ORNL/TM-2019/1110
CRADA/NFE-17-06741

VOLTTRON-enabled Home Energy Management System

CRADA final report for CRADA number NFE-17-06741

Approved for public release. Distribution is unlimited.

Helia Zandi
Teja Kuruganti
David Fugate
Edward Vineyard

January 20, 2019

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2019/1110
CRADA/ NFE-17-06741

Computational Sciences and Engineering Division

Home Energy Management System- VOLTTRON Integration

Helia Zandi
Teja Kuruganti
David Fugate
Edward Vineyard

Date Published: January 20, 2019

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Home Energy Management Systems (HEMS) consist of software and hardware systems that provide communication, control, and management among different electronic devices, such as heating, ventilation, and air-conditioning (HVAC) systems. HEMS are a growing market sector in the era of smart grids and smart homes. HEMS can help both homeowners and utilities save money and energy by monitoring and controlling smart devices in a house. Based on the Department of Energy (DOE) *Building Energy Data Book*, 41% of the energy consumed in the United States is used by buildings [1] (see Figure 1). This shows the importance of saving energy in the building sector.

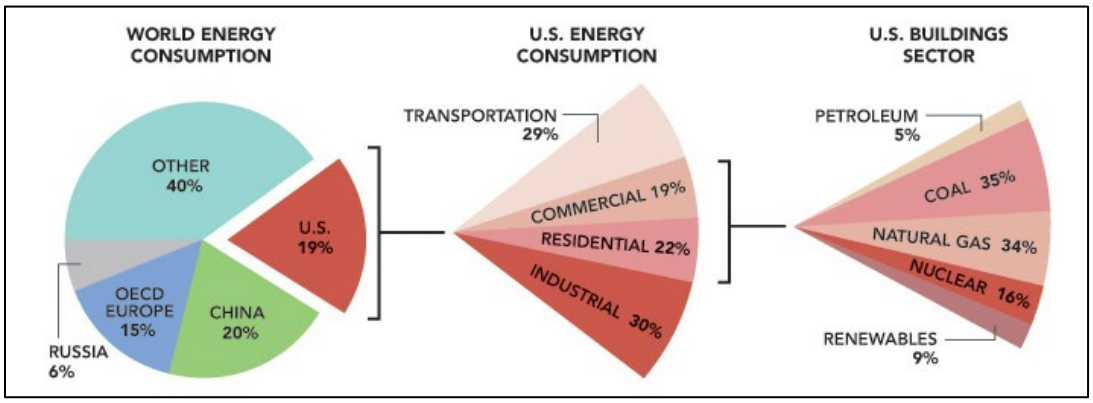

Figure 1. Building energy consumption.[1]

HEMS can monitor and control devices that use different communication protocols. They can provide feedback regarding energy usage to homeowners to help them adapt their energy consumption behavior and help reduce peak demand by supporting utility demand response (DR). They can also support home automation to help homeowners to make smart decisions regarding their energy consumption by providing automatic schedules and rules based on energy consumption patterns, DR signals, and system goals.

Because of the important role of HEMS in saving energy in the building sector, much HEMS research has been conducted and many systems have been introduced to the market [2] [3] [4] [5]. Each of these systems offers different features and has its particular advantages and disadvantages. In an extensive literature review by Oak Ridge National Laboratory (ORNL), different HEMS were examined and compared [6]. In this study, HEMS were grouped into two categories, open-source and proprietary. The former category enables additional flexibility because additional functionality can be added to the software, but they are challenging for a non-technical user to deploy and use. Devices in the proprietary category are user-friendly and easy to deploy; however, their users are limited to the functionalities provided by the specific vendor.

This report introduces two approaches that existing HEMS vendors can use to leverage VOLTTRON as their control and interoperability engine. The goal of this project is to make open-source, user-friendly HEMS available to the market so that everyone can benefit from them. The HEMS introduced as a result of this study should satisfy HEMS specifications such as device monitoring and control, seamless communication between devices, DR, intelligence, data management, and security [6].

[1] https://openei.org/doe-opendata/dataset/buildings-energy-data-book

2. SYSTEM OVERVIEW

In this section, the project goal and different open-source software systems used are described. As mentioned in the previous section, the main goal of this project is to encourage HEMS vendors to leverage VOLTTRON as the control and interoperability engine for their systems. VOLTTRON is a distributed agent platform with an architecture based on Multi-Agent System (MAS) [7]. In MAS architectures, different agents can work together and coordinate their decisions to satisfy the system goals. In VOLTTRON, each agent can publish information or subscribe to information on a common message bus using ZeroMQ [8].

Most commercial HEMS provide communication and management of various devices, but these different devices manufactured by different vendors and running different communication protocols do not interact with one another. One approach to overcome this problem is to integrate the HEMS with VOLTTRON and make all devices' information available on the VOLTTRON message bus (see Figure 2). Different agents on the VOLTTRON platform can subscribe to the information that they are interested in, and adjust their behavior based on the information they receive, to satisfy the system goal.

