Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

VOLTTRON Technical Overview and Features

BRANDON CARPENTER
JEREME HAACK

Pacific Northwest National Laboratory

DOE Building Technologies Office: Technical Meeting on Software Framework for Transactive Energy
July 23-24, 2014

U.S. DEPARTMENT OF ENERGY

Presentation Outline

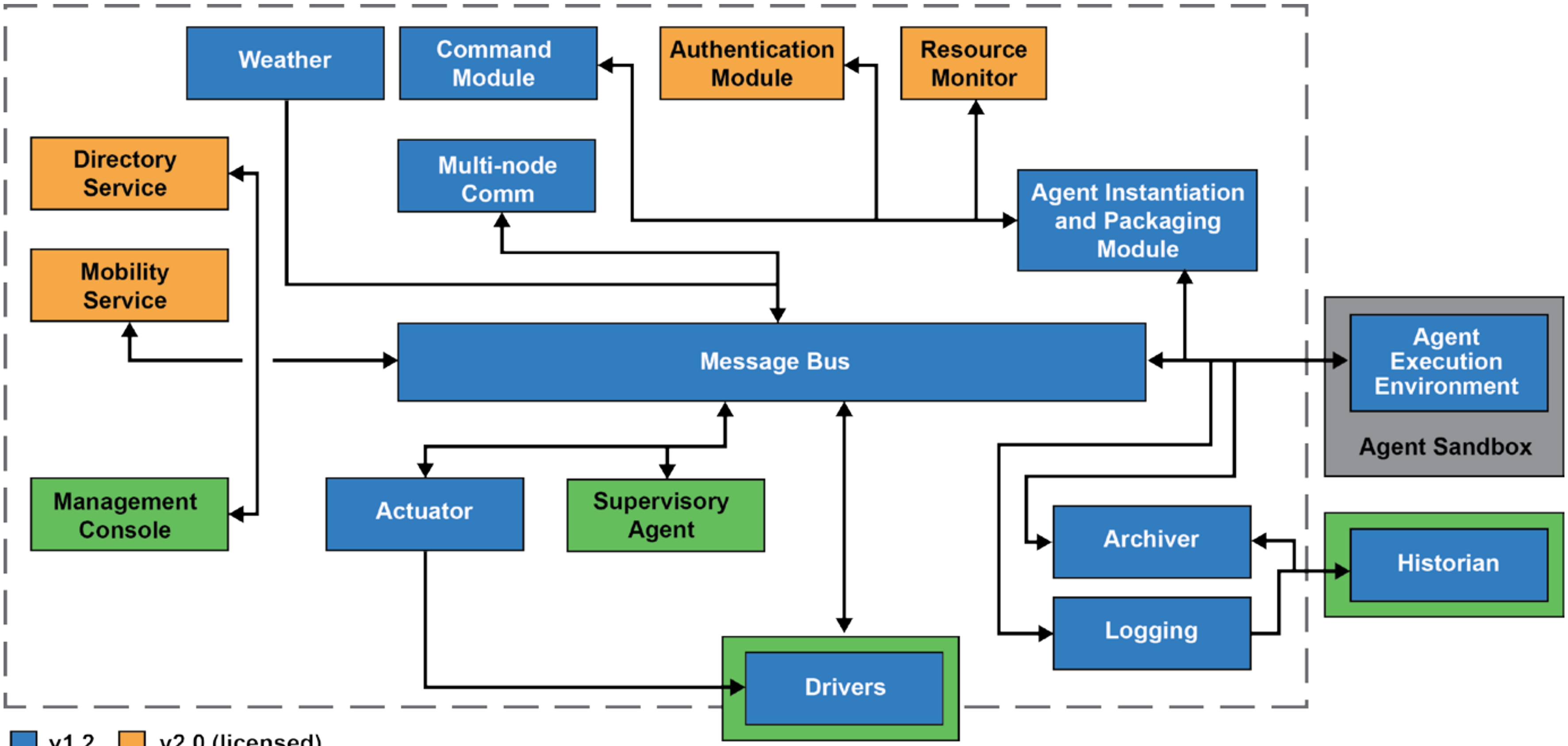
Platform Details

Platform Services

Agent Development

Future Plans

Platform Components



🟦 v1.2 🟧 v2.0 (licensed)

🟩 v3.0 ⬜ v3.0 (licensed)

Pacific Northwest
NATIONAL LABORATORY

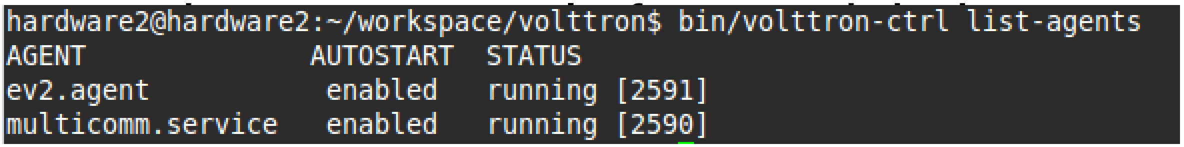
Proudly Operated by Battelle *Since 1965*

Technical Details

- Code is open source and available on github
  - https://github.com/VOLTTRON/volttron/
- Linux as target platform
- Designed to run on small form factor computers
  - PandaBoard*
  - Beagle Bone Black
  - Intel NUC
  - Desktop computer/server
- Python 2.7
  - 0MQ: Message bus (http://zeromq.org/)
  - sMAP: Used for historian and basis for drivers (https://code.google.com/p/smap-data/)
  - PyModbus: Used by driver to enable interaction with MODBUS devices. (https://code.google.com/p/pymodbus/)
  - BACPypes: Python module for BACnet communication (http://bacpypes.sourceforge.net/)
  - Twistd: Library used to start sMAP drivers (http://twistedmatrix.com)
  - Wheel: Used for agent packaging in 2.0 (https://pypi.python.org/pypi/wheel)
  - Several other libraries used

