

Live, Continuous Energy Auditing of Miscellaneous Electrical Loads: Methodology and Results

Jorge Ortiz, Jason Trager, Gavin Saldanha, David E. Culler, Paul K. Wright

University of California, Berkeley

ABSTRACT

We present the architecture, implementation, and initial results from a live, continuous energy auditing system. By combining QR codes and mobile phones, we are able to iteratively collect building plug-load information and couple it with live, streaming meter data. The system consists of a deployment of inexpensive tags (QR codes) as well a smartphone-based auditing application. The mobile phone reads a tag to associate an electrical plug load with meter data and deployment information. The application also provides *energy data services* over the deployed infrastructure. These services allow the users to observe energy consumption patterns over time. We deployed our system in a 141,000 square foot, seven-story building, and tagged over 351 items over 139 rooms. We present the architecture of our system and the results of our initial audit. We also describe our application and future directions of this work.

Introduction

Miscellaneous electrical loads (MELs) consume about a third of the energy consumed in buildings in the United States [doe]. According to Lanzisera et al. [LBNL_MELS], MELs data are typically collected from national surveys of several thousand homes and commercial buildings. The data is statistically disaggregated to estimate energy use and the residual from these models is classified as “other” and referred to as miscellaneous. It consists of end-use electrical loads that are not accounted for in the traditional heating, cooling, ventilation, and lighting. With the recent rise in MELs, there has been more interest in characterizing the energy use of the equipment in this category. Porter et al. [Porter2006] sampled 50 homes, metering 17 devices per home over the period of a week. Moorefield at al. [Moorefield2008] sampled 47 office buildings, metering 10 devices per building for a period of two weeks. In both studies, data was collected at one-minute intervals and analyzed offline.

Although these initial studies provide more insight about MELs use, the per-building deployment scale was limited, largely due to cost and the technological limitations of the tools available at the time the studies were done. The last study used a plug-load meter called a ‘Watts-up’ with no networking capabilities and it costs \$200-300. This makes sampling every device in the building cost-prohibitive. Also, the ‘Watts up’ contains a limited amount of local memory, so the sampling and collection frequency must be carefully scheduled and managed. You need to sample frequently and long enough to capture the underlying trends but not so quickly that you have to constantly *walk* over and collect the data from each meter. Lanzisera et al. [LBNL_MELS] used a much cheaper power meter with low-power wireless networking capabilities called ACme [acme]. These meters are significantly cheaper than the ‘Watts-

up', costing in the range of \$20 in volumes of 10,000. Not only does the lower cost allow for much large deployments but their wireless networking capabilities remove the need to physically "milk" the meters when space in local memory is exhausted.

Another aspect of deployment setup that makes scaling difficult is the amount of setup work that precedes data collection. Before collecting any meter data, deployment information must be collected to set the context for interpreting the data. For example, the analyst must note that power stream X came from meter Y which was attached to device Z of building A, floor B, room C. During this phase, one of the most time-consuming tasks is compiling a list of devices and categorizing them. This process involves a walk through the entire building listing each plug-load item, classifying it, and recording some item-specific descriptive information. Typically, this information is written on paper and transferred manually into a database for analysis. The associated database schema should be designed to match the information about the deployment, devices, and meters. It must also capture information about the association between the meter, the item, and its location; allowing the analyst to submit cross-cutting queries for each view.

At large scales, this process becomes even more consuming and increases with the size of the building and the number of devices metered. This is largely because inventory collection is typically completed by members of the investigation team; which can only grow so large given a limited budget. The study done by Lanzisera et al. [LBNL_MELS] consisted of a data-collection team of three members. They deployed over 455 sensors throughout an entire building. However, the information setup and gathering phase took several weeks and is discussed in great detail in [cmels]. We argue that information technology should be used to scale this process.

The infrastructure, in the aforementioned studies, was only used for the associated study. It provides *no direct value to building occupants*. We believe that **energy data services** should be offered to building occupants, provided over the deployed infrastructure. For example, occupants should be allowed to directly observe the energy consumption of their devices or see how much they have consumed over a given time period. We also believe that occupants can help track the evolution of the infrastructure (i.e. meters added, removed, replaced) through these services and that they should be indirectly tasked with *distributed consistency maintenance of deployment state* as captured in the accounting system, in return for real-time energy data services. State-consistency management is a fundamental problem and we attempt to build a system that is capable of leveraging the occupants to address it.

In summary, our work attempts to:

- Build a system that decreases the time it takes to gather and collect MELs information by using cost-effective, scalable information technology.
- Allows the integration of a wider range of sensors streams.
- Provides energy data services to building occupants.
- Enables occupants to help maintain a consistent view of the deployment as it evolves.

In the rest of the paper, we describe our architecture and explain our design choices for achieving the aforementioned goals. We also describe the energy services we have

built on top of our deployment infrastructure, go through the initial energy audit results we obtained through our system, and discuss plans for future work.

