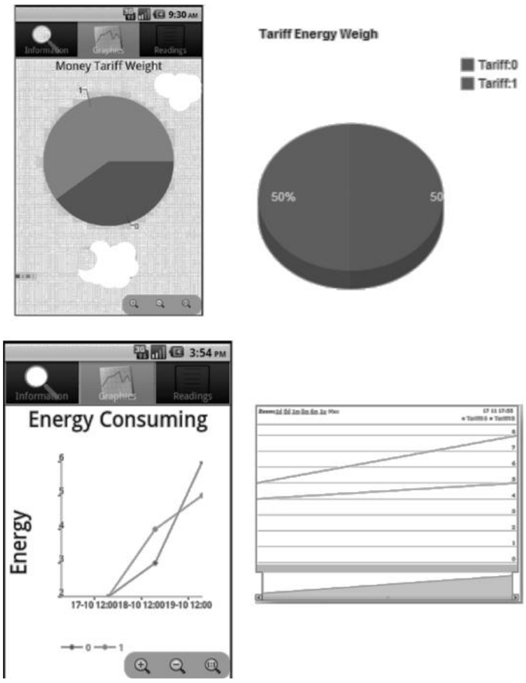

Figure 2. Energy consumption/production charts.

The users then established a consumption ceiling for each meter corresponding to 80 % of the last reading. Since the consumer ceilings generate warnings when the user consumption reaches 75 % of the established value, the system issues a warning and displays the percentage of the ceiling value reached so far – see Figure 3.

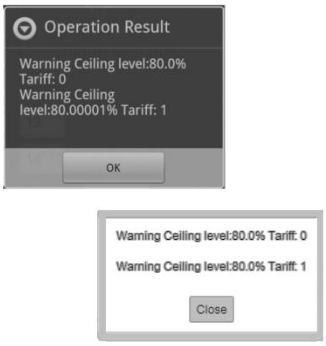

Figure 3. Ceiling warnings.

Figure 4. illustrates the creation of a consumption comparison filter between two meters located in different regions. The colour of the marker indicates if the consumption of the selected meter is above (red) or below (green) the chosen reference meter.

The next step was to test the forecasting capabilities. A consumption/production forecast requires a sample of, at least, 30 day consecutive readings (observations). The sample data is first smoothed using a moving average filter and then the forecast, which is based on a polynomial regression model, determines the function that best describes the smoothed data sample. The model forecasts up to 50 % of the provided sample period. In this case, with a 30-day sample, a 15 day forecast was obtained. To test this functionality, the user defined a new production meter with the desired profile. The result of the smoothing algorithm is illustrated on Figure 5. The left and right charts display the raw and the smoothed data.

Figure 4. Meter consumption comparison.

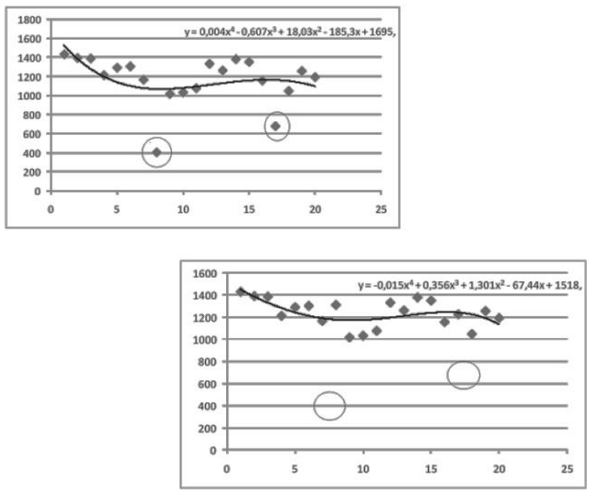

Figure 5. Data smoothing.

Finally, and since mobile communication entails costs, a series of tests was conducted to determine the exchanged data size and the time elapsed during the information submission and retrieval operations between the mobile application and the GAE. The information submission tests involved the: (*i*) elimination of all information stored on Datastore; (*ii*) creation of a user; (*iii*) definition of an installation with two meters (a residential and a C&I meter); (*iv*) submission of 10 readings; and (*v*) specification of a consumption ceiling. The information retrieval tests included the following requests: (*i*) ten readings from a residential meter; (*ii*) one month load curve with a daily update rate from a C&I meter; (*iii*) three months of the end of invoice and maximum power; (*iv*) a meter alarm set (22 instances); (*v*) a one month forecast model; and (*vi*) a comparison map. The last test included the deletion of all data stored in the mobile device followed by the recovery of the user data from the Datastore.

