Accepted to IEEE INTERNET COMPUTING, November/December 2013.

Fine-Grained Access to Smart Building Energy Resources

Eun-Kyu Lee, Peter Chu, and Rajit Gadh

Abstract

Smart city ecosystems provide fundamental resource services such as water and electricity through an intelligent management system. Here, the authors focus on energy resources for smart grids and smart buildings, investigating how a smart grid infrastructure accesses the energy resources in a smart building to obtain or control data. The energy service interface (ESI), which plays a communication gateway role at the boundary of a customer building, hasnt received much study.The authors examine its design principles for enabling fine- grained access, implement and deploy an ESI prototype, and illustrate interactive energy services between the grid infrastructure and smart buildings.

A smart grid aims at balancing power consumption with power generation to make a power infrastructure sustainable without blackouts. To this end, tiny sensor systems are embedded into every energy resource, ranging from power plants to transmission lines to home appliances. These systems measure resources status and intercommunicate with other systems to detect potential imbalances in advance and respond quickly. Their seamless interoperation is thus key.

The US National Institute of Standards and Technology (NIST) presents a conceptual model comprising seven domains to facilitate such interoperability [1]. Most electricity is consumed in the customer domain, which includes industrial, commercial, and residential sectors. This domain has two gateway actors that exchange customers resource data with external domains: the *utility meter* and the *energy service interface (ESI)*. Utility meters reside within the customer facility but are owned and managed by a utility company. They directly communicate with the utilitys meter data management system (MDMS). The meter measures and communicates an aggregated energy usage for customer billing. Its functionally limited and cost sensitive, and is therefore barely upgradeable. The ESI, on the other hand, is designed as a communication interface to serve demand response (DR) signaling. Considering the utility meters functional limitations, the ESI is expected to become the primary transmitter of advanced energy service data streams, and thus the most important entity for smart grid interoperability in the customer domain. Few researchers have investigated more than its conceptual idea. To develop and deploy the ESI in a smart grid, we must have a clear understanding of its requirements and corresponding design issues. Here, we discuss practical issues and challenges for realizing the ESI.

Energy Service Interface

Few researchers have investigated the ESI, so it isnt well understood. We first review its definition from previous literature and investigate several energy services that we can realize on top of it.

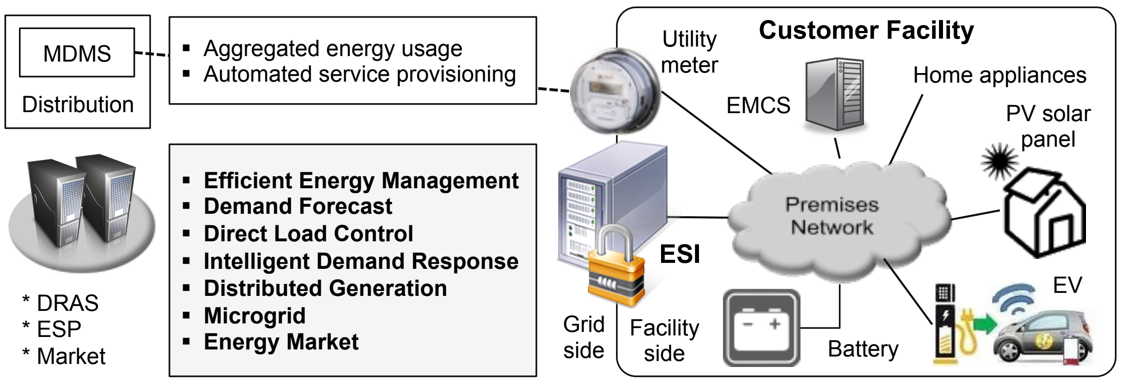
Definition and Characteristics

A couple of public reports have conceptually defined the ESI [1], [2] and from them we observe three common characteristics.

First, the ESI represents the boundary of the customer domain, logically distinguishing the domains operation from that of the outside world. To this end, the ESI consists of two sides: *facility* and *grid*. The facility side communicates directly with internal energy resources (home appliances, solar panels, batteries, and so on) or is connected to data storage and processing modules in an energy management and control system (EMCS). The grid side interacts with external systems such as energy service providers (ESPs) and utilities. As a bridge placed at the domain boundary, the ESI supports inter-domain communications.

