Accepted to IEEE Journal on Selected Areas in Communications, 31(7), July 2013.

Energy Service Interface: Accessing to Customer Energy Resources for Smart Grid Interoperation

Eun-Kyu Lee, Rajit Gadh, and Mario Gerla,

***Abstract*—The Energy Service Interface (ESI), sitting at the boundary of a customer facility, plays a communication gateway role - interconnecting internal customer energy resources to external systems. A number of customer energy services are realized over the interconnected communications, which then contributes to smart grid interoperation eventually. In this paper, we examine the design issues of the ESI. To facilitate bi-directional customer energy services, the ESI must serve as both a service consumer and a service provider. At the same time, it must protect the customer energy resources from external threats and maximize the interoperation. To verify the issues, we build and deploy two ESI testbeds. Throughout experiments with a couple of energy service scenarios, we verify that the ESI plays the service "prosumer" in an interoperable manner. We also evaluate the performance of the security mechanism applied to the ESI and examine the potential of exploiting the Cloud technology for the ESI deployment. To the best of authors' knowledge, this is the first deployment of the ESI that addresses the fundamental, functional requirements.**

***Index Terms*—Smart grid, customer domain, energy service, energy resource, energy service interface, interoperation, energy management system**

I. Introduction

SMART grid consists of a myriad of heterogeneous systems, and their seamless interoperation is the key element for the success of our future power system. To facilitate interoperability, National Institute of Standards and Technology (NIST) presents a conceptual model consisting of seven domains [1]. Among them, the customer domain - industrial, commercial, and residential sectors - is the primary energy consumer; it consumes 72% of total energy in the U.S. [2]. Therefore, it is essential to interoperate with energy resources in the customer domain in order to achieve the goal of smart grid - balancing the power demand with supply.

The customer interoperation is generally realized via a number of customer energy services. To enable the services across the customer domain, NIST defines two gateway actors - utility meter and Energy Service Interface (ESI) [1]. They place at the boundary of the customer domain and exchange energy data between the customer domain and other external domains (e.g., distribution, operation, and market). That is, they take the bridge role for inter-domain communications. Owned and fully controlled by a utility company, the utility meter measures and collects aggregated energy usage data for the customer billing purpose. However, as the utility meter is designed to handle the aggregated data only, the ESI is expected to process most emerging energy service data. In this way, the ESI would exceptionally contribute to the customer interoperation, and there is no doubt that smart grid interoperation cannot be accomplished without its help. However, as yet, few research investigated the development of the ESI in depth.

To address the growing concern of the ESI, this paper examines its design issues and implements and deploys a prototype that realizes the customer energy services. For the ESI design, we classify the issues into four categories. First, the ESI consumes energy services that external domains provide to the customer domain. To this end, it translates the semantics of external services into internal contexts as well as exchanges messages in a standard way. Next, the ESI provides energy services, as a service provider, to external domains. It creates a new service using internal energy data and defines communication interfaces through which external systems access the service in an efficient and interoperable way. Third, the ESI implements a security mechanism for inter-domain communications. It protects internal energy resources and prevents a security failure from being spread across domains. Last, the ESI is a logically-defined communication interface and can be implemented in any physical system. This property allows various types of ESI implementation architectures to be deployed.

This paper also builds an ESI testbed that includes an ESI, an energy management system, customer energy resources, and a demand response service server. We carefully take the design issues into consideration and implement our ESI on the energy management system. The ESI implements a demand response client module. It adopts a standard XML format to represent the customer energy data and implements an object-based web service interface for inter-domain communications. All the service objects are protected by a resource centric security mechanism. We build an additional testbed in which the ESI is deployed on a public Cloud. To verify the operation of the ESI and to evaluate its performance, we run experiments with several energy service scenarios. Throughout the experiments, we demonstrate that the ESI interacts with the demand response service (as a service consumer) and provides remote energy management services to external users (as a service provider). The experimental results are analyzed further to identify how the security mechanism and the Cloud architecture affect the service performance.

E.K. Lee, R. Gadh, and M. Gerla are with the Henry Samueli School of Engineering and Applied Science, UCLA, Los Angeles, CA, 90095 USA e-mail: eklee@cs.ucla.edu, gadh@ucla.edu, gerla@cs.ucla.edu.

This work has been sponsored in part by grant from Korea Institute of Energy Research (KIER) (KOREA) fund 77592, UCLA and KIER Joint Research Project on Smart Green Building.

II. Customer Interoperation via Energy Service

The customer domain will include a myriad of energy resources (load, generation, and storage). Because the customer domain is the primary energy consumer in smart grid, interoperating with the resources is the most critical concern to balance the power demand with the supply. A couple of literature list the use cases of customer energy services that accomplish the *interoperation* [1], [3], [4]. The following subsection categorizes the customer energy services that the energy resources in the customer domain are involved.

A. Customer Energy Services

Energy Usage Collection. The energy usage collection is the simplest form of services. The customer domain periodically transmits customers' energy usage data to a utility company that processes the data for customer billing.

Efficient Energy Management. The efficient energy management service informs the customers of the details of energy usage information so that they clearly understand their usage pattern. A sub-metering system in the customer facility keeps tracking of the energy usage of individual energy loads. An Energy Service Provider (ESP)[1] may analyze the usage pattern and guide the customers to consume power more efficiently. User-friendly devices such as In-Home Display show the breakdown of the usage and corresponding costs. An Energy Management and Control System (EMCS)[2] allows the customer to control the energy loads even remotely. The energy usage data is also used to validate bills from the utility company.

Customer Feedback. Customer feedback represents the deliver of customer energy information to an ESP that exploits the information to create further services. This can eventually contribute to the reliability of power infrastructure. For instance, a demand forecast service delivers information of expected customer energy demand to a power supplier so that the supplier can accurately estimate the amount of future power generation. To this end, an ESP provides power price data and weather forecast to a customer who already understands his energy usage patterns. Then, the customer examines local energy needs of the future. As an another example, the sub-metering system measures power quality, and the ESP uses it to pinpoint a potential failure spot and to maintain the health of the power supply system.

Direct Load Control. The Direct Load Control (DLC) permits external users to control internal energy loads. Currently, ESPs provide a primitive form of the service targeting at specific energy loads such as an electric water heater. They directly control the operations of the heater during the period of power supply emergencies. In the future, the ESPs will control various types of energy loads for different purposes. For instance, an ESP remotely stops charging an Electric Vehicle (EV) at a customer's garage when the power price goes beyond a contracted value, which can save the customer's electricity cost. The ESP may launch grid stabilization services such as frequency regulation and Volt-Amps Reactive (VAR) compensation. A combination of inductive and resistive loads at the customer facility can be controlled to help balance active and reactive power.

