



Software Defined Buildings



A Platform Architecture for Sensor Data Processing and Verification in Buildings

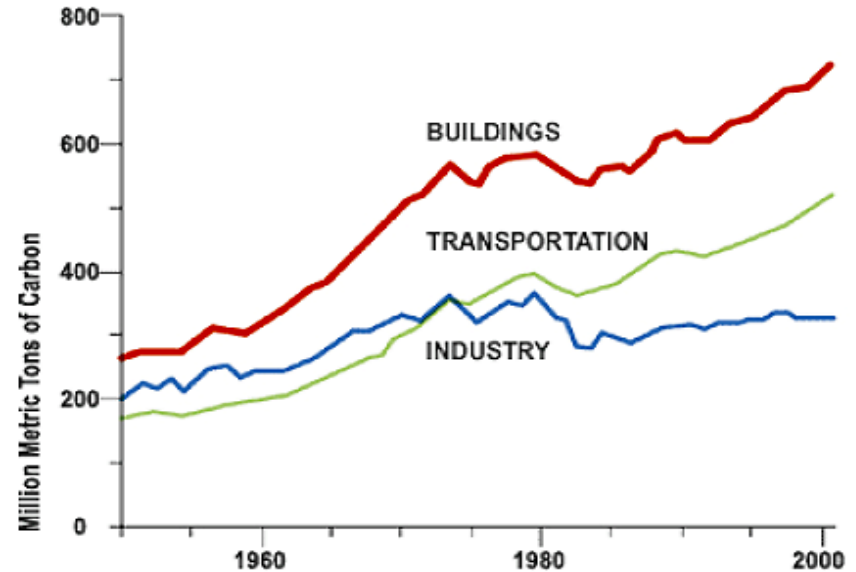
Jorge Ortiz

University of California Berkeley

<http://www.cs.berkeley.edu/~jortiz>

Buildings As Prime Target

- Buildings
 - 72% of electrical consumption
 - 40-50% of total consumption
 - 42% of GHG footprint
 - US commercial building consumption doubled 1980-2000, 1.5x more by 2025 [NREL]
- 4.8 M Commercial & 350,000 industrial buildings



Renewable energy consumption

Represent wider body of computational and information challenges of complex physical systems

Huge Sensor Networks

151 Temperature Sensors



50 Electrical Sub-meters



12 Variable Speed Fans



138 Air Dampers



> 6,000 Sense and Control Points



Sutardja Dai Hall

Built in 2009

140k sq. ft.

312 Light Relays



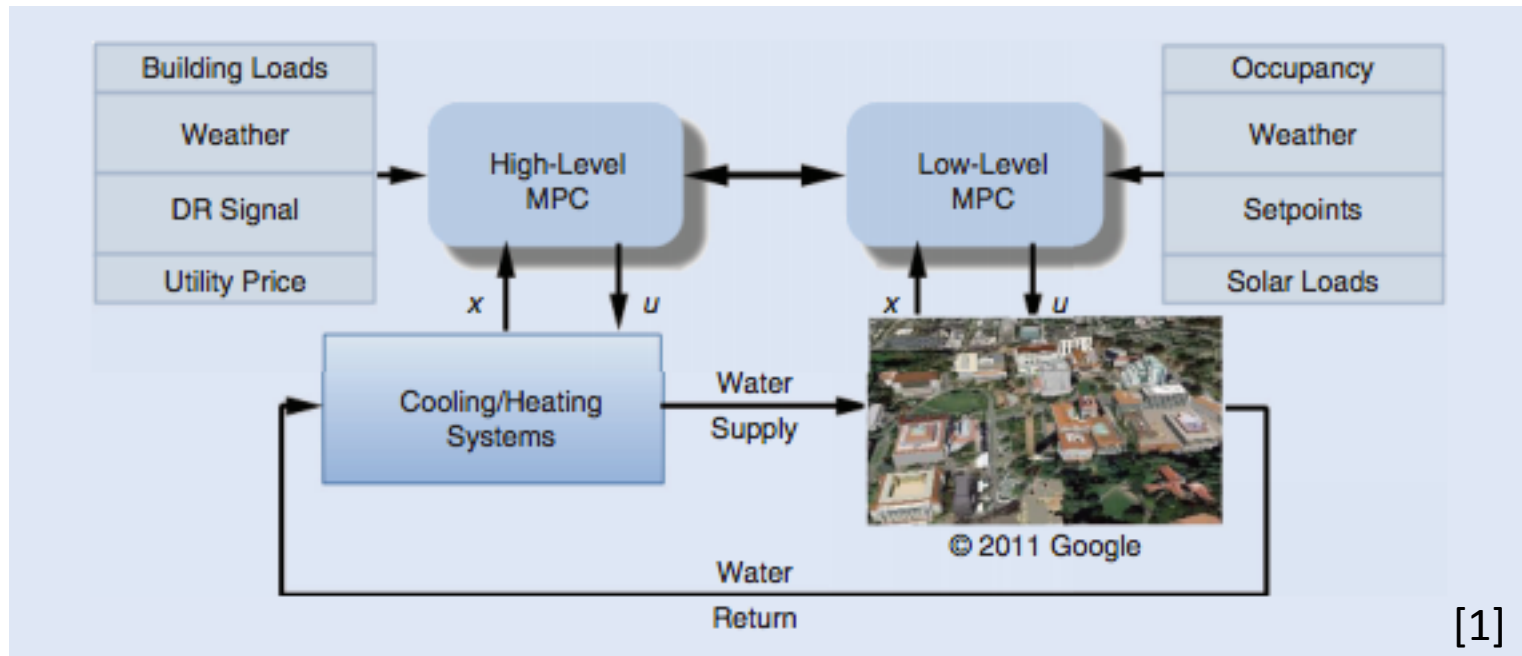
6 Variable Speed Pumps



121 Controllable Valves



Holistic Building Optimization



- **Aggregate** load decisions trickle down
- Local control based on **local conditions**

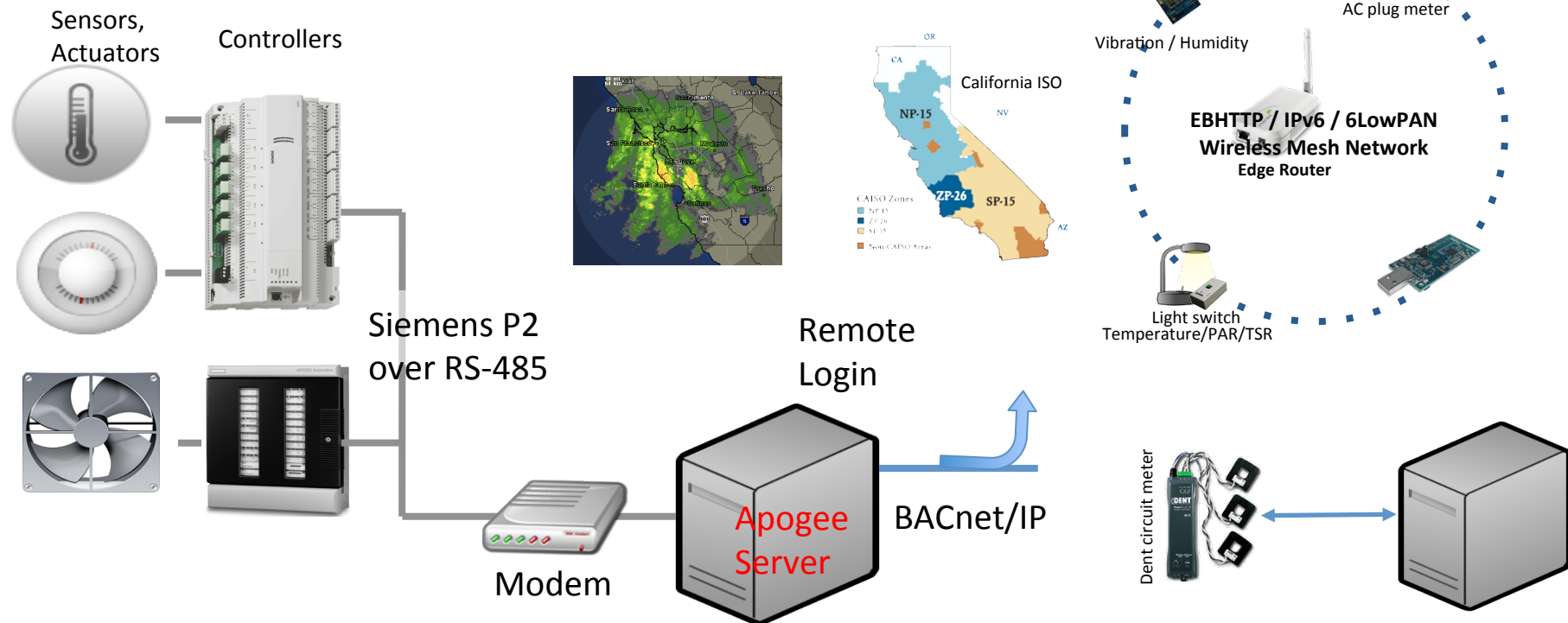
[1] Yudong Ma; Kelman, A.; Daly, A.; Borrelli, F., "Predictive Control for Energy Efficient Buildings with Thermal Storage: Modeling, Stimulation, and Experiments," Control Systems, IEEE , vol.32, no.1, pp.44,64, Feb. 2012

Diagnostics and Control with Mobiles Phones



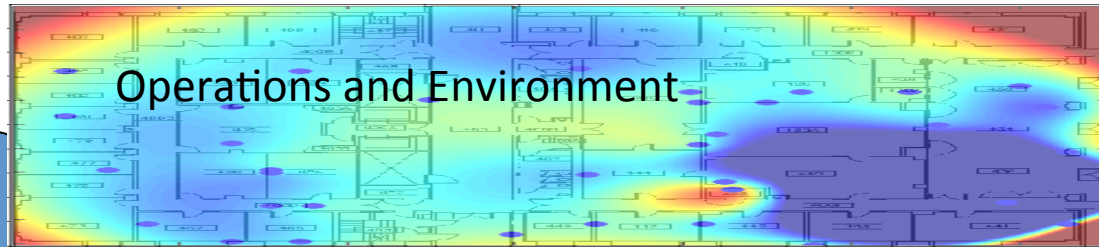
**We need to manage building sensors and
actuators in a unified fashion**

Why is this challenging?



