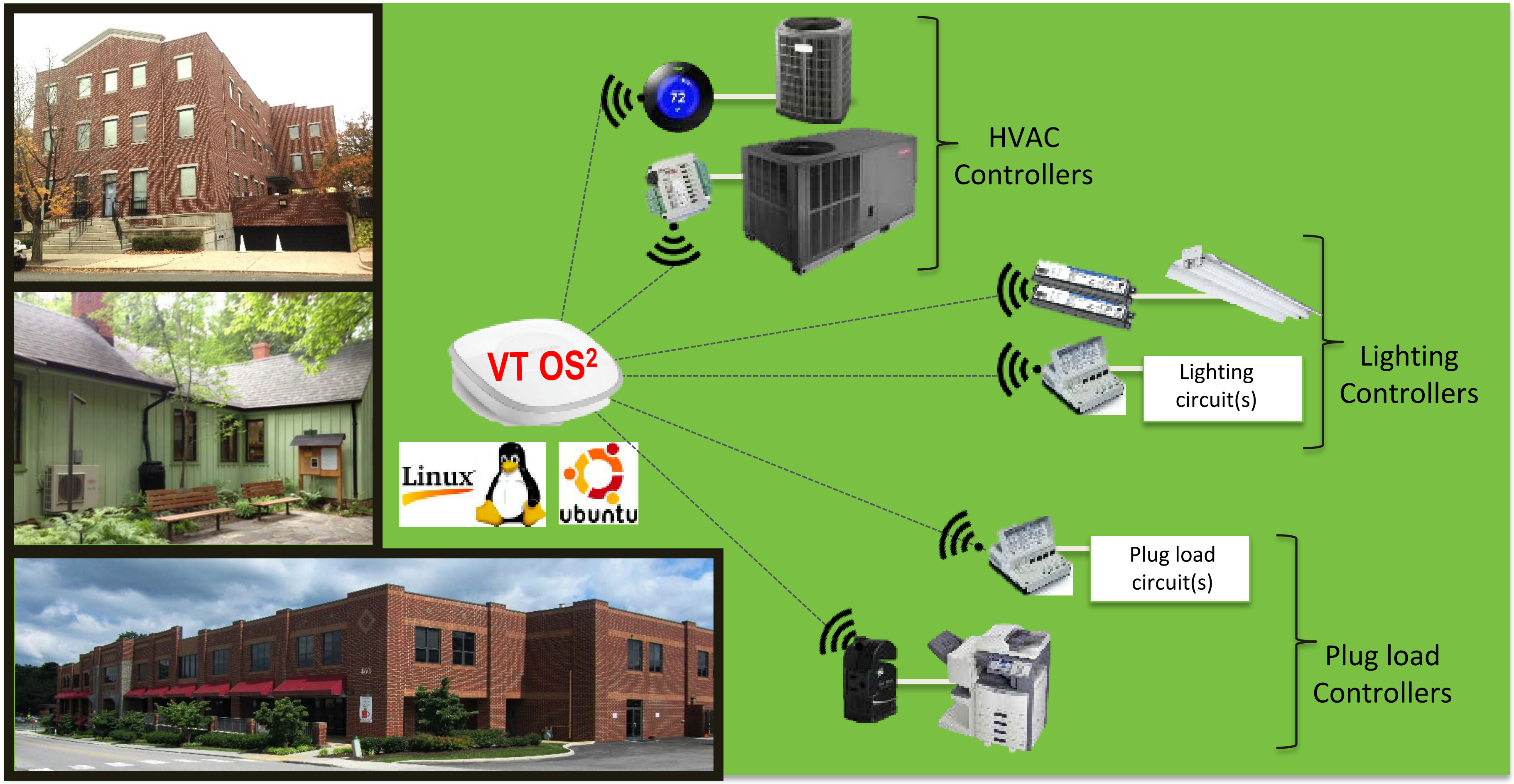
Virginia Tech Operating System (OS) built on VOLTTRON™ for Energy Management in Buildings