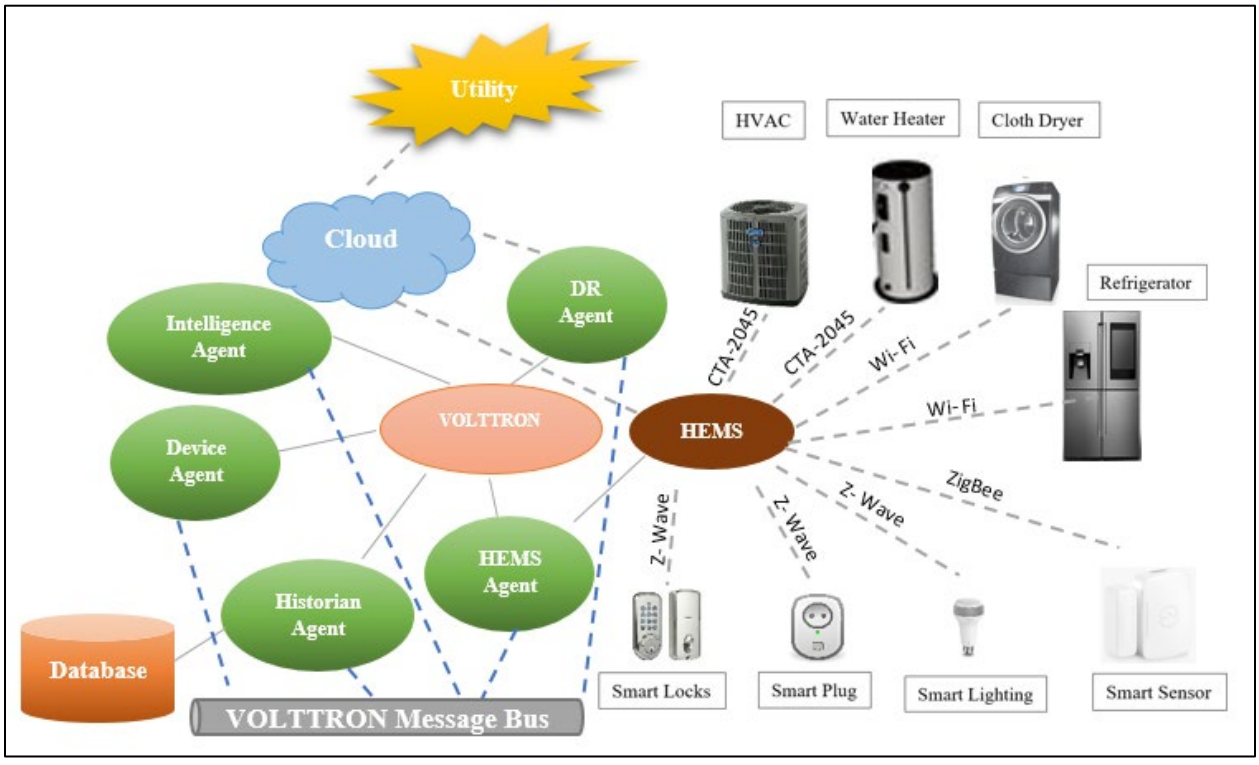

Figure 2. VOLTTRON–HEMS integration overview.

In a previous ORNL report [6], different open-source HEMS were discussed and compared. In this project, the Home Assistant (HASS) open-source home automation platform [9] was chosen as the HEMS to be integrated with VOLTTRON. HASS is implemented in Python 3 and is capable of monitoring and controlling supported smart devices. It can provide home automation features and provides a web graphical user interface (GUI) and phone application for ease of managing devices.

3. HEMS–VOLTTRON INTEGRATION

In this section, two approaches for integrating VOLTTRON with an existing HEMS are explained. Both methods explained in the following sections can be used for any HEMS, regardless of the programming languages it supports.

3.1 INTEGRATION USING RESTFUL APPLICATION PROGRAMMING INTERFACE

If the existing HEMS provides an application programming interface (API) based on the Representational State Transfer (RESTful) architecture, that API can be used for integrating the existing HEMS with VOLTTRON. To do this, an agent can be implemented in VOLTTRON to communicate with the HEMS API and receive information regarding the devices loaded on the API (see Figure 3). The agent publishes the information it receives on the VOLTTRON message bus so other agents can have access to it. The agent should also be able to call the services supported by the HEMS API to change the status of the devices.

