PNNL-25499

Pacific Northwest NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

VOLTTRON™ 2016

June 2016

B Akyol
CH Allwardt
ZW Beech
JB Chapman
JN Haack
S Katipamula
RG Lutes
KE Monson

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

This document was printed on recycled paper.
(8/2010)

PNNL-25499

VOLTTRON™ 2016

B Akyol
CH Allwardt
ZW Beech
JB Chapman
JN Haack
S Katipamula
RG Lutes
KE Monson

June 2016

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Abstract

VOLTTRON is a flexible, reliable, and scalable platform for distributed control and sensing. VOLTTRON serves in four primary roles:

- A reference platform for researchers to quickly develop control applications for transactive energy.
- A reference platform with flexible data store support for energy analytics applications either in academia or in commercial enterprise.
- A platform from which commercial enterprise can develop products without license issues and easily integrate into their product line.
- An accelerator to drive industry adoption of transactive energy and advanced building energy analytics.

Pacific Northwest National Laboratory, with funding from the U.S. Department of Energy's Building Technologies Office, developed and maintains VOLTTRON as an open-source community project. VOLTTRON source code includes agent execution software; agents that perform critical services that enable and enhance VOLTTRON functionality; and numerous agents that utilize the platform to perform a specific function (fault detection, demand response, etc.). The platform supports energy, operational, and financial transactions between networked entities (equipment, organizations, buildings, grid, etc.) and enhance the control infrastructure of existing buildings through the use of open-source device communication, control protocols, and integrated analytics.

Acronyms and Abbreviations

API	application program interface
BTO	Building Technologies Office
CETC	clean energy transactive campus
DOE	U.S. Department of Energy
IFTTT	if this than that
PLC	Programmable Logic Controllers
PNNL	Pacific Northwest National Laboratory
R&D	research and development
UW	University of Washington
VIP	VOLTTRON Interconnect Protocol
WSU	Washington State University

Contents

Figures

Tables

1.0 What is VOLTTRON?

VOLTTRON is an open-source distributed control and sensing platform for integrating buildings and the power grid. VOLTTRON connects devices, agents in the platform, agents in the Cloud, and signals from the power grid. The platform also supports use cases such as demand response and integration of distributed renewable energy sources.

VOLTTRON provides an environment for agent execution and serves as a single point of contact for interfacing with devices (rooftop units, building systems, meters, etc.), external resources, and platform services such as data archival and retrieval. VOLTTRON applications are referred to as agents since VOLTTRON provides an agent-based programming paradigm to ease application development and minimize the lines of code that need to be written by domain experts such as buildings engineers. VOLTTRON provides a collection of utility and helper classes that simplifies agent development.

Pacific Northwest National Laboratory (PNNL), with funding from the U.S. Department of Energy's (DOE's) Building Technologies Office (BTO) developed and maintains VOLTTRON as a reference transactional network platform to support buildings and grid integration. VOLTTRON source code includes agent execution software; agents that perform critical services that enable and enhance the functionality of VOLTTRON; and numerous agents that utilize the platform to perform a specific function (fault detection, demand response, etc.). The platform supports energy, operational, and financial transactions between networked entities (equipment, organizations, buildings, grid, etc.) and enhances the control infrastructure of existing buildings through the use of open-source device communication, control protocols, and integrated analytics.

To encourage development and growth of the platform, all VOLTTRON software, platform services, and agents are open-source and employ a BSD (Berkeley Software Distribution) style license, allowing the free distribution and development of the software. Contributing back to the project, which is encouraged but not required, enhances its capabilities for the whole community. The license supports organizations developing proprietary solutions on top of the open-source code.

A separately licensed (by PNNL) platform plugin provides additional capabilities to the platform for large-scale, real-world deployments. These enhancements include agent mobility, signing and verification of agents, and resource management. This plugin allows actors to sign agent packages as an acknowledgement that the code is useful, reliable, and beneficial. If the code is modified either in situ or in transit to a remote platform, it will fail verification and not be allowed to run. This mechanism is especially important for an "app store" use case in which the deploying entity may not have first-hand experience with the code and must trust the entities who have signed it. Resource management allows platforms to ensure agents do not consume more resources than are available on the hardware and exhaust the system resources. The agent mobility service allows agents to move from platform to platform (either directed by an administrator or self-directed).

An overview of the VOLTTRON platform components is illustrated in Figure 1. The platform includes several components and agents that provide services to other agents. Of these components, the Information Exchange Bus is central to the platform. All other VOLTTRON components communicate through it using the publish/subscribe paradigm over a variety of topics. For example, the weather agent would publish weather information to a "weather" topic to which interested agents would subscribe. The platform itself publishes platform related messages to the “platform” topic (such as “shutdown”). Topics are hierarchical following the format “topic/subtopic/subtopic/…/…” and allowing agents to be as general or as specific as desired with their subscriptions. For example, agents could subscribe to “weather/all” and get all weather data for a location or subscribe to “weather/temperature” for only temperature data.

The platform communicates to devices and building controllers through the driver framework. Drivers periodically read data off devices such as a building automation system controller and publish data to the message bus via a topic for each device; it also provides a means to send control commands from agents to controllers. The driver framework simplifies creation of new drivers by handling platform interaction and allows developers to focus on methods for communicating with devices. Modbus and BACnet implementations are included with VOLTTRON.

The Actuator/Scheduler agent allows other applications on the platform to schedule times to interact with devices. This Scheduler agent ensures that only one agent can control a device at a given time and allows the user to set the relative priority of each application.