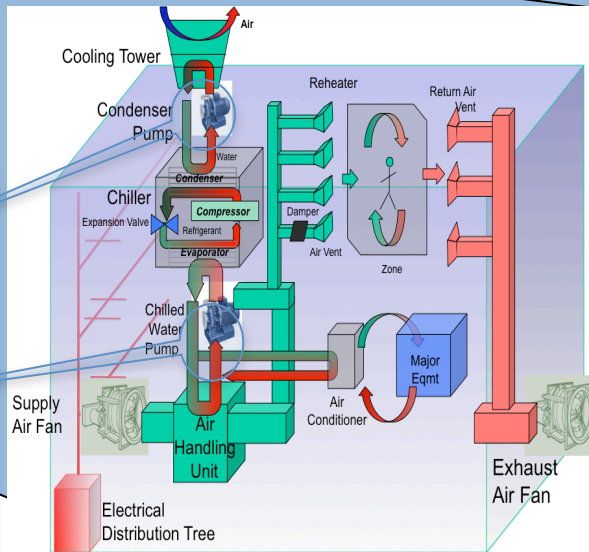
Protocol	Year	Network	Example Applications
Modbus	1979	RS-485, TCP/IP	Panel monitoring, alarms
Fieldbus/HART	1988	various	Industrial Control
BACnet	1995	ARCNET, Ethernet, IP, RS-232, etc.	HVAC, Lighting, Fire...
WirelessHART	2007	802.15.4e	Industrial control, wire replacement
Zigbee SEP 2.0	2011?	802.15.4	Plug-load monitoring

Why is this challenging?



Climate Plant

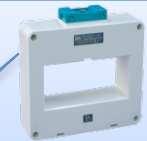
Electric Load Tree



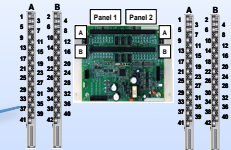
Building Environmental Manufacturing Infrastructure

Vibration

Humidity
Temperature
Pressure



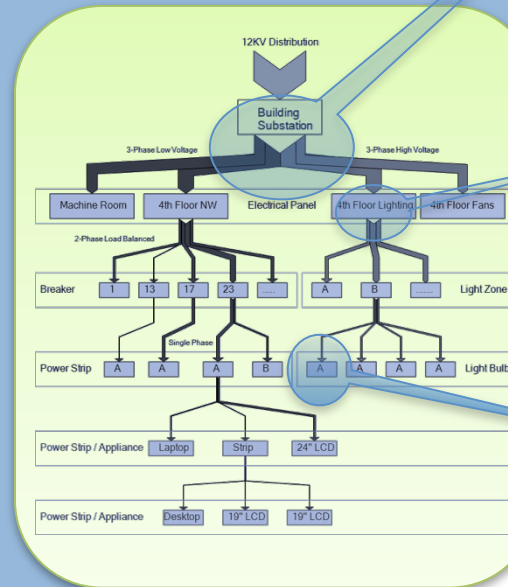
CT: mains power monitoring



panel level power monitoring



ACme: plug load energy monitor and controller



Complications

- Thousands of sensors in **separate silos**
 - No holistic view, **information sharing or extensibility**
- Many legacy/emerging **standards and interfaces**
 - No integration, portability, **scale**
- Buildings are living entities, **constantly evolving**
 - Difficult to **manage**

Solutions

- Systemic view through *OS Abstractions*
 - Focus on **filesystem** abstraction, **data services**
- Continuous *verification* of physical relationships
 - Office re-arranged, walls moved

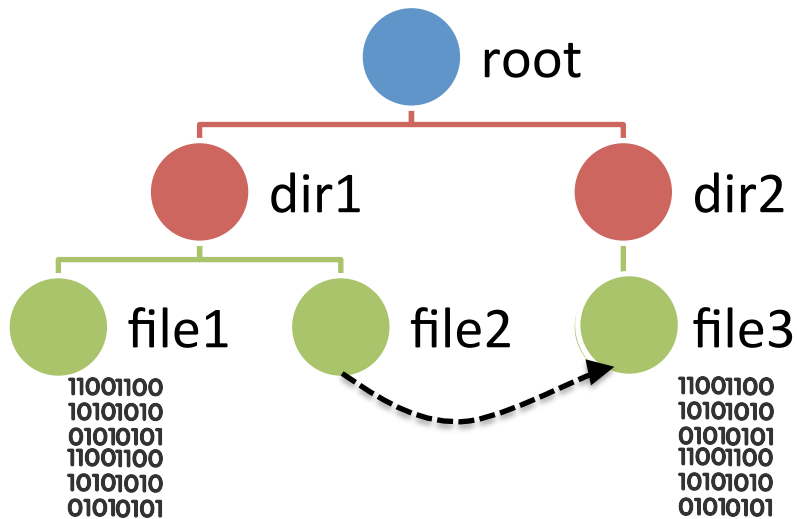
Solutions

- Systemic view through *OS Abstractions*
 - Focus on **filesystem** abstraction, **data services**
- Continuous *verification* of physical relationships
 - Office re-arranged, walls moved

Sensors/Actuators As Resources

- Data **producing**, data **consuming**
- Uniquely **identifiable**
- User-driven **multi-categorization**
- Direct, **uniform access**

Where have we seen this before?



- Unix **filesystems**!
- Resource exposed via namespace
- FS controls access to resources and collections