Intelligent Demand Response. Power suppliers often confront a shortage of generation capacity during peak-demand periods. Instead of constructing additional power plants to cope with the shortage, a Demand Response (DR) service tries to reduce energy consumption in the customer domain by sending DR signals to customers. A utility operator calls contracted customers who, then, manually stop their building operations by expecting financial incentives. New challenges of the *intelligent* DR service are automation and invisibility. When the customer domain receives DR signals from the utility, it automatically performs a predefined DR strategy (shedding and shifting the loads) to achieve the energy curtailment of a service contract. The strategy must be invisibile: It minimizes interference with ordinary building operations. Potential inclusion of EV, storage, and generation at the customer facility will play an important role for the invisible DR strategy.

Distributed Generation and Microgrid. Distributed Generation (DG) represents electricity generation that feeds into the distribution grid directly, not through the transmission infrastructure. The DG can reside within a customer facility as a form of renewable sources - micro-turbines, wind-powered generators, and Photovoltaic arrays. They can be managed and controlled directly by an ESP as a part of the DLC service. A customer may lease the DG and storage, and the ESP remotely monitors the health of the energy assets and controls them directly when necessary. With the DG, the concept of microgrid enables the customer facility to rely less on bulk power sources. In this sense, the microgrid can be leveraged for the intelligent DR service. Upon receiving the DR signal, the customer prepares to operate his own DG. This can compensate some portions or all of energy needs that must be reduced by the terms of the DR service.

Energy Market - Power Trading. With sophistication of the storage and generation technologies at the customer facility, the customer will actively interact with the wholesale and/or retail energy market. He can choose to buy power from one or more power suppliers or generate power on-site for sale in the market. That is, he buys or sells electricity in more flexible ways as a "prosumer". In the trading, real time exchange of price, schedule, and location information across domains becomes of the most importance, because it is critical to dispatch power to the right place at the right time.

Environmental Monitoring. The environmental monitoring service deals with an increasing concern of environmental sustainability. The service measures the greenhouse gas emission of bulk power generation, and the information is shared among the entire smart grid. An on-site generator and building operations also report their emission contents, e.g., carbon dioxide. This way, a customer can enumerate the influence of their energy usage activities on the environment, when he consumes power sourced from various suppliers. He may choose to buy clean energy for higher prices or participate in

[1] An ESP can be a third party service provider, a utility company, an operator, or a power supplier. This paper uses their meanings interchangeably.

[2] The concept of the EMCS includes those of building automation system (control) and energy management system (measurement).

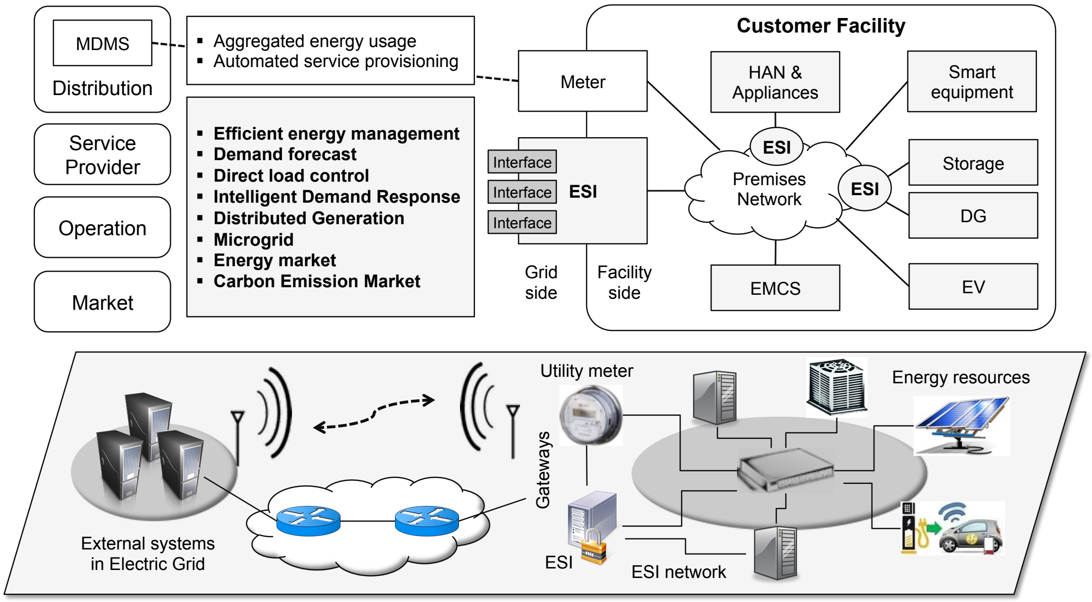


Fig. 1. Smart grid interoperation with the customer domain. Two gateway actors are responsible for providing customer energy services bi-directionally.

an emission trading market with his on-site renewables.

B. Gateway Actors for Customer Interoperation

To facilitate the interoperable energy services, the smart grid conceptual model [1] defines two gateway actors - utility meter and ESI. Sitting at the boundary of the customer domain, they are responsible for inter-domain communications. That is, external systems (utility companies and ESPs) interconnect with the customer energy resources through them. Fig. 1 shows an overall system architecture that represents the smart grid interoperation with the customer domain.

1) Utility meter: The primary function of the utility meter (a.k.a. smart meter) is to provide an accurate, remote measurement of energy usage for customer billing. To this end, a utility company installs the utility meter at a customer facility in which the meter periodically measures aggregated energy usage data. The data is then collected by an aggregator via RF communication within a neighborhood area network, and then delivered to Meter Data Management System (MDMS) in the utility via a cellular network[3]. These hardwares and communications constitute an Advanced Metering Infrastructure (AMI). As an element of the AMI, the utility meter is under full control of the utility while residing within the customer boundary. A meaningful number of utility meters already have been deployed in the U.S., and thus the AMI system architecture has been understood well. Many researches have also investigated on meter data processing and its communications.

2) Energy service interface: The ESI is initially designed as a communication interface to serve demand response signaling. But, because the utility meter is functionally limited[4], the ESI is expected to transmit data streams of most customer energy services. When expecting that a great amount of energy resources will be involved in the energy services, the ESI becomes the most important entity in the customer domain in terms of smart grid interoperability. Unlike the utility meter, however, few literature of smart grid has, as yet, investigated more than its conceptual idea. As of today, a couple of public reports simply discuss the definition of the ESI - "The ESI serves as the information management gateway through which the customer domain interacts with ESPs" [1], [5]–[7]. The discussions share common characteristics summarized below.