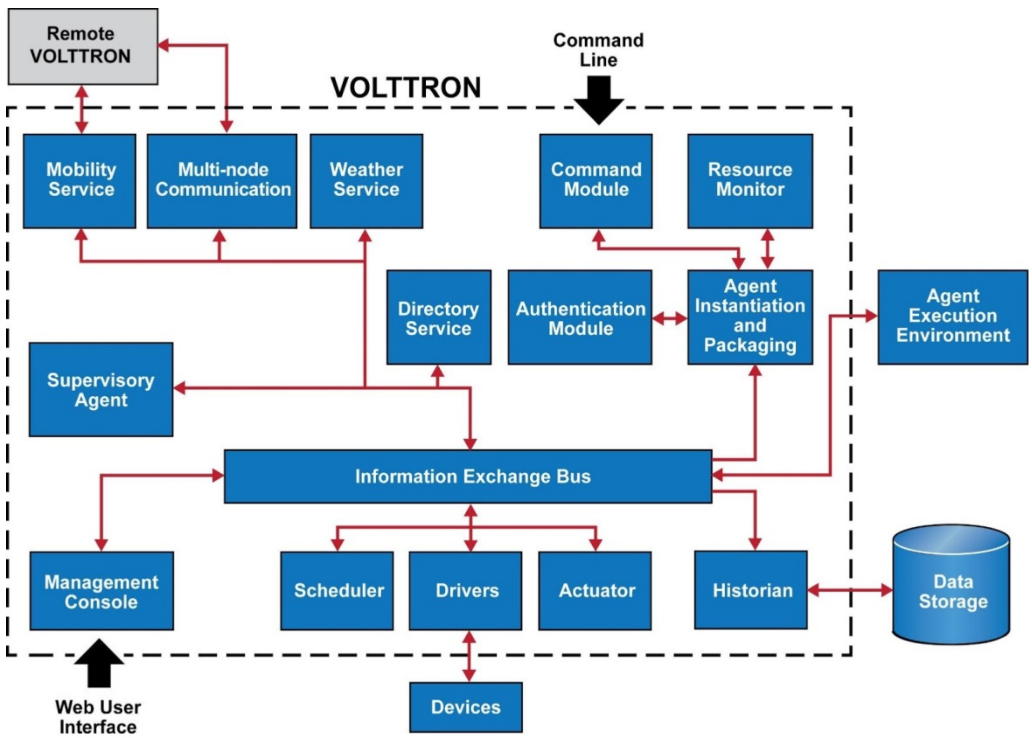

Figure 1. VOLTTRON Platform

Historian agents collect data from the message bus for storage and later retrieval, isolating agents from the specifics of the storage solution used. This data archival architecture allows multiple Historians to be used. For instance, data can be stored to a local database as well as to an external web service. Similar to drivers, historians are implemented using a framework that simplifies the addition of new concrete implementations.

VOLTTRON incorporates several open-source projects to build a flexible and powerful platform:

- **ØMQ:** The VOLTTRON message bus, which allows agents and services to exchange data, uses Zero MQ. This free software is used by National Aeronautics and Space Administration, Cisco, and other large organizations to provide scalable, reliable, and fast communication. The VOLTTRON team are active members of this open-source software community, both reporting and developing code to fix software bugs.
- **BACPypes:** The VOLTTRON BACnet driver utilizes BACPypes to interact with devices supporting the BACnet protocol.
- **PyModbus:** The VOLTTRON Modbus driver builds on PyModbus, which enables Python code to easily interact with Modbus devices.
- **SQLite:** The VOLTTRON BaseHistorian uses SQLite for its local data cache. SQLite can also be used as a main data storage solution. SQLite provides a simple, reliable, and user-friendly solution for local data storage.
- **MongoDB:** VOLTTRON supports MongoDB as another historian for archiving data. MongoDB support is released in VOLTTRON 3.5.
- **sMAP:** VOLTTRON uses sMAP as an option for a data historian, a database to store device data (historical trends) and agent analytic results. While sMAP is currently supported, the VOLTTRON team is not actively developing new features for sMAP.
- **Other open-source Python modules** being used include 'avro', 'configobj', 'gevent', 'flexible-jsonrpc', 'numpy', 'posix-clock', ‘pyopenssl', 'python-dateutil', 'requests', 'setuptools', 'simplejson', 'zope.interface', pandas, tornado, and ply.

2.0 Example Use Cases for VOLTTRON

The following section summarizes a few example use cases for the VOLTTRON platform. VOLTTRON has been used in large deployments such as the clean energy transactive campus (CETC) demonstration and in small deployments where there is one VOLTTRON instance collecting data from a small commercial building. Finally, VOLTTRON is being actively used in educational context at multiple universities including an entire class at Portland State University based on VOLTTRON for their laboratory assignments.

2.1 Clean Energy Transactive Campus

The transactive campus demonstration project is a major milestone for Washington State. The Clean Energy Fund, matched by DOE, is funding the $4.5 million project. The project is one of the first use of transactive energy controls at this scale, spanning multiple campuses including PNNL, University of Washington and Washington State University across hundreds of miles. CETC project combines energy generation, renewables, battery storage, and demand response. The result will improve energy generation and consumption across the state while also improving grid intelligence, durability, and disaster response. Insights gained from the project will inform grid modernization efforts across the United States.

The three campuses are pursuing unique but interconnected experiments and activities:

- PNNL is developing network infrastructure to connect the three campuses and enable testing of transactive methods. PNNL also is deploying VOLTTRON and its associated agents in five buildings to monitor operating conditions, run diagnostics, and improve building performance. This capability will help enable active power load control, allowing loads to be optimized and adjusted to balance fluctuations from solar energy and other sources. VOLTTRON also serves as a central technology platform in projects with University of Washington (UW) and Washington State University (WSU). PNNL will be generating a transactive coordination signal to coordinate experiments across all campuses.
- UW is installing multiple smart inverters to control power production from campus solar panels and eventually will feed campus-produced photovoltaic power into the power grid. UW also is installing a lithium-ion battery energy storage system to increase flexibility of the power consumption across campus and using its data analytics expertise to develop strategies to make buildings responsive to transactive control signals.
- WSU is installing photovoltaic modules on campus for the first time and integrating them with Pullman's "Smart City" test bed and WSU's micro-grid system. Experiments will show that campus energy generation can power critical city infrastructure in the event of an outage. WSU also is developing strategies for sharing energy between WSU's smart buildings and the solar modules.

2.2 Building-Grid Integration Research and Development Innovators Program

The BTO within DOE is continuing its engagement with universities through the Building-Grid Integration Research and Development Innovators Program. The goal of this program is to explore building-grid integration research and development (R&D) technology concepts that can improve the operating efficiency of buildings and increase penetration of distributed renewable energy generation, leading to more efficient buildings and cleaner generation of electricity. The intent of this program is to engage university professors and students directly to support BTO goals. Furthermore, this program links faculty members, graduate students, DOE national laboratory scientists and engineers, and BTO to bring the innovation and discovery process of the university together with unmet needs in the building-grid integration marketplace. This program encompasses 14 student teams using VOLTTRON to achieve their research goals.

2.3 VOLTTRON as A Data Collection and Management Framework

Driven by changing regulations and the desire to cut costs and reduce carbon emissions, demand is increasing to use energy analytics to find equipment faults, to deploy control strategies that save energy, and to identify opportunities for equipment upgrades with a high return on investment. This increased demand has resulted in a number of start-up energy analysis service companies. These companies collect data from buildings, store it in the Cloud, and run analytics to identify faulty equipment, inefficiencies in configuration, or potential energy savings.

