OpenBAS—Software-Defined Solutions for Managing Energy Use in Small to Medium-Sized Commercial Buildings

DOE Award DE-EE0006351

July 4, 2014

David Culler, Tyler Hoyt, EECS, UC Berkeley
Mark Modera, Marco Pritoni, UC Davis
Alan Meier, Francis Rubinstein, Anna Liao, LBNL
Stephen Dawson-Haggerty, Building Robotics
Therese Peffer, Karl Brown, Carl Blumstein, CIEE

The problem….

Buildings consume over 40% of the total energy consumption in the U.S.

Over 90% of the buildings are either small- (<5,000 sf) or medium-sized (between 5,000 sf and 50,000 sf).

Very few of these buildings use Building Automation Systems to monitor and control their building systems from a central location.

Thus a significant amount of energy is wasted.

– Katipamula, P et al. 2012. Small- and Medium-Sized Commercial Building Monitoring and Controls Needs: A Scoping Study. Report Number: PNNL-22169. Pacific Northwest National Laboratory: Richland, WA.

A proposed solution….

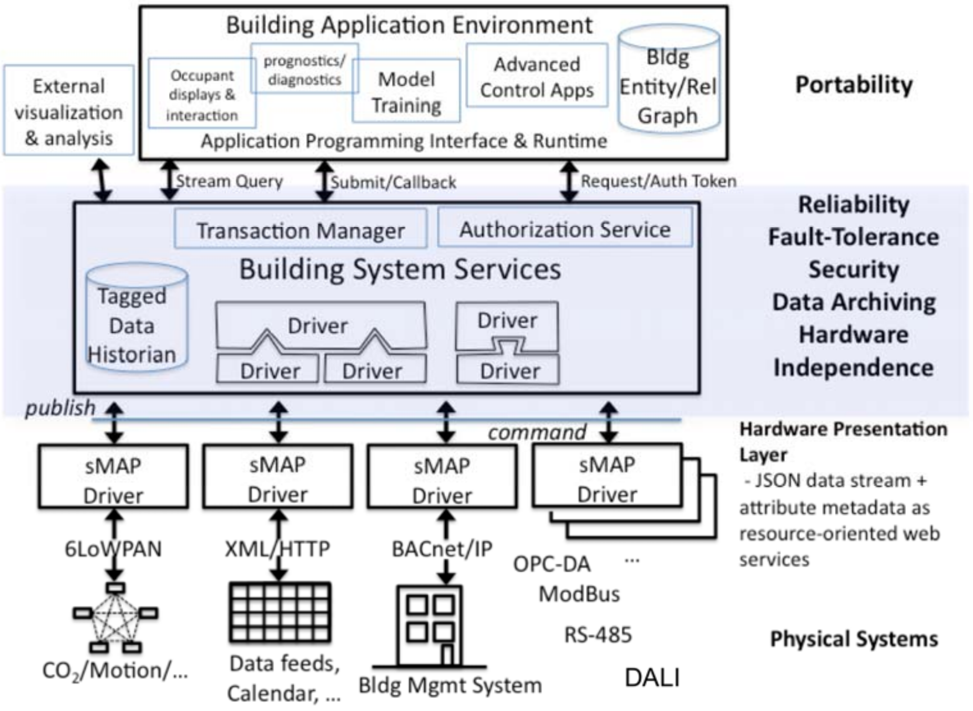
Inspired by the Internet, OpenBAS is an open software-architecture, open source Building Automation System for medium commercial buildings that uses a layered, horizontal approach to foster innovation among third party vendors.

OpenBAS:

Software-Defined Solutions for Managing Energy Use in Medium-Sized Commercial Buildings

Objective:

- Develop an open source open architecture Building Automation System (BAS) for commercial buildings < 50,000 sf.
- Develop three open source plug 'n play devices (HVAC, lighting, general)
- Develop open source user interface with system set-up, status display and auto-mapping.