Platform Commands/Admin

- Platform provides commands for agent lifecycle
  - Build
  - Deploy
  - Start and stop agents
  - Set as autostart (start with platform starts)
  - Sign*
  - Send agent to another platform*



```
hardware2@hardware2:~/workspace/volttron$ bin/volttron-ctrl list-agents
AGENT                  AUTOSTART  STATUS
ev2.agent               enabled   running [2591]
multicomm.service       enabled   running [2590]
```

* Denotes a feature new to VOLTTRON 2.0

Presentation Outline

Platform Details

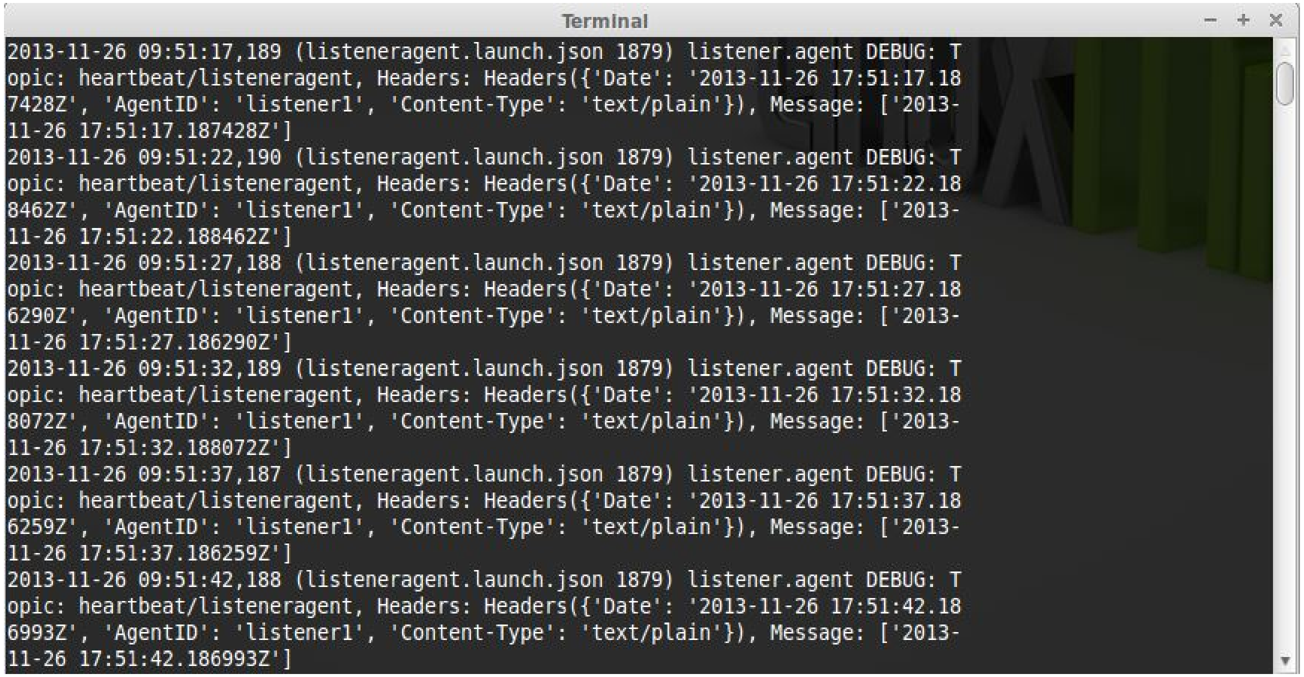
Platform Services

Agent Development

Future Plans

Message Bus

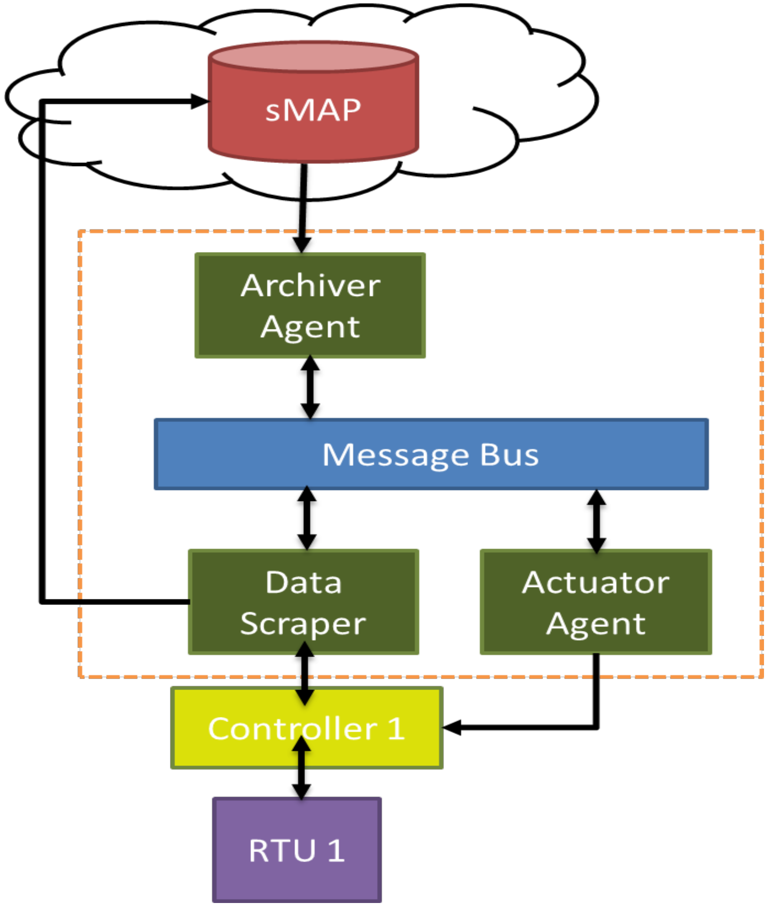
- Serves as integration point for agents, devices, external resources
  - Abstracts details of devices and agents from each other
- Uses topic based publish/subscribe
  - Campus/building/device/point
  - Can be created dynamically