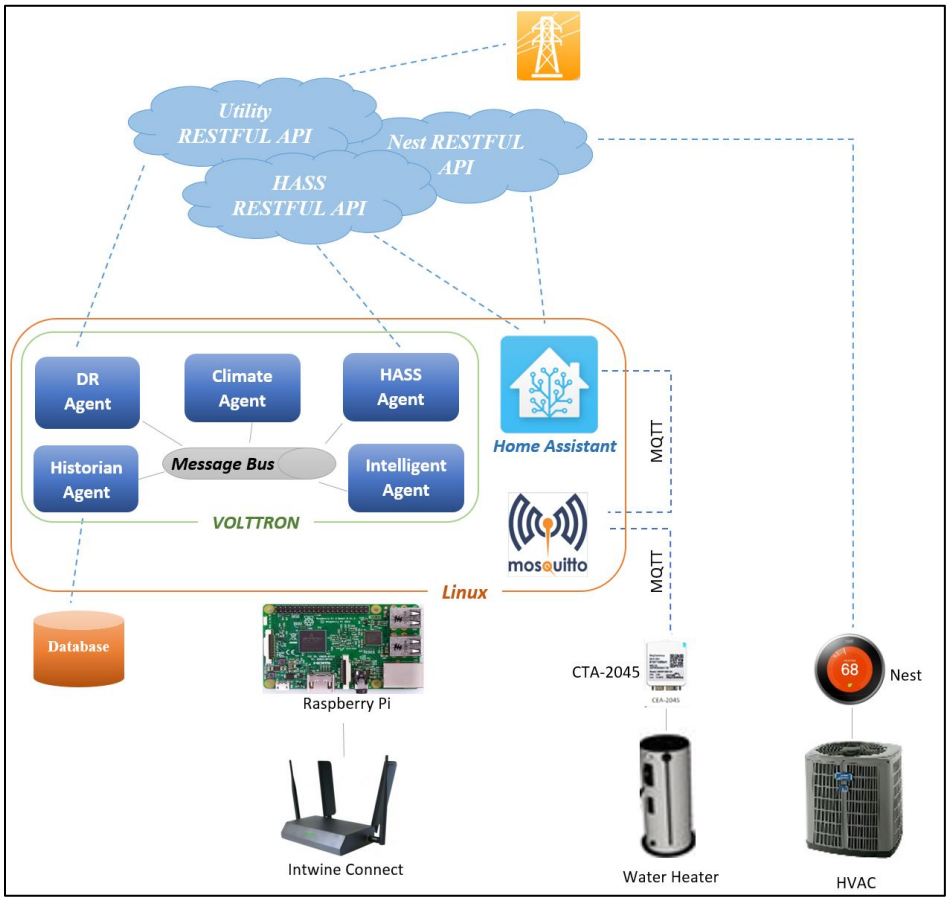

Figure 3. VOLTTRON and Home Assistant integration using RESTful API.

In the case of a thermostat, the HEMS agent receives information about the thermostat using the HEMS API. The HEMS agent then publishes this information on the VOLTTRON message bus, and a thermostat agent implemented on top of VOLTTRON listens to the messages published about the thermostat on the message bus. The thermostat agent can also publish messages on the message bus to make some types of changes to the thermostat. These control messages are sent to the thermostat by calling HEMS API services. The details of implementing the integration using HEMS API can be found in Section 4.

3.2 INTEGRATION USING MQTT

When the existing HEMS doesn't provide an API, HEMS and VOLTTRON can be integrated using the machine-to-machine protocol MQTT [10]. To do this, two additional software pieces need to be implemented: an MQTT agent on the VOLTTRON side and an MQTT component on the HEMS side. Both classes should be configured to communicate with an MQTT broker (see Figure 4). There are many open-source MQTT brokers available that can be used for this purpose, such as Mosquitto [11].