Building Operating System Services (BOSS)

Team:

David Culler, Michael Andersen, Gabe Fierro, Jonathan Fuerst, Tyler Hoyt, **EECS, UC Berkeley**
Mark Modera, Marco Pritoni, **UC Davis**
Alan Meier, Francis Rubinstein, **LBNL**
Stephen Dawson-Haggerty, **Building Robotics**
Therese Peffer, Karl Brown, Carl Blumstein, **CIEE**

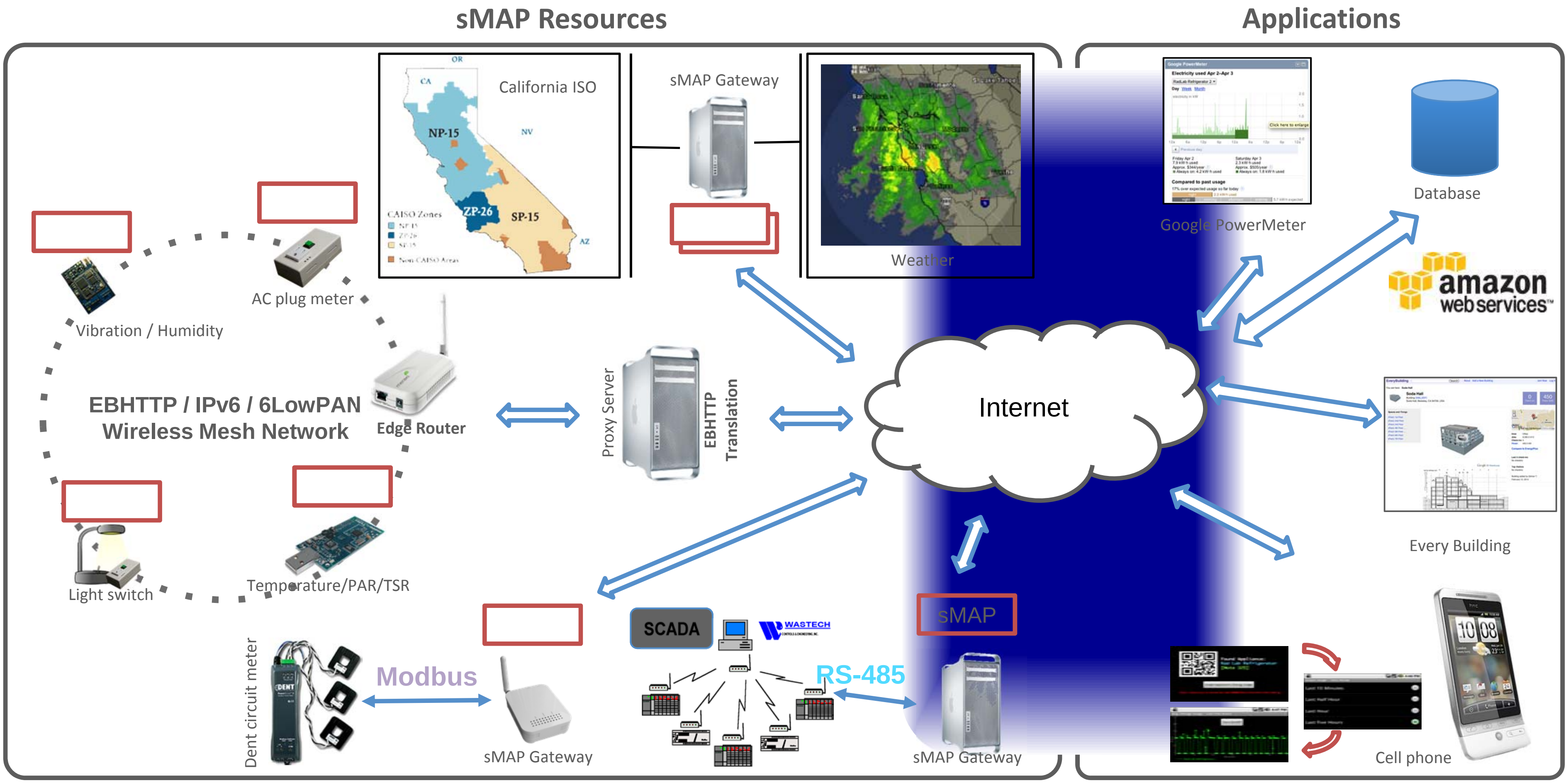
U.S. DEPARTMENT OF ENERGY

A runtime for the building

- Hardware presentation layer
  - sMAP (simple Monitoring and Actuation Profile)
  - Integrate heterogeneous monitoring, actuation, & communication substrates
- Hardware abstraction layer
  - Map between physical and virtual resources
  - Write applications in terms of relationship between hardware elements
- Time series data service
  - Archiving and querying
- Application layer
  - Portable, robust

Hardware Presentation Layer: sMAP

(Development began in 2009, now has 40+ drivers, active users group)



sMAP

- Universal information representation for physical data
  - Self-describing, compact JSON schema, transportable over UDP/TCP
  - Integrated metadata
- Software Architecture for physical data processing and actuation
  - Real-time and archival data, time-series database
  - Adapters/Drivers for legacy and direct streams
  - Subscription, syndication, distillates
  - Query processing, visualization interface
- Resource-oriented web-service framework for embedded applications

http://code.google.com/p/smap-data

Hardware Abstraction Layer

```
proc = BossProcess(timeout=15min, auth_token=ABC)
while True:
  for dmp in hal.find('#OUT_AIR_DMP > #AH'):
    for vav in hal.find('#VAV < $%s' % dmp.name):
      occ = model.estimate_occupancy(vav)
      vav.set_min_airflow((vav.min_fresh_air() /
           dmp.get_percent_open()) * occ)
  time.sleep(15*60)
```