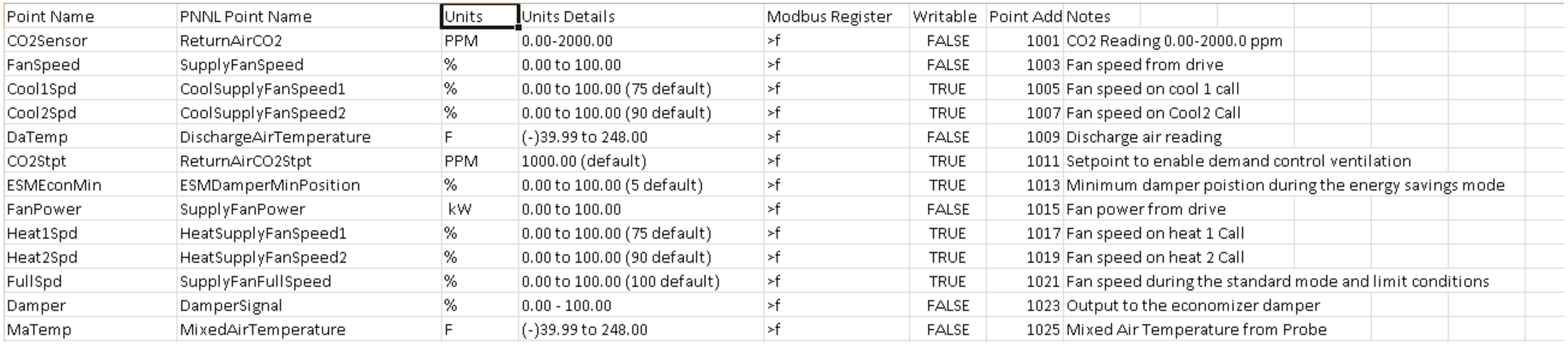
```
2013-11-26 09:51:17,189 (listeneragent.launch.json 1879) listener.agent DEBUG: T
opic: heartbeat/listeneragent, Headers: Headers({'Date': '2013-11-26 17:51:17.18
7428Z', 'AgentID': 'listener1', 'Content-Type': 'text/plain'}), Message: ['2013-
11-26 17:51:17.187428Z']
2013-11-26 09:51:22,190 (listeneragent.launch.json 1879) listener.agent DEBUG: T
opic: heartbeat/listeneragent, Headers: Headers({'Date': '2013-11-26 17:51:22.18
8462Z', 'AgentID': 'listener1', 'Content-Type': 'text/plain'}), Message: ['2013-
11-26 17:51:22.188462Z']
2013-11-26 09:51:27,188 (listeneragent.launch.json 1879) listener.agent DEBUG: T
opic: heartbeat/listeneragent, Headers: Headers({'Date': '2013-11-26 17:51:27.18
6290Z', 'AgentID': 'listener1', 'Content-Type': 'text/plain'}), Message: ['2013-
11-26 17:51:27.186290Z']
2013-11-26 09:51:32,189 (listeneragent.launch.json 1879) listener.agent DEBUG: T
opic: heartbeat/listeneragent, Headers: Headers({'Date': '2013-11-26 17:51:32.18
8072Z', 'AgentID': 'listener1', 'Content-Type': 'text/plain'}), Message: ['2013-
11-26 17:51:32.188072Z']
2013-11-26 09:51:37,187 (listeneragent.launch.json 1879) listener.agent DEBUG: T
opic: heartbeat/listeneragent, Headers: Headers({'Date': '2013-11-26 17:51:37.18
6259Z', 'AgentID': 'listener1', 'Content-Type': 'text/plain'}), Message: ['2013-
11-26 17:51:37.186259Z']
2013-11-26 09:51:42,188 (listeneragent.launch.json 1879) listener.agent DEBUG: T
opic: heartbeat/listeneragent, Headers: Headers({'Date': '2013-11-26 17:51:42.18
6993Z', 'AgentID': 'listener1', 'Content-Type': 'text/plain'}), Message: ['2013-
11-26 17:51:42.186993Z']
```

Drivers

- Drivers
  - Utilizes sMAP driver framework
  - Runs off a csv file
  - Publishes data to message bus as well as historian
  - Allows applications to interact with devices without dealing with specific protocols
- MODBUS
- BACnet
  - Configuration detection for easy configuration file construction
- Custom driver
  - Custom device interface supported as long as it publishes to the message bus