July 24, 2014

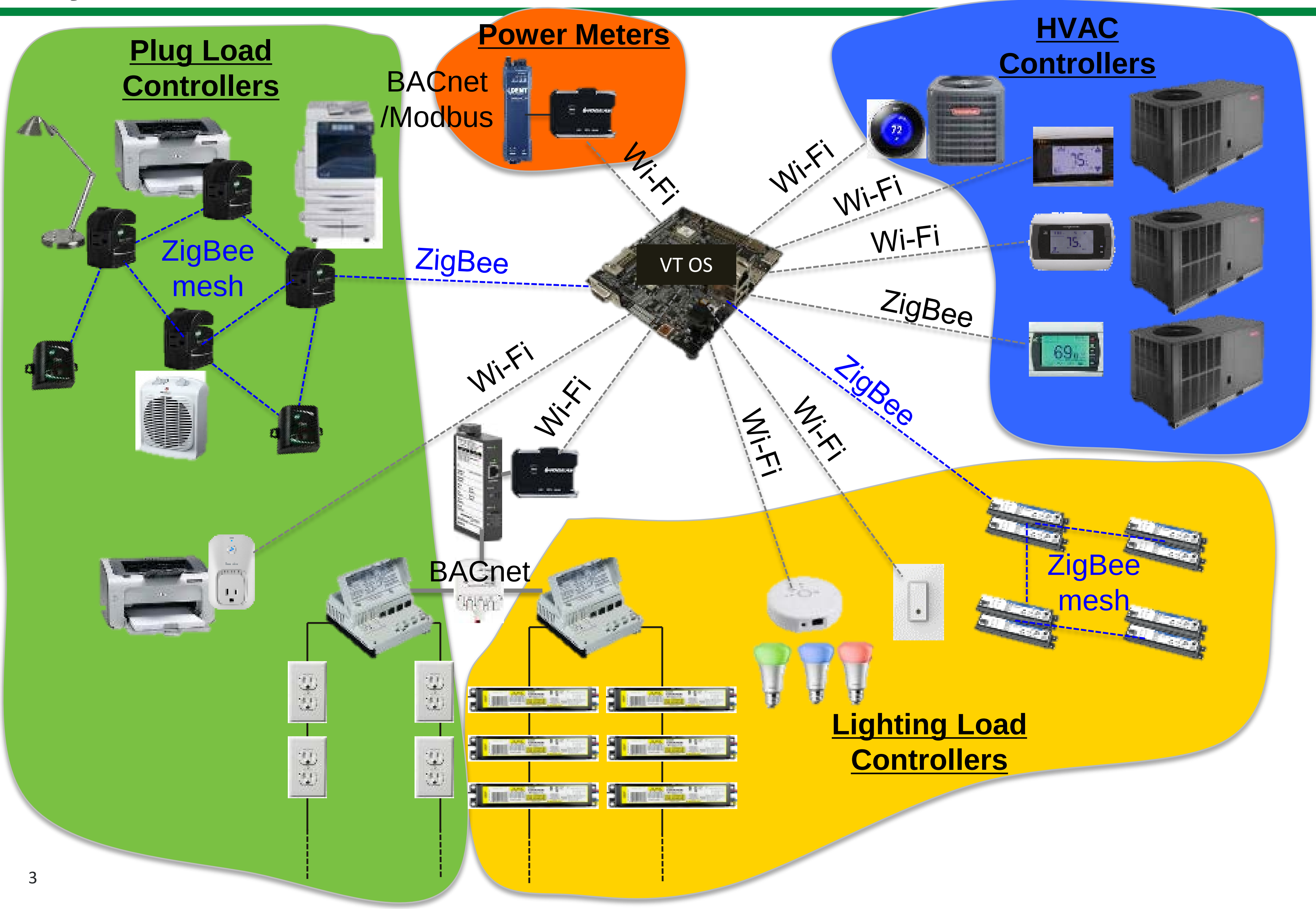
Manisa Pipattanasomporn
(mpipatta@vt.edu)
Virginia Tech

