U.S. Department of Energy Open Energy Information System (OpenEIS) Project:

Summary of Outcomes of Workshop #2, Data Requirements, held on May 14, 2013

June 5, 2013

Project Manager, George Hernandez
Emerging Technologies
Building Technologies Office

Prepared by the OpenEIS Project Team:
George Hernandez, Building Technologies Office, Project Manager
Jessica Granderson, Mary Ann Piette, Lawrence Berkeley National Laboratory
Srinivas Katipamula, Michael Brambley, Pacific Northwest National Laboratory
Samuel Jasinski, Robert Zogg, Navigant Consulting, Inc.

NOTICE

Table of Contents

Project Overview

The U.S. Department of Energy, Building Technologies Office (DOE/BTO) believes there is an opportunity to achieve significant energy savings in small commercial buildings (i.e., less than 100K sq.ft.) by increasing access to analytical approaches that are known today, but not yet widely deployed. DOE seeks to provide a low-risk opportunity for owners, service providers, and managers to explore the use of analytical methods for improved building control and operational efficiency, and to increase market demand and adoption of commercial solutions. It seeks to increase market demand for and adoption of advanced analytics by developing a software requirements specification for OpenEIS: an open, cloud-based platform that can be used to upload and perform batch analysis of building energy and operational data.

Workshop Objectives

Workshop #2 was the second of two stakeholder workshops that have been hosted by DOE for the OpenEIS project. The objective of the workshop was to gather feedback on data formats that OpenEIS should accommodate, as well as data import, database, and cloud requirements. The topics the facilitators focused on are:

- Common data formats from monthly utility bills, interval electric and gas meters, BAS and other control system trend logs, and temporary data loggers;
- Existing naming conventions, data models, or data specifications that OpenEIS should include to support the desired data import functionality;
- Types of databases, management tools, and schema elements that OpenEIS should use; and
- Cloud-based performance and architectural requirements such as data security, privacy and database hosting.

The OpenEIS Project Team (team) provided participants with examples of the requirements listed above to serve as the starting point for collaboration between DOE and the workshop participants (See Appendix B). The team compiled these examples based on a review of literature and the team's collective experience in building analysis and controls. The team focused on data requirements that would be relevant for small commercial buildings (i.e. less than 100K sq.ft.) and that would enable OpenEIS to support the high-priority algorithms identified in workshop #1 (See Appendix C).

Workshop #1 was held on February 26, 2013. The objective of workshop #1 was to gather feedback from participants on high-priority algorithms that the team should consider for including in OpenEIS. The team developed a summary of outcomes report similar to this one for workshop #1.

Purpose

The purpose of this document is to summarize the outcomes of workshop #2 for the OpenEIS project. The summary includes a list of participants and a section dedicated to each of the four topics of focus listed above that highlights the common feedback received from participants. The agenda, background materials, and detailed notes from the workshop are provided in Appendices A, B, and D, respectively.

Workshop Participants

Facilitators

George Hernandez, Building Technologies Office, Project Manager
Jessica Granderson, Mary Ann Piette, Claudine Custodio, Guanjing Lin, Lawrence Berkeley National Laboratory
Srinivas Katipamula, Michael Brambley, Pacific Northwest National Laboratory
Samuel Jasinski, Navigant Consulting, Inc.

Participants

Name	Organization
Alex Swindler	NREL
Christian Kohler	LBNL
Chuck McParland	LBNL
Mike Spears	LBNL
John Elliot	LBNL
Steven Lanzisera	LBNL
Bruce Nordman	LBNL
Devan Johnson	KW Engineering
Ed Spivey	EnerNOC
Eugene Gutkin	IBS
Indran Ratnathicam	FirstFuel
James King	Pulse Energy
Jason Freeman	McKinstry
Jibo Sanyal	ORNL
Kelly Smith	Wegowise
Mangesh Basarkar	PG&E
Mikhail Gorbounov	UTRC
Steve Kromer	SKEE
Thomas Gall	Honeywell (Novar)
Ron Chapek	Honeywell (Novar)

Common Feedback During Breakout Sessions

Data Formats and Protocols

Participants identified the data formats that are currently the most common for small commercial building data sources. In addition, participants commented on emerging data formats that are likely to be common in the future. Some participants recommended that DOE should consider including support for these emerging data formats in OpenEIS to achieve forward compatibility. The table below shows the currently most common and emerging data formats for the data sources that are typically available in the target small commercial building range, according to participants.

Data Source	Currently Most Common Formats	Emerging Formats Likely to be Common in the Future
Monthly Utility Bills	csv, pdf*	
Interval Electric and Gas Meters	csv, xls(x)	xml, html
BAS and Other Control System Trend Logs	csv, xls(x), proprietary	xml, html
Temporary Data Loggers	csv, ASCII, binary, proprietary	

*Participants agreed that monthly utility billing data is commonly available in PDF format, but that PDF data is the least desirable because it is hard to convert into a useful form.