System design

Although sufficient for their specific study, we believe that we can further leverage IT to design a system that has the following properties:

- Cost effectiveness
- Scale
- Generality and ease of use
- Incrementally deployable

Our architecture consists of four main components: the sensing tier, the data-management tier, the application tier, and the tagging infrastructure. Figure 1 shows the components of our architecture. The sensing tier is shown in the ‘plug-loads’ box. Notice the small acmes with QR codes attached to them. These acmes are placed on plug-loads throughout the building. The data management tier is presented in the box on the upper left-hand corner of the figure. The cloud component represents the fact that our data tier resides in a datacenter and is made available as a cloud service. The application tier lives on the mobile phone and any computing client with network access (not shown in the figure). Finally, the tagging infrastructure is indicated by the QR codes spread throughout the physical environment. QR codes bridge the physical world with the data management tier and the application.

BMAS system overview

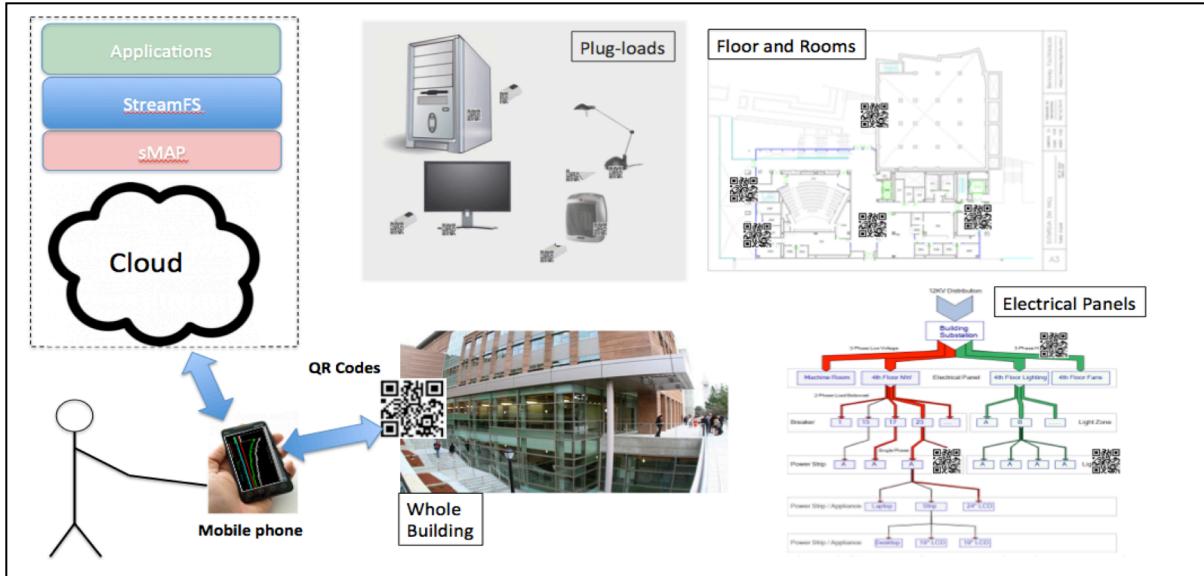


Figure 1. This figure presents the Berkeley mobile auditing system (BMAS) architecture. The figure shows the sensing tier, the data management tier, the application tier, and the tagging infrastructure that bridge the other three through the application on the mobile phone.

Cost effectiveness

To achieve cost effectiveness we examined ways of cutting the cost of each component of the architecture. The plug-load meter we chose are the same ones used by Lansizera et al. [LBNL_MELS], where the large bulk cost was in the range of \$20 per device. The price per meter was slightly higher since the bulk number of manufactured meters was much lower than 10,000. However, the cost per meter was still in the range of \$50. The data tier was set up in Amazon EC2 as a large Unix-based instance, which costs \$0.36/hour. The application tier is free. We provide a code base that can be downloaded on an android phone. The tagging tier is also virtually free. Tags can be printed on demand from QR-Code generation sites. We also provide a site that automatically generates tags.

Scale

Cost plays a large role in scalability. Although not a technically fundamental limitation, it prohibits deployments from growing too large. The decision to use Acmes allows us to deploy many sensors and stratify a much larger set of plug-loads. By choosing to run the data management tier in the cloud we are able to scale our data tier up or down. The cost scales with the number of the machines we spawn to manage the underlying data and support applications. We run a service called StreamFS [sfs] as the main data management service upon which we build our android application.

Choosing the mobile phone as the application delivery and system interaction interface allows you to scale up the number of simultaneous inputs to the system for collecting deployment information. It removes the paper-transcription step and lets you directly input deployment information through a mobile phone. It also allows you to include building occupants by having them participate through their personal devices and serves as a good platform for personalized energy services. Finally, QR codes are easy to print and can be generated and printed by building occupants themselves. They are cheap, easily replaceable, and can be placed throughout the building.