Second, the ESI represents a service interface, acting as a platform to provide energy services to both the facility and grid sides. It exposes data formats and a set of energy services that the customer domain provides to external

• *E.-K. Lee, P. Chu, and R. Gadh are with University of California, Los Angeles, Los Angeles, CA 90095.*
E-mail: eklee@cs.ucla.edu, peterchu@ucla.edu, and gadh@ucla.edu



DG: Distributed Generation	**EMCS**: Energy Management and Control System	**MDMS**: Meter Data Management System
DLC: Direct Load Control	**ESI**: Energy Service Interface	**PV**: Photovoltaic
DR: Demand Response	**ESP**: Energy Service Provider	
DRAS: Demand Response Automation Server	**EV**: Electric Vehicle	

Fig. 1: A customer domain system including the energy service interface (ESI), and customer services for smart grid interoperation. The energy management and control system (EMCS) includes a building automation system (control) and an energy management system (energy measurement).

domains - for example, 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 through the *grid service interface*.

Finally, the ESI doesnt represent a physical device. As a software component, it can be implemented on various physical systems such as an EMCS or energy gateway, or it can be realized as a stand-alone server system. Generally, the ESI is implemented in the customers EMCS, which communicates with external systems via the Internet.

Fig. 1 shows the overall architecture of a customer domain system, including an ESI.

Energy Services over the ESI

The ESI exchanges energy data across domains to form existing and emerging energy services that contribute to smart grid interoperation. We envision and examine six general categories of customer energy services.

In an *efficient energy management* service, a submetering system in a customer facility tracks usage for individual energy resources. An EMCS lets the customer read the usage breakdown and even control the resources remotely. An ESP might analyze the usage pattern and guide customers to consume electricity more efficiently. For instance, given dynamic power pricing information delivered from the ESP, the EMCS would schedule the operations of energy resources in a fine-grained manner to minimize the total electricity cost.

A *demand forecast* service gives a utility company information about expected customer energy demand (consumption) so that it can more accurately estimate the amount of power supply required in the future. To this end, the utility delivers power pricing information and weather forecasts to customers, who already understand their energy usage patterns. Then, customers examine future local energy needs.

A *direct load control (DLC)* service lets external entities (such as an ESP) control internal energy resources remotely. Currently, utility companies provide primitive services that target specific energy loads (resources), such as electric water heaters, and thus control the loads power supply during occasional high- demand periods. In the future, ESPs will control various types of energy resources for different purposes. For instance, an ESP might temporarily stop charging an electric vehicle (EV) in a customers garage when the power price goes beyond a contracted threshold value.

Energy suppliers sometimes confront a shortage in generating capacity during peak-demand periods. Instead of constructing additional power plants to cope with this shortage, an *intelligent DR* service decreases energy demand by sending DR signals to customers. On receiving the signal, the customer reduces energy consumption. This concept isnt new; utility operators currently call contracted customers who then manually stop their building operations in return for financial incentives. A new challenge for the intelligent DR service is how to automate load curtailment while minimizing its interference with ordinary building operations.

A *distributed energy resource (DER)* feeds power generation into the distribution grid directly rather than through the transmission infrastructure. DER can reside in customer facilities such as wind-powered generators or photovoltaic (PV) arrays. DER can be leveraged for intelligent DR services, compensating for some or all energy needs that must be reduced according to the DR services terms. The future smart grid could see a new business model -

leasing the equipment of a DER and power storage. An equipment service provider would constantly monitor the health of the leased energy assets and try to control them directly.

Finally, in the *energy market*, customers can trade energy and emissions. With the increasing sophistication of storage and generation technologies at customer facilities, customers actively participate in the wholesale and retail energy market. They can choose to buy power from one or more energy suppliers, or generate power for sale in the market. In other words, they buy or sell electricity in more flexible ways as “prosumers.” Similarly, customers participate in an emission market with their onsite renewable DERs.

ESI Design Principles

As we can see, energy services range from simply accessing energy resources to electricity trading, which leads to a variety of potential ESI implementations. Such diversity, however, makes it difficult to derive a common reference model for developing interoperable ESIs. Moreover, the existing definition and characteristics are still only conceptual. Given these two extremes, we determine five elementary design principles that we must consider to realize the ESI.