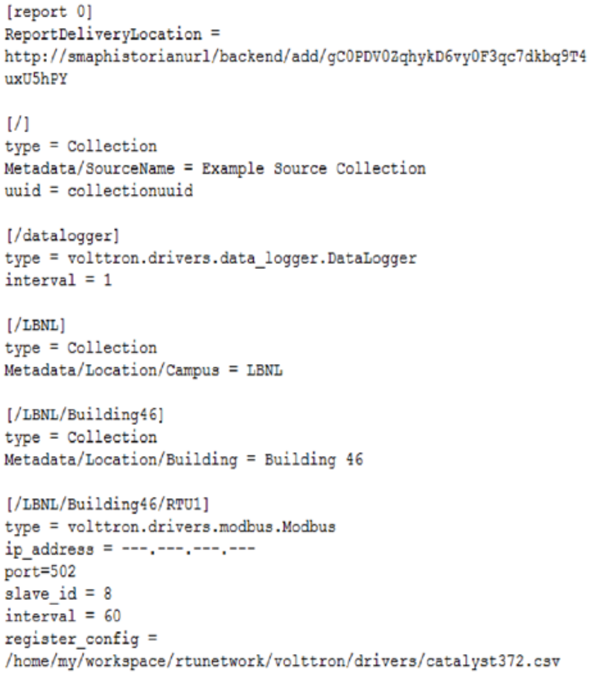


Configuring Drivers



Point Name	PNNL Point Name	Units	Units Details	Modbus Register	Writable	Point Add	Notes
CO2Sensor	ReturnAirCO2	PPM	0.00-2000.00	>f	FALSE	1001	CO2 Reading 0.00-2000.0 ppm
FanSpeed	SupplyFanSpeed	%	0.00 to 100.00	>f	FALSE	1003	Fan speed from drive
Cool1Spd	CoolSupplyFanSpeed1	%	0.00 to 100.00 (75 default)	>f	TRUE	1005	Fan speed on cool 1 call
Cool2Spd	CoolSupplyFanSpeed2	%	0.00 to 100.00 (90 default)	>f	TRUE	1007	Fan speed on Cool2 Call
DaTemp	DischargeAirTemperature	F	(-)39.99 to 248.00	>f	FALSE	1009	Discharge air reading
CO2Stpt	ReturnAirCO2Stpt	PPM	1000.00 (default)	>f	TRUE	1011	Setpoint to enable demand control ventilation
ESMEconMin	ESMDamperMinPosition	%	0.00 to 100.00 (5 default)	>f	TRUE	1013	Minimum damper poistion during the energy savings mode
FanPower	SupplyFanPower	kW	0.00 to 100.00	>f	FALSE	1015	Fan power from drive
Heat1Spd	HeatSupplyFanSpeed1	%	0.00 to 100.00 (75 default)	>f	TRUE	1017	Fan speed on heat 1 Call
Heat2Spd	HeatSupplyFanSpeed2	%	0.00 to 100.00 (90 default)	>f	TRUE	1019	Fan speed on heat 2 Call
FullSpd	SupplyFanFullSpeed	%	0.00 to 100.00 (100 default)	>f	TRUE	1021	Fan speed during the standard mode and limit conditions
Damper	DamperSignal	%	0.00 - 100.00	>f	FALSE	1023	Output to the economizer damper
MaTemp	MixedAirTemperature	F	(-)39.99 to 248.00	>f	FALSE	1025	Mixed Air Temperature from Probe

- MODBUS and BACnet drivers operate off a comma separated value (CSV) file specifying the point addresses and their configuration
- As long as the device follows the standard, there is no additional code needed. Just create this file. BACnet configuration can be auto generated
- Setup another file telling the platform to use the CSV file and name of the data topic



```
[report 0]
ReportDeliveryLocation =
http://smaphistorianurl/backend/add/gC0PDV0ZqhykD6vy0F3qc7dkbq9T4
uxU5hPY

[/]
type = Collection
Metadata/SourceName = Example Source Collection
uuid = collectionuuid

[/datalogger]
type = volttron.drivers.data_logger.DataLogger
interval = 1

[/LBNL]
type = Collection
Metadata/Location/Campus = LBNL

[/LBNL/Building46]
type = Collection
Metadata/Location/Building = Building 46

[/LBNL/Building46/RTU1]
type = volttron.drivers.modbus.Modbus
ip_address = ---.---.---.---
port=502
slave_id = 8
interval = 60
register_config =
/home/my/workspace/rtunetwork/volttron/drivers/catalyst372.csv
```

Actuator and Scheduling

- The actuator agent is responsible for sending commands to devices and for scheduling access to the device
  - Isolates agents from the specific protocol for a device
- Agents can set a priority on their device schedule request
  - High - this task cannot be preempted under any circumstance. This task may preempt other conflicting pre-emptible tasks
  - Low - this task cannot be preempted **once it has started**. This task may **not** preempt other tasks
  - Low pre-emptible - This task may be preempted at any time. This task may **not** preempt other tasks
- Actuator publishes the device schedule so all agents are aware
- In case of schedule rejection, agents are notified who is conflicting, which allows those agents to negotiate

Actuator Example Case

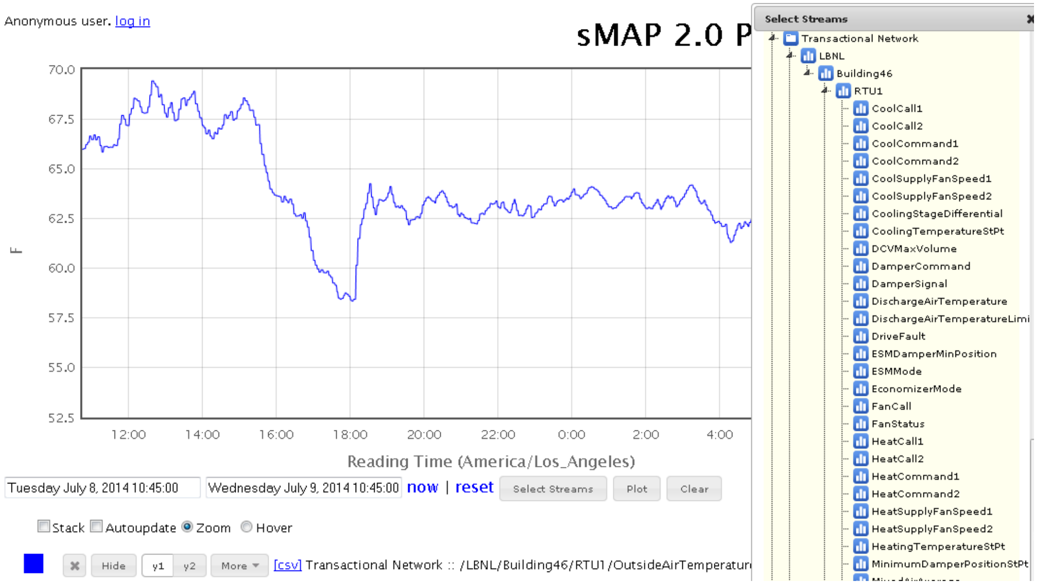
- Two agents need to control the same device
  - Fault detection agent
    - Runs at a set time at low priority
  - Demand response agent
    - No set time, but needs to run when a DR event occurs
- Fault detection is set to run at 6 p.m. at low priority
- DR event occurs at 5:30 p.m.
  - DRAgent schedules 2 hours at high priority
  - FDAgent reservation is canceled
- DR event occurs at 6:02 p.m.
  - DRAgent's reservation is rejected
  - If FD was set to Low_Preempt, then DRAgent would get priority

