VOLTTRON™ Technical Overview

JEREME HAACK

Pacific Northwest National Laboratory

VOLTTRON™ 2016

Overview

► Motivation

► Internal Development

► DOE Development

► Future

What is the VOLTTRON™ Platform?

► VOLTTRON™ is an application **platform** (e.g., Android, iOS) for distributed sensing and controls

- Open, flexible, and modular
- Scalable
- Built in features to streamline application development
- Secure communication (e.g., security libraries/cryptography)
- Platform services

Enabling Hardware and Software

► 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/)

  ■ 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/)

  ■ Wheel: Used for agent packaging in 2.0 (https://pypi.python.org/pypi/wheel)

  ■ Several other libraries used

Pillars of VOLTTRON™

► Flexibility - The platform should be flexible to meet requirements for a varied set of solution spaces

  ■ Deployment – Can be installed on a variety of hardware with differing capabilities

  ■ Topology – Can be arranged in differing topologies to meet the needs of specific implementations

  ■ Services – Components of the platform can be easily added to/replaced

► Usability – The platform should be both easy to use and straightforward to develop

  ■ Development – It should be clear how to develop agents and services for the platform. Developers should have the insight and feedback to ease development

  ■ End User – The platform should provide services that enable the development of high quality user interfaces to simplify deployment, installation, and management of the end solution

Pillars of VOLTTRON™ (cont'd.)

► Scalability – The platform should enable deployments at scale through proper deployment and division of resources

 ■ Number of platforms

 ■ Number of agents

 ■ Number of devices

► Security – The platform must be secure to protect the devices being controlled and not provide a "backdoor"

 ■ Platform integrity – The platform must protect itself from subversion

 ■ Infrastructure integrity – Recommendations for securing the underlying resources used by the platform

► Interoperability – The platform must work across vendors and protocols and provide capabilities to simplify these interactions

 ■ Data standard – A standard data format and naming convention would allow applications written by different organizations to easily talk with each other and the devices being controlled.

 ■ Interface library – A library of interfaces allowing the platform to communicate with a variety of devices through standard (Modbus, BACnet, etc.) or custom protocols.

Terminology

► Platform – VOLTTRON™

► VAgent – A process executing within the VOLTTRON™ platform communicating on the message bus.

- ■ Python VAgents extend the BaseAgent class in the VOLTTRON™ code base

► Application – One or more VAgents working with each other and platform services to achieve some goal

- ■ Fault Detection Agent
- ■ Intelligent Load Control

► Service – A VAgent which provides a capability to applications and the platform. Services *enable* applications but are not applications themselves.
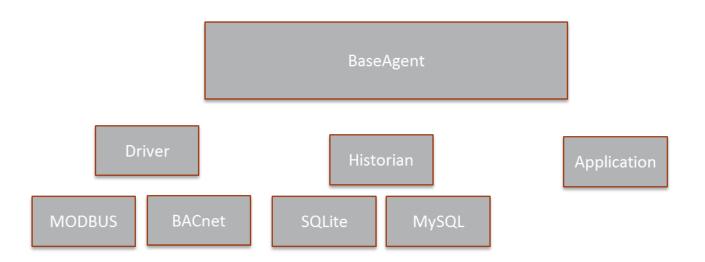
- ■ Historians
- ■ Drivers

Terminology (cont'd.)

▶ Driver – A VAgent which wraps communication with a device or devices.

  ■ Handles specific protocols required

  ■ Allows applications to interact with devices via the message bus

▶ Historian – A VAgent which subscribes to the message bus and stores messages for later retrieval.