Interface Abstraction

The ESI defines communication interfaces to support energy services in an interoperable manner. To this end, it must consider interface abstraction, which determines the appropriate level of internal details that the interface exposes to external domains. Low abstraction levels expose internal business logics with more detail, whereas high levels expose less detail. DLC and event-based DR show two extreme examples with respect to control capability. With DLC, the ESI receives command messages that control individual energy resources, thus realizing the lowest level of abstraction. With event-based DR, the ESI receives a DR event signal from a DR automation server (DRAS), leading the customer facility to reduce power consumption without disclosing the list of connected energy resources. A well-designed abstraction transmits only the necessary service data to external domains while hiding the changes that occur within the customer domain.

Data Representation and Service Interaction Model

For seamless energy services across domains, the ESI must interact with external systems in an interoperable way. To this end, it must first represent energy resource data in a standard format so that other systems easily understand the contexts. Several standard organizations have defined such data models, such as a Boolean model to denote an energy devices on/off status. Examples include the Facility Smart Grid Information Model (FSGIM), the Home Electronics System Gateway (HES; ISO/IEC 15045), Building Automation and Control Networks (BACnet; ISO 16484-5), the Open Building Information eXchange (oBIX), OLE for Process Control Unified Architecture (OPC/UA), and the ZigBee Smart Energy Profile (SEP). We expect that multiple standards will coexist to accommodate various types of energy information.

Second, the ESI must support efficient inter-action models that determine how it communicates with external systems. Traditionally, middleware has performed this task; recent standardization efforts extensively leverage a Web service approach that fulfills the interoperability requirement. In this way, such efforts develop communication specifications for potential energy services. Examples include Energy Interoperation (EI), Open Automated Demand Response (OpenADR), the Energy Services Provider Interface (ESPI), the Common Information Model (IEC CIM) family, and the HES Application Model (ISO/IEC 15067-3).

Fine-Grained Access to Energy Resources

The aforementioned standardization efforts demonstrate three interesting characteristics. First, XML is used for data representation. oBIX and BACnet use it as a data format, and OpenADR and EI define their specifications in the XML schema. Second, the definition of customers’ energy resources follows an object- oriented (OO) design pattern. Given predefined primitive objects such as “int” and “list,” an object in oBIX is modeled with data types (values) and operations (method signatures). Finally, the Web service approach is exploited for inter-domain transportation of customer data. OpenADR specifies two types of bindings (SOAP and REST) and two types of message exchange patterns (push and pull). Two common goals of these three characteristics are interoperability and automation, which we can achieve as follows. Each energy resource distinguishes its values and operations, and the Web service exposes them by implementing three access methods to the resource - *read*, *write*, and *invoke*. We can interpret these methods as “fine granularity of data access and resource controls” - that is, they have a low level of abstraction. For instance, a user accesses a read interface to read the energy usage of an air conditioning system, but might turn off the system via an invoke service interface.

Security

The relative importance of confidentiality, integrity, and availability (CIA) is reversed with the smart grid - that is, it's more fundamental that authorized messages be delivered to the right place at the right time. The ESI must implement an access control policy to protect valuable energy resources. Access control permits only authorized users to read customer data or control resources. An ideal access control realizes the principle of least privilege - that is, it permits a user to access only the information and computing resources that are necessary to complete a given task. For example, say that an ESP is permitted to perform DLC on an energy resource; the granted privilege must prohibit it from taking other actions and must be revoked immediately after use.

Availability primarily relates to the network connection at the ESI, which can undergo external attacks or internal system failure. Such threats prevent customer energy data from being consistently available and eventually lead to unreliable provisioning of energy services. One interesting solution is to move the ESI and data storage to the cloud, which is generally more reliable in terms of data access and system reliability. The ESI ensures message integrity. It must be able to detect whether messages have been tampered with and quarantine compromised services. A forged DR signal might misinform a large group of customers about an urgent DR event, which can lead them to fail to reduce energy consumption during an on-peak period. Confidentiality is also critical, but its importance varies according to the provided services.