Deployment Considerations for Actuator

- In current deployments, building controllers expose virtual points for VOLTTRON to write to
  - Heartbeat
  - VOLTTRON_Enable flag
  - Control points
- Provides fault tolerance and also assists in acceptance

Historian/ArchiverAgent

- Currently based on sMAP
- External storage of historical data
- Accessible through the ArchiverAgent
  - Publish and receive requests over Archiver topic
  - Isolates agents from details of historian
- Back-end of the deployer's choice will be supported in VOLTTRON 3.0



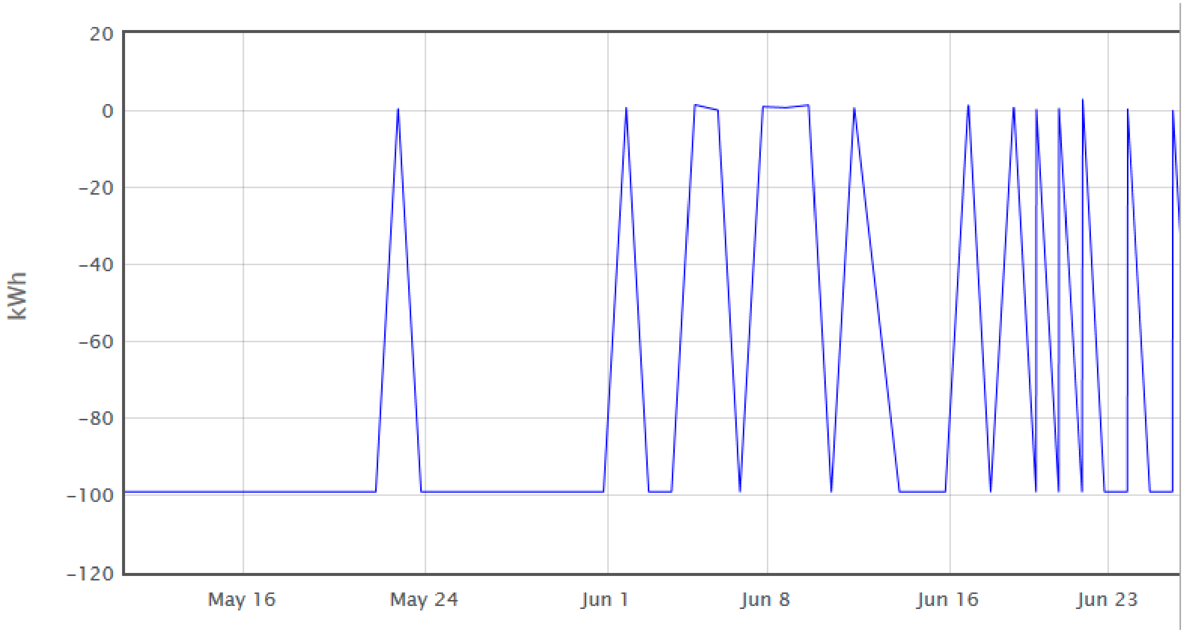
Tuesday July 8, 2014 10:45:00 | Wednesday July 9, 2014 10:45:00

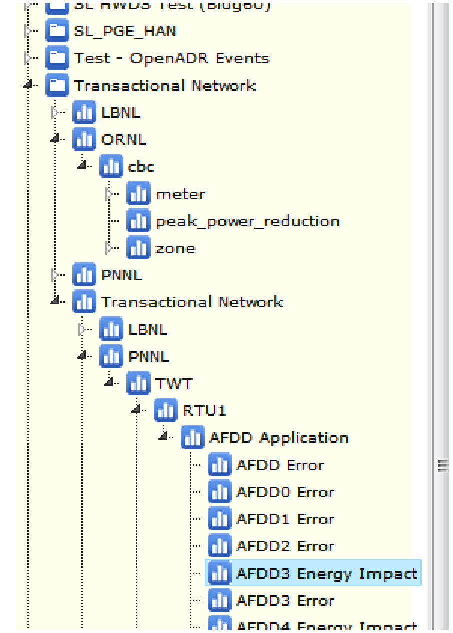
□ Stack □ Autoupdate ● Zoom ○ Hover

Transactional Network :: /LBNL/Building46/RTU1/OutsideAirTemperatur

Logging Service

- Allows agents to write to arbitrary topics for storing results, errors, etc.
  - New topics created automatically
- Other agents could retrieve results for their own use





- SL HWDS Test (Bldg60)
- SL_PGE_HAN
- Test - OpenADR Events
- Transactional Network
  - LBNL
  - ORNL
    - cbc
      - meter
      - peak_power_reduction
      - zone
  - PNNL
  - Transactional Network
    - LBNL
    - PNNL
      - TWT
        - RTU1
          - AFDD Application
            - AFDD Error
            - AFDD0 Error
            - AFDD1 Error
            - AFDD2 Error
            - AFDD3 Energy Impact
            - AFDD3 Error
            - AFDD4 Energy Impact