This collection of tests involved 205.112 KiB of downloaded data and 34.551 KiB of uploaded data. In the case of a standard monthly mobile data transfer package, that includes 500 MiB of download data, this traffic is irrelevant (≈ 0.04 %).

V. FUTURE WORKS AND CONCLUSIONS

This section discusses the results, proposes future improvements and presents the final conclusion.

A. Discussion

The ISMRT system monitors the consumption/production of electric energy and promotes the rational usage of the resources. It allows the definition of users, installations, consuming ceilings (warnings) and alarms, the submission of readings, the comparison of the consumption/production profiles between users with identical profile, the graphical representation of consumption/production of electric energy and, in case of C&I installations, the forecast of the consumption/production. It is a distributed system composed of back-end and front-end. The back-end, that includes the persistent data storage and the data processing Web service, is hosted in a cloud computing platform – the Google App Engine – ensuring interoperability, scalability and robustness. The front-end includes two user interface applications: the mobile application for Android devices and the Web application. The system was fully implemented using open source technologies and a free usage service-level-agreement.

The operational tests conducted demonstrated the correct operation of both the back-end and front-end functionalities. The data communication tests made with the mobile application concluded that the information volume exchanged with the server is reduced. The largest traffic volume occurs when obtaining maps from the Google Maps server (≈ 180 KiB). In any case, the traffic generated corresponds to insignificant financial charges for monthly subscription holders. In terms of time response, the worst case corresponded to the user data recovery operation that took ≈ 22 s. The tests were performed with real data but few users. Residential readings were collected manually and submitted by the consumers and industrial readings were collected and submitted automatically to the AceVision application.

B. Future Implementations

In terms of future development suggestions, improvements can be introduced in the layout, visual presentation, data encryption and management, forecast models and automatic meter identification, *e.g.*, using Quick Response (QR) code labels.

C. Conclusion

The ISMRT prototype allows users to monitor their energy consumption/production profile from a mobile Android device or a fixed computer and to adopt a pro-active role towards the rational use of the electric energy resources and, thus, reduce the energy bill.

The adoption of the GAE cloud computing platform to host the back-end ensured the interoperability, scalability and robustness of the solution. The processing and storage resources offered by Google were enough for the development and testing phases. The Web and mobile applications illustrate the back-end's interoperability. Furthermore, the GWT allowed the Web application development to be performed fully in Java.

The prototype was handed to Itron – Sistemas de Medição, Lda. However, prior to moving into a production environment, the GAE service level agreement should be negotiated and the final user assessment, involving a significant number of users, should be performed.

ACKNOWLEDGMENTS

The authors thank their home institution for the facilities provided and to Itron – Sistemas de Medição, Lda. for the information and support provided.

This work was partially supported by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «FCOMP – 01-0124-FEDER-022701».

REFERENCES

[1] The European Parliament and The Council Of The European Union, "Directive 2006/32/EC of the European Parliament and of the Council of 5 April 2006 on energy end-use efficiency and energy services and repealing Council Directive 93/76/EEC," 2006.

[2] SmartRegions, "European Smart Metering Landscape Report," Tech. Rep., 2012. [Online]. Available: http://www.smartregions.net/.

[3] A. Tanenbaum, Andrew and M. Steen, "Distributed Systems Principles and Paradigms." Pearson Prentice Hall, 2007.

[4] Fischer, J. E. *et. al*., "Recommending Energy Tariffs and Load Shifting Based on Smart Household Usage Profiling". Accepted at the International Conference on Intelligent User Interfaces, Santa Monica, CA, USA, 2013.

[5] T. Hargreaves, M. Nye, and J. Burgess, "Making Energy Visible: A Qualitative Field Study of How Householders Interact with Feedback from Smart Energy Monitors," Energy Policy, vol. 38, no. 10, pp 6111-6119, 2010.

[6] Itron, Ace Vision, 2011. [Online]. Available: https://www.itron.com/aunz/PublishedContent/0043-EN-ACE_Vision-0907b.pdf.

[7] Ecore, EcoreAction, 2010. [Online]. Available: http://www.ecore.fi/temp_eng/images/EA_cases_2010_eng.pdf.

[8] Kolle. Help pages for MeterReading, 2011. [Online]. Available: http://kolle.mobi/apps/meterReading/help/help.html.

[9] S. Rusitschka, K. Eger and C. Gerdes, "Smart Grid Data Cloud: A Model for Utilizing Cloud Computing in the Smart Grid Domain," In First IEEE International Conference on Smart Grid Communications (SmartGridComm), pp.483-488, 2010.