The ESI represents a logical boundary of the customer domain. The ESI logically distinguishes the operation of the customer facility from the outside world. To this end, the ESI consists of two sides: the facility side and the grid side. The facility side communicates with internal energy resources directly, or is connected to data storage and processing modules in an EMCS. The grid side interacts with external systems such as ESPs and operators.

The ESI represents a bi-directional service interface. The ESI serves as a platform that provides energy services to both the facility side and the grid side. It exposes raw data and a set of energy services that the customer domain provides to external domains - e.g., direct load controls and demand forecasts. We call this function a *facility service interface*. In the same way, the ESI delivers energy services and data from external systems to the customer domain - *grid service interface*.

The ESI does not represent a physical device. The ESI is a logical interface in the form of software component and thus can be implement on various physical devices such as EMCS and an energy gateway. Or, it can be realized in a standalone server system. It is generally assumed that the ESI is implemented in a customer's EMCS that communicates with external systems via the Internet.

[3] Different utilities may use different communication channels. Our description is based on the system specification of a utility company in California.

[4] The utility meter has a very long lifetime and is cost sensitive so that it is barely upgraded. Moreover, a utility company is unwilling to collect customer energy data other than from the utility meter due to privacy issue.

Development of the ESI is still at an enfant stage, and there are many issues to be considered before its real world deployment. This paper tackles those issues - we discuss their designs and eventually implement a prototype.

III. ESI Design Issues

Reviewing the customer energy services helps us understand the functional requirements of the ESI. This section investigates how to design an ESI system.

A. Grid Service Interface - Interconnecting with External Energy Services

At the core of the ESI are communication and interoperation with the smart grid infrastructure. The ESI must be able to interoperate with external energy services that other domains in smart grid provide to the customer domain. For instance, a Demand Response Automation Server (DRAS) provides a DR service by generating and delivering DR event signals to the customer. Then, the ESI accepts and understands the service signals; delivers event information to internal energy resources; and sends a DR event report back to the DRAS. Many standardization efforts have defined communication protocols of such grid services. Examples include Open Automated Demand Response (OpenADR) [8], Energy Market Information Exchange (EMIX)[5], Energy Interoperation (EI)[6], IEC Common Information Model (CIM) family of standards, and Weather Information Exchange Model (WXXM) [9].

To support the grid service interface, it is essential that the ESI translates the information delivered from external domains into the semantics and protocols that are used within the customer facility. Data mapping is especially critical, because most existing customer systems have used proprietary protocols that were not designed for interoperation. The ESI also translates contexts for security. Since it sits at the boundary between two independent domains, the ESI seamlessly interconnects dissimilar security mechanisms so as to balance their security policies. Misconfigured mapping of security contexts may create vulnerable points and increase the possibility to leak unauthorized private information.

B. Facility Service Interface - Serving Energy Services to External Domain

The ESI provides energy services to external domains, which primarily delivers data that the customer domain generates. In a customer feedback service, for instance, it transmits customer energy information to ESPs. It also accepts command messages that eventually control the customer equipment - e.g., remote energy management service and direct load control. From the viewpoint of the entire smart grid, these interactions make the customer domain appear as a service provider. Thus, the ESI needs to define what and how specific service data is transferred through itself. This subsection discusses five topics.

Interface abstraction: The ESI must consider interface abstraction that determines the appropriate level of internal details that the interface exposes to external domains. High level of abstraction exposes internal business logics with less details, while low level of abstraction exposes more details of internal operations. DLC and event-based DR show two extreme examples with respect to control capability. In DLC, the ESI allows external systems to control energy loads directly, which realizes the lowest level of abstraction. On the other hand, it receives a DR event signal from a DRAS, from which the facility determines its own control strategy without revealing the list of controlled energy loads. The abstraction level must match to the requirements of energy services and applications [7], [10]. A well-designed abstraction transmits only the necessary service data to external domains, while shielding them from changes that occur within the customer domain. In this way, the ESI maintains consistent views of internal energy services, which is critical to the interoperability goal of smart grid.

Separation of concern: An ESI consists of several service interfaces, each of which is responsible for one functionality of energy services as shown in Fig. 1. The separation of concerns indicates that an interface is functionally independent of other interfaces so that any changes on the interface hardly affect the others. For instance, an interface for an event-based DR service must be distinguished from a bidding DR service. In the same way, sub-functions within an interface must be separated in an appropriate way.

Data representation: The ESI must provide energy data in a standardized format. Various customer equipment generates different formats of data that must be semantically understandable over smart grid. To cope with the heterogeneity, smart grid takes a Canonical Data Model (CDM) approach [11]. A data producer transforms its output to a standardized information model, and then a consumer transforms it back to own terminology. This approach is especially crucial in the customer facility, because there are still many legacy systems generating data in the proprietary format. The ESI must be able to handle and transform such data into a standard form. Standardization efforts[7] touching this issue include Facility Smart Grid Information Model (FSGIM), ISO/IEC 15045, Open Building Information eXchange (oBIX), Building Automation and Control networks (BACnet), OPC Unified Architecture, and ZigBee Smart Energy Profile (SEP).

Service interaction model: The ESI must support an efficient interaction model that determines how it communicates with external systems. Traditionally, a middleware has performed this task. In order to fulfill the interoperability requirement, recent standardization efforts consider web services. For instance, OpenADR specifies two types of bindings (SOAP and HTTP) and two types of message exchange patterns (PUSH and PULL). However, some energy services may require a constraint of high performance. In this case, we may consider an API based asynchronous messaging system.

[5] https://www.oasis-open.org/committees/emix/

[6] https://www.oasis-open.org/committees/energyinterop/

[7] FSGIM - http://spc201.ashraepcs.org/standards.html;
oBIX - http://www.obix.org; BACnet - http://www.bacnet.org;
OPC/UA - http://www.opcfoundation.org;
SEP - http://www.zigbee.org/Standards/ZigBeeSmartEnergy

Another consideration is a content-centric middleware that fully supports contextually-driven associations amongst energy resources [12].

Extensibility: The extensibility and flexibility are important requirements for the ESI, because the customer-engaged energy services evolve over time. New applications will be introduced, and corresponding service interfaces are newly added into the ESI. Some of them may reuse existing interfaces as sub-functions, and existing interfaces become deprecated or remain for backward compatibility. Implementation of internal functions also keeps changing with technological advancements. The design of the ESI must be extensible enough to allow such innovations of service interfaces, while ensuring the level of abstraction to be consistent.