Participants identified csv files and other delimited text file types as the current de-facto standard for all small commercial building data sources. For monthly utility bills, participants explained that the most common format is pdf files. Participants stressed that pdf files are the least preferable for data analytics because optical character recognition (OCR) methods that are not very accurate are needed to extract the data. Participants also stated that, to a lesser extent compared to csv files, interval electric and gas meter data is currently common in xls format. For meter data, participants recommended that OpenEIS also support xml files because the increasing popularity of Green Button will make xml-formatted interval data common in the near future. Participants added that interval meters and BAS systems are also increasingly exporting xml-formatted and sometimes html-formatted files because oBIX is gaining popularity. However, participants warned that BAS data that is accessible to building owners/operators is often in proprietary formats, which will require a vendor-provided key for uploading and analyzing. Participants commented that data from temporary loggers is usually in ASCII, binary, or a proprietary format, but can usually be exported as csv.

Participants also commented on whether the data required for the high-priority algorithms identified in workshop #1 (See Appendix C) would be time-series data or change-of-value (COV) data. Participants stated that most of the data required for the high-priority algorithms would be time-series data and some would be COV. Participants added that time-series and COV data comes in varying formats. Participants offered ASCII text variations and seconds from January 1, 1970 UTC as common examples. In addition, participants stated that time-series and COV data will come in varying sample frequencies. Participants recommended that OpenEIS include some method of syncing data sets and converting to a standard time interval. Participants prefer a year, date, time and time zone format.

Data Mapping, Normalization, and Taxonomy

Many industry groups have launched information initiatives that concern issues of data naming conventions, building energy information, data schema, and taxonomies. Workshop participants suggested that the most critical requirement for OpenEIS is that it be internally consistent, with a structure that delivers the desired functionality and user outputs. There was also agreement that it could be advantageous for the overall efficiency community to explicitly align with and integrate emerging conventions. There was consensus among participants that Project Haystack is gaining traction and significant industry participation, and is directly relevant to OpenEIS. While it would be simple to adopt Haystack naming conventions within OpenEIS, some participants expressed a preference for shorter variable names.

In addition to Project Haystack, participants suggested that the team also investigate other taxonomies and data models, such as those used in IFCs, gbxml, Title 24, IEP, and ASHRAE projects. Most participants indicated that the site-building-system-component point mapping hierarchy was likely sufficient, perhaps with the addition of a 'client' level to support OpenEIS users who provide third party services. However, one breakout group offered that such one-dimensional hierarchies may not suffice to characterize complex HVAC systems and the relationships between various building systems, for example, a boiler that might serve an air handler, which serves a set of VAV boxes. This dual perspective is an example of a common theme regarding primary versus secondary target uses of OpenEIS and long-term capabilities versus those implemented upon initial release of the platform and associated algorithms.

Across the breakout groups, there was consensus that well-designed user guidance was a critical requirement for point mapping and configuration as well as use of the analysis applications. Graphical and/or interactive mapping of data and physical points within the building system hierarchy was suggested as one alternative. Relative to the analysis applications, participants specifically mentioned documentation of data requirements that directly impact the quality and usability of the output, for example, minimum historic duration and sampling frequency.

Database Type and Schema

The objective was for the participants to identify and define database requirements for the OpenEIS platform, including type and instances of database that meet those requirements. A strawman schema based on a relational database was presented before the discussions began. Three questions were asked of the participations and their answers are summarized in this section.

What is a preferred database type for the OpenEIS application, i.e., relational vs. hierarchical vs. time-series?

Most participants preferred open source databases; however, most felt that flexibility, maintenance and upkeep were more important than which database is eventually selected for the platform. There was a divergence of opinions on whether a relational database is need or non-relational time-series database is needed. Some participants felt that relational databases suffer from performance issues when a large amount of time-series data are stored. However, some participants saw a need for a relational database to store the metadata associated with the points. While others suggested a non-relational database (e.g. mongo DB) may provide more flexibility and also are faster than relational database to store and retrieve time-series data. So, it appears that the OpenEIS may need to support an hybrid approach to store both the metadata and the time-series data.

What are a couple of examples of open source databases of the preferred type (MySQL, noSQL, ProgreSQL, etc.)? Are there database management tools sufficient for the OpenEIS application?

Most participants were familiar with MySQL, some suggested other open sources databases such as mongo DB, time-series DB, etc. But, most felt that good database management tools should be provided and also many felt that most of the database management should be "automated" because an average user may not be able to do this function easily. If the OpenEIS is hosted on the customer desktop or a private Cloud, installation of the database should take minimal effort.