Sensors as Files



uid:d73a3040e095

temp

owner:jorge

installed:06/28/10

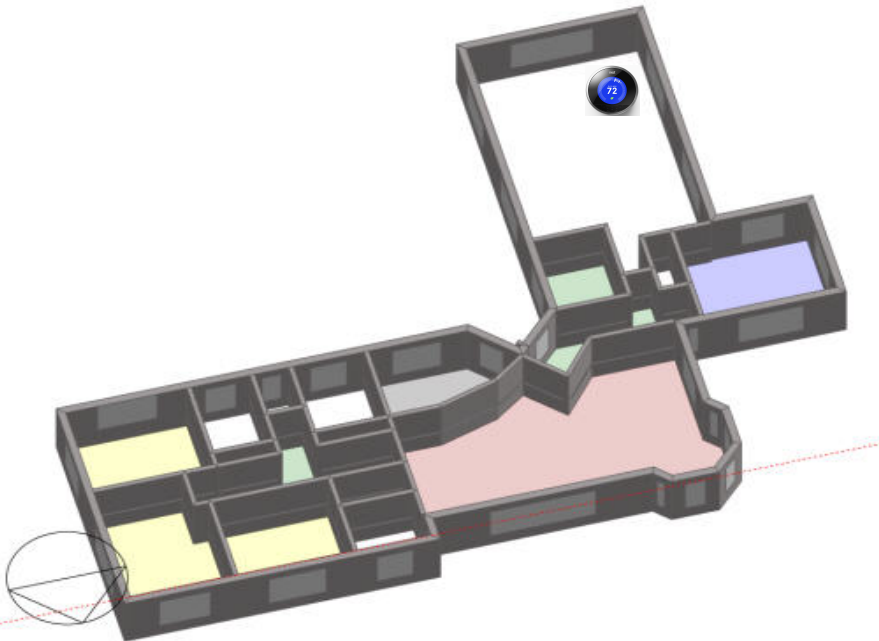
/room/410/temp

/room/420/temp

*

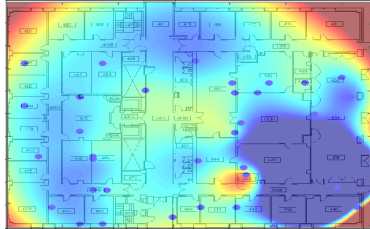
*

*

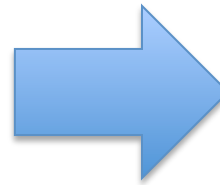


Views Captured in Namespace

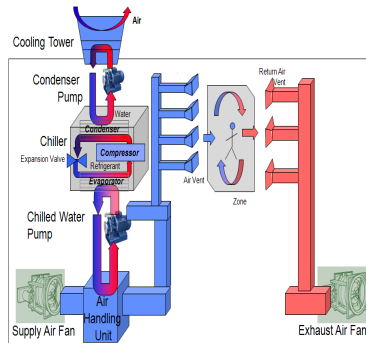
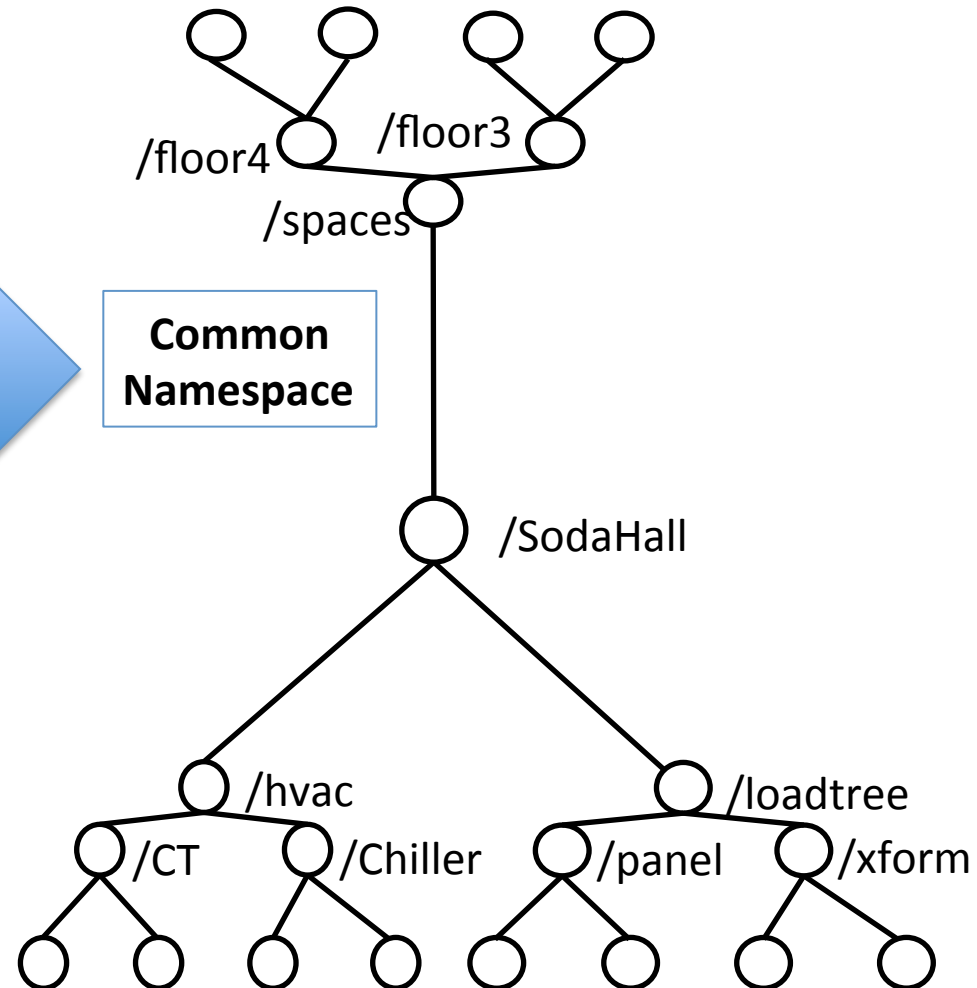
Environment and Activity



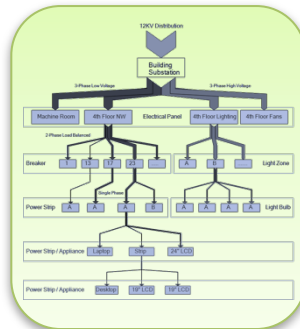
Multiple building views



Common Namespace

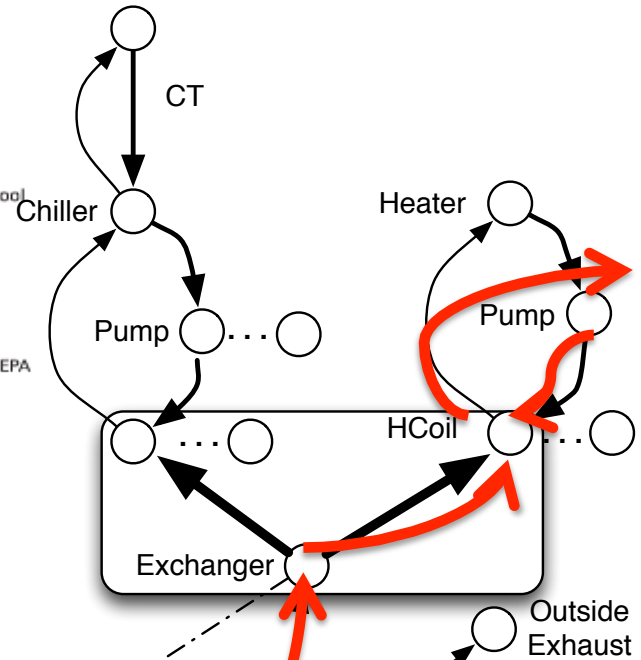
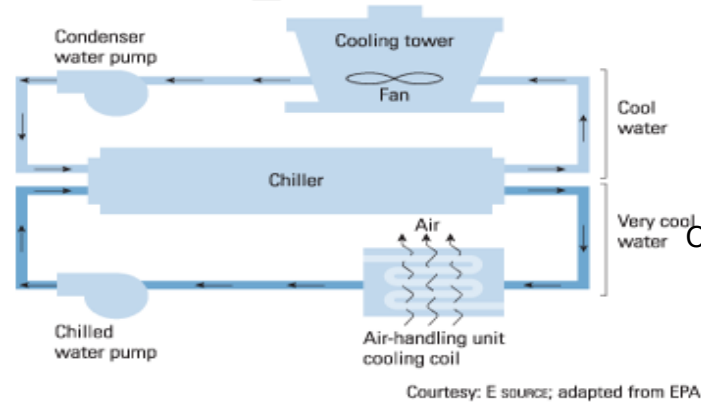


Climate plant



Electrical Load Tree

Capturing Inter-relationships

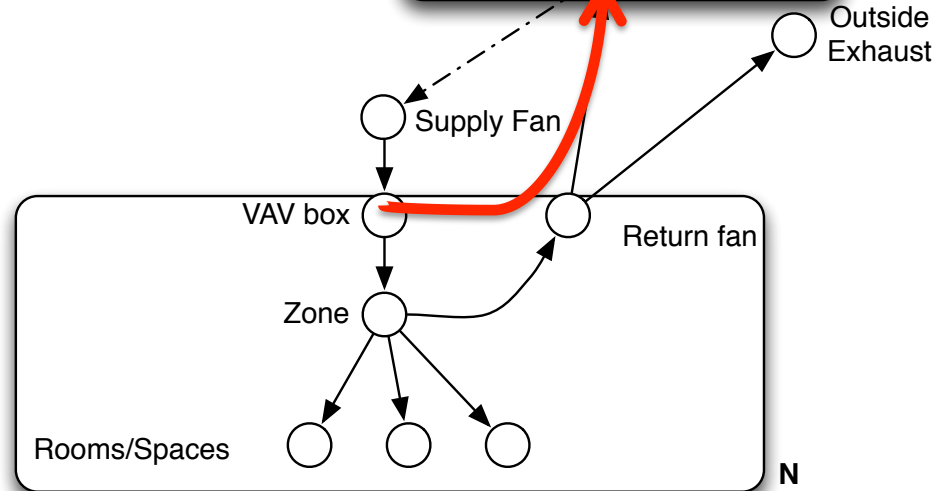
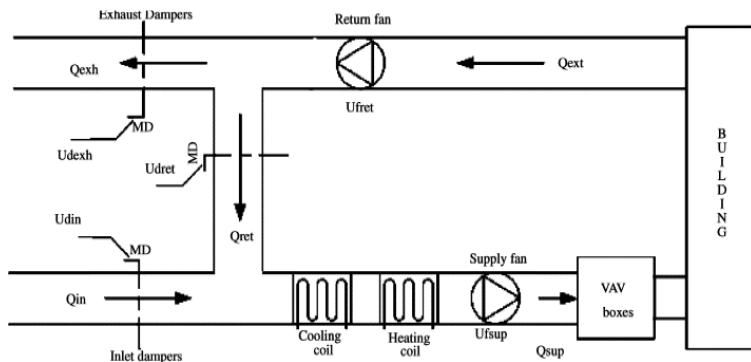


/vav/zone/room/410/return/exchanger

/exchanger/hcoil

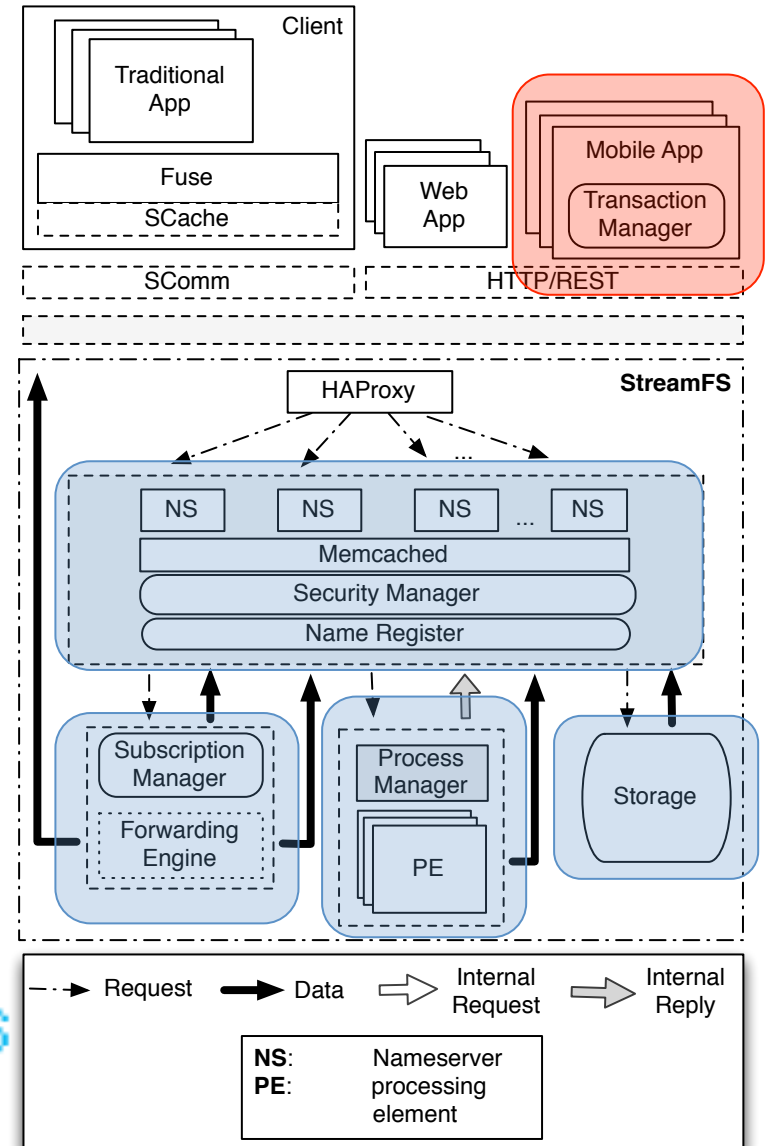
/hcoil/heater/pump

/pump/hcoil



Implementation


- ~20k lines of code (Java)
- 4 core components
- Web service architecture



Application: The Mobile Energy Lens

Energy Lens

[Home](#) / [About](#) / [Grapher](#) / [QRC Generator](#) / [Download](#)

Track Your Energy.  on Android

Tag

Generate QR Codes to place on your personal items. Tagging items allows you to create a virtual representation of your device that enables the coupling of virtual services on your devices, such as the device energy viewer or the personal energy view or control it remotely with the remote control app.