Murat Kuzlu
(mkuzlu@vt.edu)
Virginia Tech

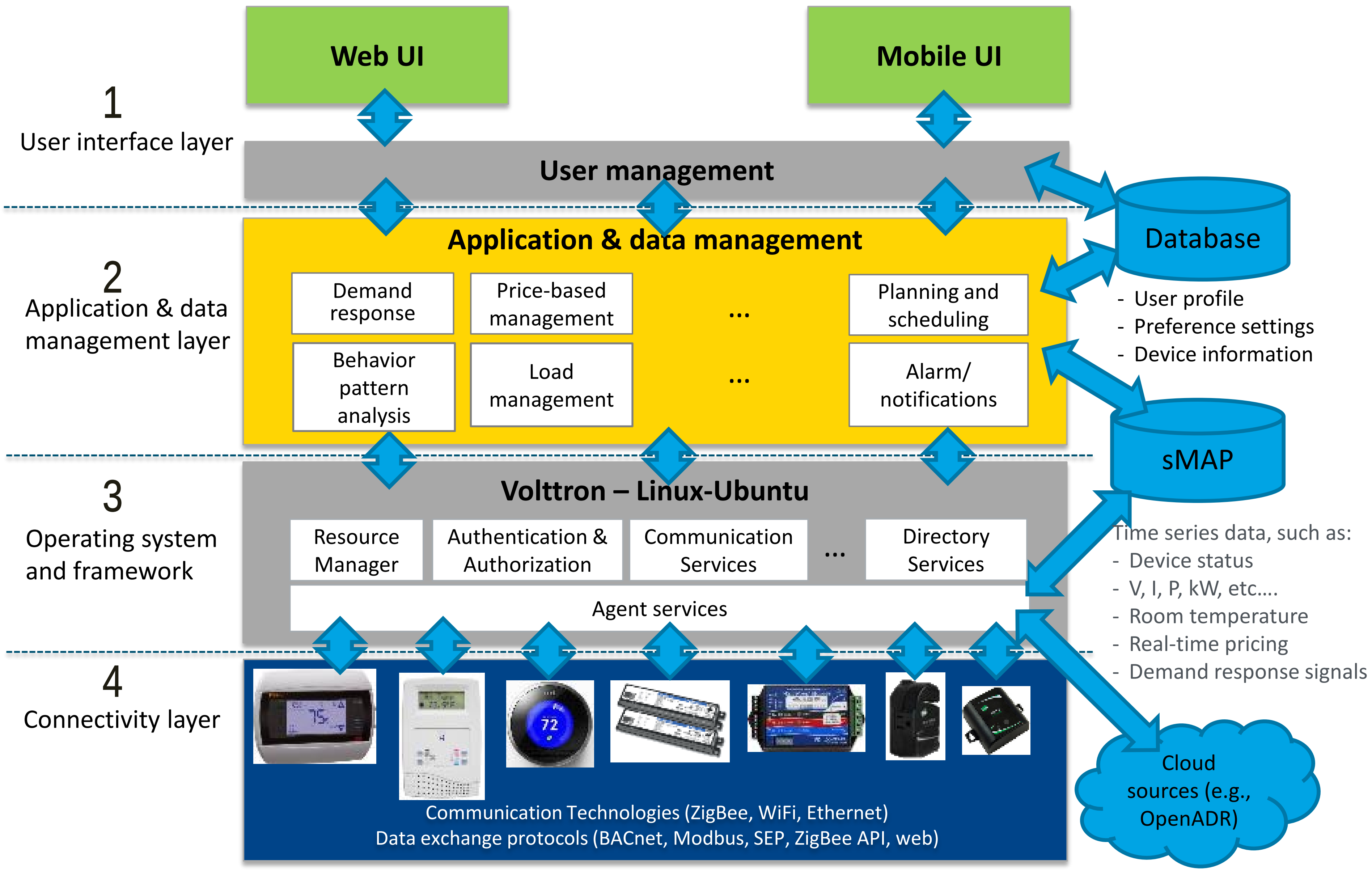
Outline

- System architecture – VT OS
- Software architecture – VT OS
- Operating system and framework layer implemented using VOLTTRON™
- Device Discovery Agent
- Live Demonstration