VOLTTRON provides a way for these energy analytics companies to reduce their costs by providing a free, open-source solution that can run on commodity hardware. Additionally, since nothing in

VOLTTRON is proprietary, companies can modify or enhance the platform themselves (if capable) instead of working with a vendor who may have other priorities.

2.4 Portland State University

Portland State University is offering an experimental graduate Electrical and Computer Engineering class entitled "Distributed Control Systems for Grid Applications." This class focuses on the emerging capabilities being researched and developed for distributed energy resource control and management. The students are building an energy management system that emulates a microgrid consisting of a central control system and distributed control agents.

VOLTTRON was selected as the software framework for the microgrid energy management system. The central control agent will be where the agents are managed and centralized control decisions are made. VOLTTRON Central will be used for the managing the agents in the system, and VOLTTRON Interconnect Protocol (VIP) is used to connect and communicate between the distributed agents (over Wi-fi).

For the distributed agents, Raspberry Pi's running in headless mode will be used. Each agent is connected to a Programmable Logic Controller (PLC)-based synchronizing relays for synchronous generators. The objective is for the central controller to link the generators together to meet generation objectives. The PLCs are accessed using a ModBus TCP-IP interface (using the VOLTTRON ModBus driver). The register format and content of the PLCs that control the synchronous generators are being designed by students taking ECE358.

Students are expected to design the communication network between the central and distributed agents for controllability as well as observability. Once the basic system is in place, future class offerings may take advantage of the system, expand it to include a richer mix of assets, and explore different management policies.

3.0 Services and Components of VOLTTRON

VOLTTRON’s services are built on top of the core VOLTTRON platform and are illustrated in Figure 2.

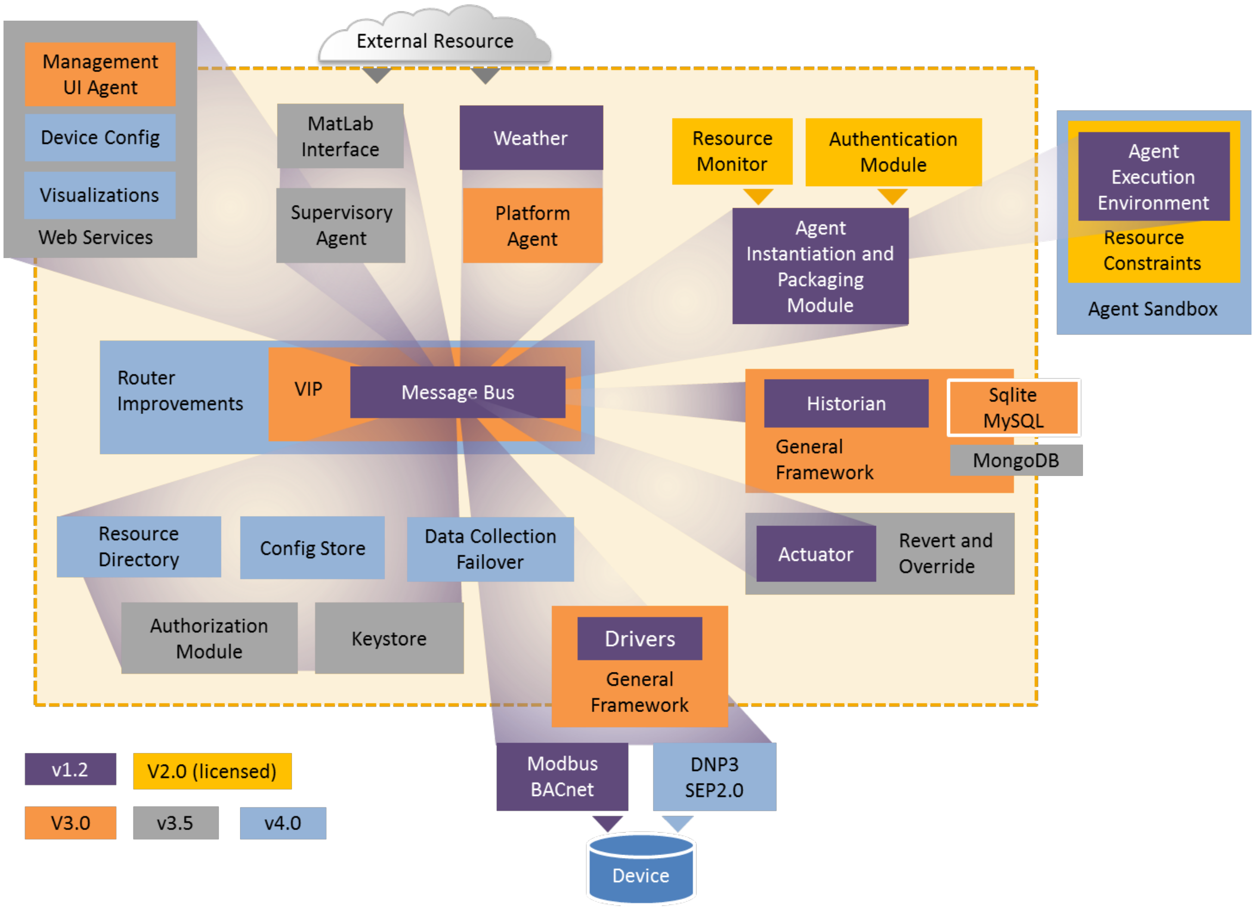

Figure 2. VOLTTRON Features and History

Services/applications provided by VOLTTRON include:

- **Actuator Agent**: This platform service is deployed in the form of an agent running on VOLTTRON. The Actuator agent manages the control of external devices by agents within VOLTTRON.
  - **Device control**: The Actuator agent will accept commands from other agents and issue the commands to the specified device. Currently Modbus- and BACnet-compatible device communication is supported.
  - **Device Access Scheduling**: This service allows the scheduling of agents' access to devices to prevent multiple agents from controlling the same device at the same time.
  - **Revert**: The Actuator agent sends a signal to the drivers to reset all controlled points to a default value contained in their configuration file. This can be used in case of agent failure or at the end of an experiment to reset devices for the next round.
  - **Lockout Signal:** A lockout signal is sent to the platform and prevents any control actions from being taken. This allows building managers to perform actions/maintenance without VOLTTRON agents sending commands at the same time.
- **Drivers:** VOLTTRON driver agents communicate with devices being controlled by the platform. They isolate device-specific protocols from the rest of the platform by publishing device data to the

message bus and taking commands from the message bus. VOLTTRON currently includes supported drivers for BACnet and Modbus protocols.