Presentation Outline

Platform Details

Platform Services

Agent Development

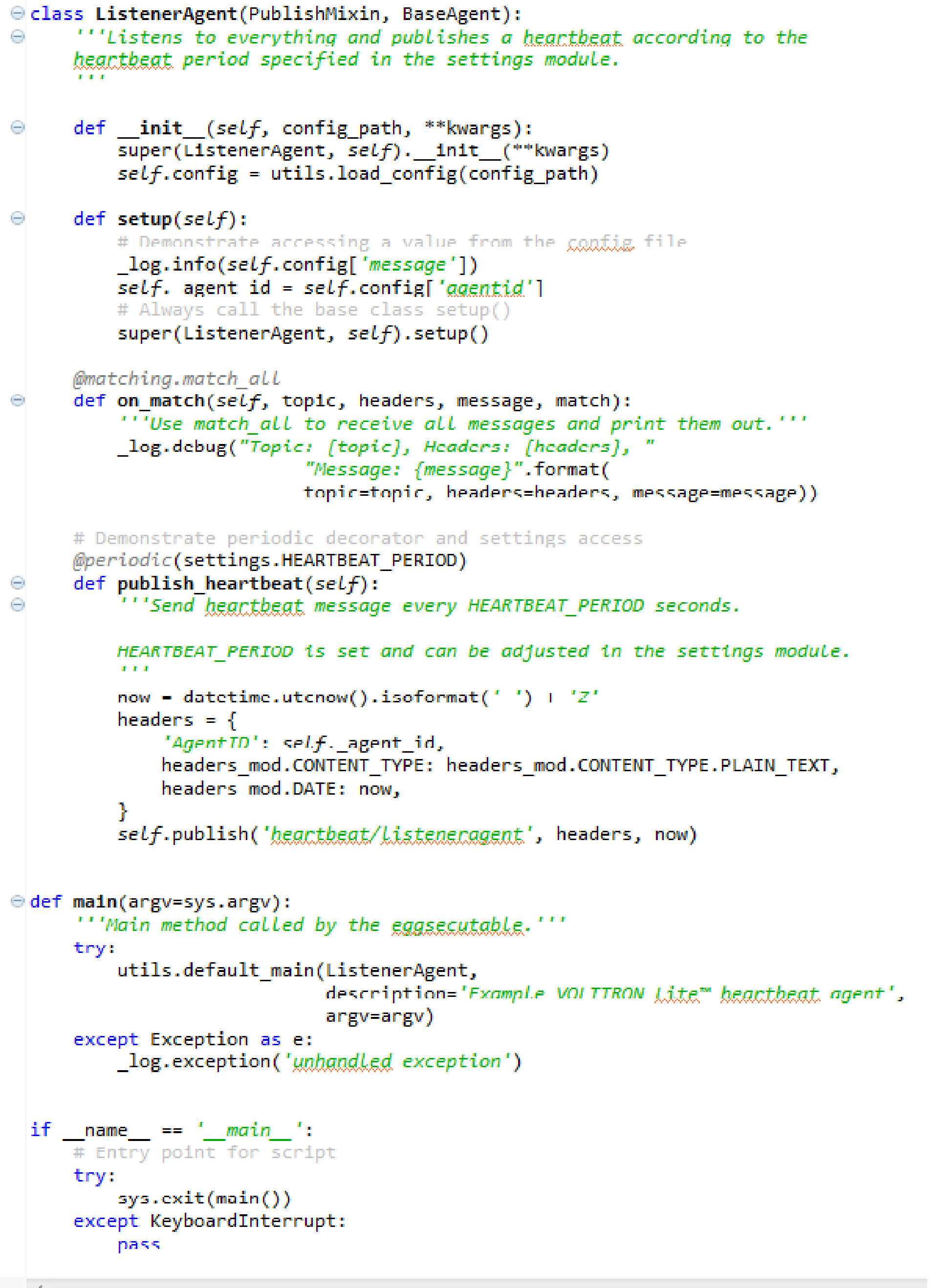
Future Plans

Agent Types

- Platform Agent
  - Provides platform services to other agents running on the platform
- Passive Agent
  - Subscribe to certain data from the buildings and other agents to perform analysis and create knowledge. Publishes results and stores in the historian
- Control Agent
  - Using data from buildings and other agents, these agents make decisions and interact with devices and other resources to achieve a goal
- Proxy Agent
  - Acts as a bridge between a remote service and agents running on the platform
  - Forwards information from the message bus and publishes messages from the remote service