Is flexibility of database more important or performance (or both)?
Most participants felt that both the performance and flexibility were important. Some felt that there need not be a tradeoff if we chose the right database. Some suggested that we optimize the database by storing derived values (e.g. create stored procedures).

Requirements for Cloud-Based Delivery of OpenEIS

Roughly half of the workshop participants reported prior experience in the development of cloud-based applications. In general discussions of security, the distinction between security and privacy requirements was noted, with participants offering that security was related to protection of hacking, whereas privacy was more related to data permission and access rights. Participants were not aware of standard industry security tests that could be include in the OpenEIS software requirements specification (SRS) however the OpenEIS project team was encouraged to look into DOE's recent data security certification process for Google, and industry standard practices from banking IT security. The Department of Defense was mentioned as an example of the most rigorous security requirements, as was the fact that, in addition to secure storage, secure transfer of data may be a necessary requirement.

There was broad consensus that the industry trend is to use Amazon, Google, or Microsoft cloud services, although it may be useful for the project team to explore Open Stack. Moreover, it was suggested that the project team carefully investigate the differences between those offerings to identify distinguishing factors that might impact the specific contents of the SRS. For example, Google uses non-relational databases. Reliability, cost, stability, and backup may be other relevant considerations.

Workshop attendees suggested that ease of use considerations could impact decisions related to browser-based versus desktop-based delivery of the OpenEIS application. For example, the need to install a database instance on a local PC might require a level of IT familiarity that is not consistent with the target user group. Two examples that arose in discussion were Google Maps vs. Google Earth, and hosting Wordpress in a local server vs. a Wordpress server.

Other Recurring Discussion Topics

As in the workshop #1, participants emphasized the importance and challenges of ensuring high-quality data inputs to the OpenEIS algorithms. As the OpenEIS project progresses, the OpenEIS project team and DOE project manager will determine how issues of data quality should be addressed. One solution may be another application or algorithm that analyzes the data for validity. In both workshops participants noted that maintaining an uncorrected history of 'raw' uploaded data would be desirable for several reasons, including assuring integrity of project savings calculations, and of algorithm outputs in diagnostic cases. Data accessibility was another common theme that emerged in both workshop #1 and workshop #2. There is widespread consensus among participants that data availability will be critical to the OpenEIS solution, particularly given the target building size of under 100,000 square feet.

Participants in workshop #2 pointed out that a useful exercise for the team would be to develop more detailed use cases for the primary user groups that OpenEIS intends to target. There was agreement those use cases could then be valuable in driving some of the specific elements of the software requirements specification (SRS). For example, provision of open interfaces is critical for providers of commercial tools to integrate OpenEIS outputs into their platforms.

The final common theme expressed in many of the comments throughout the day was the need for the SRS to accommodate:

- Practices and technology elements that are most common today
- Functionality and algorithms that are most critical for OpenEIS' initial release
- Emerging trends and long term functionality and applications.

For example, csv data is common today; however, green button data, and therefore xml, may be increasingly common the future. Similarly the highest priority applications for initial release do not require specific utility tariffs; however, in the future that may be a desired feature.

Next Steps

The OpenEIS project team will consider participant feedback from the two workshops in developing the OpenEIS Software Requirements Specification. The final specification will be completed by October 2013.

Appendix A: Workshop Agenda

8:00 – 8:30:	Welcome, Introductions
8:30 – 9:15:	Project Overview, Workshop Overview, and Objectives
9:15 – 10:00:	Q&A
10:00 – 10:15:	Break
10:15 – 11:00:	Breakout sessions background and overview presentation
11:00 - 12:00:	Breakout Session #1: Data formats and protocols
12:00 – 1:00:	Lunch
1:15 – 2:30:	Breakout Session #2: Point mapping, data interpretation, normalization, and taxonomy
2:30 – 3:15:	Breakout Session #3: OpenEIS database design (i.e., type and schema)
3:15 - 3:30:	Break
3:30 – 4:00:	OpenEIS team report back on key findings, highlights from breakouts
4:00 - 4:30:	Discussion of cloud issues – attendee experiences, performance, architecture, security needs
4:30 – 5:00:	Wrap up, Thanks, and Next Steps

Appendix B: Background Materials

Breakout Session #1

Table 1: Expected Common User Data Formats by Source

Data Source	Common Formats
Utility Billing Data	CSV, TXT (other delimiting) , HTML, doc, docx, XLS, xlsx, pdf
Meter Data	CSV, TXT (other delimiting) , dat, XML, OBIX, DAT, XLS, xlsx
Temporary Logger Data	CSV, TXT (other delimiting) , HTML, DAT, XLS, xlsx, pdf
BAS Data	CSV, TXT (other delimiting) , doc, docx, XML, OBIX, DAT, XLS, xlsx, pdf

Breakout Session #2

Translator requirements
Many groups and information initiatives are grappling with issues of naming conventions, building energy information standards, data schema and taxonomies, etc. One objective is understand what existing work OpenEIS should leverage or align with, and how. Three illustrative examples follow.