- **Historian**: Enables the storage of device data obtained by the drivers and application analysis results in a database (currently SQLlite, MySQL, MongoDB, and sMAP databases are supported). Multiple historians can run on the platform at the same time.
- **Management Interface:** Web-based user interface that allows the administration of VOLTTRON nodes (and the agent/applications) running on the VOLTTRON nodes on one or more networks.
- **Message Bus**: All agents and services can publish and subscribe to topics on the message bus. This provides a single and uniform interface that abstracts the details of devices and agents from each other. At the most basic level, agents and components running in the platform produce and consume messages and/or events. The details of how agents produce events and how they process received events are decided by the agents.
- **Remote Procedure Calls:** VOLTTRON agents can interact via Remote Procedure Calls. This is especially useful when interacting with service agents such as the Actuator or Historians and simplifies the code required to use those services.
- **VOLTTRON Interconnect Protocol (VIP):** VIP is a protocol designed to provide secure communications within and between VOLTTRON platforms. It allows for attribution of messages and restriction of access. It also makes it easier to address messages to agents on other platforms. VIP is a wrapper around the message bus.
- **Multi-Node Communication:** VIP provides multi-node communication allowing agents to publish and subscribe to the message bus of a remote VOLTTRON platform. This communication can be encrypted using 0MQ Curve.
- **Supervisory Service:** VOLTTRON's supervisory service comprises several subsystems that the VOLTTRON Central Management User Interface uses to give an administrator situational awareness of the health and status of devices in their deployment. These systems can also be used to trigger emails and other automatic notifications.
  - **Status:** Agents periodically publish their status as simply either Good or Bad and include string containing further context. The context should provide an error message in case of a Bad status and allows for easier debugging of issues on platforms.
  - **Alerts:** Developers can use this subsystem to declare high priority errors that should be immediately brought to the attention of administrators. For instance, if a driver cannot reach a device then it should produce an alert that will cause VOLTTRON Central Management to send an email to the administrator.
  - **Threshold Agents:** These agents monitor the message bus for certain points or messages and trigger an alert when receiving data that fit their criteria.

- **Proxy Agents.** A proxy agent is an agent that runs on the platform with the sole purpose of interfacing with a service that does not natively participate in the VOLTTRON messaging bus.
  - **Weather Information**: This platform service is deployed in the form of an agent running on VOLTTRON. This agent periodically retrieves data from the Weather Underground site. It then reformats the data and publishes it to the platform on a weather topic accessible to other agents.
  - **MATLab:** The MATLab interface allows VOLTTRON to interact with models in MATLab. Agents validated through this interface can then transition to work against a real building without any changes to the agent code.
  - **Framework for Network Co-Simulation (FNCS):** FNCS is a co-simulation framework that can interact with the VOLTTRON platform and allows agents to work with real and simulated devices through the same platform interface.

VOLTTRON enhancements distributed under license by PNNL include the following:

- **Agent Signing and Verification:** Agent code and configuration information is signed by several entities to ensure that it has not been tampered with in transit or while on the server.
- **Resource Management:** Agents present an execution contract with a resource requirements estimate. The platform only allows agents to run if it can support their requirements.
- **Agent Mobility:** Agents can be sent to other platforms via an administrative command or they can request the move themselves. The receiving platform performs verification of the agent package and resource requirements before allowing it to execute.

Agents deployed on VOLTTRON can perform one or more roles, which can be broadly classified into the following groups:

- **Platform Agents:** These agents provide services to other agents running on the platform such as weather information, device scheduling, etc.
- **Proxy Agents**: These agents act as a bridge to remote applications that need access to the messages and data on the platform. A Proxy agent subscribes to topics of interest and forwards messages to the remote (or Cloud) application. These Cloud applications can then publish data to the platform via the Proxy agent.
- **Control Agents**: Using data from buildings and other agents, these agents make decisions and interact with devices and other resources to achieve a goal.
- **Passive Agents**: These agents subscribe to data from the building systems and perform diagnostics to create actionable information for the operators (faulty operation). The information and knowledge that these agents create is posted to the Historian or in a local file.

3.1 Communications and Networking Features

Residential and commercial buildings use a range of protocols and communications infrastructure. To facilitate successful integration of transactive energy into existing buildings, VOLTTRON includes built-in support for a range of devices and BACnet/MODBUS protocols and is easily extensible to support other protocols. A key feature of VOLTTRON is the ability to easily integrate new device control systems and data gathering protocols.

VOLTTRON communicates over many types of communication networks including IEEE 802.11, IEEE 802.15.4 and IEEE 802.3 networks. VOLTTRON utilizes a gateway or a protocol converter to communicate over legacy serial links. VOLTTRON includes support for both IPv4 and IPv6 communications and can use both TCP and UDP over IPv4 and IPv6.

3.2 VOLTTRON Security

Security is a primary feature for VOLTTRON. VOLTTRON provides security against unauthorized access to system data and unauthorized exercise of control functions. VOLTTRON isolates applications running on the platform from each other (if needed) and enforces resource utilization limits on the applications to ensure stability. VOLTTRON uses well-established and widely accepted security mechanisms including elliptic-curve encryption, authentication, and authorization.

Authentication and authorization go hand in hand. When a peer (agent) authenticates to a platform, the peer is proving they are who they say they are. In other words, *authentication* is the process of binding a peer to an identity. *Authorization* is the processing of granting permissions or capabilities to a peer based on their identity. VOLTTRON agents use authorization to selectively limit which peers can call which methods based on the peer's granted permissions. VOLTTRON authorization gives agent authors and platform owners fine control over who can use their agents and how their agents can be used.

Additionally, communications with other platforms use authentication and authorization functions to ensure that only legitimate transactions are performed. Access to the system through local management interfaces are also protected by similar security measures.

The security requirements may differ between transactional energy applications depending on the particular aspects of the service and the participants involved. Each transactional energy application must specify its security characteristics and requirements. These security characteristics include:

- Level of data integrity and encryption required for communication transmissions and for data storage.
- Identification requirements of parties participating in the transactions. For example, attributes such as non-repudiation, cryptographic mechanisms being used to establish identities (e.g., X.509 certificates), and the establishment of trust roots may need to be specified.
- Security requirements/features for the communication patterns being used by the application. For example, are there any firewalls or is the network address translation expected to be in the communication paths? Does it require push or pull for messaging? Is the application event driven? Does the application require cross-domain transactions? How are cross-domain transactions handled? Is there a third party broker involved? How do we accommodate different cyber security authorization policies? Is there a trusted intermediary (e.g., escrow)?
- Transaction validation requirements. Is there an instantaneous enforcement required to ensure against incorrect or malicious actions? Is there a reconciliation mechanism to ensure compliant behavior? If the transactive energy messages are modified during transit, how does the system ensure that the modifications are not erroneous or malicious?