System Architecture – VT OS

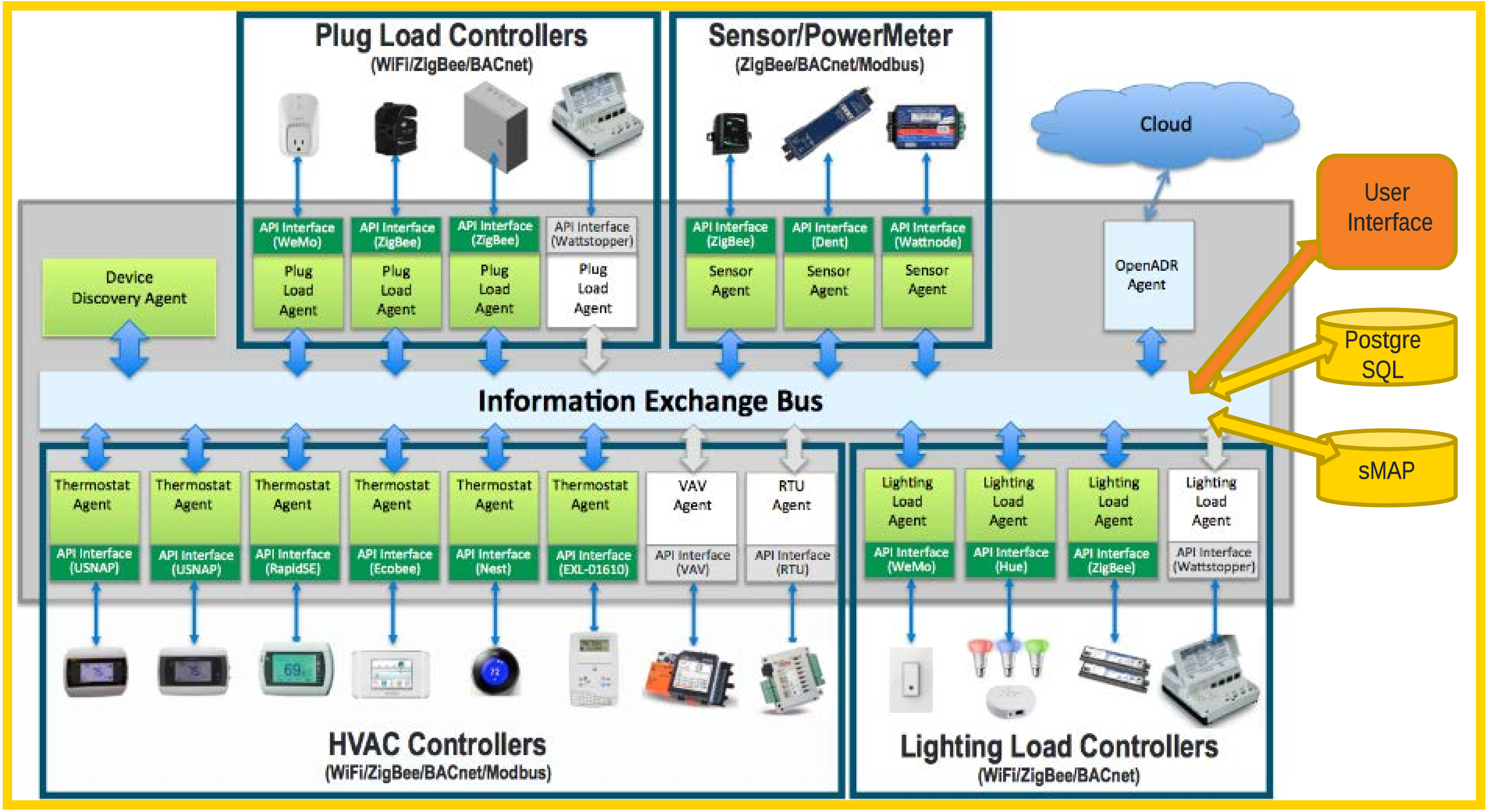


Software Architecture – VT OS



VT OS: Operating System and Framework Layer

VT OS is built upon VOLTTRON^{TM}



Device Discovery Agent

Step 1: Detect the presence of a device in the building

- Wi-Fi devices
- ZigBee devices
- ZigBee SEP devices
- BACnet devices
- Modbus devices

"I am here"

Step 2: Query the device to find out its address and model information

"This is my model number"

Step 3: Look up the device's API

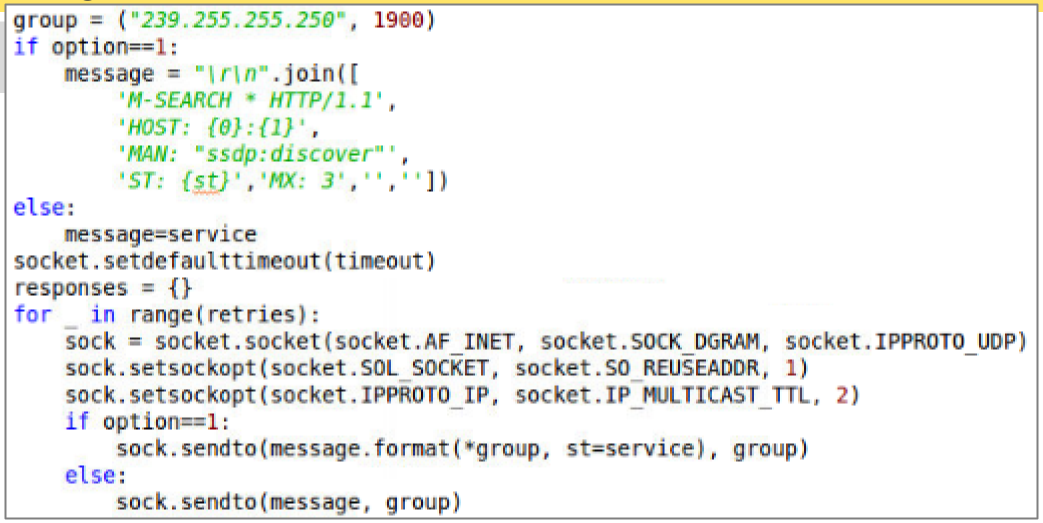
"This is what I can do"