Architectural Considerations

The ESI is a logically defined specification for a communication interface; it plays a gateway role across domains, and its physical location is less important. This property introduces several new design issues. A large customer facility might have multiple EMCSs that form a network. In a campus scenario, for instance, an EMCS manages the operations of an onsite DER installed in an independent building, while another EMCS controls a classroom building with various energy loads in the rooms and solar panels on its roof. ESIs implemented within each EMCS enable automated communications and interactions among energy resources across those EMCSs. A central EMCS is responsible for interdomain communications.

ESIs can also be located remotely; with advancements in cloud technologies, many drivers exist for moving the ESI or EMCS to the cloud. First, a local ESI might not be universally accessible because many customer systems reside behind firewalls. This limits external systems' access to customer data. Next, many small facilities still can't afford a local EMCS for the ESI. Even with one, the system might not be able to store and process the increasing volume of energy data. Finally, we can't expect that required security patches will be applied to all systems in a timely manner.

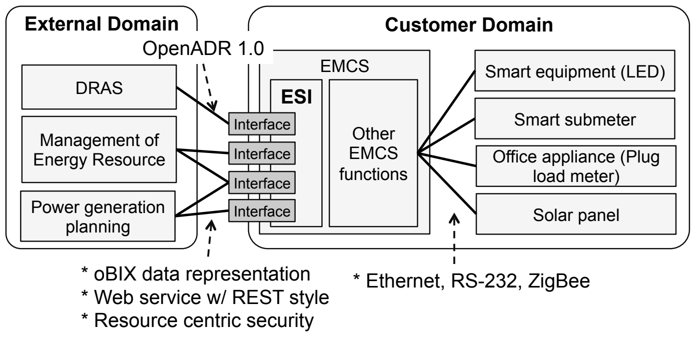
Prototype Implementation

We developed an ESI testbed for a small customer facility, in which we implemented a single ESI on an EMCS. In addition to supporting ESI functionality, the EMCS performs data collection, energy resource management, database management, and service data generation. We deployed three types of energy resources in our laboratory. We plugged office equipment such as monitors into a plug load meter that measures energy usage and turns the input power on and off. We instrumented a smart submeter on each circuit in a circuit break panel. An LED light represents smart equipment that can adjust its own operations beyond simple on/off status. The LED operates with eight brightness and temperature levels that directly affect its power consumption level. The energy loads connect to the EMCS via Ethernet, RS-232, and ZigBee. In addition to energy loads, a PV solar panel is connected to the EMCS. We're currently installing a new one on the roof; in the meantime, we've developed a solar panel simulation that uses the same hardware specifications as installing the real panel. A core part of this simulation panel is its prediction algorithm [4]. As an external energy service, our testbed realizes the DR service using OpenADR 1.0 [3]. Exploiting an open source toolkit [5], we deploy a DRAS. The ESI implements an OpenADR client that accepts event-based DR signals from the DRAS. Fig. 2a illustrates the testbed system that contains the ESI, the EMCS, energy resources, and an external energy service.

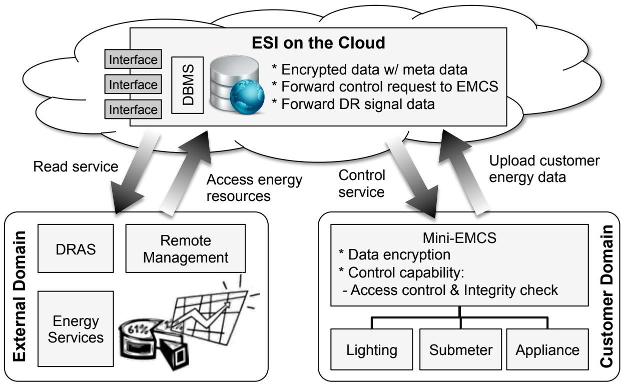
Remotely Managing Customer Energy Resources

To realize an ESI acting as a service provider, we consider a scenario that remotely manages customer energy resources. It includes various sub-services that let an external user obtain energy usage information on individual energy resources, retrieve historical data, and control the resources directly.

Open Building Information eXchange. For standard data representation, we leverage the oBIX specification (www.obix.org), which helps us model energy resource information as an object. Each object is modeled using a set of `value` objects and a set of `op` (operations) objects. The object model allows inheritance to model complex energy data via a contract mechanism. Realized by the `is` object, inheritance establishes a conventional "is a" relationship


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

(a) The first testbed system contains the ESI, the energy management and control system (EMCS), energy resources, and energy services.