C. Security

Cyber security is one of two cross-cutting issues in smart grid, and it is well known that the relative importance of Confidentiality, Integrity, and Availability (CIA) is reversed in smart grid [1]. This is because it is more fundamental that authorized messages must be delivered to the right place at the right time. Confidentiality is also critical, but its importance varies according to the provided services. Thus, this article omits discussion on it. This section discusses three important topics of availability, access control, and integrity.

Availability: Customer energy data must be consistently available for the successful provisioning of energy services. And such availability is primarily related to the network connection at the ESI. The connection can be threatened by external attacks such as Denial of Service (DoS). Or, data may be unavailable due to internal reasons such as the ESI failure. One interesting solution is to move the ESI to the Cloud that is generally assumed to be more accessible and reliable. An external system contacts the Cloud to access customers's energy resources, without knowing the location of the customer facility. The availability is also affected by traffic congestion in an emerging converged network, where the energy data is transmitted over the same IP network that has delivered residential data. A strong logical separation of traffic and proper QoS mechanisms can minimize the impact of non-critical traffic on the energy data [13].

Access control: An access control permits only authorized users to read customer data and to control energy resources. To this end, access control verifies the identity of an accessing user and grants him access permission to specific energy resources. An emerging issue of the access control to the resources is the increasing complexity of an access control rule. The customer domain will include a myriad of equipment, each of which is equipped with an embedded system. This enables external systems to access individual resources. Moreover, an access for data reading must be distinguished from that for resource control. The access control rule must take the new condition into account, which makes it complicated. In any scenario, the rule must be carefully designed, because any abuse of privilege makes the smart grid system unreliable potentially as well as violates privacy policy. An ideal access control realizes the principle of least privilege. Say, an ESP is permitted to perform a DLC to an energy load. The granted privilege must prohibit it from taking other actions and must be revoked immediately after its use.

Integrity: The ESI takes care of both message integrity and system integrity. It must be able to detect any tampered messages and to quarantine compromised interactions. A forged DR signal may misinform a large group of customers of an urgent DR event, which can lead to the failure of their reducing energy consumption during on-peak period. This can potentially cause a serious blackout. A proper usage of a message integrity code can mitigate the risk. System integrity represents the capability of preventing the cascade effect of a security failure. Say, a service interaction is compromised. This must not affect other interfaces and functionalities. A well-designed separation of concerns can mitigate the impact by removing an unforeseen chain of events among interfaces.

D. Architectural Consideration

The ESI is an interface specification for inter-domain communications, and its physical location is of less importance. This property introduces several new design issues as below.

The ESI is generally implemented on an EMCS, but its functionality can be realized by a coordination of several subsystems. Or, a customer facility may have multiple EMCSs, and thus multiple ESIs. For instance, the owner of a multi-tenant office building may want to distinguish sub-domains, each of which is managed through internal EMCSs. Moreover, solar panels on the roof and EV charging stations may be under control of another EMCS. There is still a central system running an ESI responsible for inter-domain communications, and the internal systems organize a network in a hierarchy.

The ESI can be deployed in a remote site. Especially, with the advancement of the Cloud technologies, there are many drivers to move the ESI to the Cloud. First, a local ESI may not be universally accessible, because many customer systems reside behind firewalls and Network Address Translations (NATs). This limits external systems' access to the customer data. Next, there are still many small facilities that cannot afford a local EMCS system for the ESI. Even if there are some, the systems may not store and process the increasing volume of energy data. Moreover, it is very hard to expect that all the security patches are applied to all the systems in a timely manner. Last, the Cloud enables efficient data accesses. Customer systems are located over geographically distributed areas, and some of their connections are quiet slow. When an ESP collects energy information from multiple customers and generates a new benchmark, it can do the task more quickly and efficiently once all the data is already on the Cloud.

IV. Implementation of the ESI

A. ESI Testbed

The integral part in developing and deploying the ESI is to realize two service interfaces in an interoperable and secure manner. In the grid service interface, the ESI implements the counterpart module of an external energy service that can understand the service's context as well as translate the context to the terminology that the customer domain uses. Implementing

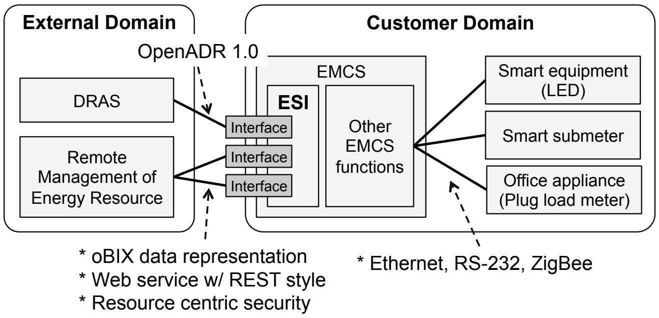
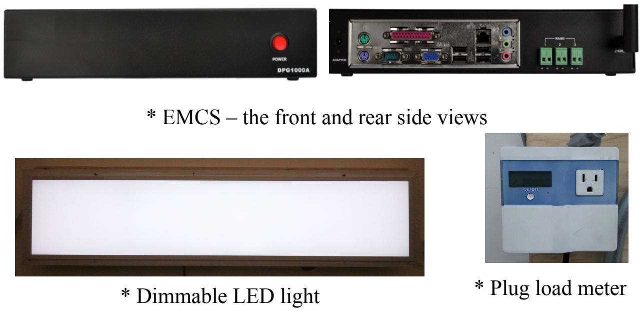

External Domain
Customer Domain
OpenADR 1.0
EMCS
DRAS
ESI
Interface
Interface
Interface
Other EMCS functions
Remote Management of Energy Resource
Smart equipment (LED)
Smart submeter
Office appliance (Plug load meter)
* oBIX data representation
* Web service w/ REST style
* Resource centric security
* Ethernet, RS-232, ZigBee

Fig. 2. ESI testbed consists of ESI, EMCS, energy loads, and energy services.


* EMCS – the front and rear side views

* Dimmable LED light

* Plug load meter

Fig. 3. ESI testbed - pictures of the EMCS and two types of energy loads.

the facility service interface is more challenging: it requires prior deployment of energy resources and implementation of EMCS in the customer facility. Then, the ESI models resources as service objects, abstracts their communication interfaces according to the separation of concern principle. To ensure interoperation, the objects and the interfaces must be implemented in a standardized way. A security mechanism being deployed must take into account the abstraction level.