Write applications in terms of relationship between hardware elements

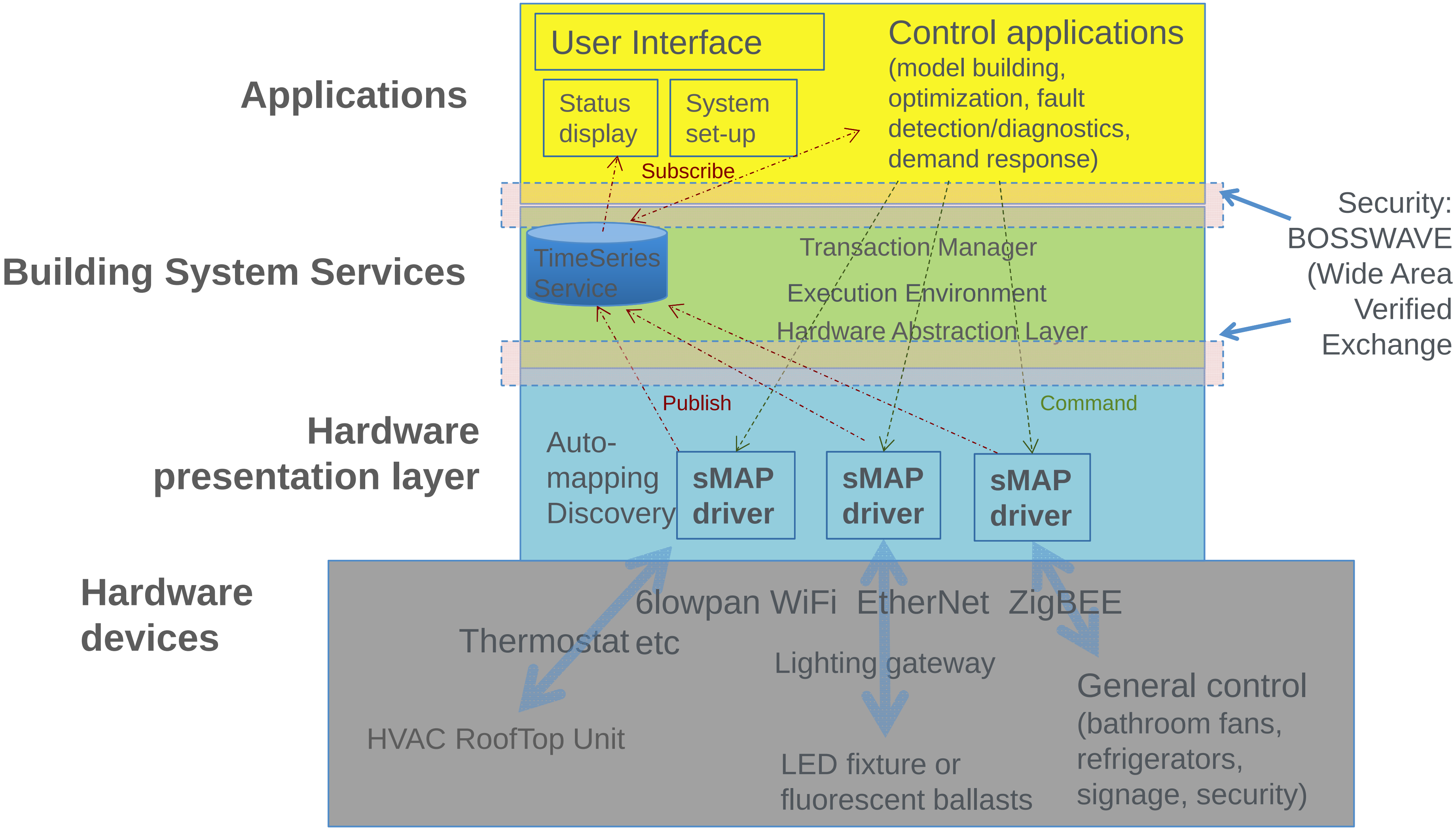
Goals and Challenges

- Portability of Applications
  - Write once, run anywhere for buildings
  - *Current practice: hand-coded logic*
- Fault tolerance
  - Partial failures of controllers
  - Network partitions
  - *Current practice: really tough hardware*
- Multiple processes
  - Concurrent applications and users
  - *Current practice: none*
- Federation
  - Multiple heterogeneous systems
  - *Current practice: lots of stovepipes*
- Scale
- Security & privacy

Security: BOSS Wide Area Verified Exchange (BOSSwave)

- Web of trust model
- Decentralized
- Push to (multiple) subscribers – not poll
- Revocation
- Verify
  - Origin, Authorization of Operation, Target
- Limit
  - Processing of unauthorized ops, bandwidth of fanout
- Tolerate
  - Intermittent connection

BOSS Software platform = backbone of OpenBAS



Applications

User Interface

Status display

System set-up

Control applications (model building, optimization, fault detection/diagnostics, demand response)