Generality and ease-of-use

Generality can be achieved with the right abstractions and interfaces. For this application we chose sMAP [smap] and StreamFS [sfs] in the data management tier, QR codes in the tagging tier, and the mobile phone in the application tier. sMAP provides a generalized, uniform interface for accessing physical data from sensors. StreamFS provides a generalized interface for organizing sensor data and coupling the organizational structure with streaming data directly. It also provides aggregation facilities that are useful for building energy services; such as an aggregate view of spatial energy consumption. Both provide HTTP/RESTful APIs, simplifying the integration processes between them and with the application. Figure 2 shows the StreamFS architecture.

StreamFS system architecture

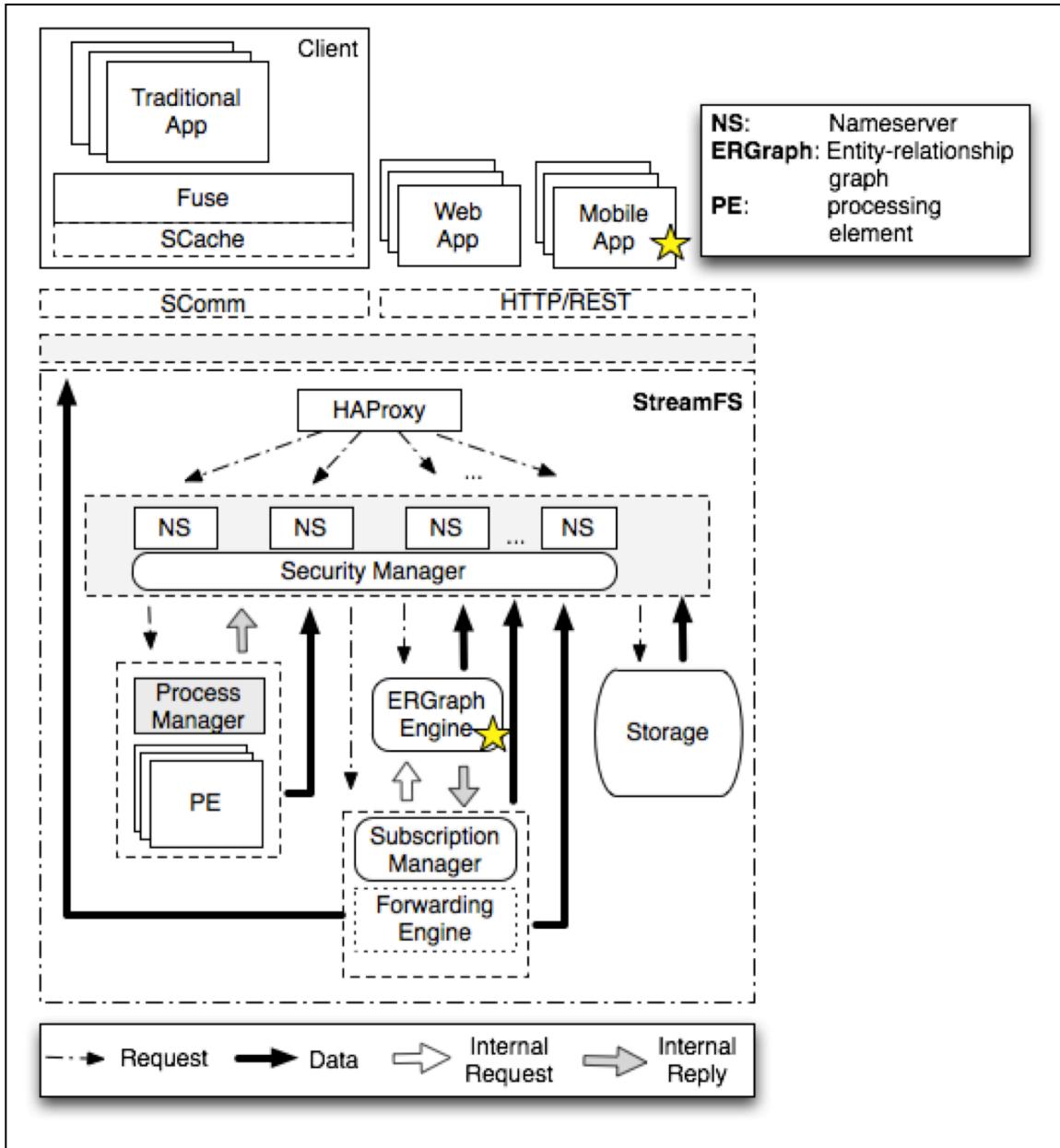


Figure 2. This figure shows the components of the StreamFS system. The main component of the system that interacted with our application was through the HTTP/REST interface. Our application also made use of the entity-relationship graph component as a tool for building energy-management services. Both are starred in the diagram.

The mobile phone serves as the main platform for enabling interaction between users and the infrastructure. By using the on-board camera as a generic tag scanner we are able to bridge the physical world with information about the physical world as represented in the data management tier. Our data tier generalizes to any kind of physical data. Resources are accessible through a uniform resource identifier (URI) whose access is presented by sMAP and the data managed by StreamFS. A meaningful naming

structure is managed in StreamFS, that allows us to couple any application-specific naming convention with streaming physical data fed through sMAP. These pieces **do not limit the notion of physical data** as coming only from Acmes, but from any sensor device that can access these services through a network either directly or via a proxy. The mobile application attempts to preserve the general notion of context in a building and sensor data streams. It does not bind to a specific type of sensor and accepts the binding of QR codes to any type of device.