Agent Development

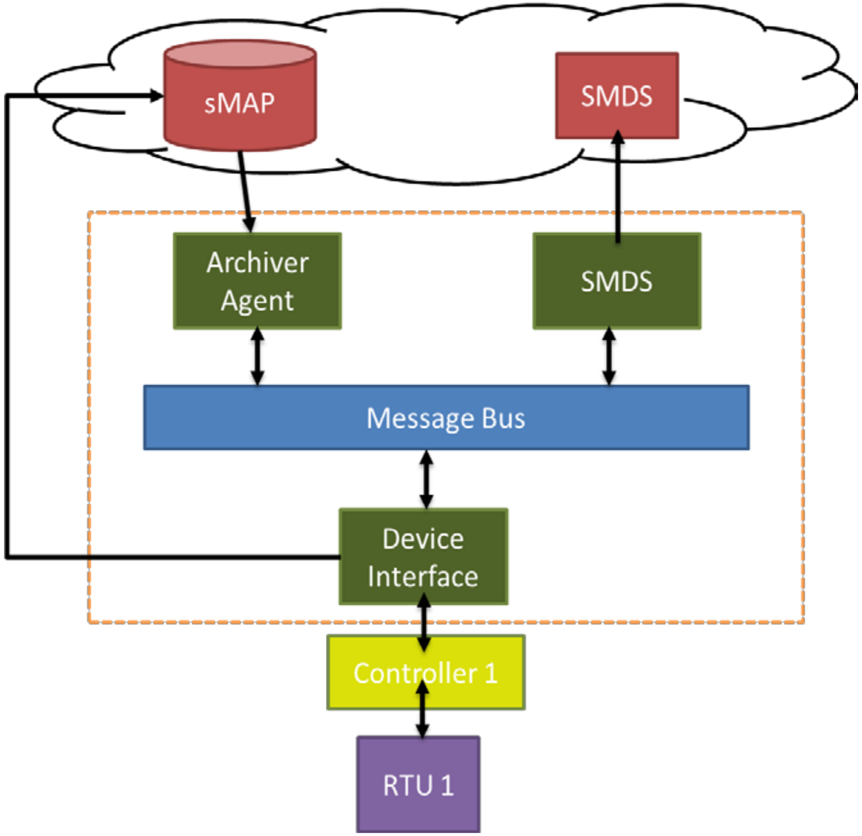
- Applications can be written in any language (even executables or scripts) as long as they can utilize the message bus
- Base class and numerous utilities for applications in Python
- BaseAgent
  - Handles basic platform interaction
  - Provides methods for agents to override for their behavior



```
class ListenerAgent(PublishMixin, BaseAgent):
    '''Listens to everything and publishes a heartbeat according to the
    heartbeat period specified in the settings module.
    '''

    def __init__(self, config_path, **kwargs):
        super(ListenerAgent, self).__init__(**kwargs)
        self.config = utils.load_config(config_path)

    def setup(self):
        # Demonstrate accessing a value from the config file
        _log.info(self.config['message'])
        self._agent_id = self.config['agentid']
        # Always call the base class setup()
        super(ListenerAgent, self).setup()

    @matching.match_all
    def on_match(self, topic, headers, message, match):
        '''Use match_all to receive all messages and print them out.'''
        _log.debug("Topic: {topic}, Headers: {headers}, "
                   "Message: {message}".format(
                   topic=topic, headers=headers, message=message))

    # Demonstrate periodic decorator and settings access
    @periodic(settings.HEARTBEAT_PERIOD)
    def publish_heartbeat(self):
        '''Send heartbeat message every HEARTBEAT_PERIOD seconds.

        HEARTBEAT_PERIOD is set and can be adjusted in the settings module.
        '''
        now = datetime.utcnow().isoformat(' ') + 'Z'
        headers = {
            'AgentID': self._agent_id,
            headers_mod.CONTENT_TYPE: headers_mod.CONTENT_TYPE.PLAIN_TEXT,
            headers_mod.DATE: now,
        }
        self.publish('heartbeat/listeneragent', headers, now)


def main(argv=sys.argv):
    '''Main method called by the eggsecutable.'''
    try:
        utils.default_main(ListenerAgent,
                           description='Example VOLTTRON Lite™ heartbeat agent',
                           argv=argv)
    except Exception as e:
        _log.exception('unhandled exception')


if __name__ == '__main__':
    # Entry point for script
    try:
        sys.exit(main())
    except KeyboardInterrupt:
        pass
```

Application Example

- Data collecting device interface takes readings from the HVAC controller every minute and both pushes that data to sMAP and publishes out on the Message Bus
- Every hour, the SMDS Proxy agent publishes a request to the Archiver agent for the last hour of controller data for the points: unit power, supply fan speed, and outdoor air temperature
- Archiver agent queries sMAP and publishes the results on the Message Bus
- The SMDS agent receives its data, reformats it, and the pushes it to the SMDS application in the Cloud

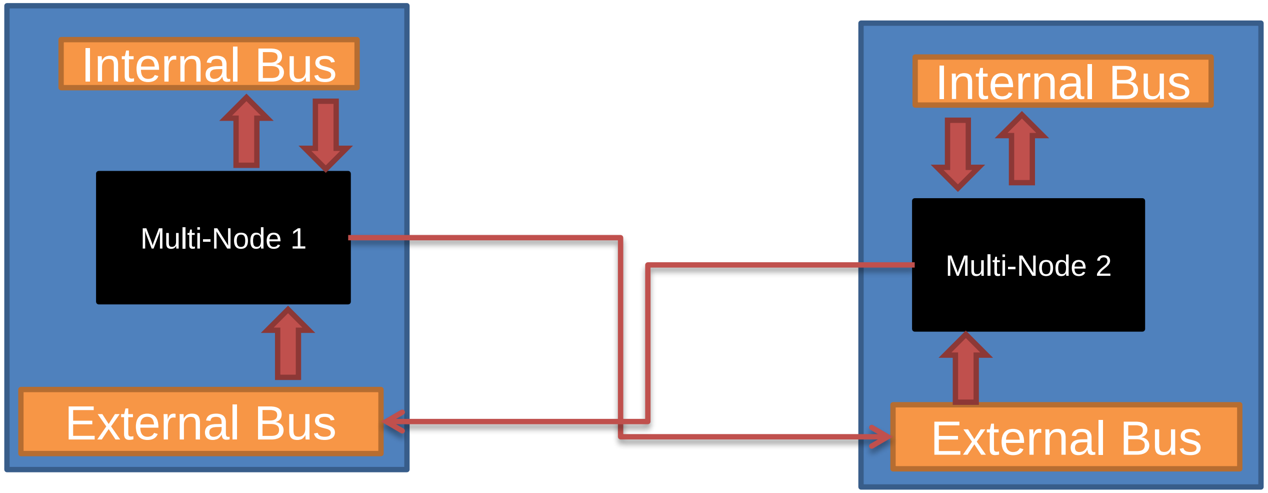


Security Synopsis

- VOLTTRON Open Source
  - Encrypted multi-node communication
  - SSL to external resources supported
- VOLTTRON PNNL IP
  - Agent validation – Signed agent code validated before execution
  - Agent packaging – Agent code and files signed at each stage of development/deployment
    - Python Wheel supports x509 certs
  - Resource management – Agents present resource contract. Platform determines if it can support agent and manages resource utilization during agent execution
    - Uses Linux control groups (cgroups)
    - Current managed resources include memory and CPU utilization
    - Future resources could include disk and network
- VOLTTRON 3.0+ Proposed Security Enhancement
  - Hardening and penetration testing
  - Sandboxing applications