When considering the security properties of confidentiality, integrity, and availability, the control systems community typically takes a much different approach than would typically be taken by the government, military, banking industry, etc. which typically places the highest priority on confidentiality and the lowest on availability. The viability of the control systems depends on placing the highest priority on availability and the lowest on confidentiality. Between these two extremes is the security property of integrity (source integrity/identity and data integrity) that contributes to reliable operation by preventing system compromise and by rapidly detecting any system compromise that occurs. The authentication component of VOLTTRON directly addresses integrity.

There are four broad classes of threats: disclosure, deception, disruption, and usurpation of some part of a system[1]. Disclosure is a violation of confidentiality. Deception and disruption are generally countered by integrity services. Disruption is also countered by availability mechanisms. Availability, integrity, and confidentiality features for VOLTTRON are discussed below.

3.3 Availability Features

- VOLTTRON is based on top of Linux and can detect resource exhaustion (either malicious or due to defective software) using system management utilities and take corrective action. VOLTTRON can also be connected to a monitoring system such as "Big Brother[2]" or "Xymon[3]" and integrate with an enterprise-wide alerting solution.
- As of VOLTTRON 4.0, the VOLTTRON Central platform will deploy a set of agents to specifically detect anomalous system behavior based on historical trends and alert the system operator.
- VOLTTRON continues to operate when network connections are temporarily disrupted. Historians are engineered to locally cache data and upload when network connectivity resumes.
- VOLTTRON (licensed code) detects and terminates an application that is not behaving according to its application/platform contract.
- VOLTTRON has the ability to implement a "liveness" check for each application hosted on the platform in order to detect and terminate malfunctioning applications. Applications must be written to respond correctly with this check or they may be stopped prematurely.
- VOLTTRON supports sending security-related logs to a remote security event information management system.

3.4 Integrity Features

- Agents running on VOLTTRON are authenticated and authorized at every step of the agent life cycle including agent instantiation and movement.
- VOLTTRON authorization features are configurable via policy statements.
- VOLTTRON detects if code or data have been tampered. Typically, VOLTTRON is configured not to execute code that has been tampered.

[1] https://www.ietf.org/rfc/rfc2828.txt
[2] https://en.wikipedia.org/wiki/Big_Brother_(software)
[3] http://xymon.sourceforge.net

- Since VOLTTRON runs on top of Linux, it can lock certain memory locations as read-only once the boot process is completed. This may include all cryptographic and authentication libraries. However, current VOLTTRON versions do not implement this feature.
- As of VOLTTRON 4.0, part of the anomaly detection feature is going to be able to detect bad or poorly written software agents. Once such an agent is detected, based on configuration, the agent may be stopped or prevented from restarting.
- Since VOLTTRON is based on top of Linux, it offers memory protection to hosted agents and platform services that run as separate processes.
- When configured for automatic software updates, VOLTTRON automatically checks and receives updates for both security and maintenance from a known, secure resource on the network.
- VOLTTRON notifies administrators of all agent-related security failures.

3.5 Confidentiality Features

Confidentiality concerns arise in regard to business interests and personal privacy and also secrecy if the agent is, for example, monitoring for illegal usage. Confidentiality features implemented by VOLTTRON include:

- All communications between VOLTTRON platforms are encrypted according to the configuration set by the administrator.
- VOLTTRON supports cryptographic algorithms with key strengths higher than what is recommended in NIST SP800-131Ar1[1].

3.6 Security and Usability

It is notoriously difficult to make user-friendly secure systems. VOLTTRON has made strides in usability by reducing the amount of user-input necessary to securely configure and connect platforms. VOLLTRON introduced a Discovery protocol to easily find and connect platforms. Rather than requiring the user to enter multiple platform encryption keys in addition to a server address, the Discovery protocol utilizes key-stores to select the correct keys to establish secure communication. Managing encryption keys and key-stores has been simplified via the VOLTTRON Control interface.

3.7 VOLTTRON Agent Execution Environment Features

Software agents are entities that can either be stationary or roam between platforms in the system. Agents are task driven and can be used to accomplish a task that involves many platforms. The following features are provided by the VOLTTRON agent execution environment:

- Agents run on multiple CPU architectures as long as agent resource requirements are met. VOLTTRON provides the agents with a runtime environment independent of the underlying CPU architecture.

[1] http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf

- The agent execution environment determines whether it has the capabilities to support a particular agent. In essence, the platform and the agent enter into a binding contract when a platform hosts an agent.
- The agent execution environment determines if the agent requesting "hosting" is authorized to execute on that platform.
- The agent execution environment determines if the agent code has been tampered.
- The agent execution environment provides robust mechanisms to allow agents to store information on-board.
- The agent execution environment provides access to information collection and dissemination services for the agent by means of the message bus and historian services.
- When using the licensed software, the VOLTTRON agent execution environment can guarantee (stochastically) a requested slice of its resources such as memory, storage, and processor cycles to an agent.
- When using the licensed software, the VOLTTRON agent execution environment detects and terminates an agent that is not behaving according to its agent/platform contract.
- The agent execution environment collects and transmits an inventory of agents hosted by the platform to VOLTTRON Central.

4.0 VOLTTRON Central

VOLTTRON Central (Figure 3) is a web-based interface for managing deployed VOLTTRON systems. The interface improves VOLTTRON's accessibility as a technology solution by providing simple, visually intuitive ways to interact with platform instances. The feature extends technical interactions that previously could only be conducted through the command-line terminal.