Incrementally deployable

Finally, we aim for the architecture to allow incremental deployment so that it can grow over time to capture more than just plug-load information. Both sMAP and StreamFS follow the principle of horizontal scalability – if you need more compute or storage, simply add more servers, incrementally, on EC2 (or your favorite cloud-service provider). Storage grows with the amount of data collected, compute power grows with the number and complexity of running applications. QR codes are also incrementally deployable. The more items you have, the more QR codes you print. Furthermore, as smart phones penetration increases, you will have more occupants interacting with the infrastructure through their mobile phone. All of the pieces are meant to build up as demand increases on multiple fronts: more data, more compute, more users, more applications.

Finally, the input protocol implemented by the mobile application incrementally adds components to the infrastructure in small units that include the meter, the item, and the location of the meter/item combination. We go through each step of the protocol in the upcoming sections.

Berkeley mobile auditing system (BMAS)

We call our system the Berkeley mobile auditing system (BMAS). We implemented a version of the application using the aforementioned components. The main pieces consist of an android-based application, acme sensors, QR codes, sMAP, and StreamFS.

Metering devices

In our deployment we used Acme power meters [acme] to take measurement of devices. Acmes transmit data periodically to a local IPv6 router. We set up a network of Acmes in various parts of the building, had them report their data at one-minute intervals to a sMAP server in the cloud. Our architecture does not restrict you to using only Acme meters. The RESTful API allows the developer to include any data stream she wishes to integrate into the system to be represented and managed in a uniform fashion. We chose Acmes for our initial deployment because it provides many benefits based on price-point and distributed/remote management.

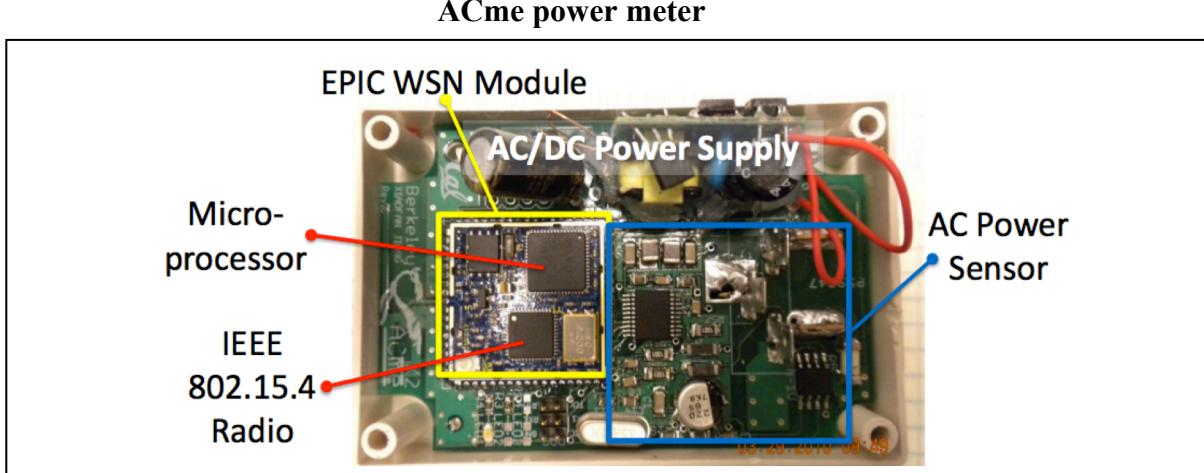


Figure 3. Acme power meter. The hardware includes a microprocessor running an entire IPv6 stack, an AC power meter, and a low-power wireless radio.

Cloud-storage tier

The cloud storage tier consists of sMAP machines and StreamFS machines. sMAP provides uniform representation and access for retrieving physical data and StreamFS organizes that information into a common namespace, capturing the entity relationships that are interpreted by the application. StreamFS manages metadata like a filesystem; organizing the data into a hierarchical namespace that consists of files and folders. It also supports *symbolic linking* and a mechanism inspired by Unix *pipes*. The main differences between StreamFS and a traditional filesystem are that files and folders can be tagged with extra metadata, and these tags can be queried as a way to quickly find the files you are trying to locate quickly. StreamFS represents streams as files. *Stream files* can be subscribed to or piped through processing elements that perform cleaning and aggregation on the data.

These features are main benefits of StreamFS. When constructing the namespace for a building we refer to most objects via their spatial configuration. The root of the namespace is the building itself (referred to by its name), followed by each of the floors and rooms. Inside each room or space we include a folder for each metering device. The contents of the meter folder are the collection of streams produced by the metering device, included as *stream files*. This captures the deployment context so that we can interpret the data. We transform the physical deployment into a structured, traversable namespace as shown in Figure 4.