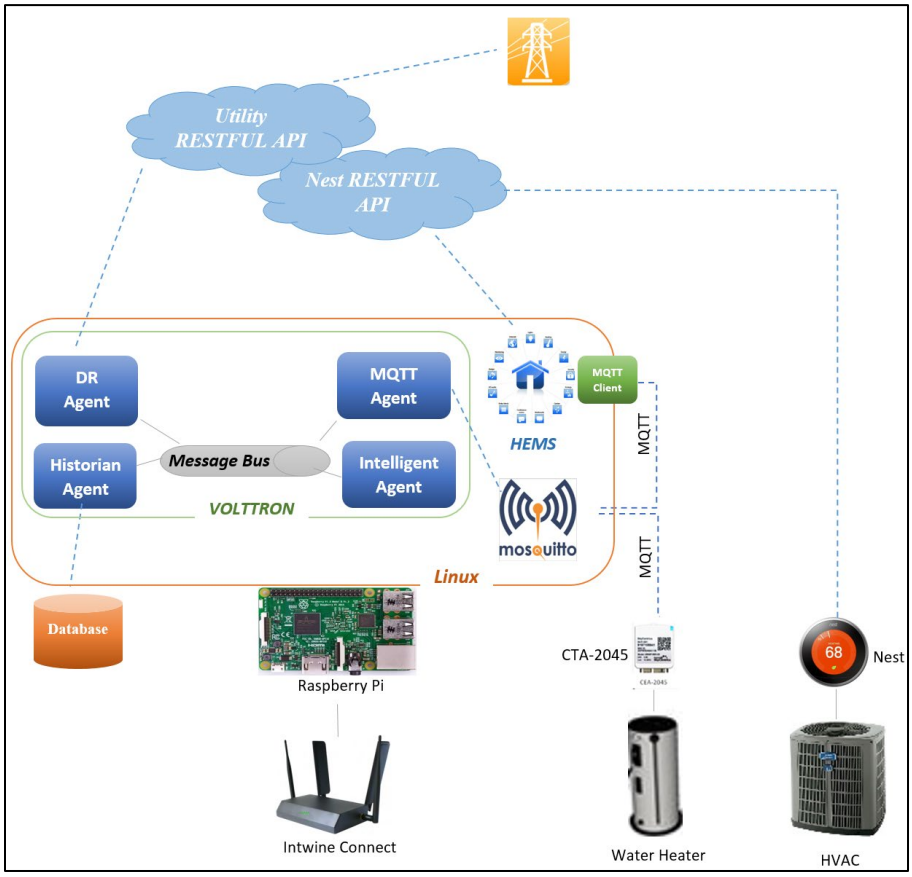

Figure 4. VOLTTRON–HEMS integration using MQTT.

By default, Skycentrics CTA-2045 modules are configured to send the device information to Skycentrics RESTful API. To enable local communication with CTA-2045 modules, Skycentrics must reconfigure the module to send its information to the Internet Protocol (IP) address of a MQTT broker that is running on the same network as the devices. To do this, Skycentrics needs the IP address of the broker and the module media access control (MAC) address. After the reconfiguration is completed, the device information will be sent to the broker when a change is detected or every 60 seconds.

On the VOLTTRON side, an MQTT agent should be implemented on top of the platform. This agent listens to the MQTT broker and is subscribed to the messages received from the existing energy management system. This agent also publishes the information it receives from the broker on the VOLTTRON message bus so other agents also can have access to it. The MQTT agent also publishes messages on the MQTT broker to change the state of the devices loaded on the HEMS.

On the HEMS side, an MQTT component needs to be added. This component publishes the information it has regarding each device on the system to the MQTT broker. It is also subscribed to the messages regarding

its devices on the MQTT broker. Whenever it receives any command message, it directs those messages to the proper device and then publishes feedback on the MQTT broker regarding the result of applying the control command.

4. VOLTTRON–HASS INTEGRATION USING RESTFUL API

In this section, the details of the process of integrating VOLTTRON with an existing HEMS using the RESTFul API is explained. Since HASS was chosen as the representative open-source HEMS, the details of the integration process refer to HASS requirement. However, the integration process can be generalized for any existing HEMS. The code can be found in https://github.com/heliazandi/volttron-applications/tree/master/ornl/HomeAssistant-VOLTTRON-Integration-Agents.

Note that although both VOLTTRON and HASS are implemented in Python, these software systems cannot be integrated merely by implementing another class and running them in the same project. VOLTTRON is implemented in Python 2.7 and is dependent on some dependencies supported only in Python 2.7; HASS is implemented in Python 3 and uses some dependencies supported only by Python 3. Therefore, the integration process is carried out as if they were implemented in two different programming languages.

4.1 INTEGRATION USING VOLTTRON AGENTS

In this section, different agents implemented in VOLTTRON for supporting the integration are explained (see Figure 5).[2] For each of these agents, the HASS API address, the API password, and the agent identification (ID) are provided to the agent in the agent configuration file.

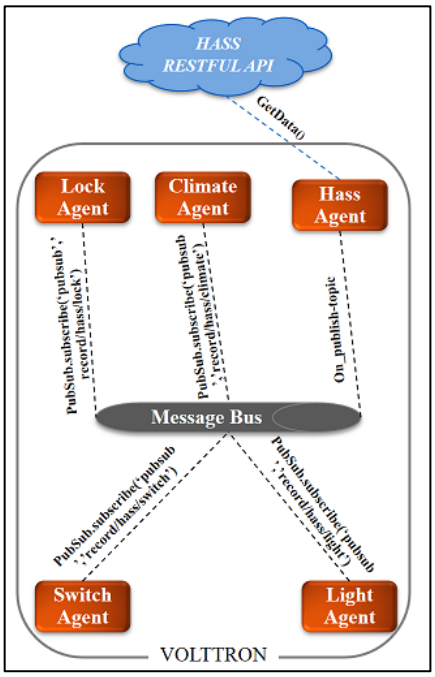

Figure 5. Agents used for HASS–VOLTTRON integration.

[2] The code is available on https://github.com/heliazandi/volttron-applications/tree/master/ornl/HomeAssistant-VOLTTRON-Integration-Agents.

1. **HASS Agent:** This agent is implemented on top of VOLTTRON and can communicate with the HASS API. The agent checks the state of the components loaded on HASS every 30 seconds (this value is configurable) and publishes the information regarding each device on the VOLTTRON message bus (Figure 6). Table 1 lists the topics used by the HASS agent.