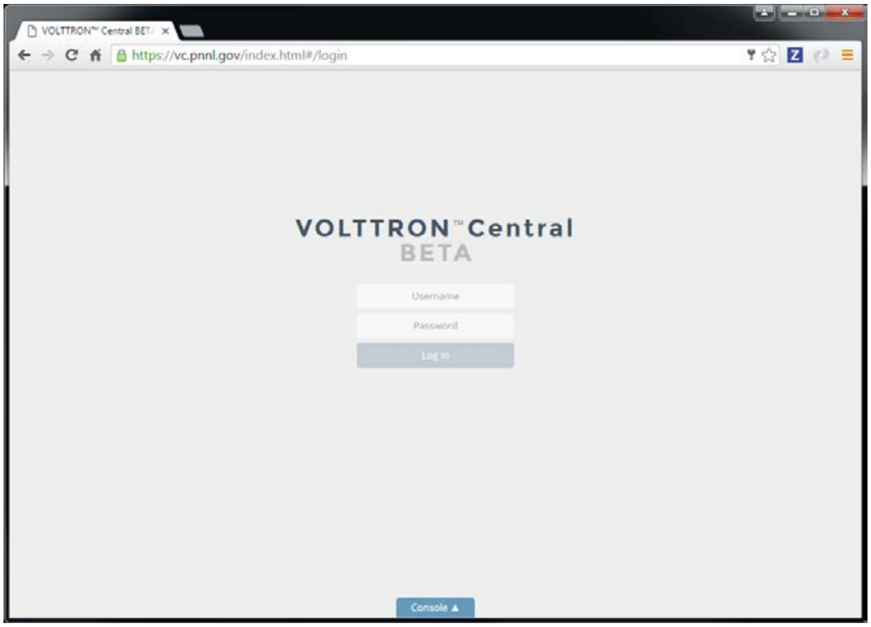

Figure 3. VOLTTRON Central

With VOLTTRON Central, administrators and users can access any VOLTTRON instance that has been registered through the interface. Instances that are managable through VOLTTRON Central include the local platform hosting VOLTTRON Central and remote platforms running on Internet-connected machines that could be anywhere in the world (Figure 4).

Figure 4. Remote platforms

VOLTTRON Central is a portal for information related to the monitoring and health of platform instances. Monitoring refers to the data points generated by devices in a VOLTTRON system. Health is information describing the state of those devices as well as the software agents installed in a system to collect and publish the monitoring data.

An overview of the platforms shows each registered instance and includes a summary of its installed agents. At this level of the interface, users can register additional platforms and deregister platforms to remove them from the portal (Figure 5).

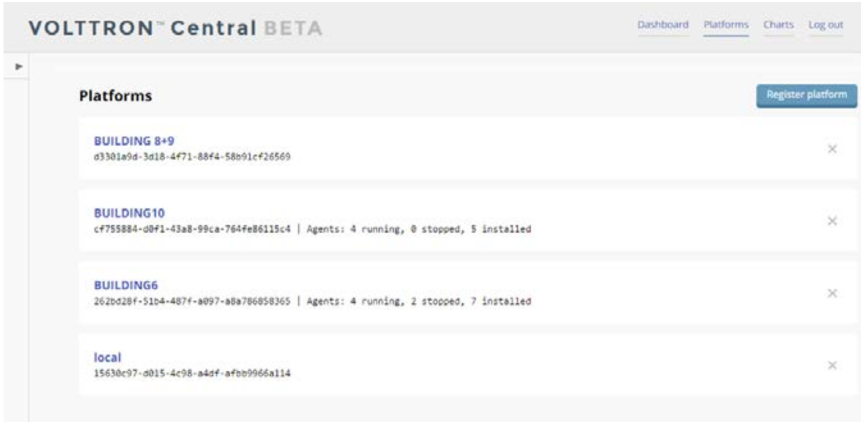

Figure 5. Registering platforms

Each VOLTTRON instance has a Platform agent that allows it to interact with VOLTTRON Central. The local VOLTTRON instance also has a VOLTTRON Central agent that allows it to run the portal. Users can click on a platform's name link to view and manage its agents.

The agents view shows the agents that have been installed and indicates whether they are running. Users, depending on their permissions, can start, stop, and remove agents (Figure 6). Crucial agents, such as a remote instance's Platform agent or the local instance's VOLTTRON Central agent, cannot be stopped or removed, but they can be restarted in the event of a problem. Agents also can be installed by users with the appropriate permissions, and non-crucial agents can be uninstalled.

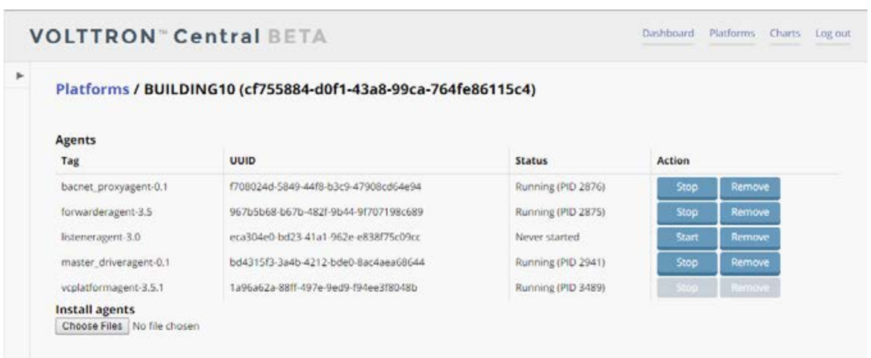

Figure 6. Agents view

VOLTTRON Central also includes a tree view of platforms, agents, and devices that displays additional information about health and status (Figure 7).

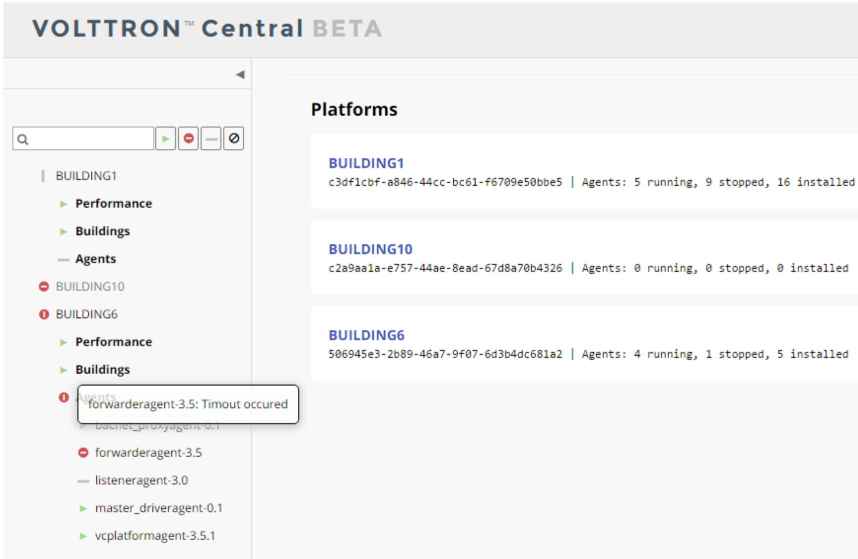

Figure 7. Tree view of platforms

Additionally, the tree view is a dynamic component for launching the display of monitoring data (Figure 8).