Namespace construction

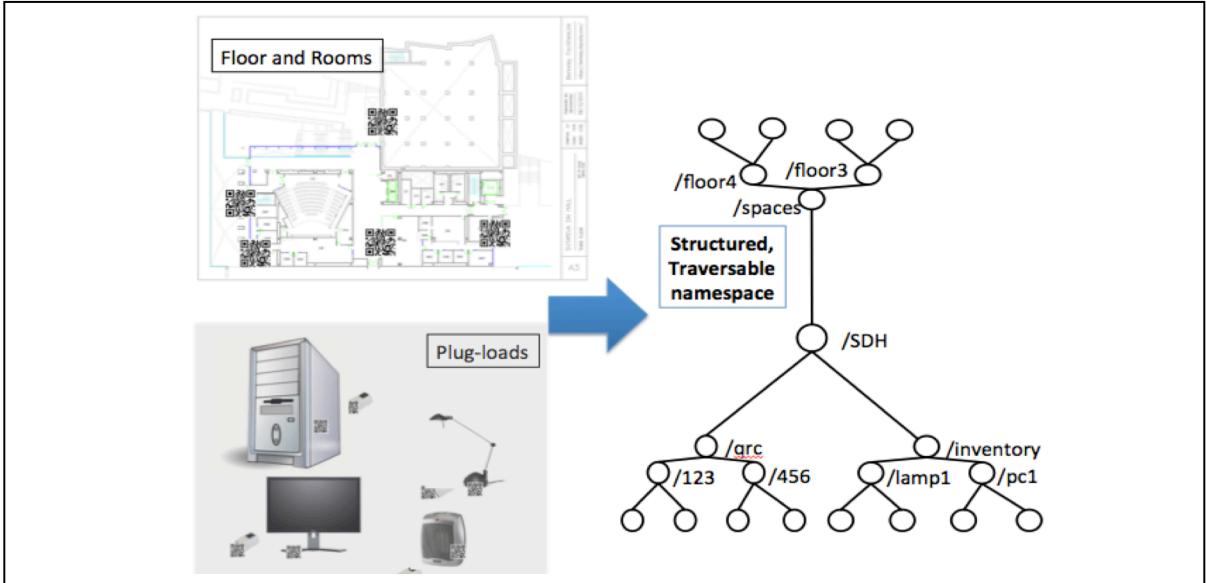


Figure 4. This figure shows how StreamFS is used to manage the deployment in a common namespace. We take items from the building, including the QR codes, and build up the namespace and entity-relationships using the mobile application.

To access an acme in room 410 on the fourth floor of Soda Hall we fetch the data from /soda/4/410/acme123. This naming structure encodes the location but we may also wish to name our sensors according to the panel it is attached to, or by device owner. For this access pattern we simply create a new hierarchy and use the *symbolic linking* mechanism. This allows us to capture the entity relationships by symbolically linking nodes from multiple sub-namespaces. In the example above, we can create a new tree rooted at /soda called /soda/electrical and access the meter from /soda/electrical/root/panel15/load83/acme123, where this is a symbolic link to /soda/4/410/acme123.

Our application uses three namespaces: /dev, /qrc, /spaces. The /dev namespace is used to group all devices folders, the /qrc folder holds all QR code ids (encoded in the QR codes we generated), and the /spaces folder groups the floors, rooms, and spaces throughout the building. Symbolic paths are traversed as a lookup mechanism. A QR code file links to the device it is attached to and the location folder links to the device that is in that location. Devices are also linked to meters. To fetch device info, the QR code device link is resolved. To get associated data, the device's meter link is resolved.

Mobile application

The mobile phone application is mainly used to input data about the deployment but can also be used to fetch and display information. The primary operations supported are 1) registration, 2) bind/attach, and 3) scan. Figure 5 shows a few screen shots of the menu in the mobile application.

Mobile application screen shots



Figure 5. Screen shots from the Berkeley mobile auditing application. The figure on the left shows the menu items that the user chooses from. The figure on the right shows the results of a scan on a Lamp.

Before using the application, a user must generate a QR code. The QR code contains a URL that is encoded with a unique identifier. This identifier is placed in the /qrc namespace, as described in the previous section.

Registration

To register an item, the user generates, prints, and attaches a QR code to the item. The QR code is scanned and item information is entered by the user. The registration process lets the user enter information about the item and permanently associates the QR code with the item. A folder is created for the item (/devices/[device]) and a symbolic link is added to the /qrc/[id] folder to the /devices/[device]. Descriptive metadata is attached to the device as a tag and any future QR code scans display information about the item that was entered during the registration phase. Every item in the inventory, including devices and meters must be explicitly registered.

Location management

We require that all rooms be registered as well. Room registration provides a course-grained mechanism for locating items spread throughout the building. We also simplify the registration process by *implicitly* associating the “current location” of the user with the location of the last item that was scanned. The user’s current location is *explicitly* set when they scan a QR code that’s attached to a location in the building. Registering a room is similar to registering an item.