Table 1. List of topics used by HASS agent.

Component name	Message topic
Climate	record/hass/climate/entityId
Sensor	record/hass/sensor/entityId
Light	record/hass/light/entityId
Fan	record/hass/fan/entityId
Lock	record/hass/lock/entityId
Switch	record/hass/switch/entityId
MQTT	record/hass/mqtt/entityId

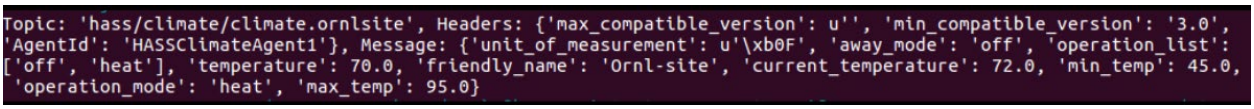

Figure 6. Sample message published by the HASS agent on the VOLTTRON message bus regarding the ORNL Nest thermostat.

2. **HASS Climate Agent:** This agent subscribes to all messages published about climate components by the HASS agent. Climate components (https://home-assistant.io/components/climate/) are devices that can manage HVAC units. It can also change the state of a specific climate device (e.g., temperature, set points, fan mode, operation mode) by sending appropriate service calls to the HASS API.

3. **HASS Light Agent:** This agent subscribes to all messages published about light components loaded on the HASS API. Light components (https://home-assistant.io/components/light/) provide a means to control various lighting systems (e.g., light bulbs). This agent can also change the state of lights in the residence (e.g., turn lights on/off, toggle) by sending service calls to the HASS API.

4. **HASS Lock Agent:** This agent subscribes to all messages published by the HASS API regarding lock components. In HASS, lock components (https://home-assistant.io/components/lock/) allow the user to control door locking devices. This agent also can lock/unlock door locks by sending appropriate service calls to the HASS API.

5. **HASS Switch Agent:** This agent subscribes to all messages that are published regarding switch components. The switch components (https://home-assistant.io/components/switch/) allow the user to manage the state of the switches. The agent can turn on, turn off, and toggle switches by sending appropriate service calls to the HASS API.

4.2 REFACTORING HOME-ASSISTANT INTO THE VOLTTRON DRIVER FRAMEWORK

The functionalities of the agents discussed previously can be refactored to the VOLTTRON driver framework. After we refactored the agents to the VOLTTRON framework, the components status could be queried or changed by the actuator agents. Also, all of the information about the devices is automatically collected and save by the Historian agent. This architecture can be seen in Figure 7.