Step 4: Initiate an agent associated with the discovered device

Device Discovery Agent Process – WiFi Devices (1 of 4)

Step 1: Use SSDP to send a discovery message to the multicast address 239.255.255.250:1900

Code to send the multicast command:



```python
group = ("239.255.255.250", 1900)
if option==1:
    message = "\r\n".join([
        'M-SEARCH * HTTP/1.1',
        'HOST: {0}:{1}',
        'MAN: "ssdp:discover"',
        'ST: {st}','MX: 3','',''])
else:
    message=service
socket.setdefaulttimeout(timeout)
responses = {}
for _ in range(retries):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, 2)
    if option==1:
        sock.sendto(message.format(*group, st=service), group)
    else:
        sock.sendto(message, group)
```

Service message for RadioThermostat:

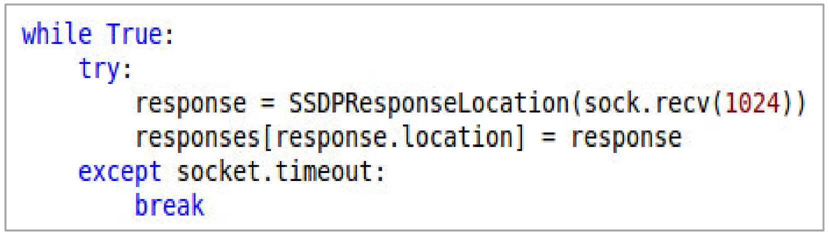
"TYPE: WM-DISCOVER\r\nVERSION: 1.0\r\n\r\nservices: com.marvell.wm.system*\r\n\r\n"

A response from RadioThermostat:

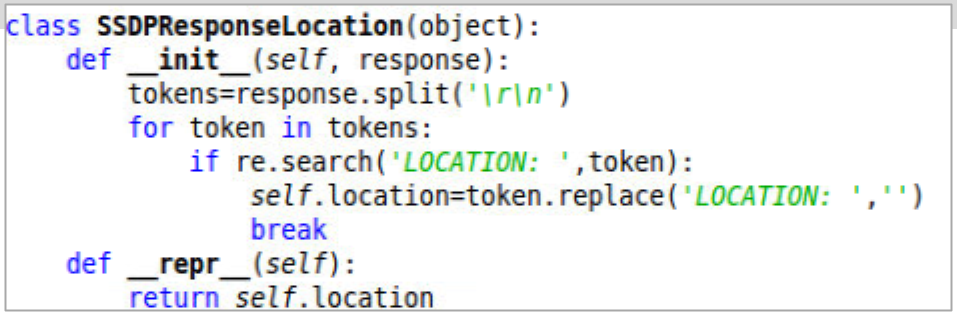
"TYPE: WM-NOTIFY\r\nVERSION: 1.0\r\n\r\nSERVICE: com.marvell.wm.system:1.0\r\nLOCATION: http://38.68.232.113/sys/\r\n\r\n"

Response includes IP

Code for parsing responses from devices:



```python
while True:
    try:
        response = SSDPResponseLocation(sock.recv(1024))
        responses[response.location] = response
    except socket.timeout:
        break
```



```python
class SSDPResponseLocation(object):
    def __init__(self, response):
        tokens=response.split('\r\n')
        for token in tokens:
            if re.search('LOCATION: ',token):
                self.location=token.replace('LOCATION: ','')
                break
    def __repr__(self):
        return self.location
```

Device Discovery Agent Process – WiFi Devices (2 of 4)