  ■ Can be implemented to work with any storage solution

▶ Message Bus – The integrating service which allows the actors in the platform to communicate with each other

▶ VIP – A protocol built on top of ZeroMQ which allows for secure communication between platforms
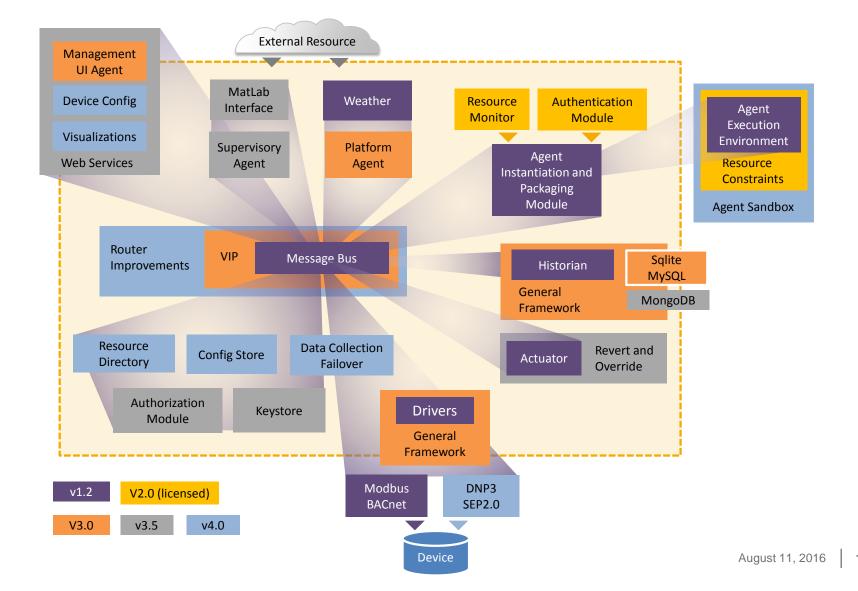
Agents and Services

▶ Base agent provides the boiler plate code for interacting with the platform

■ Provides a base for implementing applications

▶ Services such as drivers and historians are also built upon the base agent

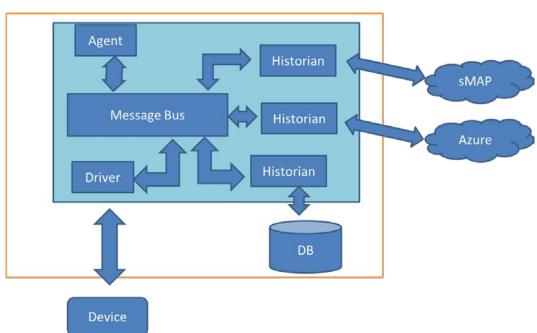■ Provide their own frameworks for implementing specific instances

Features Overview

External Resource

Management UI Agent
Device Config
Visualizations
Web Services

MatLab Interface
Supervisory Agent

Weather
Platform Agent

Resource Monitor
Authentication Module
Agent Instantiation and Packaging Module

Agent Execution Environment
Resource Constraints
Agent Sandbox

Router Improvements
VIP
Message Bus

Historian
General Framework
Sqlite MySQL
MongoDB

Resource Directory
Config Store
Data Collection Failover

Authorization Module
Keystore

Actuator
Revert and Override

Drivers
General Framework

Modbus BACnet
DNP3 SEP2.0

Device

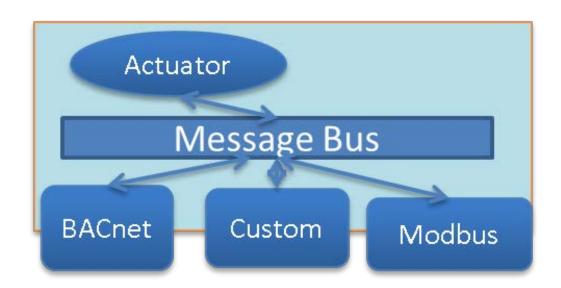| v1.2 | V2.0 (licensed) |
| V3.0 | v3.5 | v4.0 |

Modularized Historian

► Historians can be built for any storage solution

► Previous versions did not have option for local storage

► BaseHistorian

- ■ Can be extended for any solution
- ■ Handles subscribing to Bus
- ■ Local cache

Modularized Drivers

► Standardized creating custom drivers to scrape data and publish to the message bus.

► Simplify developing drivers and contributing new capabilities back to VOLTTRON™.