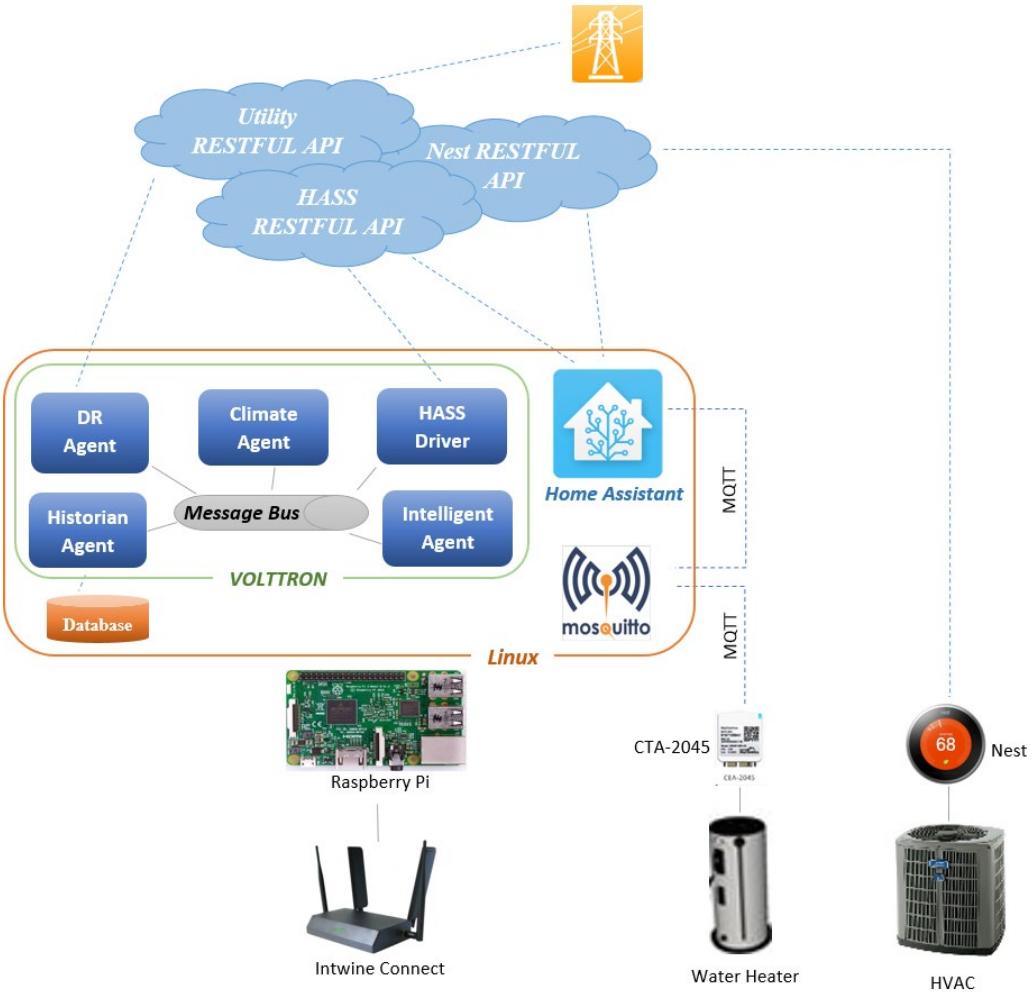

Figure 7. HASS–VOLTTRON integration by refactoring the HASS agent to the VOLTTRON driver framework.

4.3 HOME ASSISTANT COMPONENTS

The devices that are communicating with HASS are known as components. The components are climate, fan, sensor, switch, lock, light, MQTT, and so on. Each component provides a means to control and monitor a specific group of devices; for example, the climate component supports management of devices that monitor and control the HVAC systems. The HASS components website (https://home-assistant.io/components/) [12] provides a full list of supported components.

To add a device in HASS, the information for the device needs to be added to a HASS configuration file, which follows the YAML syntax [13]. Detailed information about adding supported devices to HASS is provided on the HASS website (https://home-assistant.io/) [9]. When a device is added to HASS, the information regarding that device becomes available to the HASS API, and other software can access that information through the API (see Figure 7).

If a device is not supported by HASS, then the developer can add support for that device on HASS by following the information on the HASS website and implementing a new class for that device. In this project, a new class was implemented and added to the HASS API to provide support for managing a CTA-

2045–enabled water heater using MQTT [14].[3] In Figure 8, this component is shown as a CTA-2045 module. This modular communication interface provides a means to communicate among different residential devices such as thermostats, water heaters, and so on.

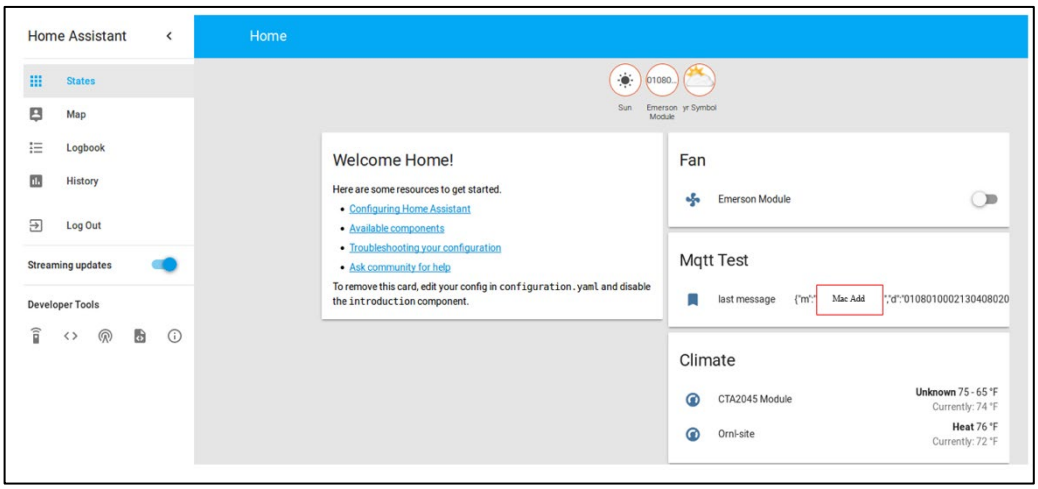

Figure 8. Components loaded on the HASS API.

5. DEMONSTRATION AT YARNELL STATION RESEARCH HOUSE

The new software was deployed for testing at the Yarnell station research house in Knoxville (Figure 9).

Figure 9. ORNL Yarnell Station research house.

[3] The code is available on https://code.ornl.gov/hz6/home-assistant-dev/tree/master. Once openly available, it will be checked in to github.

The house major loads are (Figure 10).

1. Carrier variable-speed HVAC system with variable-speed fan, compressor, and condenser fan. The Carrier thermostat is used to control the HVAC unit. The Carrier RESTful API is used for controlling the unit on HASS.
2. Rheem hybrid electrical water heater. The Rheem RESTful API is used for controlling the water heater.
3. Siemens CTA-2045–enabled electric vehicle charger. The charger is being controlled using local communication with the module.
4. Pentair CTA-2045 enabled pool pump.
5. A.O. Smith water heater with support for CTA-2045.

A. O. Smith

Rheem Water Heater

Pentair Pool Pump

Siemens Electric Vehicle Charger

Carrier Thermostat

Carrier HVAC

Figure 10. Yarnell Station house major loads.