Categorization

As previously mentioned, part of the inventory collection processes involves categorizing the item. We used the taxonomy of miscellaneous and lower-power products [tax] as a separate namespace and symbolically linked between categories and devices upon registration. This makes it easy to “query” by category by simply traversing that namespaces. We use it to generate several graphs displayed in the results section.

Bind/Attach

Registration performs a binding between the QR code and the item it is attached to. We implemented two explicit associations that are important, *binding* a meter to a device and *attaching* two devices together. The bind operation involves a pair of swipes. 1) Swipe the QR code of the device and 2) swipe the QR code of the meter. This creates an association (symbolic link) between the meter and the item so that user can scan the item later and retrieve meter measurements associated with the item. It also creates an explicit association for detailed accounting after the fact. ‘Attach’ is very similar to bind, except that you attach two devices together. These relationships are important to capture, since there are situations where a meter is actually attached to a power strip which has multiple items attached to it. The association allows you to follow the link path between a meter and an item that has other items attached to it and these associations inform how dis/aggregation is done.

Initial audit results and applications

We tagged and registered a total of 351 items and 139 rooms throughout a 141,000 square foot, seven-story building. For each load, we recorded power/current rating, to compare the actual readings with plated readings. Figure 6 gives a summary of the different types of items we collected.

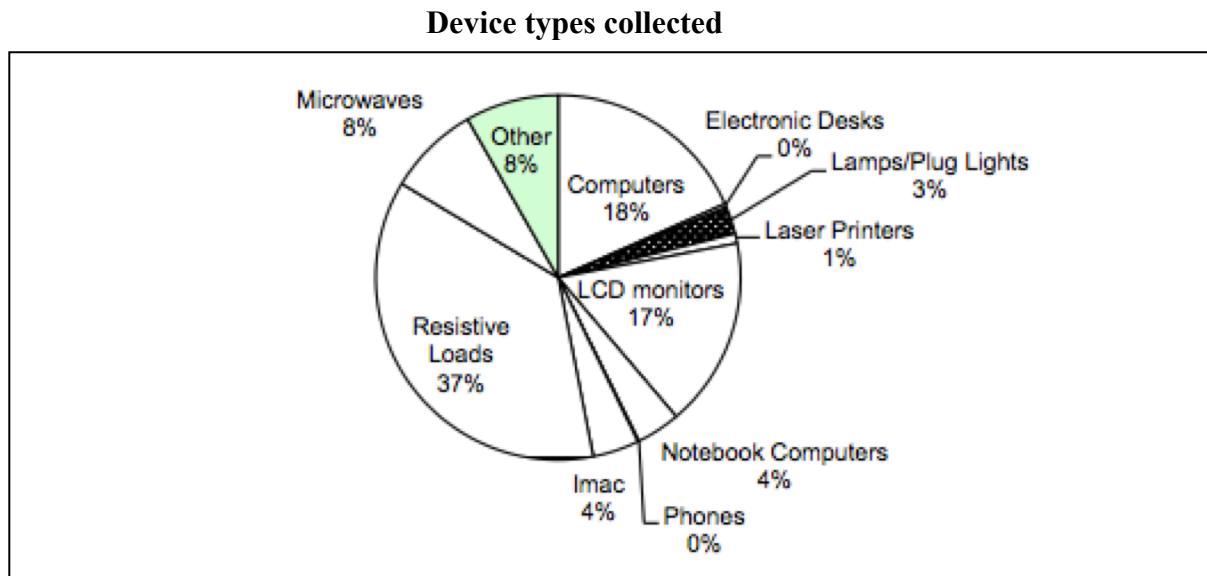


Figure 6. The figure shows the decomposition of various types of devices captured in the initial audit.

The vast majority of items are resistive loads, such as space heaters and coffee machines. The next largest category consists of computing equipment, such as laptops, LCD screen, and desktop machines. These were generated directly from the namespace and symbolic links from the taxonomy namespace.

The BMAS architecture allows you to incrementally deploy live metering and associate it with devices. In our initial iteration, we deployed a set of Acmes in a one of the student offices and used the bind/attach operations to associate the meter feeds to the specific items inside that room. The students in the room tagged and recorded the information with the mobile application. We used the real-time feed to drive a usage visualization application. Figure 7 shows two screen shots of this visualization. The basic goal of the application is to inform users which items are on and which are off. Occupants sometimes leave the lab without turning off. This application allows them to see when items are on that they normally would not be able to observe visually. The figure shows the fridge in various operational modes, derived from the real-time power stream.