Register

Register your tagged items using the Android-based mobile application. Using the mobile application you can register your item, input information about the item and attach virtual services on it as well. Make sure to set the deployment configuration on the phone before getting started.

Swipe

Swipe the QR codes in the world around you! By **swiping QR codes** in the physical world, you can pull up the virtual services associated with that item. If an item moves, make sure to re-swipe the item in the new location, that way we can provide you with context-aware virtual services.



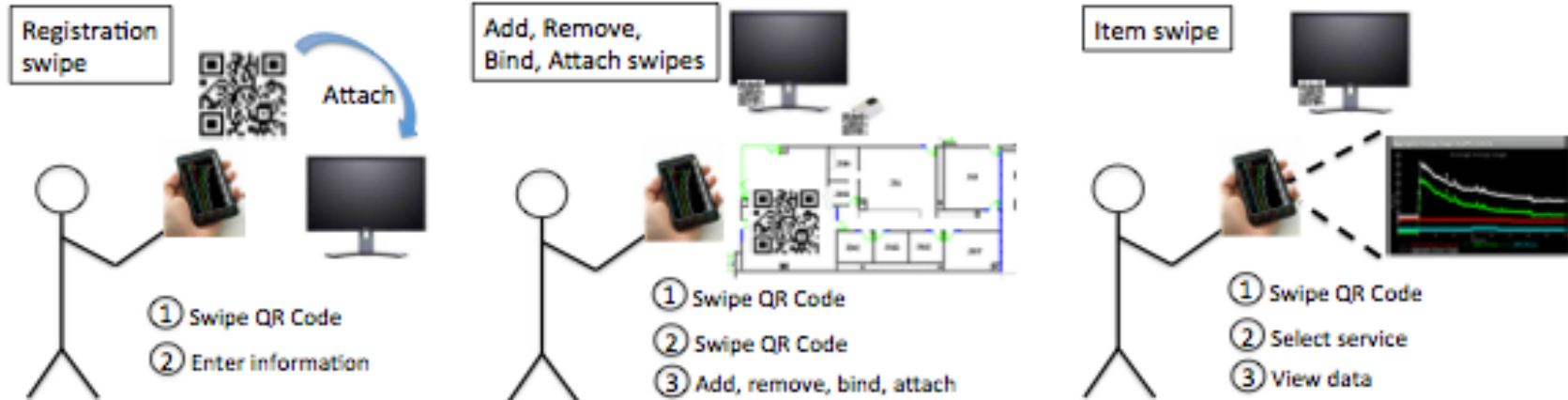
Energy Lens Challenges

- Tracking people and things
 - Active Aggregate Statistics
- Consistency Management
- Disconnect Operation

File Abstraction

- **4 Types**
 - container, stream, ctrl, special
- **Canonical container names**
 - /dev/ctrl
 - /dev/strm
 - /proc

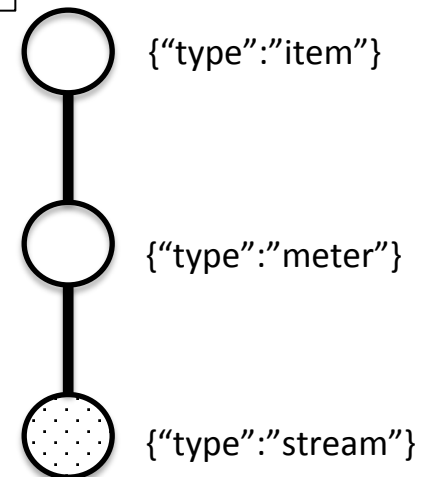
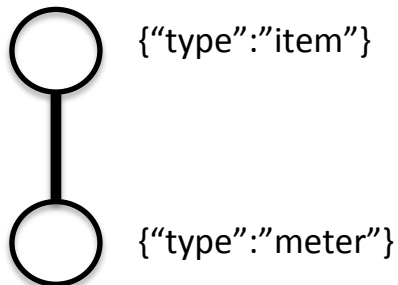
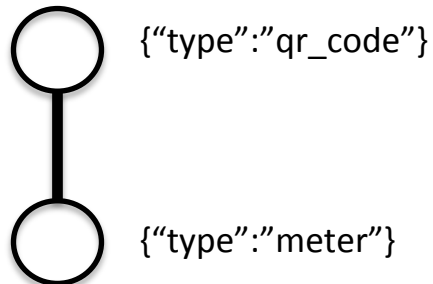
Swipes Capture Links



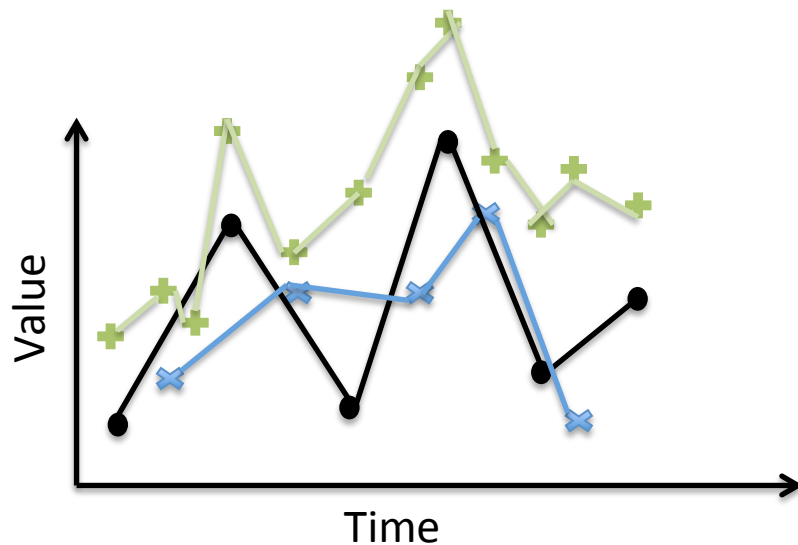
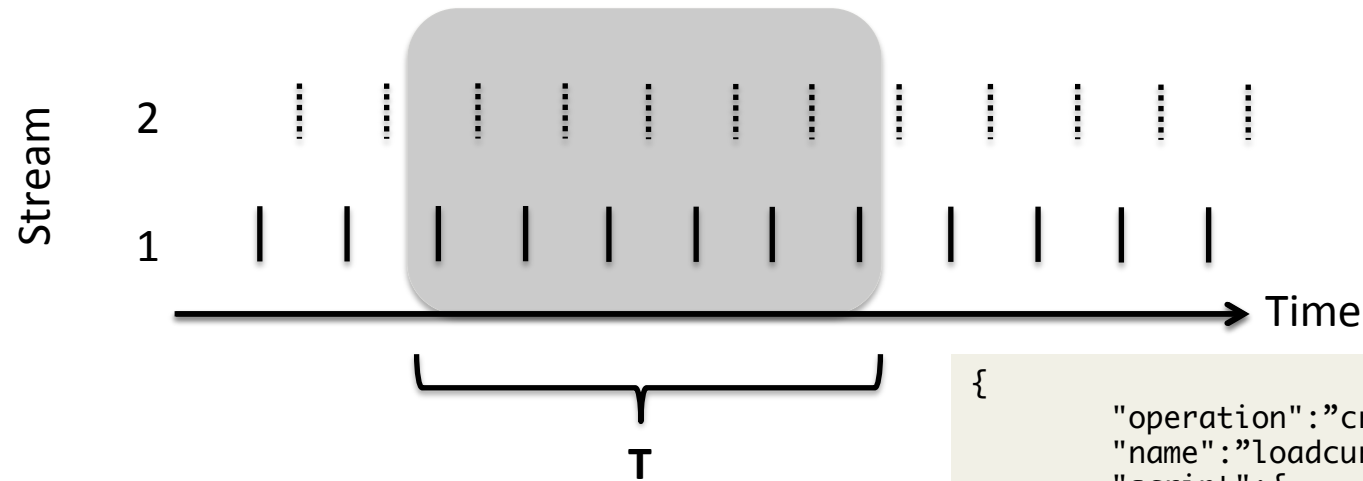
**/qrc
/inventory**