Project Haystack 'tags' 'markers' and other conventions
Project Haystack is an open source initiative to develop naming conventions and taxonomies for building equipment and operational data. We define standardized data models for sites, equipment, and points related to energy, HVAC, lighting, and other systems.

damper

Kind: Marker
Used With: point, unit

Damper position measured from 0% (fully closed) to 100% (fully open).

area

Kind: Number
Used With: site

Floor area of a site measured in ft² or m².

rooftop

Kind: Marker
Used With: ahu

Used with `ahu` to mark an AHU as a packaged rooftop unit (RTU).

elecMeterLoad

Kind: Ref
Used With: point, equip

Equip or point which consumes energy as electrial load. Value should reference the meter (or submeter) it is associated with.

Tags

AHUs should always be marked as `equip`. The following HVAC specific tags are used on AHU records:

- `hvac`: always specified to mark as an HVAC asset
- `rooftop`: if the AHU is a packaged rooftop unit (RTU)
- `dualDuct`: if the AHU has separate cold air and hot air ductwork. Dual duct AHUs should have two sets of discharge points identified with the `coldDeck` and `hotDeck` tags
- `boilerPlantRef`: to associate the AHU with the boiler plant supplying its hot water or steam
- `chillerPlantRef`: to associate the AHU with the chiller plant supplying its chilled water

Sections

Most points in an AHU are associated with one of the following sections of the unit:

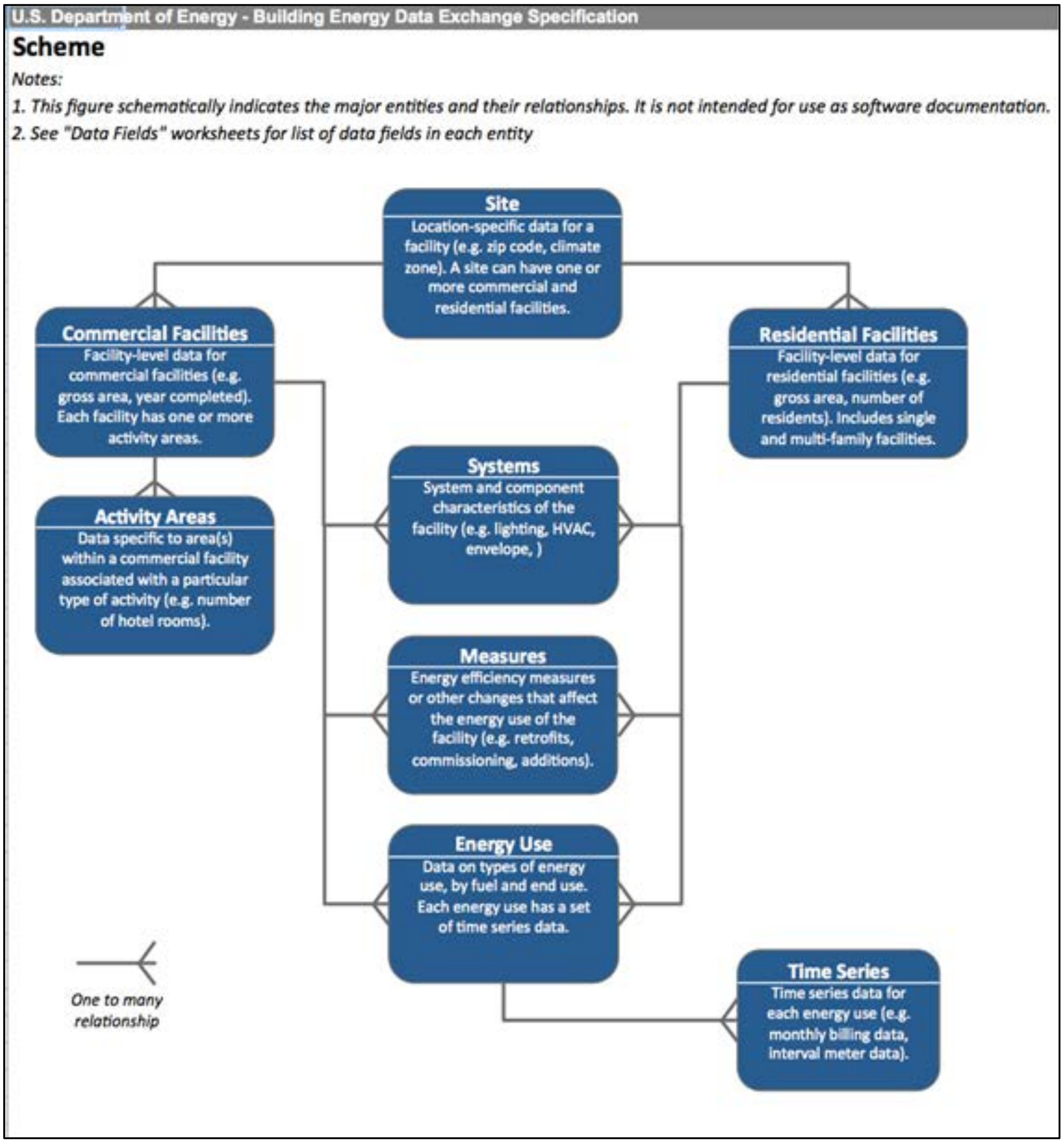
- `discharge`: air exiting the unit to be supplied to the zones/terminal units
- `return`: air returning from the zone back into the unit
- `outside`: fresh, outside air entering the unit for air quality and economizing