In office appliance-state application

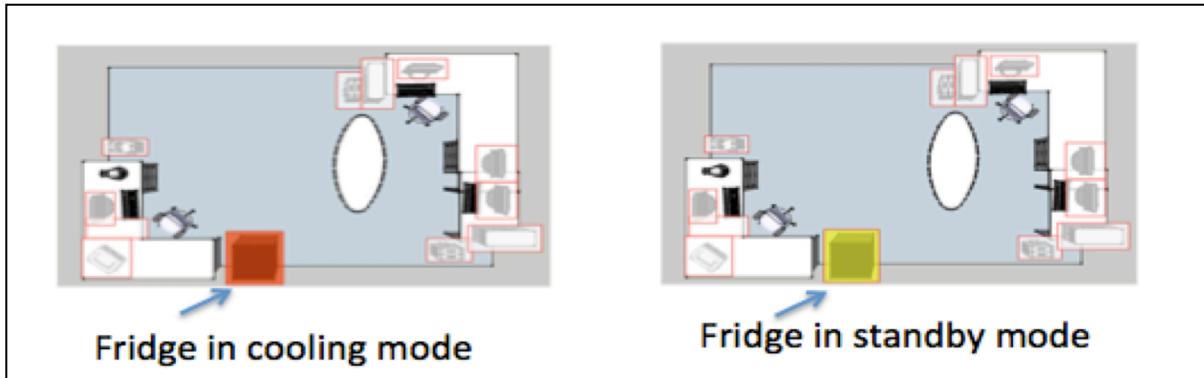


Figure 7. Live data associated with specific device in an office. This is an application built on top of the auditing infrastructure. It shows when appliances in the office are in/active.

Finally, we wrote a script that queries the feeds associated with power usage and by generated a graph showing times of the day when usage was at its peak. Figure 8 shows this graph. Using the graph we can pick out the weekdays and weekends. Weekdays are identifiable by the clusters of red made up of four or five horizontal red strips. Weekends and off-peak hours are the rest, colored either green or yellow. We can see that the typical weekday work hours are between 10am-8pm. Weekdays see the highest power-draw and the weekends see about half the power draw of weekdays. Furthermore, we can see the peak hours of usage during the day. The times correspond to a typical workday between 10am and 8pm. We also noted that the baseline consumption is approximately 30 kW.

Energy consumption: October, 2011

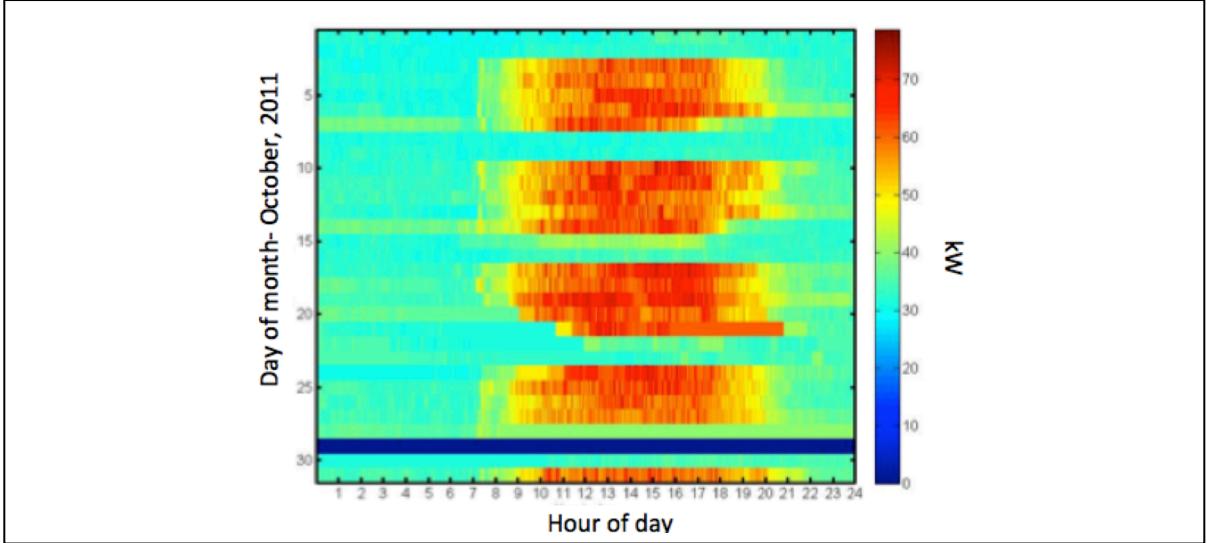


Figure 8. This figure shows the cumulative power draw for plug-load devices in a single room in October 2011. We can differentiate between weekdays and weekends. Weekdays are identifiable by the clusters of red made up of 4-5 strips horizontally. Weekends and off-peak hours are the rest. We can see that the typical weekday work hours are between 10am-8pm.

Energy-related services

We started experimenting with a couple of energy related services. The right-most screen shot of Figure 5 shows the energy scanning services. This lets the user scan an item and view historical power traces associated with the item. We also support spatial views of energy, whereby an occupant scans the QR code for a room or floor and see the aggregate power/energy trace over time that corresponds to it. We are currently working on a personalized energy services as well. This service allows users to tag items that belong to them and view their items in aggregate or grouped by location. This is made so occupants can keep track of their own energy footprint within the spaces that they spend the most time.