(b) The second system moves the ESI and a database to the cloud. Energy data is stored on the cloud after encryption. The external user contacts the cloud to access customer energy resources.

Fig. 2: Two energy service interface (ESI) testbeds.

with various overriding rules. Thus, an object not only represents a physical unit directly but also a particular functionality as a collection of sub-objects. Leveraging oBIX's advantages, we implemented data and service models for our scenario. The box below illustrates a `History` object. The `is` attribute in the `obj` element indicates that it's extended from a standard oBIX object, `obix:History`. The example also shows that the query operation for reading history records takes an argument whose object type is defined as `psxml:HistoryFilterEx` and returns history records in the object format of `obix:HistoryQueryOut`.

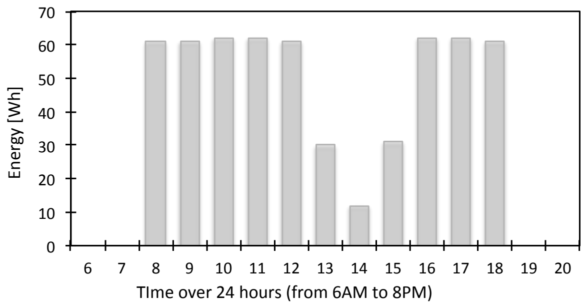
```
<obj href="http://myESI/History/" is="obix:History">
  <int name="count" val="541"/>
  <abstime name="start" val="2012-01-02T00:00:00.000-08:00"/>
  <abstime name="end" val="2013-01-06T00:00:00.000-08:00"/>
  <op name="query" href="query" in="psxml:HistoryFilterEx" out="obix:HistoryQueryOut"/>
  <op name="rollup" href="rollup" in="obix:HistoryRollupIn" out="obix:HistoryRollupOut"/>
</obj>
```

Web service. To realize the interaction model, we exploited the HTTP binding and implemented the ESI in the REST style [6]. More specifically, we mapped the read, write, and invoke request types in oBIX to HTTP methods. The read request uses GET for any object with the `href` attribute and returns an object's information as an oBIX document. Write is targeted at any object whose writable attribute is set to `true`, and is implemented using PUT. Invoke supports any object's operations using the POST method. The box below 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 an external user can control the load in two ways: write and invoke. To turn it on or off, an external system sends a PUT request directly targeting the `connectLoad` object within the same URI, or sends a POST request to the hyperlinked URI targeting the operation object, `controlLoad`.

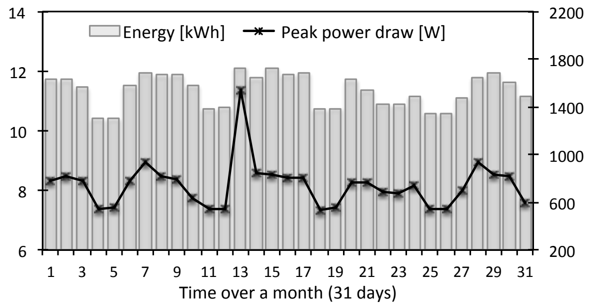
```
<obj href="http://myESI/Points/zigbee/plug1/">
  <str name="deviceName" val="BSPE12SOYZM43001"/>
  <bool name="connectLoad" writable="true" val="true"/>
  <op name="controlLoad" href="control" in="obix:WritePointIn" out="obix:Point"/>
  <ref name="power" href="power"/>
</obj>
```

Resource Centric Security. A security mechanism must satisfy the security requirements we discussed earlier. To address the integrity issue, we implement a hash-based message authentication code (HMAC) with SHA-256. To resolve the fine granularity issue in access control, we adopt Resource Centric Security (RCSec) [7], which performs authorization at the action level (read, write, and invoke). Each object maps to an attribute with a three-digit privilege level. For instance, the object "`plug1`" in the box above is related to an attribute "`plug1=111`". The first digit indicates read permission, and the following two indicate the rights of write and invoke, respectively. Users maintain their own set of attributes in their private key. When a user tries to access the object with a key having an attribute of "`plug1=100`", he or she can read energy usage data but not change any values or control the energy resource.