Energy Information Standards Alliance

The Alliance has published a specification and uses cases to define vendor-neutral method of exchanging energy information for communication between the user and the grid.

ID	Name	Description
1	Weather	Forecast of weather, and related conditions.
2	Power Quality	Various attributes of Power Quality on both an instantaneous as well as historical or scheduled (future) perspective
3	Pricing Information	Customer-specific real-time pricing (RTP) data tables and demand interval data (e.g. demand rules, tariffs, previous highest demand etc)
4	Energy Emissions	Information used to quantify the environmental burden created during the generation of the power.
5	Present Demand-Aggregated	Instantaneous energy usage of the facility or production site, as an aggregate.
6	Present Demand-Loads	Instantaneous energy usage of the sub loads being monitored.
7	Available Shed-able Load	Individual loads that may be shed to reduce power demand
8	Critical Load	Individual loads deemed to be critical, and their aggregate value, which may not be shed
9	State Change Interval	How quickly sub loads can respond to a shed command to reduce power demand.
10	Existing Demand Thresholds	Existing peak demand in current billing period; existing maximum demand during ratchet demand period
11	Onsite generation capabilities (and availability--spinning and non-spinning reserves)	The on-site generation capabilities, current and future status are summarized.
12	Onsite energy storage (and availability)	The on-site energy storage capacity as well as current and future status is summarized.

DOE Building Performance Database
DOE's newly released (beta) tool establishes a data specification and hierarchy for building energy use and characteristics data: site-->facility-->systems-->energy use and time series.



Appendix C: High-Priority Algorithms Identified in Workshop #1

The following graphic lists the high-priority algorithms, the data points required for those algorithms, and the sources for that data.

Algorithms

- Heat maps or carpet plots to visualize load vs. schedule
- Time series load profiling
- Base to peak load ratios, and peak load benchmarking
- Longitudinal and cross-sectional benchmarking, e.g. Energy Star
- Economizer operations, adequacy of ventilation, and sensor and damper functionality
- Evaluation of indoor comfort
- Identification of excessive mode transitions

Data Points

- Interval gas and electric meter data
- Building occupancy schedule
- Building characteristics (sf, zip code, # occupants)
- Outdoor, mixed, supply, return air temperatures
- Outdoor air, mixed air, and return air enthalpies
- Zone indoor air temperature
- System control set points
- Equipment schedules, status, capacity
- Damper signal, supply fan status
- Heating and cooling on/off status, valve signal
- VAV status

Data Sources

- Whole building gas meters
- Whole building electric meters
- Electric submeters
- Temporary data loggers
- HVAC system trend logs
- Other control system trend logs
- Weather feeds
- OpenEIS user inputs

Appendix D: Detailed Notes from Each Breakout Group

Group 1

Summary of Breakout 1: Data Formats and Protocols

- The group felt that CSV format was the most commonly available data format for any type of data acquisition (meters, loggers, building automation systems).
- The group also felt that even if we restrict the data imports initially to CSV format, we should make sure that we allow of other types of data formats that may become popular in the future.
- Also, the group felt that regular interval time-series data is more common as opposed to times-series change of value (COV).

Summary of Breakout 2: Data Mapping, Normalization and Taxonomy

- The group felt that rather is doing an one-dimensional mapping that was used as an example, we use a two or three dimensional object (e.g. RTU, AHU) while mapping the points. So, it is intuitive to the user what they are mapping.
- The group also felt that there may be open source mapping tools (extract translate and load) that could be useful to the project. They recommend that we look at those tools.
- The group also felt that hierarchal data model presented as an example doesn't capture the relations of various systems in buildings (especially building with complex HVAC systems). For example, a chiller or a boiler can be connected to number of AHUs, while AHUs can be serving a number of zones/VAV boxes, etc. They encouraged us to consider presenting the metadata in different formats to highlight these relationships.
- The also encouraged us to consider capturing the nameplate information of the equipment.