Accommodating the issues above, this section develops and deploys an ESI testbed in a small customer facility where one ESI is implemented on an EMCS. In addition to the ESI functionalities, the EMCS performs data collection; management of energy resources; database management; and service data generation. We deploy three types of energy loads in our office. Office appliances such servers are plugged into plug load meters that measure energy usage and turn on/off the input power. A smart submeter is attached to each circuit within a circuit break panel. An LED light represents a smart equipment that can adjust own operations beyond a simple on/off status. The LED operates with 8 steps of brightness and temperature that affect its power consumption directly. The energy loads are connected to the EMCS via Ethernet, RS-232 and ZigBee. Fig. 2 illustrates the testbed system that includes the ESI, the EMCS, energy loads, and an external energy service. Fig. 3 shows pictures of our testbed - the front and rear side views of the implemented EMCS; the dimmable LED light; and the plug load meter.

B. *Open Automated Demand Response*

The ESI implements an OpenADR client as a grid service interface that realizes an automated DR service [8]. OpenADR is expected to be the first grid service based on real-time power price that will appear in the near future. Thus, examining it helps us understand how a customer domain interoperates with external energy services. To illustrate the interoperation, we implements and deploys a DRAS server system in our testbed [14]. The OpenADR client accepts and interprets DR signals from the DRAS. At the moment, the client module handles event programs only, not the bidding program.

A DR event initiates when the DRAS issues a message, *EventInfo*, and then, generates an *EventState* signal[8] that represents the DR event. The DRAS supports connections from both smart and simple clients. The smart client is capable of dealing with the *EventInfo* information within the *EventState* signal. Included in *SmartClientDREventData* entity, it contains event details. For instance, the *eventInfoTypeID* identifies the type of event information and takes one of values of PRICE_ ABSOLUTE, PRICE_RELATIVE, LOAD_AMOUNT, etc. For the simple OpenADR client, the DRAS translates the *EventInfo* information into a simpler form, named *SimpleClientEventData*. In the entity, two variables describes the event state. The *EventStatus* element indicates the temporal state of the event (FAR, NEAR, or ACTIVE), whereas *OperationModeValue* denotes the operational state of the energy loads in the event (NORMAL, MODERATE, or HIGH). Our ESI implements both smart and simple clients and periodically "pulls" the *EventState* message from the DRAS. This PULL mode is typically used, since the OpenADR client has more control over the communications, e.g., firewalls and private networks using NAT. A PUSH mode can be considered in scenarios where very low latency is required.

To address the security issue of OpenADR, in particular the message integrity, we implement a Message Authentication Code (MAC) on top of the existing OpenADR system. Following the NISTIR 7628 guideline [15], our testbed takes a hash-based MAC (HMAC) with SHA-256.

C. *Remote Management of Customer Energy Resource*

To show the ESI serving as a service provider, we consider a scenario of "remote management of customer energy resources". It includes various types of services - an external user is able to obtain energy usage data of individual energy loads; retrieve historical data; and control the loads directly. The ESI leverages two technologies of oBIX and web services to implement the facility service interface supporting the services above. This subsection describes how they can realize the services while fulfilling the design issues discussed earlier.

1) Open building information exchange: The ESI takes the oBIX specification as a standard data representation because of several outstanding advantages.

First, it supports an Object-Oriented (OO) design pattern with low-level abstraction. This helps us model information of electrical and mechanical systems in a facility as an object. Like the OO programming, each object is modeled by a set of *value* objects like "int" and "str" and a set of *op* objects that defines an operation with input and output objects. The object

[8] All the data in OpenADR is represented as XML form, and their schemas are available at http://openadr.lbl.gov/src/1.

model allows inheritance so as to model complex energy data by means of a contract mechanism. Realized by *is* object, it establishes the conventional "is a" relationship with various overriding rules. In this way, an object cannot only represent a physical unit directly, but also a particular functionality as a collection of sub-objects. This capability allows us to abstract service interfaces in a more flexible manner according to the service requirements. Next, oBIX exploits XML to express its underlying object model, which maximizes the interoperation property of standard data format. To this end, it specifies four syntaxes - each object type maps to one XML element; an object's children are mapped as children elements; the XML element name maps to the predefined primitive object type; and every other objects are expressed as XML attributes. Last, the OO design of oBIX is also beneficial to the design issues of extensibility and separation of concern. The reusability of the object model helps make the ESI more flexible to accomodate future innovation of the services. The inheritance property simplifies the development of complex energy services, and each service and interface implementation can be easily separated from others.

Leveraging the advantages of oBIX, we implement data and service models for our scenario. The box below illustrates a *History* object, a historical archive of an energy usage data over time. The *is* attribute in the *obj* element indicates that it is extended from a standard oBIX object *obix:History*. The example also shows that the *query* operation to read history records takes an argument whose object type is defined as *psxml:HistoryFilterEx* and returns history records in the object format of *obix:HistoryQueryOut*.

```
<obj href="http://myESI/History/" is="obix:History">
  <int name="count" val="541"/>
  <abstime name="start" val="2012-01-02T00:00:00
      .000-08:00"/>
  <abstime name="end" val="2013-01-06T00:00:00
      .000-08:00"/>
  <op name="query" href="query" in="
      psxml:HistoryFilterEx" out="
      obix:HistoryQueryOut"/>
  <feed name="feed" href="feed" in="
      obix:HistoryFilter" of="obix:HistoryRecord"/>
  <op name="rollup" href="rollup" in="
      obix:HistoryRollupIn" out="
      obix:HistoryRollupOut"/>
</obj>
```

In a similar way, the box below represents a *power* object that contains information of power draw data, power factor, and communication quality regarding plug1 (a plug load meter) connected to the EMCS via ZigBee. As shown in the *href* attribute, the object is a sub object of a *plug1* and *zigbee*.

```
<obj href="http://myESI/Points/zigbee/plug1/power/">
  <real name="voltage" val="120.2" unit="obix:units/
      volt"/>
  <real name="current" val="1.21" unit="obix:units/
      ampere"/>
  <real name="power" val="160.03" unit="obix:units/
      watt"/>
  <real name="powerFactor" val="0.996" unit="
      obix:units/power_factor"/>
  <int name="rssi" val="-72" unit="obix:units/
      decibel"/>
</obj>
```

2) Web service: Our data representation model enables to design data in an object oriented way, and then each service is mapped into an object having both values and method signatures. The ESI takes Web Service (WS) as a service interaction model, because it accommodates this property as well as exposes service interfaces in an interoperable manner. In terms of WS, each energy object is accessed via a URI, and its data is passed around as an oBIX document. To realize this access activity, we take HTTP binding and implement the ESI in the REpresentational State Transfer (REST) style. Supporting a resource centric access, instead of method centric one, REST utilizes a small set of verbs to transfer an object's state via XML [16]. Its resource-oriented access mechanism best fits to our data model. Similarly, The RESTful paradigm has been used in recent literatures of smart grid [17].