ESI on the Cloud. Although moving the ESI to the cloud has many benefits, we must solve the security concerns in the cloud technology. Our fundamental approach 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 implemented the ESI and a database in a cloud system hosted by Amazon; the control and security functionalities remained in the mini-EMCS. Fig. 2b represents the new system architecture. More specifically, all the energy data is first encrypted at the



(a) A DR event affects the operation of LED. The bars indicate 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 day.



(b) Aggregated energy usage over 31 days. The average energy consumption is 11.35 KWh, while the average over the weekends (day 4, 5, 11, 12, 18, 19, 25, and 26) is 10.63 KWh. The maximum peak power demand is hit on 13th of the month.

Fig. 3: Two energy services of automated DR and remote energy management, and their experimental results.

mini-EMCS, after which the cloud stores encrypted energy data with meta information. So, external users retrieve historical data by contacting the cloud directly, whereas the cloud doesn't access the plaintext. It forwards users' requests for the control operations to the mini-EMCS, which performs access control and verifies message integrity.

Numerical Result

In an OpenADR experiment, the DRAS server generates a DR event of a real-time pricing (RTP) program - that is, an `EventState` message. The event starts at 1 p.m. and lasts until 4 p.m. During the event, the unit power price changes every hour: it becomes two times, three times, and two times more expensive than a normal price. This event information is generated one hour before the event, so it's an hour-ahead DR program. 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 in our strategy; as the price goes up, the LED gets dimmer proportionally. Because the LED's power consumption is proportional to the brightness level, we can observe the change in energy usage during the DR event, as Fig. 3a shows.

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

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

The next experiment demonstrates that the ESI provides customer energy services. A user retrieves aggregated historical energy usage data from the ESI. To this end, the user prepares a request message (the box above) and contacts the service object via the URI to trigger the invoke operation. 3b shows the retrieved energy data. The bars illustrate that energy consumption (KWh) decreases on weekends, but the difference is slight. This is mainly attributed to the type of energy loads deployed. Our testbed has many servers and switches running 24 hours, but few on/off loads. The figure also shows a curve of peak power draw (maximum instantaneous power consumption) ranging from 500 to 1600 W. The maximum peak was hit on 13 August 2012, when 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 affected the energy usage on that day, because the experiment was short. However, this high peak matters in practice, because most utility companies charge customers based on their peak values. Finding energy usage patterns and encouraging customers to reduce peak values to save energy costs is the energy management system's primary goal.

Smart grid interoperation requires fine-grained accesses to energy resources in smart buildings via the ESI. Here, we illustrated fundamental forms of the ESI that would let external users read data and control resource devices. The ESI's functional requirements will keep updating as the advancement in IT technologies realizes sophisticated energy services. For instance, standardization efforts have defined communication protocols for the energy market in which each smart building participates to sell or buy energy capabilities (consumption, generation, and storage). In this "transactive energy" service, the ESI communicates with another ESI representing a neighboring smart building to make the energy transaction. The emergence of such new energy services also changes how we access energy resources. Integral future research will include an in-depth investigation on new functional requirements of the ESI in transactive energy and focus on how such requirements change the pattern of accessing energy resources.

Acknowledgment

This work has been sponsored in part by a grant from the Korea Institute of Energy Research (KIER) fund B3-2432-01, the University of California, Los Angeles, and the KIER Joint Research Project on Smart Green Building.

References

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

[2] "UCAIug HAN System Requirements Specification, v2.0," http://osgug.ucaiug.org/sgsystems/openhan/default.aspx.

[3] M. P. et al., "Open Automated Demand Response Communications Specification v1.0," *California Energy Commission*, vol. CEC-500-2009-063, 2009.

[4] R. Huang, T. Huang, R. Gadh, and N. Li, "Solar Generation Prediction using the ARMA Model in a Laboratory-level Micro-grid," in *IEEE Smart Grid Communications*, 2012.

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

[6] R. Fielding and R. Taylor, "Principled Design of the Modern Web Architecture," *ACM Transactions on Internet Technology*, vol. 2(2), pp. 115–150, 2002.

[7] 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*, 2012.