**/spaces
/inventory**

Update entity-relationships



Stream Resampling



```
{  
  "operation": "create_process",  
  "name": "loadcurve",  
  "script": {  
    "winsize": 12,  
    "materialize": true,  
    "timeout": T,  
    "func": "function(buffer, state){  
      ...  
    }"  
  }  
}
```

$$s_3(t_i) = \text{sum}(s_1(t_i), s_2(t_i), \dots)$$

Process Management Through FS

/proc/loadcurve/

```
{
  "status": "success",
  "type": "PROCESS_CODE",
  "properties": {
    "operation": "save_proc",
    "name": "slope_intercept_tester",
    "script": {
      "winsize": 10,
      "materialize": "true",
      "timeout": 80000,
      "func": "
        function(buffer, state){
          . . .
          loadcurve.push({'ts':cleaned
[thissubid][ts_idx].ts,
'value':sum});

          . . .
          outObj.data = loadcurve;
          return outObj;
        } "
    },
    "children": [
      "eda76b7a3b23"
    ]
  }
}
```

/proc/loadcurve/eda76b7a3b23/

```
{
  "status": "success",
  "pubid":
"85e23cce-2378-4c85-9395-09be857e40ff",
  "head": {
    "value": 3271.6666672825813,
    "ts": 1345573470
  },
  "properties": {}
}
```

/sub/1b524cb/

```
{
  "status": "success",
  "pubid":
"85e23cce-2378-4c85-9395-09be857e40ff",
  "head": {
    "value": 3271.6666672825813,
    "ts": 1345573470
  },
  "properties": {}
}
```

Collection & Live Aggregation

MobileSFS 9:50 PM

Current Location: /is4/buildings/SodaHall/spaces

Change Current Location

Add Space

Add Resource

Bind Meter to Item

Unbind Meter from Item

Attach Resources

Detach Resources

MobileSFS 8:56 PM

Current Location: /is4/buildings/SodaHall/spaces

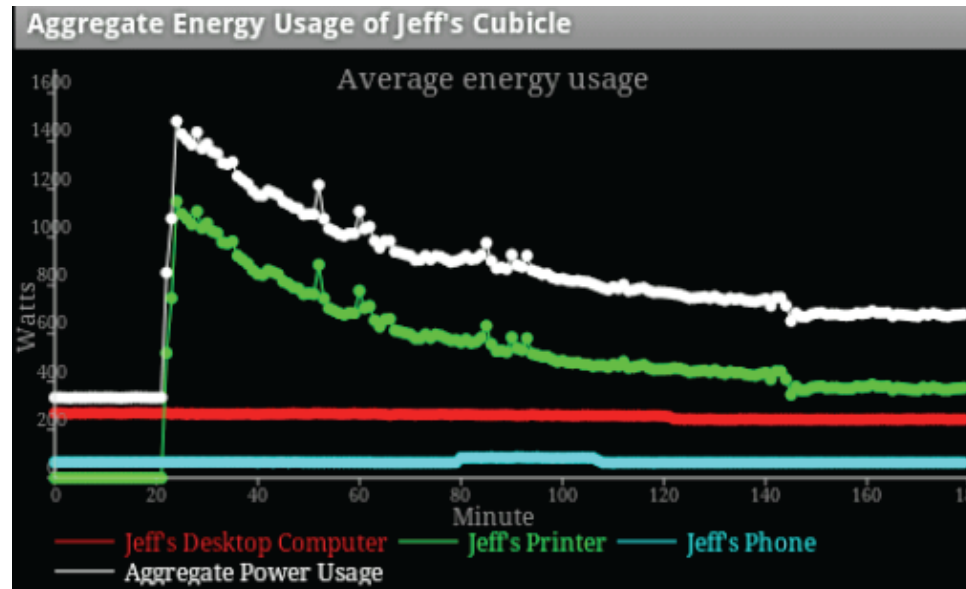
Change Current Location

Name of Item:

Power Rating (Watts): or Current Draw (Amps):

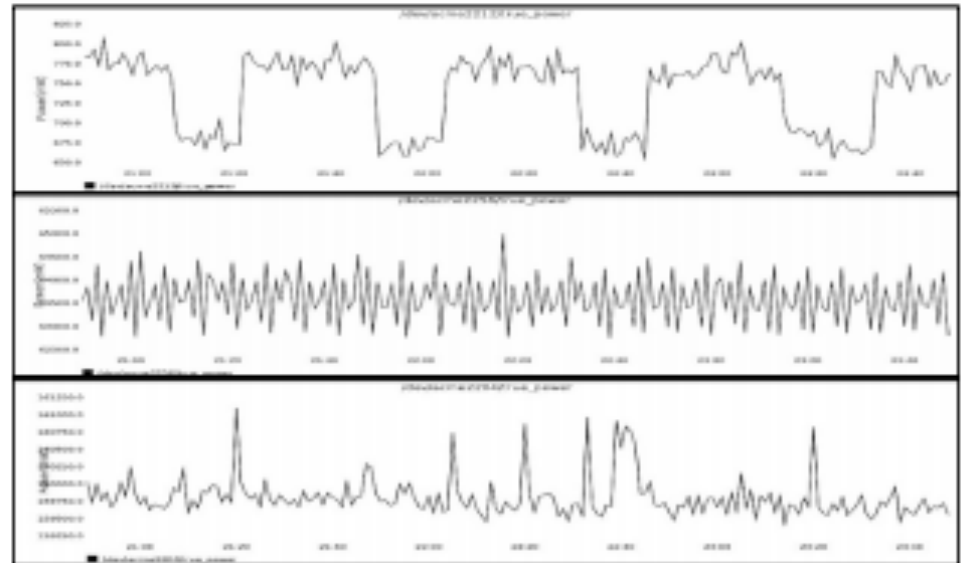
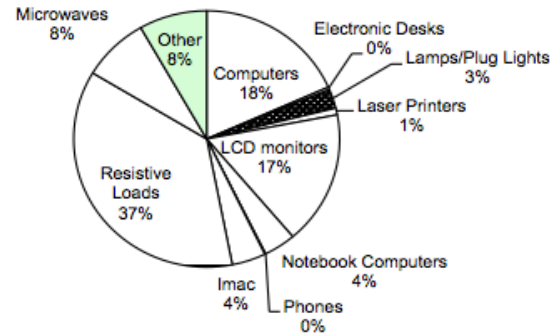
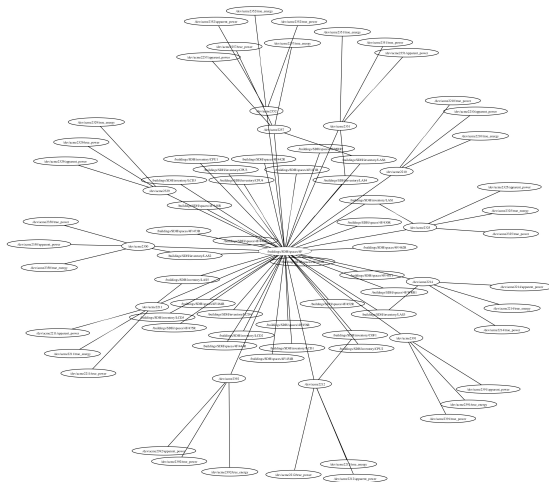
Scan QR Code

New Item will be added inside current location.



Live Statistics On Building Inventory

- SDH deployment
- Registered and classified
 - 20 meters
 - 371 loads
 - 139 rooms
 - 7 floors

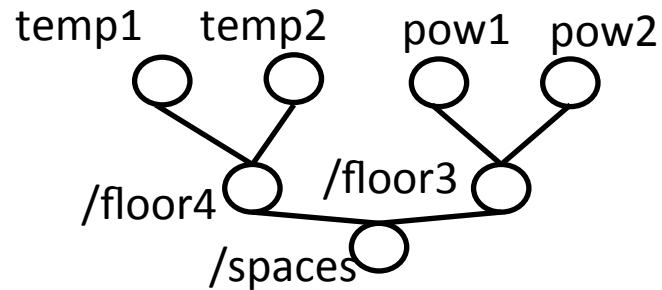


Solutions

- Systemic view through *OS Abstractions*
 - Focus on **driver** interface, **filesystem** abstraction
- Continuous *verification* of physical relationships
 - Office re-arranged, walls moved

Verification

- Inter-relationships manually specified



- **Geometric Relationship:** relative spatial associations

Can we use the sensor data to determine
geometric relationships?

