Eclipse VOLTTRON Development Update

September 25, 2019

Jereme Haack
**VOLTTRON™ Releases**

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Released by FPGA</td>
<td>FY13</td>
</tr>
<tr>
<td>2.0</td>
<td>VOLTTRON (w/ patent) - VOLTTRON Lite</td>
<td>FY14</td>
</tr>
<tr>
<td>3.0</td>
<td>Unified VOLTTRON</td>
<td>FY15</td>
</tr>
<tr>
<td>4.0</td>
<td>Security - User Interface</td>
<td>FY16</td>
</tr>
<tr>
<td>5.0</td>
<td>Improved Performance and usability</td>
<td>FY17</td>
</tr>
<tr>
<td>6.0</td>
<td>Improved Message Bus Independence</td>
<td>FY18</td>
</tr>
<tr>
<td>7.0</td>
<td>Python3 - Containers</td>
<td>FY19</td>
</tr>
</tbody>
</table>

- **PNNL EV Demo**
- **BEMOSS RTU Network First User Forum**
- **NREL ESIF 2nd User Forum** Quality Logic Implements Transactive Node
- **3rd User Forum Transactive Campus** BIRD-IP GMLC Usage Buildings Challenge TWT Deployment Intellimation deploying 100 instances
- **NW Tech Meeting** 4th User Forum VOLTTRON™ Foundation
- **5th User Meeting** Chargepoint training session ACEIoT Announcement
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.

• 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.
Driver Framework

- Framework simplifies process. Fill out methods for:
  - Setup
  - Reading values
  - Sending commands
- Growing list of existing interfaces
- Flexible options for collection rate and organization of sensor points
- Deployer edits configuration files, no need to code for different devices
Historian Framework

- Framework handles collecting data from the message bus for storage
- Simplifies creating specific instance
  - Setup
  - How to store data
  - How to retrieve data
- Numerous supported databases with more being contributed
- Data can also be sent to cloud services or another VOLTTRON instance
- Maintains a cache until data stored
Simulation and Hardware in the Loop

• Simulation connection module allows applications to interact with simulations and models as if they were real buildings.
• Applications tested against simulation need no modification to run against target buildings/devices
• Simplifies hardware-in-the-loop
Platform Security

• Platform hardening guidelines for securing underlying Linux system
• Multi-platform Message Bus
  ▪ Encrypted communication between VOLTTRON instances
  ▪ Authorization required for agents to communicate with the VOLTTRON message bus
  ▪ Pub/sub topics can be restricted to authorized agents
• Platform Security and Monitoring
  ▪ Access to VOLTTRON instances restricted to approved hosts
  ▪ System for forwarding crucial log files for analysis
  ▪ Alerts can trigger emails to administrators
  ▪ Monitor and alert on pub/sub topics for interruptions and unexpected values
• Agent Security
  ▪ Role based access to agent capabilities
  ▪ Agents execute in separate process from platform
VOLTTRON Security Analysis

SSC follows a standard process

Projects can engage at any level as they see fit

SSC and VOLTTRON™ teamwork

CIA Triad
- Confidentiality
- Integrity
- Availability
Platform Updates
Message Bus

- Message Bus
  - Message bus refactoring to support both ZMQ and RabbitMQ message bus
    - Command line wizard (vcfg) supports installing both options
  - Multiplatform communication with federation and shovel plugins
  - Management plugin integrated with volttron-ctl
  - Backward compatibility with ZeroMQ
  - Added CSR support for multiplatform and multi-bus communication
  - Added SSL support to the platform for secure communication
    - In ZMQ, this is for https not Elliptic curve
  - Integrating 3rd party tools
    - MQTT
    - Elastic Search
    - Non-VOLTTRON clients
Services

• Drivers
  ▪ Driver support for DNP3 based MESA-ESS outstation communication
  ▪ BACNet proxy and driver now support Change of Value Subscriptions

• Market Service
  ▪ Added a base MarketAgent and Market Service agent to allow simulation of market based pricing model.

• Weather Agent
  ▪ WeatherUnderground no longer has free keys
  ▪ New WeatherAgent based on a framework similar to historians
  ▪ Implemented for DarkSky and weather.gov

• Historian
  ▪ Redshift Historian
  ▪ Base historian now uses health subsystem and can report backlogged, cache_count, publishing and cache_full status keys.
Agents

• Agents
  ▪ Agent configurations can be written in yaml (no tabs in yaml) or json.
  ▪ Secure ConfigStore

• Agent Template
  ▪ Updated template generation
  ▪ Historian template

• Simulation Support
  ▪ Added fncs subsystem (Allows co-simulation FNCS)
    [https://github.com/FNCS/fncs/tree/develop](https://github.com/FNCS/fncs/tree/develop)
  ▪ New MATLab integration agent
Automated Testing

• Testing
  ▪ Added docker support to the test environment for easier RabbitMQ testing.
  ▪ Added test support for RabbitMQ installations of all core agents.
  ▪ Added multiplatform (zmq and rmq based platform) testing.
  ▪ Review of tests both for RMQ upgrade as well as Python3 work
Upcoming

• VOLTTRON 7.0
  ▪ Python 3
    ✓ Python2 approaching end of life
  ▪ Deployment “recipes”
  ▪ Addressing Security Assessment
    ✓ Agents launched with a different user from that of the platform
    ✓ CSR framework for RMQ
  ▪ Simulation integration streamlined

• Future
  ▪ Integration possibilities with other platforms
  ▪ Data curation during collection
  ▪ Simulation specific version of VOLTTRON
  ▪ Video tutorial series
  ▪ User interfaces to simplify manual processes (tell us your pain points)
• Feedback is essential
• Make use of support channels
  • Suggest topics for office hours
  • Present research
• Project participation
  • Interest/ability to try out experimental branches and release candidates
  • Trying out/contributing documentation and example agents
• What equipment do you have access to?
• Which agents in the applications repo are of interest?
  • Preferences on ‘abandoned’ agents?
• Additional drivers, historians, weather services, message buses, etc.
  • What are the needs?
  • What can be contributed back?
• Interest in a hands-on focused meeting/training session?
Thank you