Both the VOLTTRON platform and the HASS home automation system were installed on an Ubuntu-based machine. An MQTT broker was also installed and configured for local communication between the system and the CTA-2045–enabled devices. A MongoDB database was also installed and configured to be used by the VOLTTRON Historian agent.

On the HASS side, two platforms and four components were added to the source code. The Carrier platform provides a way of communicating with the Carrier RESTful API. The CTA-2045 platform encodes and decodes the commands for controlling the CTA-2045–enabled devices. Also, Carrier, water heater, electric vehicle, and pool pump components were added to provide a means to control devices seamlessly from the HASS user interface or a phone application Figure 11).

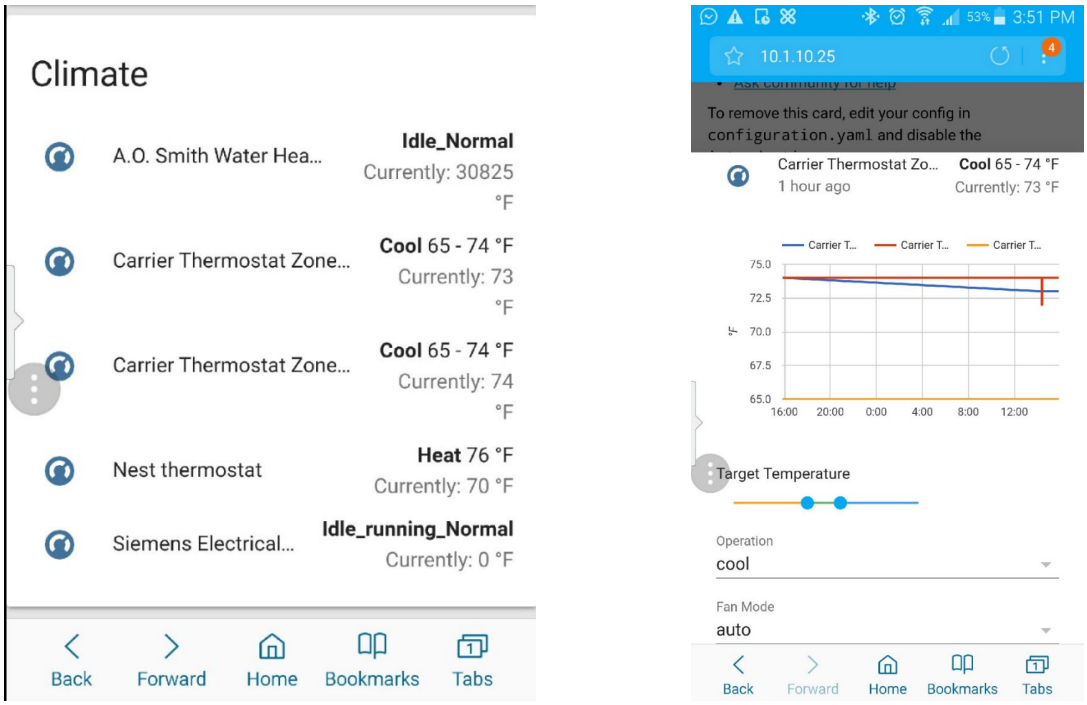

Figure 11. HASS Android phone application.

On the VOLTTRON side, the HASS interface was implemented as a driver under the VOLTTRON Master Driver Interface. The information from devices is pushed to the HASS API. The HASS driver pulls all of that information from the API and publishes them on the VOLTTRON message bus, so they become available to other agents. The Actuator agent was used to control the devices loaded on HASS from VOLTTRON. All of the information was also stored on the local MongoDB database. The setup can be seen in Figure 12.