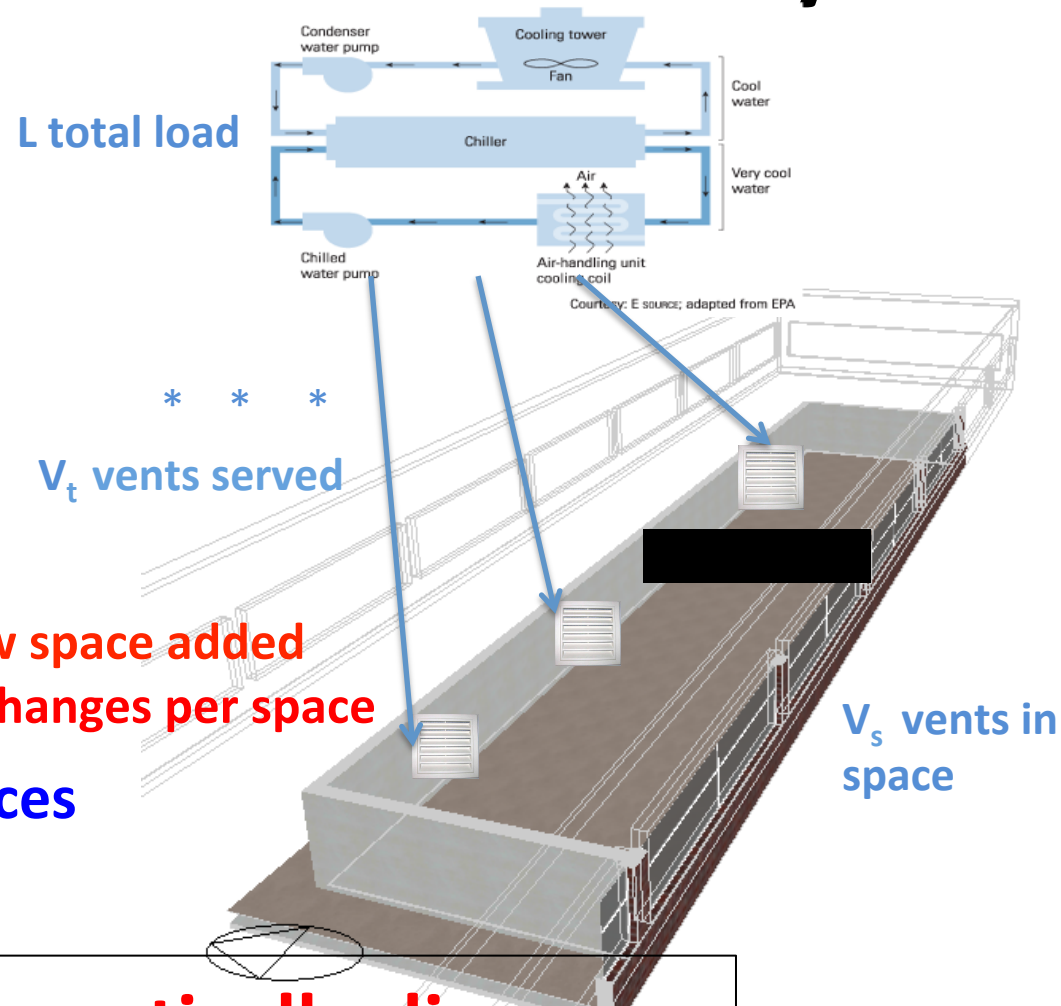
Models Depend on Geometry

$$C \cdot \Delta T = \overset{\downarrow}{u} + P_d + (T_{oa} - T)/R$$

- C thermal capacitance
- T temperature in the space
- u heating/cooling power input
- P_d occupant/computer load
- T_{oa} outside air temperature
- R is the resistivity of the walls

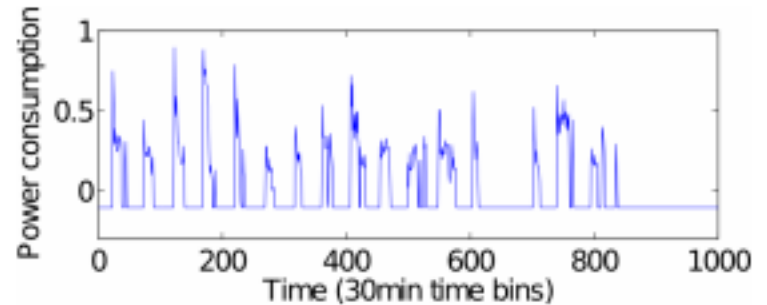
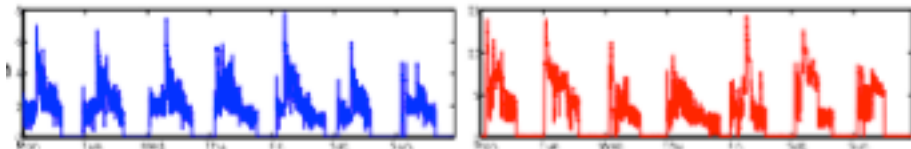
Mapping vents to zones/spaces
must be accurate

How can we automatically discover
changes made to the physical layout?

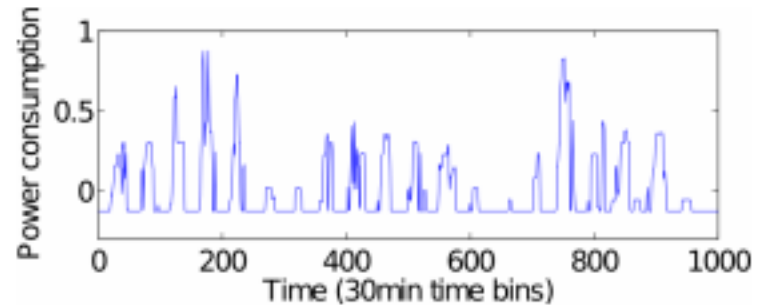


Similar Trend in Data Streams

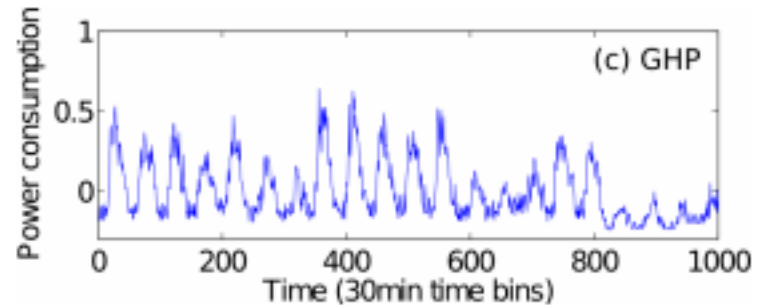
- Sensor streams driven by same phenomena
- Common trend ineffective at uncovering relationships



(a) EHP trace



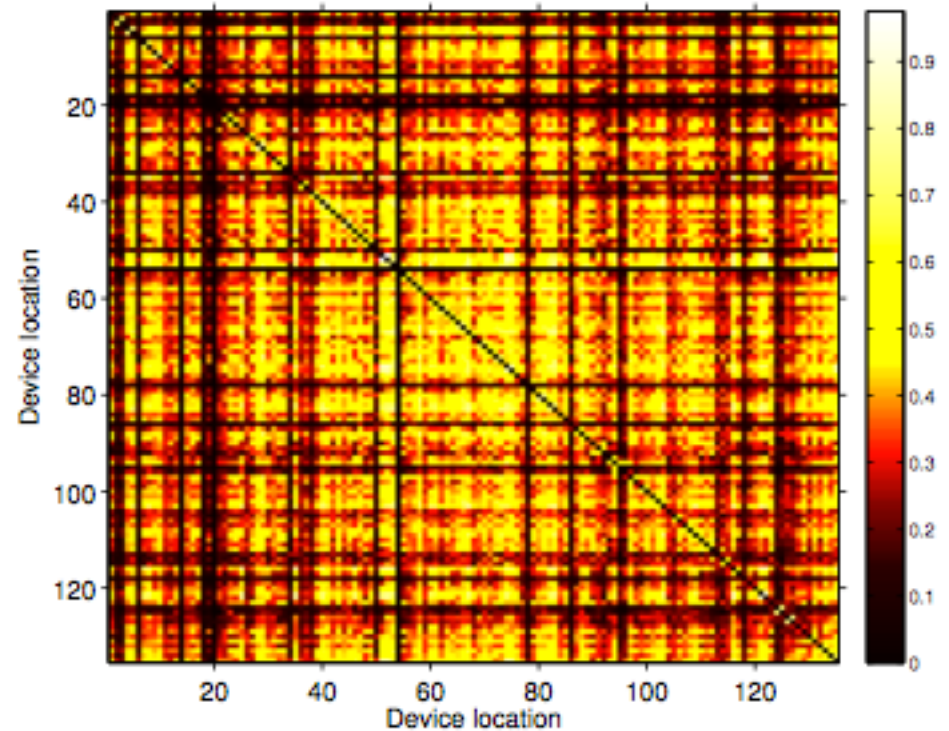
(b) Light trace



(c) GHP trace

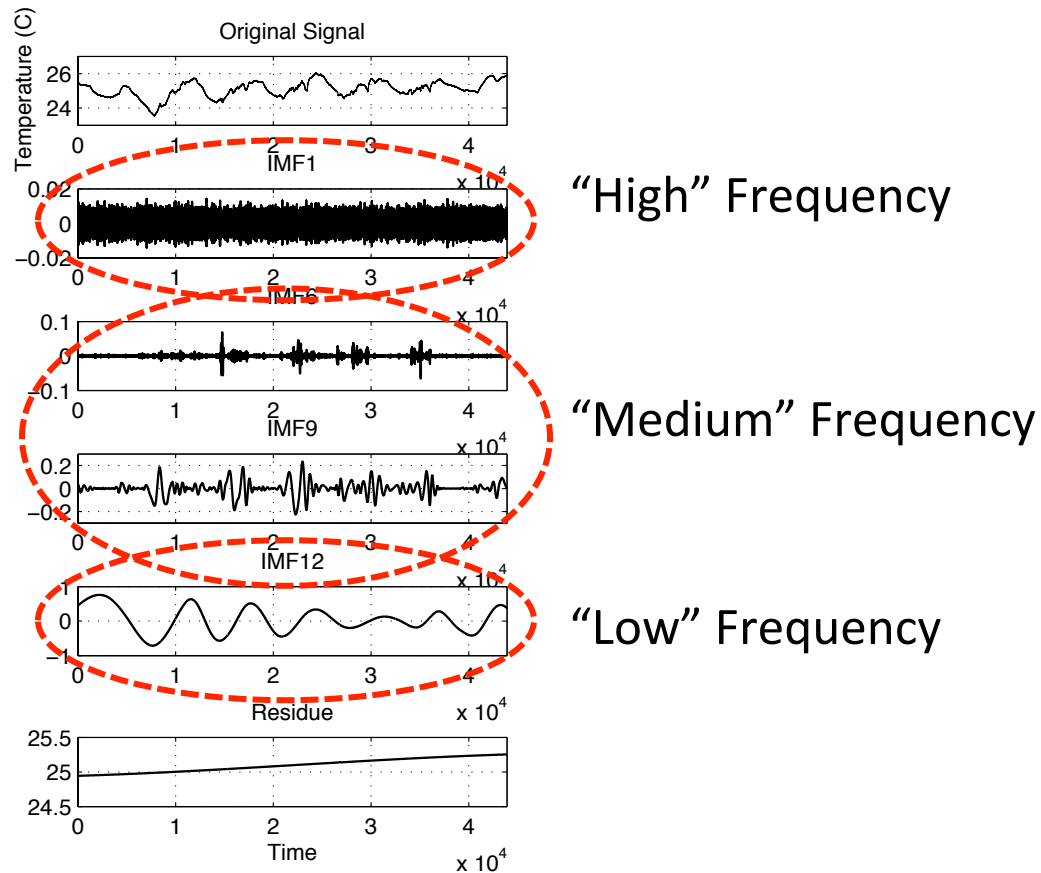
No Discernible Correlation Pattern in Raw Traces

- Each row/column is a location in the building
 - Each location has one or more sensors
- Cell (i,j) is the average device pairwise correlation between sensors at locations i and j



An example...