Subscribe

Building System Services

TimeSeries Service

Transaction Manager

Execution Environment

Hardware Abstraction Layer

Security: BOSSWAVE (Wide Area Verified Exchange

Publish

Command

Hardware presentation layer

Auto-mapping Discovery

sMAP driver

sMAP driver

sMAP driver

Hardware devices

6lowpan WiFi EtherNet ZigBEE

Thermostat etc

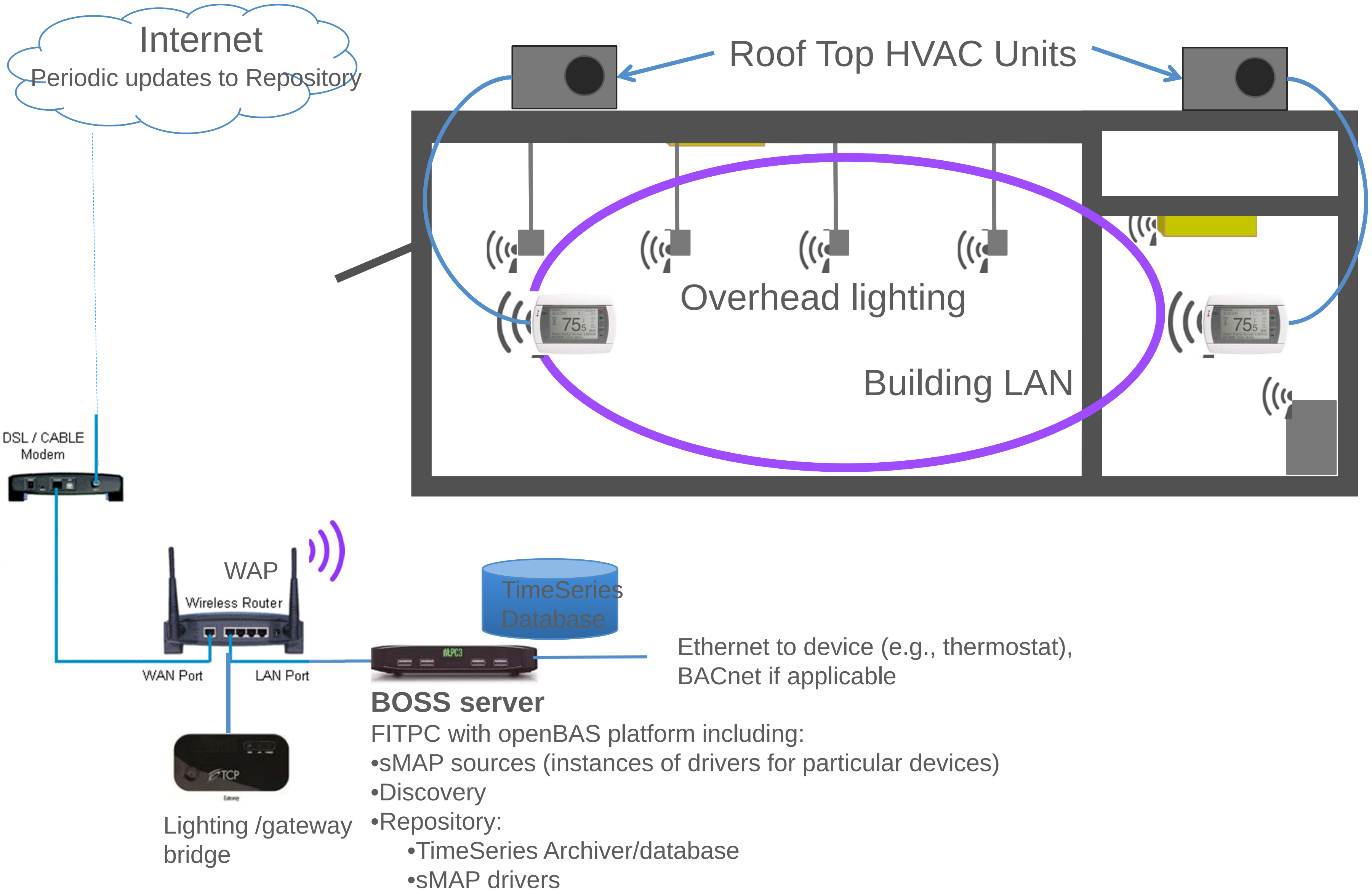
HVAC RoofTop Unit

Lighting gateway

LED fixture or fluorescent ballasts

General control (bathroom fans, refrigerators, signage, security)

Proposed openBAS



BOSS server

FITPC with openBAS platform including:

- sMAP sources (instances of drivers for particular devices)
- Discovery
- Repository:
  - TimeSeries Archiver/database
  - sMAP drivers
  - Config
  - Discovery registry

Questions?

Therese Peffer

therese.peffer@uc-ciee.org

510-289-4278