Summary of Breakout 3: Database Type and Schema

- Although initially the group felt that a relational database is appropriate, but at the end the group concluded that a non-relational database (e.g. mongo DB) may provide more flexibility and also are faster than relational database to store and retrieve time-series data.
- The group felt that database management should be "automated" because an average user may not be able to do this function easily. If the OpenEIS is hosted on the customer desktop or a private Cloud, installation of the database should take minimal effort.

Notes for Breakout 1: Data Formats and Protocols

- Utility Meters:
    - csv, tab-separated

Timestamp	Value
…	…

Vs.

Timestamp	…
Values	…

  - "shadow" meter
    - format is all-over the place
    - non-propriety, non-regulation meters
    - used for sub-metering, validation, tenant metering
  - pdf of monthly utility data
    - Extracting data is not easy
    - need to write optical character recongnizer (OCR), still need to validate data extracted by OCR
  - Greenbutton: xml
- Loggers: typically csv, format depends on what was launched
- BAS: xlsx, csv, xml (more common for oBix and ALC)
  - Tridium products all used oBix
- Database formats are less common
- xml becoming more common
- Other data formats

Timestamp	Value 1	Timestamp	Value 2	Timestamp	Value 3
…	…	…	…	…	…
Timestamp	Value 1	Value 2	Value 3	… Value 16	
…	…	…	…	…	

- (ALC, Tridium, Johnson)
- Data formats also depend on clients used to extract data from BAS (opc, oBix, BACnet, etc.)
- Regular time series are more common, but there are advantages to using change of value
- Should also consider hybrid time series where the logger reports changes in value and during a set time interval
  - captures missing data that would other be lost in a change of value format
- Synching timestamps from different datasets (possibly weather and electrical demand) is an issue
- For both time series and change of value, still have to convert into a standard time interval
- When to do point mapping in data stream? Could happen as data is written into an output file or before they enter OpenEIS application
  - Idea: to have a data logger in the system that standardizes, cleans and compresses the data into the OpenEIS format
  - This allows for minimal changes in current building system
- Consider that cost is what most people understand in terms of energy use: "saving 2 kWh" vs. "saving $100 ever month"
  - Rate structures can be downloaded from openEI
  - OpenEI is a wiki source for utility rate data
  - Participants suggest that OpenEIS not be limited to csv files.
  - OpenEIS should be in a tech path of automated file format standards
- Should have a version of OpenEIS that requires the minimal amount of data to run a simple analysis to hook clients who are not familiar with building analysis.
  - If the client wants to know more information they would be linked to service providers
  - Develop a version for service providers that requires more data mapping but has more capabilities and data analysis tools
  - The simple version could require building location, floor area, & time series
- Other tools

- Building Component Library
- Data syncing
- Weather service
- BAS clients

Notes for Breakout 2: Data Mapping, Normalization and Taxonomy

- Instead of one-dimensional mapping demonstrated in the presentation have a visual-based mapping system.
  - Have a diagram of AHU and have users click which part of the system the data points are assigned to
- Search/filtering utilities for mapping multiple points in a data set
  - Ex. Search for all values with the "RTU" tag and map all to the related tag
- For huge datasets limit the number of points displayed
- ETL tools (Extract Translate Load)
  - Open source data integration
  - Clover
  - OpenRefine
  - Data cleaning (Validation, Editing and Estimation)
- Do a rough validation/screening on data or a quick analysis to guide managers instead of going through a wizard
- The hierarchical data model does not show the relationships between different individual systems
- Better to use a semantic database not a linear chart
- Have to address different views of the data
  - Equipment level
  - Zone level
  - Whole building level
- Building plans and specifications are not clearly written.
  - The design view of the plan does not meet with a typical maintenance person's view
  - Makes maintenance and use of building automation systems difficult
- OpenEIS might need a nameplate database for equipment data
  - Possibly from equipment nameplates (QR codes off new equipment)
  - Asset spreadsheets provided by building managers
- Other Tools
  - Industry Foundation Classes
  - BOSS : Building Operating System Services - USB (Culler)
  - Greenbutton apps - programs that use Greenbutton datasets to do simple and quick analysis

Notes for Breakout 3: Database Type and Schema

- A relational database has more resources and can handle larger datasets, also most people are familiar with this type
- Hosting server would dictate the type of database
  - Setting up private database is non-trivial for small building owners
- Instead of hosting a complex app in the cloud consider having an easy to use library or share the code on GitHub
- Have good "admin" capabilities (view, edit, and add points) ex. Chartio: drag and drop visualization tool for database files
- Plan for more flexibility since database formats and types are always changing