Step 2: Query device to find its MAC address and model information

Code to send GET request; and retrieve UUID from JSON response:

```
#Send GET request to device and retrieve UUID from JSON response
deviceuuidUrl = urllib2.urlopen(ipaddress)
deviceuuid=self.parseJSONresponse(deviceuuidUrl.read().decode("utf-8"),"uuid")
```

For RadioThermostat, MAC **address** can be found at: http://ip_address/sys/

A sample JSON format response:

```
{"uuid":"88308a2231de","api_version":113,"fw_version":"1.04.84","wlan_
fw_version":"v10.105576"}
```

Response includes MAC address

Parsing UUID response:

```
def parseJSONresponse(self,data,key):
    theJSON = json.loads(data)
    return theJSON[key]
```

The agent checks the MAC ID against the database, MAC ID does not match = newly discovery device. If the MAC ID is already in the database, the device was previously discovered and an agent exists for that device.

For RadioThermostat, **device model information** can be found at: http://ip_address/tstat/model/

Code to send GET request; and retrieve model number from JSON response:

```
deviceModelUrl = urllib2.urlopen(ipaddress.replace("/sys","/tstat/model"))
deviceModel = self.parseJSONresponse(deviceModelUrl.read().decode("utf-8"),"model")
```

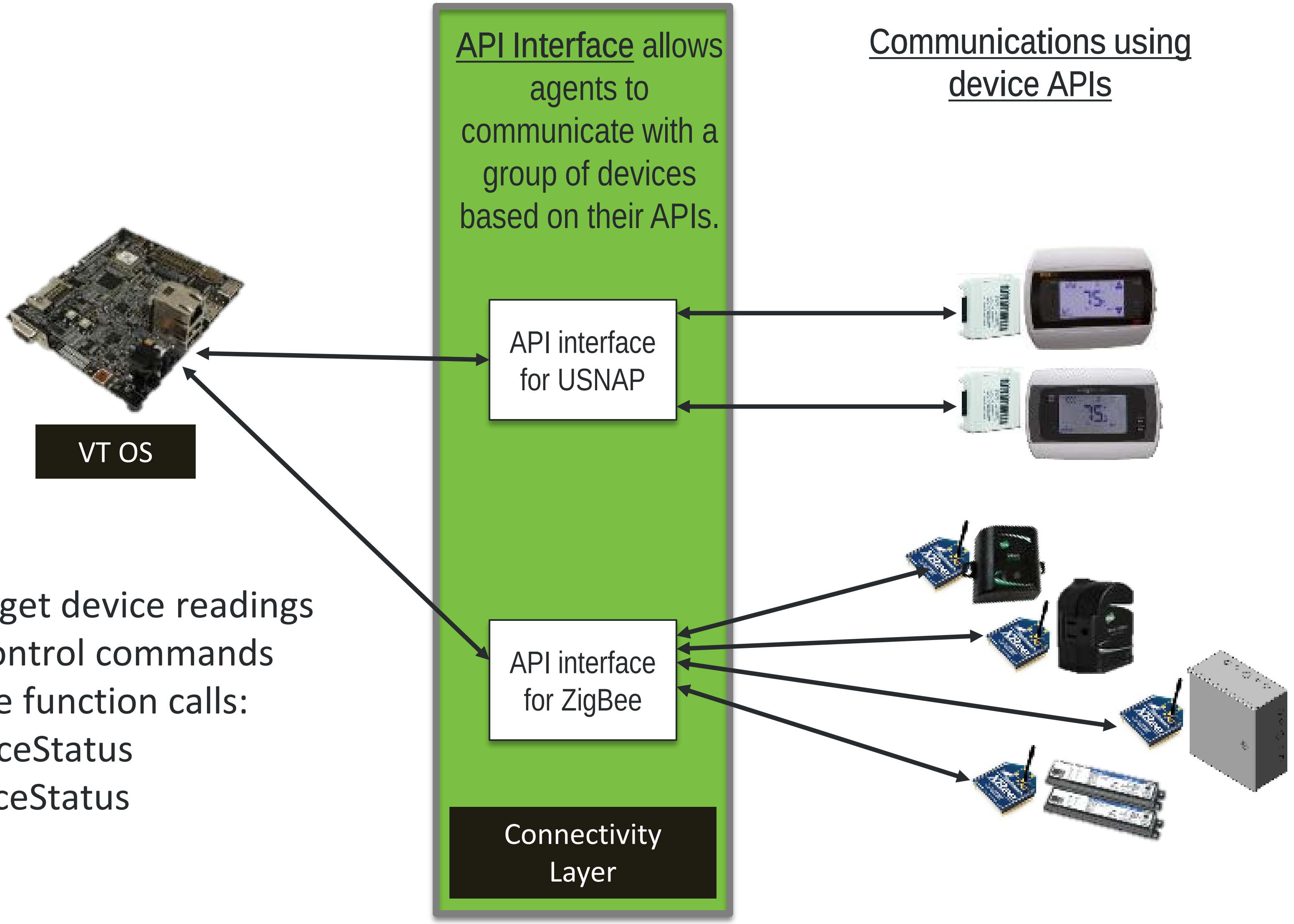
A sample JSON format response:

```
{"model":"CT50 V1.94"}
```