More specifically, three request types in oBIX are mapped into HTTP methods - Read, Write, and Invoke. Read request uses GET for any object having *href* attribute and returns information of an object as an oBIX document. Write is targeting at any object whose *writable* attribute is set to true, and is implemented with PUT. Invoke supports operations of any object by using the POST method. An oBIX document is passed to a server as an input in both Write and Invoke. Below oBIX document represents a smart plug object, "plug1". The root element indicates its access URI, and the *ref* tells the link to the associated sub-object, "power". The document also shows that the load is controllable via two ways: Write and Invoke. To turn it on or off, an external system sends directly a PUT request targeting at the *connectLoad* object within the same URI or sends a POST request to the hyperlinked URI targeting at the operation object, *controlLoad*.

```
<obj href="http://myESI/Points/zigbee/plug1/">
  <str name="deviceName" val="BSPE12SOYZM43001"/>
  <str name="version" val="C2V5.57"/>
  <bool name="connectLoad" writable="true" val="true
      "/>
  <op name="controlLoad" href="control" in="
      obix:WritePointIn" out="obix:Point"/>
  <ref name="power" href="power"/>
</obj>
```

D. Resource Centric Security

A security mechanism satisfies the security requirements discussed earlier. To address the integrity issue, we implement HMAC with SHA-256. We address the availability issue with a cloud technology in the following subsection. Access control is more challenging, because our interaction model, WS, exposes the values and operations of each energy resource via three different operations: Read, Write, and Invoke. This property is interpreted as "fine granularity of data access and load controls". Fine granularity introduces a new challenge, because different access actions induce different operation consequences and indicate different levels of privacy violation. Say, you might be okay that ESPs read the energy usage of your air conditioning system, but you do not allow them to turn it off on a hot summer day.

To resolve the fine granularity problem, authors in [18], [19] proposed a new access control concept that performs

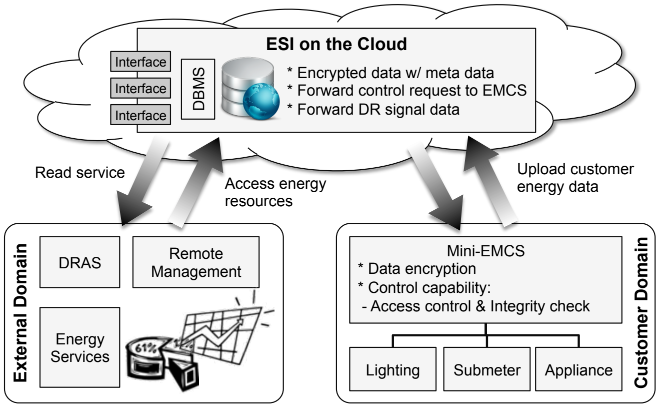


Fig. 4. Moving the ESI to the Cloud. The energy data is stored on the Cloud, while the customer still has full control over data encryption and access control. The external user contacts only the Cloud to access the customer energy resources.

authorization on the action level (e.g., Read, Write, and Invoke). We adopt Resource Centric Security (RCSec) [18] that implements both access control and encryption in a distributed manner. To address the fine granularity issue, RCSec leverages the concept of a filesystem Access Control List (ACL). That is, each object maps to an attribute with three-digit privilege level. For instance, the object “plug1” in the previous example is related to an attribute “plug1=111”. The first digit indicates permission of Read, and the following two digits indicate the rights of Write and Invoke, respectively. Each user maintains his own set of attributes in his private key. When the user tries to access the object with his key having an attribute of “plug1=100”, he is permitted to read energy usage data, but not to change any values or control the energy load. In this way, RCSec performs authorization based on what the end user has, and this does not require the ESI to maintain any information for access control such as user id-password pairs and user-privilege mapping rules.

E. *Separating Read Operation from Control Operation*

As justified in Section III-D, deploying the ESI on the Cloud is one convincing option for the real world deployment. This is also expected to help resolve the availability issue in the security context by utilizing more reliable and secured computing resource on the Cloud. Despite these benefits, however, we must solve the privacy problem of the Cloud technology, because people do not want the Cloud to read their private data or to control internal energy resources.

Our fundamental approach to address the issue is to extend RCSec so that it distinguishes the Read operation from the Control operation. Then, we move the Read service to the Cloud. To this end, we implement two service interfaces of the ESI and a Database on a Cloud system hosted by Amazon, while control and security functionalities remain in a lightweight mini-EMCS. Fig. 4 represents the new system architecture. More specifically, all the energy data is first encrypted at the mini-EMCS and then transmitted to the Cloud, and the Database in the Cloud stores encrypted energy data with meta information. So, external users retrieve historical data by contacting service interfaces on the Cloud directly while the Cloud does not access the plaintext of the energy data. The Cloud, on the other hand, forwards the users requests for control operations to the mini-EMCS that performs access control and verifies message integrity.

Separating and moving the Read operation to the Cloud is feasible mainly because RCSec is implemented based on Attribute-Based Encryption (ABE) algorithm [20] that realizes a secret sharing scheme using a pairing-based cryptography. That is, the mini-EMCS encrypts energy data using a list of attributes, not using specific user’ public key or specific symmetric key. Since RCSec decouples encryption from users’ identity, the Cloud does not authenticate and authorize the accessing users. Instead, any users can access encrypted energy data via the Cloud, but only qualified ones having the matched set of attributes (protected via the encryption algorithm) can decrypt the ciphertext.