► Abstracted out driver interfaces allowing Actuator Agent to handle controlling devices via any protocol

VOLTTRON™ Interconnect Protocol

► Increase security of the message bus and allow direct communication where appropriate

- Platform – service
- Agent – service
- Agent – Agent large transfer

► Communication model underneath VOLTTRON™ Message Bus

- Compatibility layer so changes are transparent to existing agents

► VOLTTRON™ now requires only a single socket

► Message Bus can be secured

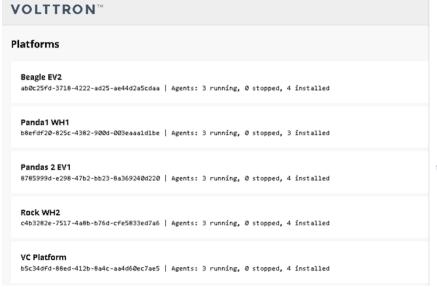- Authenticated publishers
- Publishers can limit subscribers

Management Agent

► PlatformAgent acts as manager for the platform

- Sends commands to agents
- Enables monitoring of health of agents and platform
- Exposes status to other platforms/web console
- Support for applications which analyze data and issue alerts (behavior out of norm)
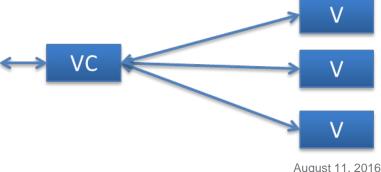
VOLTTRON™ Management Central

► Improve visibility of deployed platforms

► Previous interface to the platform "admin centric"

► Makes use of the service exposed by the Management Agent

► VOLTTRON™ Management Dashboard

■ Allow for better insight into the state of the platform and agents

■ Does not require expert user

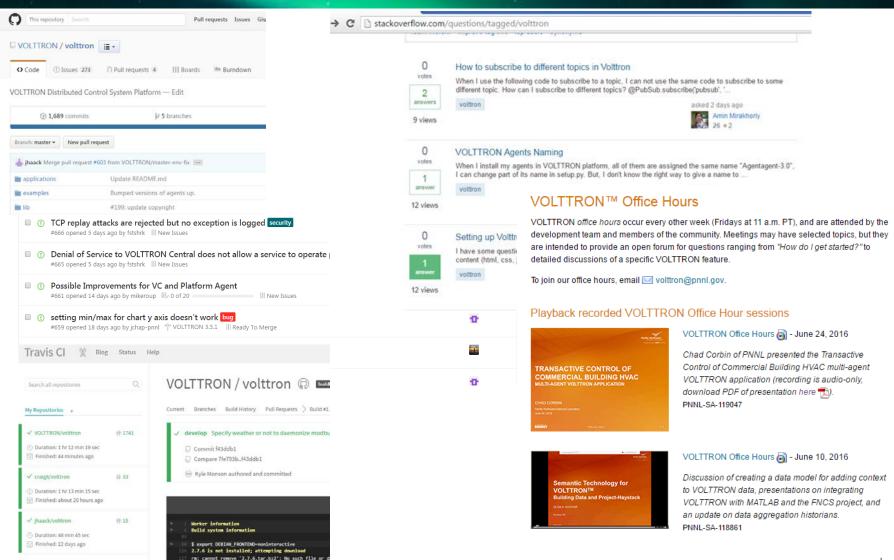■ Quickly see overview of platforms being monitored

Support Mechanisms

VOLTTRON™ Resources

► GitHub
  ■ Codebase: https://github.com/VOLTTRON/volttron.git
  ■ Issues and requests: https://github.com/VOLTTRON/volttron/issues

► Documentation: http://volttron.readthedocs.io/en/develop/
  ■ Documentation is per branch

► StackOverflow: http://stackoverflow.com/questions/tagged/volttron

► Email: volttron@pnnl.gov

► Bi-weekly office hours, email to be added
  ■ Recordings: http://bgintegration.pnnl.gov/volttronofficehours.asp