- Tools
    - Mongo database: open-source, light-weight and fast; not a relational database more of a document database that is geocoded
        - Easy to query weather to equipment
        - Has a different database for meta data
        - "handles" are used to relate a time series data point to meta data
    - Cassandra database

Group 2

Breakout 1: Data Formats and Protocols

- Most common data format:
    - Utility: paper, pdf, and csv (paper and pdf are the same)
    - Meter (utility interval meter data): csv, xml,
    - Logger: csv (the raw data installed in logger is csv)
    - BAS: csv
- Preferred data format:
    - Utility:
    - Meter: Green button (xml), meter connect to JACE (xml, csv and pdf) or BAS
    - Logger: csv
    - BAS: SQL API –database connection availability
- Least common data format: txt
- The participants said that though .txt is an available format in all of the 4 sources, it is the least common format the users will choose when exporting their data.
- Pdf is the least preferred data format attendees want to use.
- Is time series or change of value (COV) data more common?
- All agreed that OpenEIS should support both.
- What time/date formats are most common?
- ASCI text with or without time zone info; 2)Seconds since January 1, 1970UTC; 3)timestamp UTC in ASCI text4
- Year Date Time + time zone preferred way for attendees

Breakout 2: Data Mapping, Normalization, and Taxonomy

- Question 1-1: What existing naming conventions, data models, or data specifications should leveraged or aligned with in defining OpenEIS's variable names, data models/taxonomy? What standards or approaches do people see taking hold in industry, if any?
    - Existing naming conventions, data models, or data specifications:
        - One attendee pointed out three projects IEP, IFC, and gbxml.
        - The other one mentioned NET CDF data format developed by UCAR for weather.
        - csv mapping may be sufficient.
    - Naming convention discussion:
        - All attendee agreed that short names are desirable. One attendee thought the name should include the info like what the point is, where the point is, and which type the point is. Where the point mean the point is leaving or entering. Which type of the point means the point is AI/AO (Analog input/output), DI/DO (Digital input/output).
        - In naming convention, machine readable is important

- Question 1-2. Is alignment with other approaches more, less, or equally important than internal consistency within OpenEIS?
  - Internal consistency is more important
- Question 2: Is a data model and point configuration based on site-building-system-component sufficient? What else is required based on the algorithms targeted?
  - Clients can have several sites. Client-Site-BLDG-System-component
  - One attendee suggested looking at BIM for OpenEIS's schema.
- Question 3 Should OpenEIS provide the user guidance on how to identify correct BAS points – what is the scope or extent of the guidance documentation that should provide?
  - All agreed that user guidance is helpful. The guidance should be graphic map showing the data points and text description of the physical locations.

Breakout 3: Database Type and Schema

- Question 1: What is a preferred database type for the OpenEIS application, i.e., relational vs. hierarchical vs. time-series?
  - One attendee pointed out the choice of database depends on what is the schema/data structure, what kind of analysis want to do about the data, and how much data.
  - In one attendee's experience, he used relational database SQL and time series for BPD.
- Question 1-1&2: What are a couple of examples of open source databases of the preferred type (MySQL, noSQL, ProgreSQL, etc.)? Are there database management tools sufficient for the OpenEIS application?
  - An SQL (e.g. MySQL) would work if the data set is not too large.
- Question 1-3: Is there a need for a database clean-up tool?
  - Don't make changes or delete data from database. Do it in the algorithm part. The database should store the raw data.
- Question 2: Are there any comments on the examples schema presented earlier today? Is there anything specific you want to see included in the schema?
  - The schema should on the slide – site-building-meter-meter data is not the same with the schema discussed in section 2 (site-building-system-component)
  - Meter is not at the same level as site and building. The example schema should be normalized by site-building-system-component.
- Question 3: Is flexibility of database more important or performance (or both)?
  - These two may not be competing characteristics.
  - Database has ACID (Atomicity, Consistency, Isolation, Durability) properties.
  - NoSQL has extreme flexibility and may not need for OpenEIS. The relational database SQL should be adequate for OpenEIS.

Group 3

Breakout 1: Data Formats and Protocols

- Utility billing data
  - Pdf, tab delimited text and csv
  - Utilities maintain a lot of metadata - meter data, account, location, etc. - provided in a relational Access database
- Meter data
  - Almost universally csv/tab delimited text