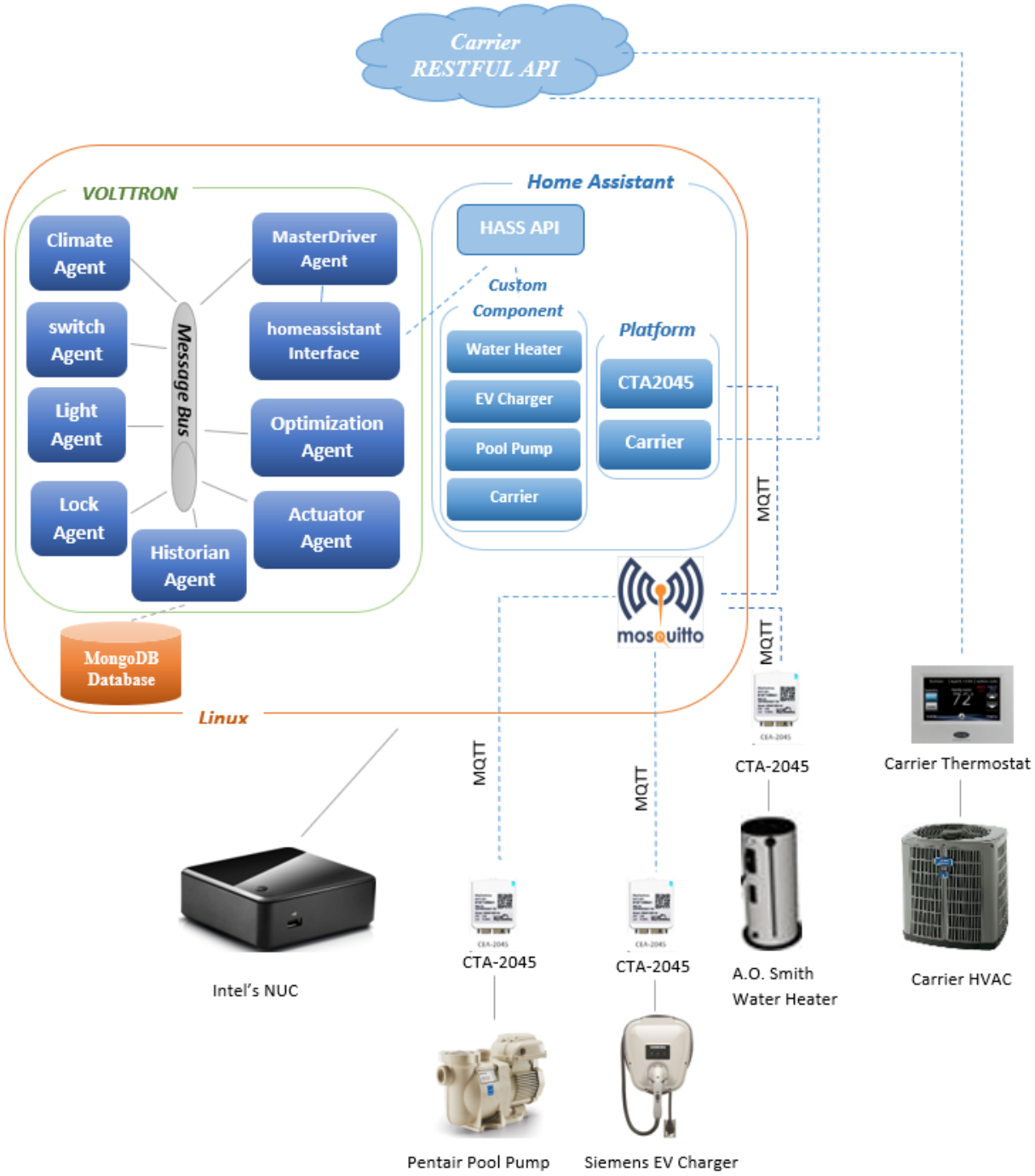

Figure 12. Hardware and *s*oftware setup and architecture.

6. REFERENCES

[1] *Building Energy Data Book*, Buildings Sector, U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Building Technologies Office, 2012.

[2] G. Lobaccaro, S. Carlucci, and E. Lofstrom, "A review of systems and technologies for smart homes and smart grids," *Energies*, 9(5), 1–33, 2016.

[3] B. Karlin, R. Ford, A. Sanguinetti, J. Gannon, M. Rajukumar, and K. Donnelly, *Characterization and Potential of Home Energy Management (HEM) Technology*, Pacific Gas and Electric, San Francisco, CA, February 2015.

[4] S. Aman, Y. Simmhan, and V. K. Prasanna, "Energy management systems : State of the art and emerging trends," *IEEE Commun. Mag.*, 51(1), 114–119, January 2013.

[5] B. Asare-Bediako, W. L. Kling, and P. F. Ribeiro, "Home energy management systems: Evolution, trends and frameworks," *Int. Univ. Power Eng. Conf. (UPEC), 47th London*, pp. 1–5, 2012.

[6] H. Zandi, T. Kuruganti, D. Fugate, and E. Vineyard, "Connected Homes: Home Energy Management Systems," Oak Ridge National Laboratory website, 2016. Available: https://www.ornl.gov/news/energy-connected-homes.

[7] J. Haack, B. Akyol, B. Carpenter, C. Tews, and L. Foglesong, "Volttron: An agent platform for the smart grid," *Proc. 2013 Int. Conf. Auton. Agents Multi-agent Syst.* (AAMAS 2013), pp. 1367–1368, 2013.

[8] "ZeroMQ," retreived from http://zguide.zeromq.org/page:all.

[9] "Home Assistant," [online]. Available: https://home-assistant.io/.

[10] "MQTT, " retrieved from http://mqtt.org/, [online]. Available: http://mqtt.org/.

[11] "Mosquitto Broker," [online]. Available: https://mosquitto.org/.

[12] "Home Assistant Components," [online]. Available: https://home-assistant.io/components/.

[13] "YAML," [online]. Available: http://yaml.org/.

[14] "CTA- 2045," [online]. Available: https://standards.cta.tech/kwspub/published_docs/.