Multi-Node Communication

- Implemented as a service agent allowing it to easily be enabled or disabled. Configured using the agent's configuration file
- Allows platforms to communicate with each other
  - Sets up a subscribing topic that other platforms can publish to
- Currently, setup in the multi-node service agent configuration file

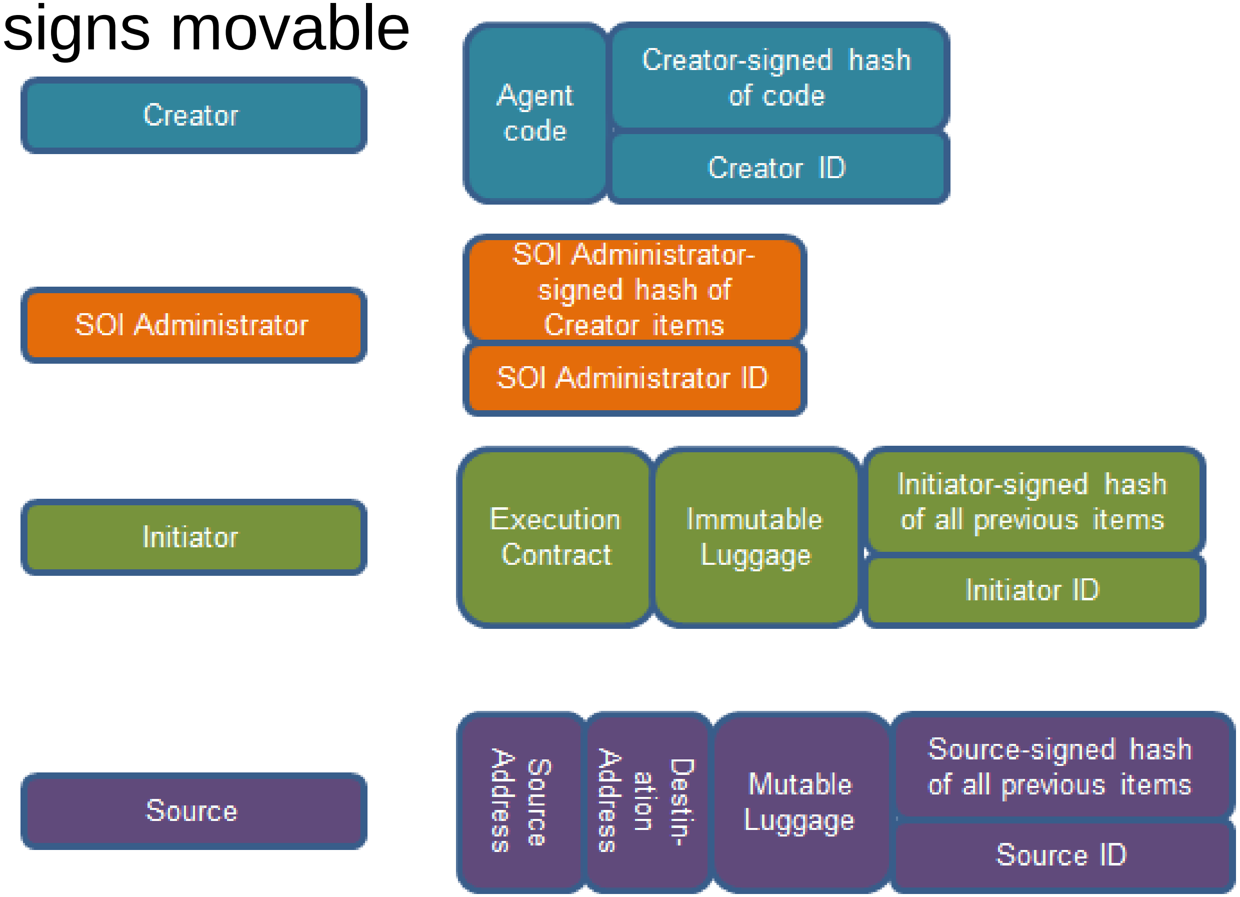


Pacific Northwest NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

Agent Transport Payload

- Entity creates agent
- Administrator signs
- Initiator adds configuration info and sends into system
- Each source signs movable content



Presentation Outline

- Platform Details
- Platform Services
- Agent Development
- **Future Plans**

VOLTTRON 2.0 and Beyond

- VOLTTRON 2.0 based on porting over and refining features from original PNNL developed version of VOLTTRON
  - Resource management
  - Enhanced security
  - Agent mobility
- VOLTTRON 3.0/FY15
  - Centralized management console
  - Enhanced modularization to easily swap out different component implementations
  - Supervisory agent
  - Scalability study
  - Penetration testing and security enhancements
  - User preferred historian
  - Large scale campus-wide testing/demonstration

VOLTTRON Development Support

The VOLTTRON software developers want VOLTTRON users to be successful. They are here to help!

- VOLTTRON community portal
  - Extensive documentation, use cases, and training materials including videos
  - VOLTTRON getting started SDK (software development kit) including available virtual machines that are pre-configured
  - VOLTTRON source code
  - VOLTTRON application store
- Periodic VOLTTRON users meetings

Questions?

- VOLTTRON Resources
  - Wiki: https://github.com/VOLTTRON/volttron/wiki
  - Email: volttron@pnnl.gov
  - Bi-monthly office hours