V. Experiment and Discussion

A. *Automated DR Service*

In this automated DR experiment, the DRAS server generates a DR event of a Real Time Pricing (RTP) program. The box below shows the event signal - the *SmartClientDREventData* entity in the *EventState* message. The event starts at 1pm and lasts until 4pm. During the event, the unit power price changes every hour; it becomes 2 times, 3 times, and 2 times more expensive than a normal price. The event data is generated one hour before the event - an hour-ahead DR program.

```
<p:drEventData>
  <p:notificationTime>2012-08-20T12:00:00.000-07:00<
      /p:notificationTime>
  <p:startTime>2012-08-20T13:00:00.000-07:00</
      p:startTime>
  <p:endTime>2012-08-20T16:00:00.000-07:00</
      p:endTime>
  <p:eventInfoInstances>
    <p:eventInfoTypeID>PRICE_MULTIPLE</
        p:eventInfoTypeID>
    <p:eventInfoName>price</p:eventInfoName>
    <p:eventInfoValues>
      <p:value>2.0</p:value>
      <p:timeOffset>0</p:timeOffset>
    </p:eventInfoValues>
    <p:eventInfoValues>
      <p:value>3.0</p:value>
      <p:timeOffset>3600</p:timeOffset>
    </p:eventInfoValues>
    <p:eventInfoValues>
      <p:value>2.0</p:value>
      <p:timeOffset>7200</p:timeOffset>
    </p:eventInfoValues>
  </p:eventInfoInstances>
</p:drEventData>
```

The ESI pulls the *EventState* message from the DRAS every 15 minutes. Once the ESI notices that the price goes up, it performs a predefined DR strategy. In this experiment, we register one LED light to our strategy so that the price change is seen through the brightness of the LED. As the price goes up, the LED gets dimmed proportionally. Since the power draw of the LED is proportional to the brightness level, we can observe the change of energy usage during the DR event as shown in Fig. 5.

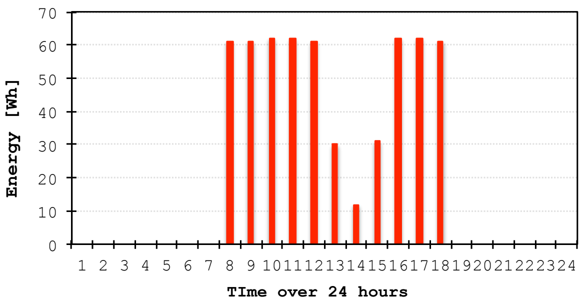


Fig. 5. DR event affects the operation of LED. The bars indicates energy usage of the LED over 24 hours. The operation of the LED is also scheduled so that it turns on at 8am and off at 7pm every week days.

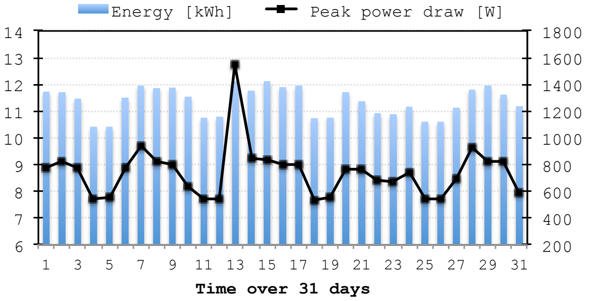


Fig. 6. Retrieve aggregated energy usage over 31 days.

We note that the experimental results demonstrate an automated DR, but not address the invisibility that the intelligent DR requires (see Section II). The invisibility still remains a big research challenge that has not been studied much yet. One plausible approach is to prioritize the energy loads so that lower-prioritized loads are shed first upon DR events [21]. The prioritization can be automated by the combination of many factors - residents' load-usage profile, time, space, and environmental context.

B. Providing Energy Management Service

Next experiment demonstrates that the ESI provides customer energy services. A user retrieves a historical energy usage data from the ESI. To this end, he prepares an request message as shown in the box below and contacts the service object via a URI to trigger the Invoke operation. The URI in the box implies that the service provides an aggregated energy usage information. The user would retrieve daily usage data of smart submeters and plug load meters for a month.

```
http://myESI/History/aggregated/rollup/

<obj xmlns="http://obix.org/ns/schema/1.0">
  <reltime name="interval" val="P1D"/>   // daily
  <int name="limit" val=""/>
  <abstime name="start" val="2012-08-01T00:00:00"/>
  <abstime name="end" val="2012-08-31T23:00:00"/>
  <str name="orderby" val="ttime"/>
  <bool name="reverse" val="false"/>
</obj>
```

Data size [KB]	10	200	400	600	800	1000
Encrypt [ms]	1383	1369	1372	1380	1389	1399
Decrypt [ms]	225	224	225	229	230	240

TABLE I
DATA ENCRYPTION AND DECRYPTION TIME [MS]

The retrieved energy data is drawn in Fig. 6. The bars show that energy consumption decreases on weekends, but the difference is slight. This is mainly attributed to the type of deployed energy loads. In our testbed, several servers and switches running 24 hours are plugged into the plug load meters. And few such loads that is affected by human activities are deployed. We note that a preliminary analysis on the types of energy loads and their operating characteristics is fundamental to establishing the right DR strategy. The figure also draw a curve of peak power draw (maximum instantaneous power consumption), ranging from 500W to 1600W. The maximum peak was hit on Aug. 13 on which we ran another experiment with power-hungry loads such as a dryer and a refrigerator. Regardless of the high value, however, the peak draw hardly affects the energy usage on that day, because the experiment lasted short. However, this high peak matters in practice, since most utility companies charge customers based on their peak values. Finding the energy usage patterns and encouraging to change them to save the energy cost is the primary goal of the energy management service. In the future, the energy data can be further analyzed for advanced energy services. Recently, data mining technology and network optimization algorithms have been leveraged to analyze the time series of energy usage data, which then performs energy efficiency benchmark [22], fault detection and diagnosis, and anomaly detection [23].

C. Running the ESI on the Cloud

In this experiment, we measure the time overhead of RCSec, once it is applied to our ESI testbed on the Cloud architecture. The mini-EMCS encrypts data using 10 attributes on average. The experiment changes the data size and measure the encryption time. The encrypted data is then uploaded to the Cloud that provides the Read service to end users. We also measure the decryption time at a user. We implement both the mini-EMCS and the user at a conventional laptop computer, running with 2.2 GHz Intel Core 2 Duo processor and 2 GB memory, that performs encryption and decryption, respectively. The box below illustrates a sample of energy data that the Cloud provides. It shows that 5 attributes are used in the header and encrypted data is attached in the data element.

```
<obj href="http://Cloud/Points/zigbee/plug1/power">
  <obj name="header">
    <str name="attributes">(ucla_esi AND expire
        >=1362978960563) AND (power>=110 OR (
        BSPE12SOYZM43001>=100 AND power>=100))</str>
  </obj>
  <obj name="data">
      FJ131tjE3yDjXdIgNtH715DVuBRUXalfkredN9OmxNyN48
      ...</obj>
</obj>
```