Response includes model name

Device Discovery Agent Process – WiFi Devices (3 of 4)

Step 3: Look up device API

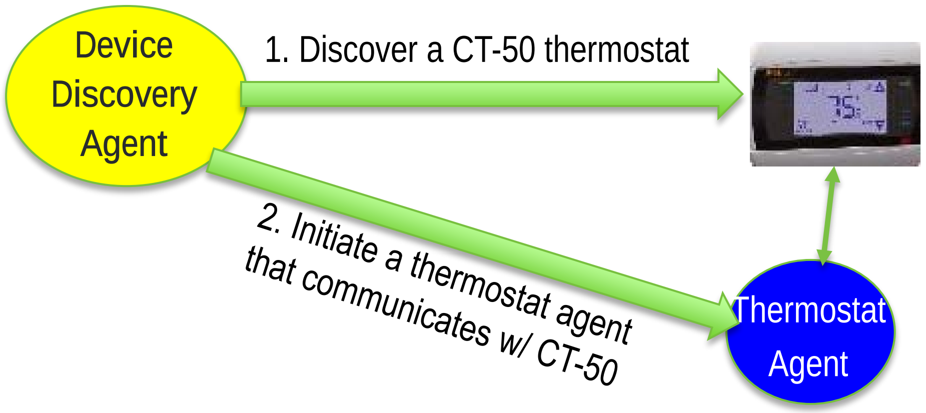

API Interface allows agents to communicate with a group of devices based on their APIs.
Communications using device APIs
API interface for USNAP
API interface for ZigBee
VT OS
Connectivity Layer

Agents can get device readings and send control commands using simple function calls:

- getDeviceStatus
- setDeviceStatus

Device Discovery Agent Process – WiFi Devices (4 of 4)

Step 4: Initiate a thermostat agent


Device Discovery Agent
1. Discover a CT-50 thermostat
2. Initiate a thermostat agent that communicates w/ CT-50
Thermostat Agent

The thermostat agent is assigned its behaviors:

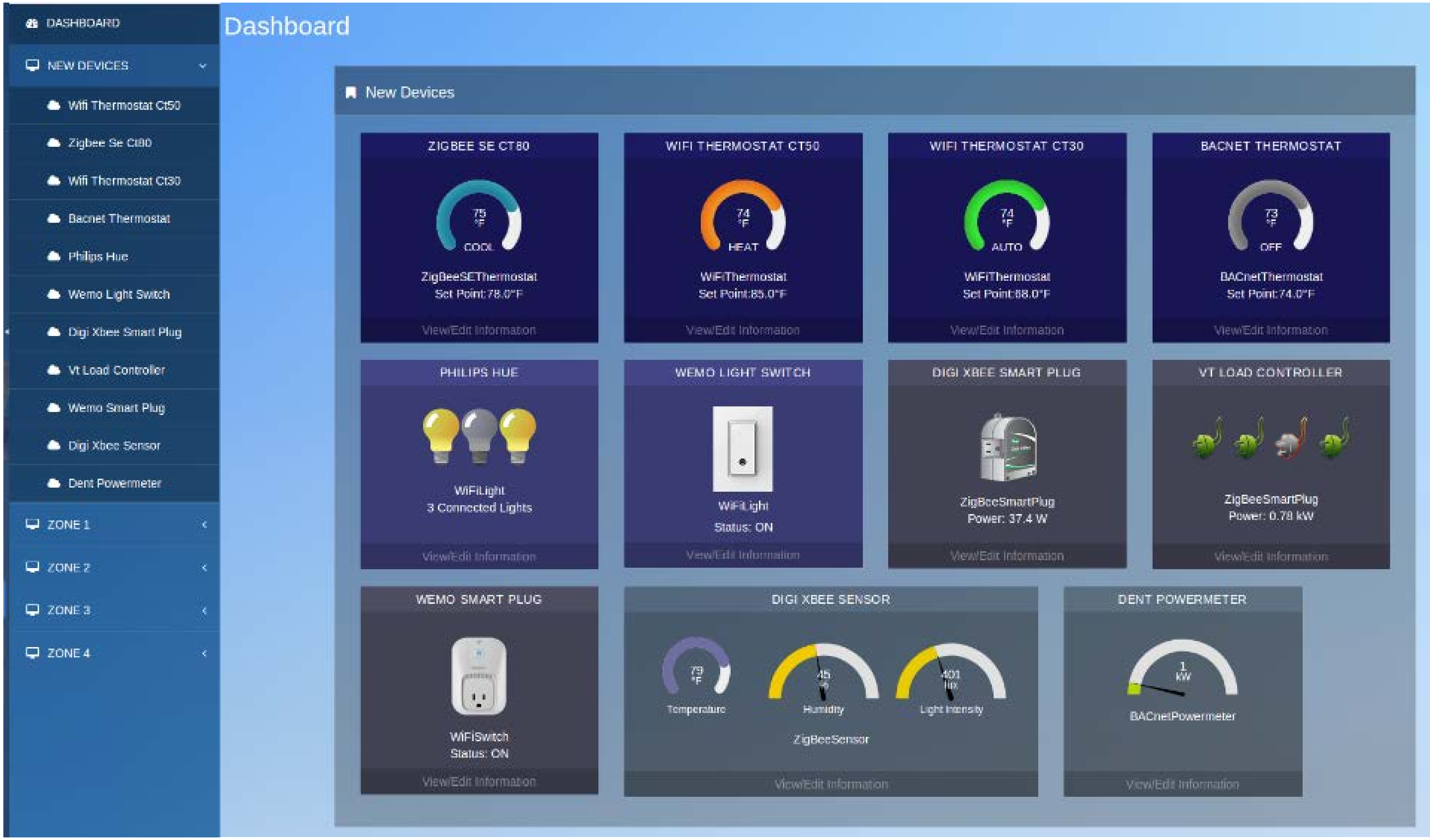
- Monitor
- Control
- Update UI

The thermostat agent is set up to:

- Publish its information in IEB
- Subscribe to relevant data sets

UI – Dashboard Page

Once the discovery agent gets device information, device discovery status is displayed in the UI.



ZIGBEE SE CT80
75 °F COOL
ZigBeeSEThermostat
Set Point:78.0°F
View/Edit Information

WIFI THERMOSTAT CT50
74 °F HEAT
WiFiThermostat
Set Point:85.0°F
View/Edit Information

WIFI THERMOSTAT CT30
74 °F AUTO
WiFiThermostat
Set Point:88.0°F
View/Edit Information

BACNET THERMOSTAT
73 °F OFF
BACnetThermostat
Set Point:74.0°F
View/Edit Information

PHILIPS HUE
WiFiLight
3 Connected Lights
View/Edit Information

WEMO LIGHT SWITCH
WiFiLight
Status: ON
View/Edit Information

DIGI XBEE SMART PLUG
ZigBeeSmartPlug
Power: 37.4 W
View/Edit Information

VT LOAD CONTROLLER
ZigBeeSmartPlug
Power: 0.78 kW
View/Edit Information

WEMO SMART PLUG
WiFiSwitch
Status: ON
View/Edit Information

DIGI XBEE SENSOR
79 °F Temperature
45 % Humidity
401 lux Light Intensity
ZigBeeSensor
View/Edit Information

DENT POWERMETER
1 kW
BACnetPowermeter
View/Edit Information

Live Demo for Device Discovery Process

Live Demo

Thank You

Manisa Pipattanasomporn
mpipatta@vt.edu

Murat Kuzlu
mkuzlu@vt.edu

Virginia Tech – Advanced Research Institute

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

Live Demo for Device Discovery Process

<u>**Step 1: Join our network**</u>

Network:	ARI_Demo
PW:	ARI_Demo

<u>**Step 2: Go to our UI**</u>

IP:	192.168.1.101:8000
Username:	admin
PW:	admin

See devices appearing on the screen as they are detected.