- Applying EMD on a temperature trace



Experimental Setting

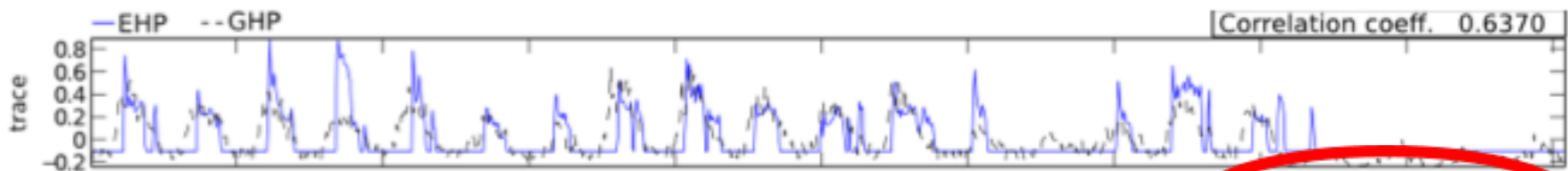
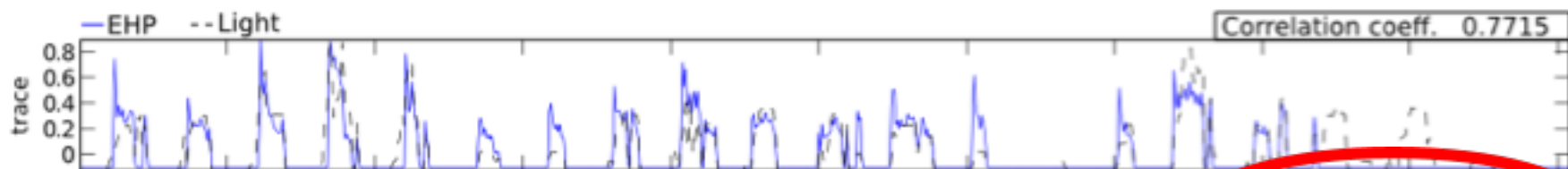
University of Tokyo



Engineering building 2

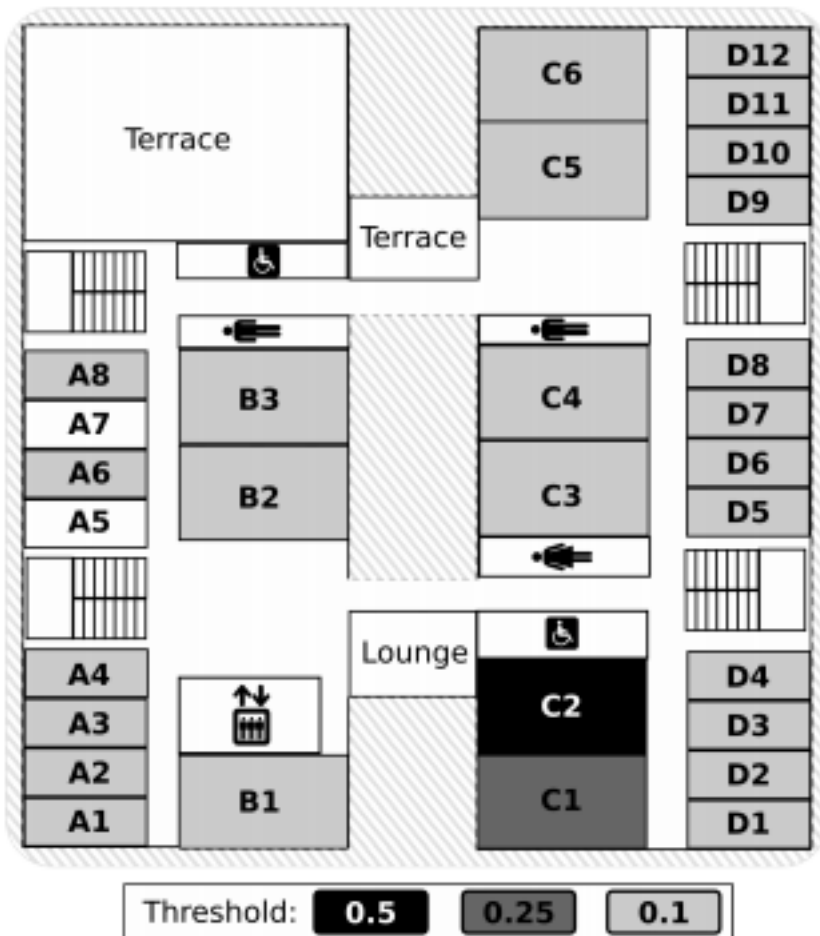
- 12-story building
 - Built in 2005
- >600 sensor stream dataset
- Lighting and HVAC for 231 rooms
- 10 weeks of summer 2011
 - Post-Fukushima Daiichi nuclear disaster

Analysis with EMD

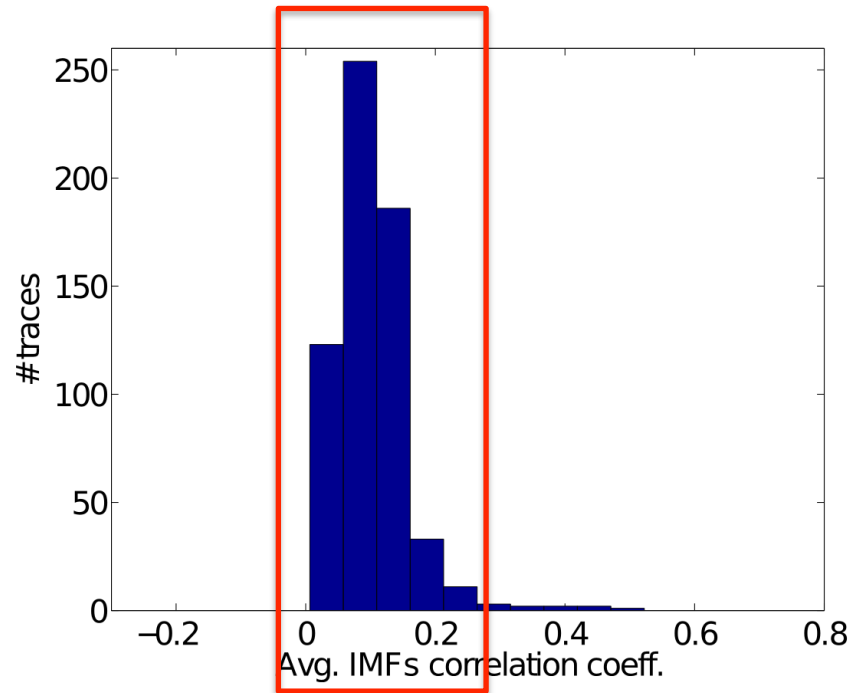


Broader validation

Run across 674 sensors



Geometric Verification Promising



Many sensors intrinsically unrelated,
Few related

Data set

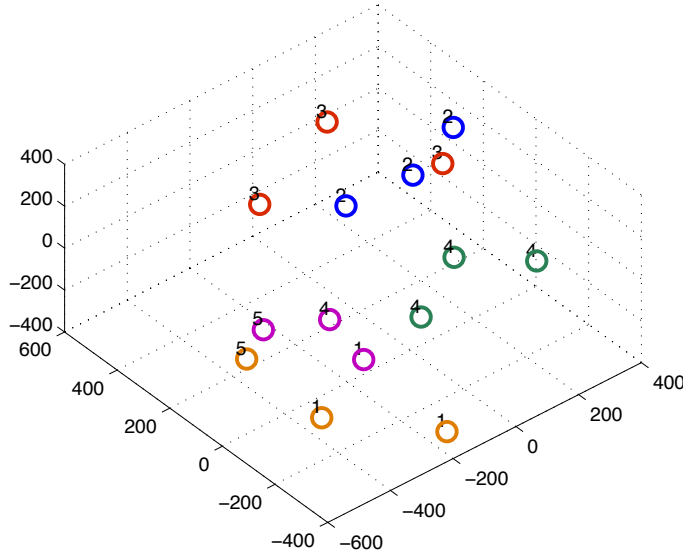
- 5 rooms, 3 sensors/room
- Sensor type: temperature, humidity, CO₂
- Over a one-month period



Results

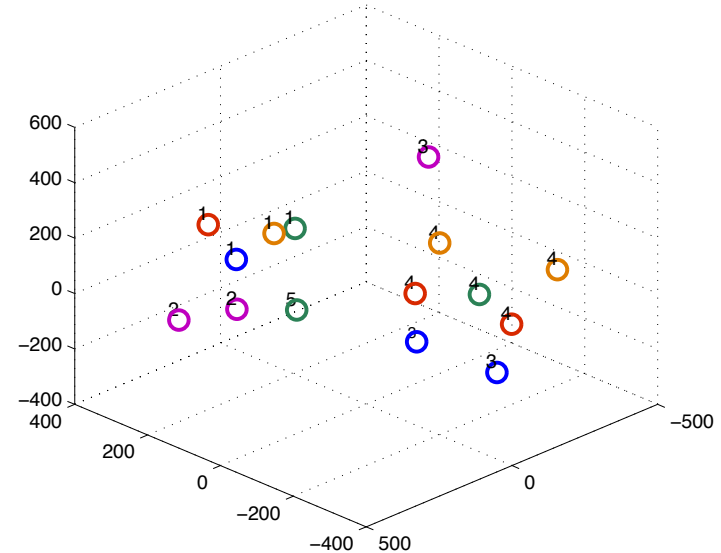
- Clustering results (MDS + k-means)

-MDS: Multidimensional Scaling



On corrccoef from re-aggr.