- Greenbutton is xml , going to be more common in the future
    - Csv is the de-facto standard, next most common is xls
    - More meters coming out with oBix
    - Smart energy profile has options for export
    - Real-time reading of register data happens in the fully automated case
- Logger data
    - ASCII, proprietary options that can 'export as' binary or csv
- BAS data
    - oBix is emerging but haystack is looking for other standards/protocols
    - Html
    - oBix is a great solution for meter data, but less so for BAS data
- Emerging tools and standards are coming out of Haystack, that might be more efficient, particularly for ---- does is come out csv??
- Data quality is from the meter, not something gap filled with computation, etc. - total system accuracy is a bigger issue
- Data quality is tied into the data source (revenue grade meter data vs. BAS-integrated panel meter)
- 'closed-open' option: data exported in a proprietary format and the vendor provides the key
- Look what universal translator supports as file types
- Zip files for really large data sets and compressed files
- Reformat data into a common time representation
- kW will always be in time series
- Change of state/value saves you space - control setpoints
- Refrigeration equipment may have a lot of change of value and should not be ignored in the algorithms
- Data loggers are fertile grounds for COV

Breakout 2: Data Mapping, Normalization and Taxonomy

- Leader in market right now with mindshare is going to be the people pushing Haystack
    - Gaining critical mass
    - If Haystack can have extensions to own naming conventions
    - Large companies recognize the benefit and will be writing extensions to be compliant
- Are there area where OpenEIS links with DOE ANSI roadmap?
- Suggested to align with Haystack but OpenEIS should be internally consistent
    - Borrow Haystack names where it is useful
    - (Same with BPD data specification)
    - Look into how Haystack compares to IFC
- Economizer is called 'energytransferdevice' in IFCs -- What is the equivalent in Haystack?
    - Close to OpenEIS application, because it's more aligned with device and asset level considerations, than simulation which is not necessarily rooted in the measured facility data
- ASHRAE dash and CEC title 24 and simulation IFC
- App should come with documentation on input data, point mapping, etc.
- Each algorithm will have a "good", "better", "best" data requirements; minimum points duration; and sampling frequency
- Set the expectation and data requirements for each algorithm

Breakout 3: Database Type and Schema

- Postgre SQL Open-source can be tough, most people familiar with MSSQL; MSSQL license is pricey
- On-premise or private cloud implementation, there could be a need to match the OpenEIS database with whatever the enterprises; It is comfortable with
- Group says performance is maybe most relevant to user, but for expanded development long-tem and extensibility, flexibility is important
- The industry trend is to use Amazon, Google, or Microsoft cloud
- Any open source database option would work.
- Maintenance and upkeep more important than the specific database
- Speed of data transfer will be more an issue perhaps than the speed of querying the database, at leat at the onset
- Suggestion to go back to use cases to identify key aspects of performance and then back out database requirements from there
- Time series vs relational vs hierarchical vs object oriented--consensus that it doesn't matter; it's a non-transactional database (no transit time, round trip time, etc.) and will be cloud based
- Length of data historically is a critical of database performance--optimize runtime for monthly analysis (30 days of data as input) vs last twu years is different
- Consider calculation and storage of derivative data, i.e. daily peak for that last 6 months

Appendix E: Detailed Notes from Cloud Discussion

- Security, privacy, and reliability are all things that should be addressed in the spec.
  - Security - prevent hacks into the system
  - Privacy - permissions and sharing
  - Reliability - upload time and stability
- Utility company is more worried about the security than the building owners.
- Department of Defense, as a client, is the higher extreme in terms of data security (with multiple levels of encryption)
- Look at DOE data security certification specification for Google, Amazon, etc.
- Ask someone from IT banking what industry standard security practices are for data security and privacy specifications
- Hand-in-hand with secure storage is secure transfer of data
- Meter data audit to make sure data is accurate and un-doctored coming from the meter source
- There is no difference for data mapping for cloud or non-cloud.
- Half attendees thought OpenEIS is browser-based. The other half thought it is not.
- Can OpenEIS be a mobile (tablet/cellphone-based) app?
- Consider browser and OS compatibility
- If OpenEIS is browser-based, there should be a list of which browsers and which versions the OpenEIS can fit in. There should be another list of compatible windows version.
- Browser-based vs desktop applications and a number of options to explore for OpenEIs, rich clients, auto-update, etc.
- In thinking about whether OpenEIS should be browser-based of installed in hardware, updates are going to be frequent and how these updates are published to users
- The advantage of browser-based OpenEIS is that when different users start the tool, they will see the same thing. There is no need for software update for the users. In contrast, if OpenEIS is not browser-based. The users will see different thing as they may have different versions of OpenEIS.
- Google Maps vs Google Earth
- Wordpress - hosting Wordpress in your server or using Wordpress servers
- Compare cloud based systems of Google and Amazon
- Ease-of-use in defining the actual implementation, for example, are they going to want to go to Amazon and set up an instance, are they going to want/need to set up a database on their PC?
- Google uses non-relational databases, 'big tables' but backup is not provided
- If users are only running their own instance of OpenEIS then cloud security is not an issue
- Private cloud storage - OpenStack
- In one attendee's experience, holding data on cloud is much expensive than just using computing service of cloud.