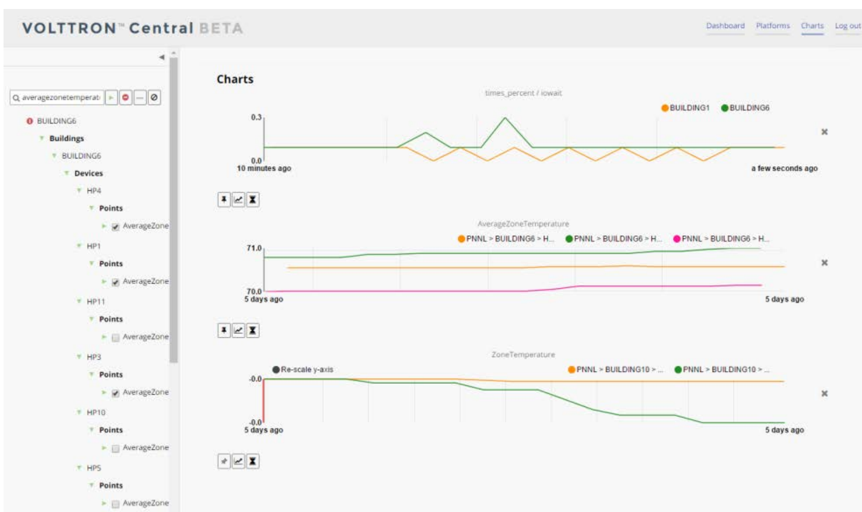

Figure 8. Tree view of platforms (continued)

In the Charts view (Figure 9), selected data points in the tree are added to configurable graphs. From there, the charts can be pinned to the Dashboard, which is the default view users are directed to when they log into VOLTTRON Central.

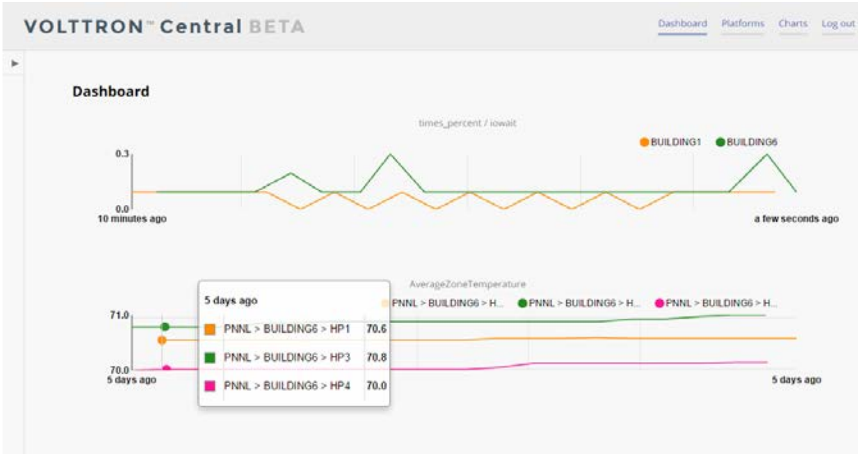

Figure 9. Customizable views

Through the workflow of registering platforms, installing agents, and configuring charts, VOLTTRON Central provides this customizable view of selected live device data—a view that is readily available each time a user accesses the portal.

5.0 Extending VOLTTRON

A major effort for VOLTTRON 3.0 was to increase the flexibility of the platform and allow developers to easily customize components for their requirements or technology preferences. The frameworks for building Drivers and Historians are the two main examples of this.

5.1 Driver Framework

A simple application program interface (API) adds support for new types of devices to the platform via the Master Driver Agent (Figure 10). Using this API ensures existing Historians, platform services, and agents will work with the new type of device. Every platform feature available for working with devices will be available to the new driver.

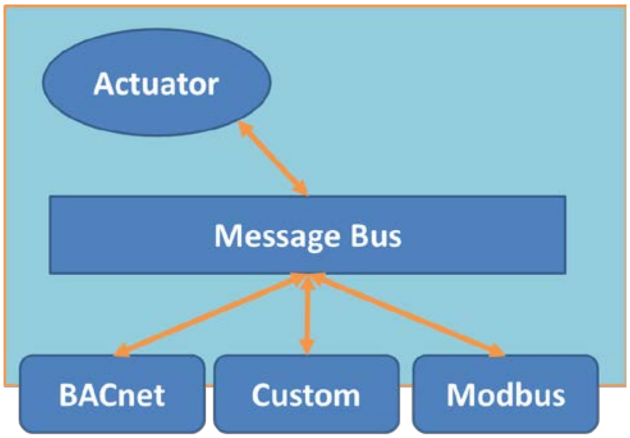

Figure 10. Driver framework

5.2 Historian Framework

A simple API adds support for new types of data stores to the platform. Using this API ensures that agents and platform services that use historical data can access it from the new store and automatically takes advantage of reliability features of the platform Historian agent (Figure 11).

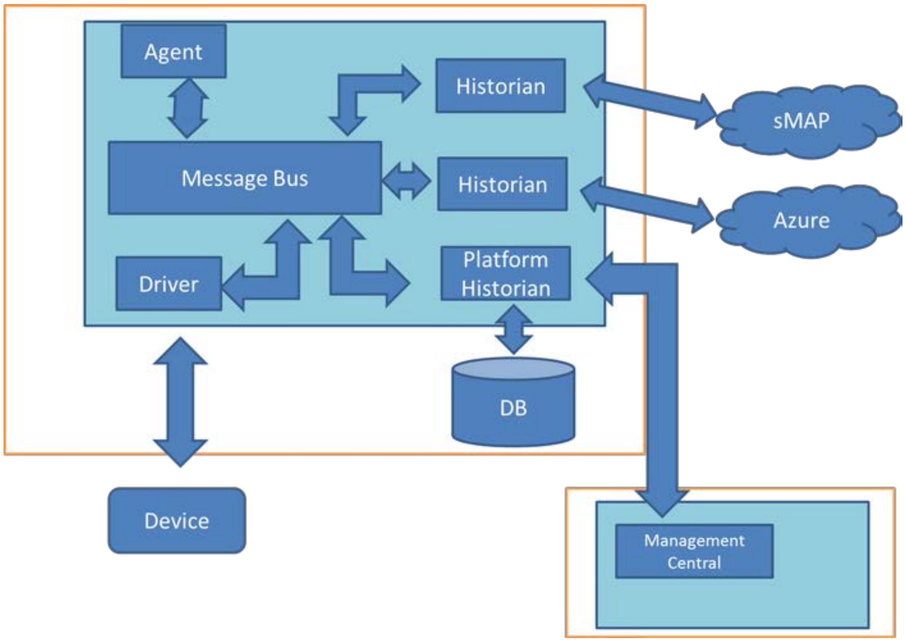

Figure 11. Historian framework

5.3 Other Services

New services can be provided in the form of Agents. For example, an agent was created to provide current and forecast weather data.