Deployment time

Doing the initial deployment took less than one day with only two people. The registration of office items took only a few minutes. Entering information about a device takes a few seconds. The scan time per QR code takes approximately 0.5-2 seconds. A bind/attach operation takes approximately 3-5 seconds, as it involves two scans and a button press. Entering information about an item takes about 30 seconds.

Continuous auditing, commissioning, iteration

The applications demonstrate a range of applications that our architecture supports. We can integrate any kind of sensor feed into a common namespace upon

which we can either run multiple studies based on historical data measurement or build applications that display real-time, aggregate data.

The cost of the system scales well with the size of the deployment and can be done incrementally. Sensors can be added when they are available. The registration process requires scanning a QR code with your phone and inputting a bit of data, QR codes are easily generated and replaceable. They can be printed by building occupants and taped on items of interest. Furthermore, and perhaps to the most important aspect of the architecture, is that it allows the deployment to scale with the number of occupant participants, sensors, and data.

For the initial phase we ran the initial building-wide deployment ourselves with a small team and later picked an office to have the occupants of that office tag and register their items. In return, we offered energy services/applications, which kept them engaged over the period of time that we ran the initial study. We are currently working on a web site that occupants can use to generate QR codes for their own items and to download the mobile application.

Hosting a cloud-based instance of sMAP and StreamFS is also inexpensive and can be scaled incrementally. We are looking to integrate, not only plug-load data, but also building data that is extracted from the building management system. We are also going to add weather feeds and other information to support a new class of modeling applications on the infrastructure. The infrastructure that is in place continues to provide benefits to the occupants and application writers. If we continue to provide interesting applications to building occupants we may incentivize them into participating in maintaining deployment consistency. This will allow us to continue to provide energy-related visualization and control services and will help us understand how energy is being used throughout the building better.

Conclusion

Miscellaneous electrical loads consume a large portion of the total building energy and it is on the rise. It is important to understand how MELs are used within buildings, so that we can learn how to better manage the ones that are deployed and potentially even design better electrical loads. In order to understand these more deeply, we must measure loads that fall into this category. Various studies have constructed multi-component instruments in order to collect and analyze data on MELs. However, there are various shortcomings in the design of such systems. They are generally expensive to put together, difficult to manage and maintain, are only useful for conducting the particular study, and do not particularly scale well.

This paper explores the use of IT to minimize the cost of doing such deployments, provides an incrementally deployable infrastructure that is useful for learning more about MELs and potentially integrates with all kinds of building data sources. The Berkeley mobile auditing system (BMAS) also makes use of building occupants to provide input and manage scalability. We also wrote several applications that not only provide a value to auditors but for potentially inducing changes through visualization and control.

References

[LBNL_MELS] Steven M. Lanzisera, Stephen Dawson-Haggerty, Xiaofan Jiang, Hoi Ying Cheung, Jay Taneja, Judy Lai, Jorge J. Ortiz, David Culler, Richard Brown. Wireless Electricity Metering of Miscellaneous and Electronic Devices in Buildings. 2011 Future of Instrumentation International Workshop.

[Porter2006] Porter S.F., Moorefield L. and May-Ostendorp P. Final Field Research Report: California Plug-Load Metering Study. Ecos Consulting, 2006.

[Moorefield2008] Moorefield L., Frazer B. and Bendt P. Office Plug Load Field Monitoring Report. Ecos Consulting, 2008.

[smap] Stephen Dawson-Haggerty, Xiaofan Jiang, Gilman Tolle, Jorge Ortiz and David Culler. sMAP – A Simple Measurement and Actuation Profile for Physical Information. 8th ACM Conference on Embedded Networked Sensor Systems (Sensys 2010

[sfs] StreamFS. <http://streamfs.cs.berkeley.edu>

[acme] Jiang X., Ly M.V., Taneja J., Dutta P. and Culler D. Experiences with a High-Fidelity Wireless Building Energy Auditing Network. Proc. of the Seventh ACM Conference on Embedded Networked Sensor Systems (SenSys'09). (Berkeley, CA, Nov. 4-6, 2009). Ed.: Association for Computing Machinery (ACM).

[qrc] "QR Code Standardization". *QR Code.com*. Denso-wave.com. Retrieved 23 April 2009.

[doe] US DOE. 2009 Buildings Energy Databook. U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, 2009. Download at:
<http://buildingsdatabook.eere.energy.gov/>

[tax] Nordman B. and Sanchez M. Electronics Come of Age: A Taxonomy for Miscellaneous and Low Power Products. Proc. of the 2006 ACEEE Summer Study on Energy Efficiency in Buildings. (Asilomar, CA, August, 2006). Ed.: American Council for an Energy Efficient Economy, Washington, DC.

[android] <http://www.android.com/>

[cmels] Berkeley National Laboratory, National Renewable Energy Laboratory, Oak Ridge National Laboratory, and Pacific Northwest National Laboratory. Monitoring and characterization of miscellaneous energy and business-process loads: Demonstration of field methods for studying diverse commercial environments. October, 2007.
http://endusefiles.lbl.gov/public/CMELs/CMELs_Joint_Interim_Report_2010_10_07.pdf