12/15 correct = 80%



On corrccoef from raw traces

8/15 correct = 53.3%

Results

- Clustering results (thresholding based)

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	
<i>Sensor A</i> ₁	1	0	0	0	0	✓
<i>A</i> ₂	1	0	0	0	0	✓
<i>A</i> ₃	1	0	0	0	0	✓
<i>B</i> ₁	0	1	0	0	0	✓
<i>B</i> ₂	0	1	0	0	0	✓
<i>B</i> ₃	0	1	0	0	0	✓
<i>C</i> ₁	0	0	1	0	0	✓
<i>C</i> ₂	0	0	1	0	0	✓
<i>C</i> ₃	0	0	1	0	0	✓
<i>D</i> ₁	0	0	0	1	0	✓
<i>D</i> ₂	0	0	0	1	0	✓
<i>D</i> ₃	0	0	1	0	0	×
<i>E</i> ₁	0	0	0	0	1	✓
<i>E</i> ₂	0	0	0	0	1	✓
<i>E</i> ₃	0	0	0	0	1	✓

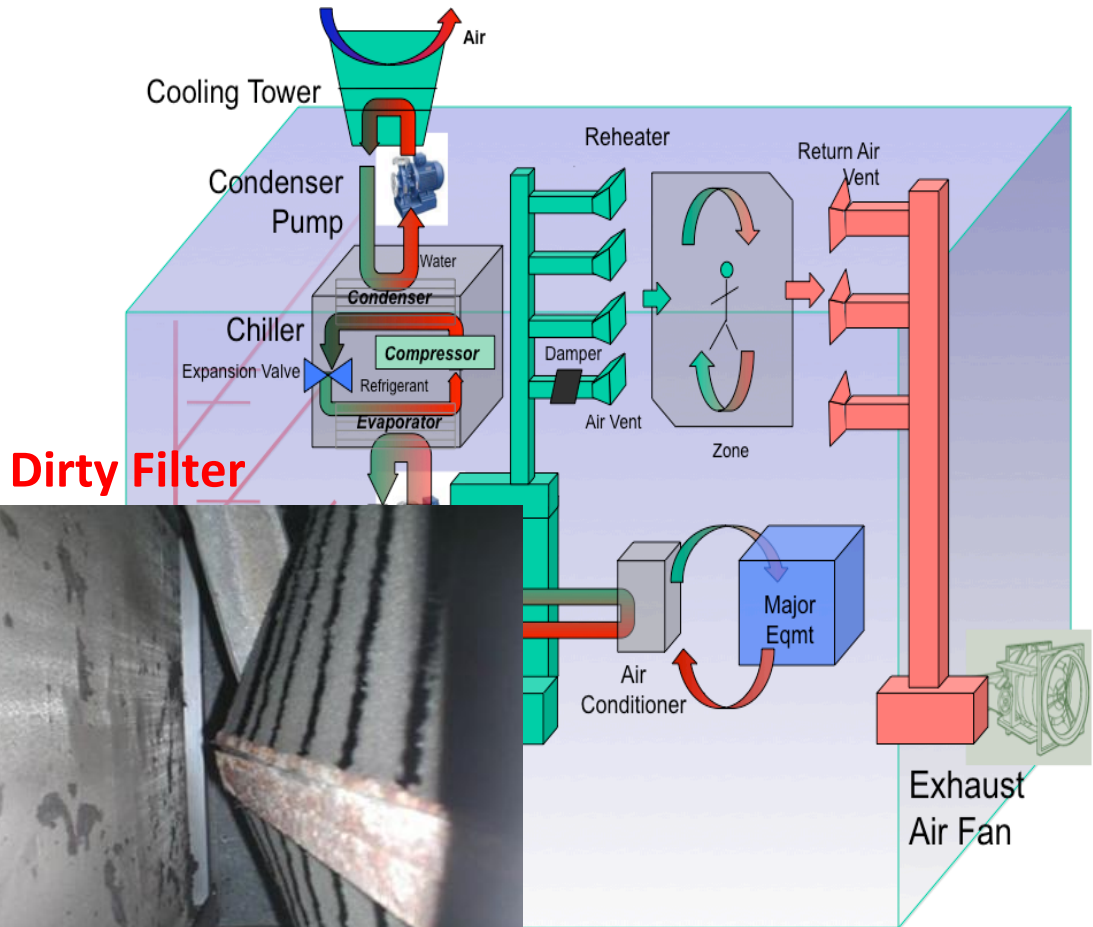
14/15 correct = 93.3%

Waste in Buildings

Lights/HVAC left on



Exhaust fan hard-wired
Always "on"



Manufacturing Infrastructure

Retro-commissioning

Table 2. Top faults causing energy inefficiencies in commercial buildings (top 13 of 100+)

	National Energy Waste (Quads, primary/year)	Electricity equivalent (BkWh/year)	Cost (\$billion/year)
Duct leakage	0.3	28.6	2.9
HVAC left on when space unoccupied	0.2	19.0	1.9
Lights left on when space unoccupied	0.18	17.1	1.7
Airflow not balanced	0.07	6.7	0.7
Improper refrigerant charge	0.07	6.7	0.7
Dampers not working properly	0.055	5.2	0.5
Insufficient evaporator airflow	0.035	3.3	0.3
Improper controls setup / commissioning	0.023	2.2	0.2
Control component failure or degradation	0.023	2.2	0.2
Software programming errors	0.012	1.1	0.1
Improper controls hardware installation	0.01	1.0	0.1
Air-cooled condenser fouling	0.008	0.8	0.1
Valve leakage	0.007	0.7	0.1
Total (central estimate)	1.0	94.6	9.6
Total (range)	0.34-1.8	32.4-171.4	3.3-17.3

Adapted from Roth et al. (2005) assuming 10,500 BTU/kWh, and \$0.10/kWh

Design Goals

1. Scalable

- Any building, any data source, everywhere
- Simple tuning

2. Range-based accuracy

- Identify segment in data to focus on and search around

Tradeoff **accuracy** for **scalability**

Expanded w/Anomaly Detection

- Learn typical correlations patterns
- Discover changes
- Return associated change points in trace

Experimental Setting

UC Berkeley

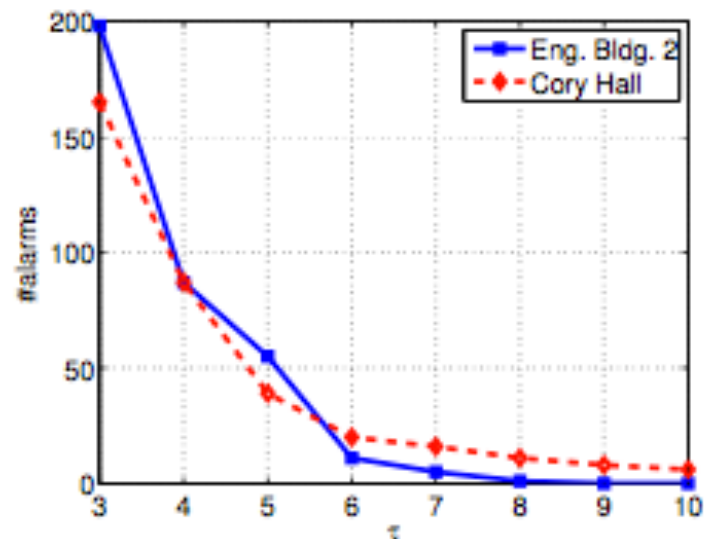


Cory Hall

- 5-story building
 - Built in 1950
- 70 sensors stream dataset
- Lighting, HVAC, elevator, electrical receptacles and panels, etc.
- 15 different HVAC systems
- 8 weeks in Spring 2011

Results

- High power usage
 - Alarms corresponding to electricity waste
- Lower power usage
 - Alarms representing abnormal low electricity consumption
- Punctual
 - Short increase/decrease in electricity consumption
- Missing data
 - Possible sensor failure
- Other
 - unknown



	High	Low	Punc.	Missing	Other
Eng. Bldg 2	9 (5)	6 (5)	1 (1)	36 (1)	3 (3)
Cory Hall	25 (7)	7 (3)	4 (4)	0 (0)	3 (3)

Alarms in Eng. Bldg 2 @Todai

Alarms in Cory Hall

Deployments



Count		Value	
No. of deployments		7	
No. of feeds and files		~10k	
Locations		University of Tokyo, UCB Cory Hall, UCB SDH, Stanford Y2E2, Intel, Nokia, Samsung	
Amount of data		~1TB (2 years)	



Future Work

- Automate verification further
 - Examine **value** and **spatial** verification
- More sophisticated **control applications**
 - MPC applications on building
- **Version control** over StreamFS
 - **Provenance checking** for streams, jobs, and control settings

Conclusion

- Device drivers
 - Standard Interfaces
- File system
 - common namespace
- Verification
 - statistical