Table I shows that the data size hardly affects the performance. It also indicates that the encryption overhead is comparatively greater than the decryption. This is mainly attributed to difference of mathematical complexity within the encryption and decryption. The encryption algorithm constructs an elliptic curve and performs bilinear maps (pair-

Num. attribute	1	5	10	15	20
Overhead [ms]	900	1648	2357	3097	3823

TABLE II
OVERHEAD OF AUTHORIZATION WITH VARYING NUMBERS OF ATTRIBUTES.

ing) based on the curve's elements. For each attribute, it constructs two bilinear group elements, which requires two exponentiations. The pairings and exponentiations dominate the encryption overhead. In the decryption algorithm, on the other hand, bilinear map operations are partially replaced by exponentiations, which is more lightweight, in a recursive manner. This optimization reduces the processing overhead. We refer [24] for detailed analysis. The observation that the decryption overhead is comparatively low benefits the Cloud architecture. The customer domain encrypts data in advance, and users do not experience it. Thus, the decryption overhead only contributes to the overall service performance. In this sense, the Cloud architecture considerably compensates the overhead of the security mechanism.

Next experiment measures the overhead of the authorization protocol for the control operation that travels to the EMCS via the Cloud. As more customer equipments are connected to the EMCS, the authorization rule would also get complex. RCSec handles such complexity by applying varying numbers of attributes in the authorization protocol. This experiment, therefore, changes the number of attributes and measures the protocol latency. The result, shown in Table II, indicates that the latency overhead is non-trivial in numbers. In particular, the processing overhead overwhelms the communication latency (89% vs. 11% on average, not shown in the table). However, when considering that the control operation occurs infrequently and does not require real time performance, the overhead is acceptable. Moreover, the overhead is compensated by benefits coming from the decentralization property of the protocol. In the authorization protocol, an end user does not register to the EMCS. Instead, he obtains his own private key from a certificate authority. Complex authorization rules are embedded into exchanged messages, instead of being managed by the EMCS. Then, authorization is carried out on-the-fly using the private key. This property makes the EMCS much simpler and more lightweight. Since the protocol completely separates the EMCS from the user, it easily scales well in a distributed environment such as smart grid.

VI. CONCLUSION

The ESI plays the most significant role in the customer domain for smart grid interoperation. In this paper, we looked into four categories of ESI design issues to support interoperable customer energy services. Two of them addressed functional requirements that the ESI supports as a service prosumer. The other two examine quality requirements that the ESI supports for better energy service - security and ESI system architecture. To verify the issues, we built and deployed two ESI testbeds that also showed how the design issues are implemented in a real world. Through experiments with a couple of energy service scenarios, we have demonstrated the service interoperation and evaluated the performance.

REFERENCES

[1] "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0," Feb. 2012.

[2] "The U.S. Energy Information Administration, Annual Energy Outlook," 2012.

[3] "EIS Alliance, Customer Domain Use Cases, v3.01," 2010.

[4] "EIS Alliance, Customer Domain Energy Services Interface (ESI) Requirements, V3.01," 2010.

[5] "UCA International user group (UCAIug) Home Area Network (HAN) System Requirements Specification, v2.0," http://osgug.ucaiug.org/sgsystems/openhan/default.aspx.

[6] D. V. Dollen, "Electric Power Research Institute (EPRI) Report to NIST on the Smart Grid Interoperability Standards Roadmap," 2009.

[7] D. Hardin, "Customer energy services interface white paper," in *Grid-interop Forum*, Dec. 2011.

[8] Piette, M. Ann, G. Ghatikar, S. Kiliccote, E. Koch, D. Hennage, P. Palensky, and C. McParland, "Open Automated Demand Response Communications Specification v1.0," *California Energy Commission - PIER Program*, vol. CEC-500-2009-063, 2009.

[9] "WXXM 1.1 Primer, Federal Aviation Administration / European Organization for the Safety of Air Navigation," 2010.

[10] "REQ.21-Energy Services Provider Interface (ESPI), North American Energy Standards Board (NAESB)," http://www.naesb.org/ESPI_Standards.asp.

[11] "Semantic Model Working Party, Smart Grid Architecture Committee," http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/SGIPSemanticModelSGAC.

[12] Y.-J. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, "SeDAX: A Scalable, Resilient, and Secure Platform for Smart Grid Communications," *IEEE Journal on Selected Areas in Communications*, no. 30(6), pp. 1119 – 1136, Aug. 2012.

[13] A. Wright, P. Kalv, and R. Sibery, "Interoperability and security for converged smart grid networks," in *Grid-Interop Forum*, Dec. 2010.

[14] "OpenADR Open Source Toolkit: Developing Open Source Software for the Smart Grid, LBNL-5064E," Lawrence Berkeley National Laboratory, Tech. Rep., 2011.

[15] Smart Grid Interoperability Panel - Cyber Security Working Group, "Introduction to NISTIR 7628 Guidelines for Smart Grid Cyber Security," Sep. 2010.

[16] R. Fielding and R. Taylor, "Principled Design of the Modern Web Architecture," *ACM Transactions on Internet Technology*, vol. 2, no. 2, pp. 115–150, 2002.

[17] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler, "sMAP-a Simple Measurement and Actuation Profile for Physical Information," in *ACM SenSys*, Nov. 2010.

[18] E.-K. Lee, R. Gadh, and M. Gerla, "Resource Centric Security to Protect Customer Energy Information in the Smart Grid," in *IEEE Smart Grid Communications*, Taiwan, Nov. 2012.

[19] "Machine-to-Machine communications (M2M), Functional architecture, ETSI TS 102 690 v.1.1.1," 2011.

[20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM CCS*, Alexandria, USA, Oct. 2006.

[21] Y. Agarwal, B. Balaji, S. Dutta, R. Gupta, and T. Weng, "Managing Plug-Loads for Demand Response within Buildings," in *ACM BuildSys*, 2011.

[22] A. Littman, G. Lyon, A. Shah, and J. Vogler, "Exploring Advanced Metering Infrastructure Deployment for Commercial and Industrial Sites," in *ASME Energy Sustainability and Fuel Cell*, 2012.

[23] G. Bellala, M. Marwah, M. Arlitt, G. Lyon, and C. E. Bash, "Towards an Understanding of Campus-scale Power Consumption," in *ACM BuildSys*, 2011.

[24] M. Pirretti, P. Traynor, P. Mcdaniel, and B. Waters, "Secure Attribute-Based Systems," *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.