6.0 Connecting VOLTTRON to Third Party Software

This section summarizes how VOLTTRON is connected to third party software platforms. The recommended method to connect VOLTTRON to any third party software platform is via the "Proxy Agent" pattern. A proxy agent is an agent that runs on the platform with the sole purpose of interfacing with a service that does not natively participate in the VOLTTRON messaging bus. A proxy agent can be as simple as the weather agent currently implemented in VOLTTRON where it gets the current weather conditions and then pushes the information to the message bus. A more complex proxy agent may be involved in collecting data from one or more VOLTTRON instances and then uploading the collected data to one or more web services such as Azure IOT. Another complex proxy agent may interface with a workflow software such as If-This-Than-That (IFTTT) to implement complex workflows.

For example, a simple IFTTT recipe is shown in Figure 12:

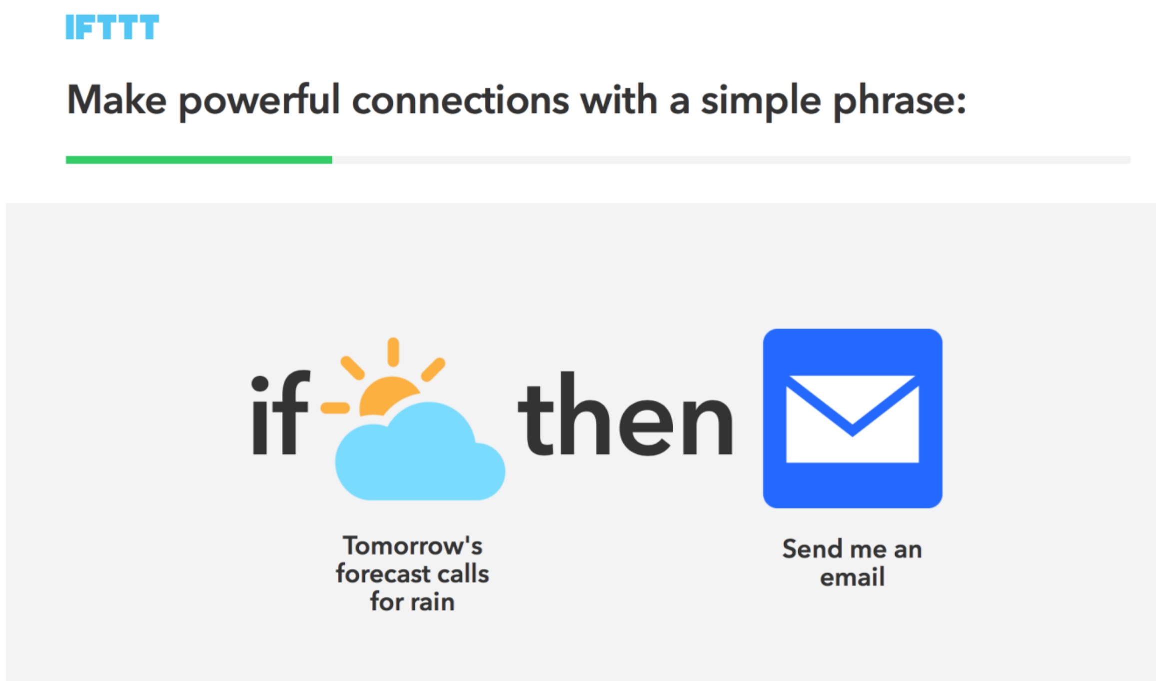

Figure 12. If-this-than-that

VOLTTRON can be easily integrated via an IFTTT proxy agent to implement an action such as “If tomorrow’s forecast calls for 90-plus degree weather AND there is a utility demand response credit available, pre-cool the building during off-peak periods.”

In this example, the intelligence to determine when to pre-cool the building is shared between IFTTT and VOLTTRON. IFTTT implements three main constructs: Trigger, Action, and Recipe. The recipe defines the workflow; the trigger (via the IFTTY Maker Channel) is a web service used to notify third party devices when a trigger occurs. A VOLTTRON IFTTT proxy agent will implement an interface to the trigger. This is done via a web services call as shown in Table 1:

Table 1. Text obtained from IFTTT Developer's Guide

To trigger an Event
Make a POST or GET web request to:

https://maker.ifttt.com/trigger/{event}/with/key/<keyID>-
With an optional JSON body of:

{ "value1" : "", "value2" : "", "value3" : "" }

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

Once the trigger fires, and the proxy agent connects with IFTTT to receive the trigger, it will post a message to the message bus indicating the desire to pre-cool the building. This message will then be picked up by a platform-native demand-response agent to be implemented as if the demand-response condition occurred natively on the platform.

As a second example, Table 2 shows how to interface with Microsoft Azure IOT platform. Azure IOT supports the interfaces for the device client.

Table 2. Text obtained from Microsoft Azure IOT Developer's Guide

Device client:
Support for C, JavaScript (node), Java and C#
Support for AMQP, AMQP over WebSockets, MQTT and HTTP/REST for the device–to-cloud communication
Support for SSL (using third party dependencies such as openSSL or WolfSSL)
Simple APIs to:
Establish a secure connection to IoT Hub
Send messages to IoT Hub
Receive messages from IoT Hub

Similar to IFTTT, VOLTTRON can be integrated to Azure IOT using a proxy agent that implements the RESTful HTTP(S) API for Azure IOT to establish and register the VOLTTRON instance, then to send and receive messages from the Azure IOT hub. Most third party applications implemented in the Cloud follow a RESTful web services model. All communication with the Cloud is completed via HTTP(S) calls. The HTTP(S) calls can be used to check on the status of an entity (e.g., get current weather conditions) or to push a data reading. Events and notifications can be implemented via a polling web interface where the proxy agent will poll a URL periodically or via an asynchronous interface such as web sockets. The VOLTTRON proxy agent is a well-understood and commonly used pattern for integrating third party services to the VOLTTRON platform.

7.0 Conclusion

VOLTTRON is a flexible, reliable, and scalable platform for distributed control and sensing that serves as:

- A reference platform for researchers to quickly develop control applications for transactive energy.
- A reference platform with flexible data store support for energy analytics applications either in academia or in commercial enterprise.
- A platform from which commercial enterprise can develop products without license issues and easily integrate into their product line.
- An accelerator to drive industry adoption of transactive energy and advanced building energy analytics.

The VOLTTRON team will continue adding features while exploring multiple avenues to transition VOLTTRON to a joint development model that includes VOLTTRON users.

Pacific Northwest
NATIONAL LABORATORY

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

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov