VOLTTRON™ Transactive Control Node: Case Study

Linda Rankin (linda.j.rankin@gmail.com)

Node Contributors:

Chris Freeman (chris.freeman.pdx@gmail.com)

Glen Cooper (gcooper@qualitylogic.com)

VOLLTRON Technical Meeting, July 2015

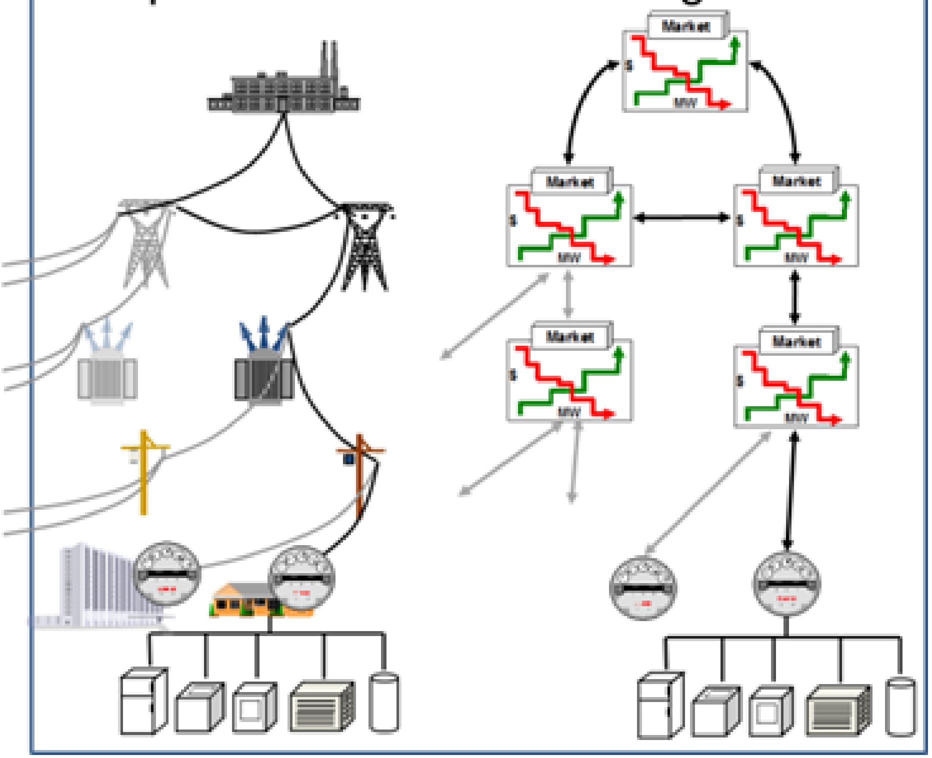
Transactive Control (TC)

- A unique distributed control and communication system demonstrated by the Pacific Northwest Smart Grid Demonstration Project (PNW-SGDP)
  - www.pnwsmartgrid.org
- Localized power generation/load decisions enabled by
  - Distribution of predicted cost and load schedules
  - Incorporating local information and requirements
- Addresses the following areas:
  - Integrating renewable energy
  - Improving reliability
  - Cost reduction
  - Empowering consumers

Transactive Control Ingredients

- Transactive Control Signals
  – Communication signals for predicted cost and load
  – TIS (Incentive), TFS (feedback)
- Transactive Control Node
  – Uses neighbor signals and local information to generate predicted costs and load
  – Manages local assets (resources and loads)
  – Flexible and efficient design allowing deployment at all levels of the energy hierarchy
  – IBM developed a proprietary node based on IEC 18012 (iCS)

Node: point in the grid where flow of power needs to be managed



PNW-SGDP: Node Behavior

TIS/TFS (incentive and feedback signal) calculation and interaction with node neighbors defined by the demonstration project

Computation flow diagram from Transactive Node Toolkit Framework, v 1.0

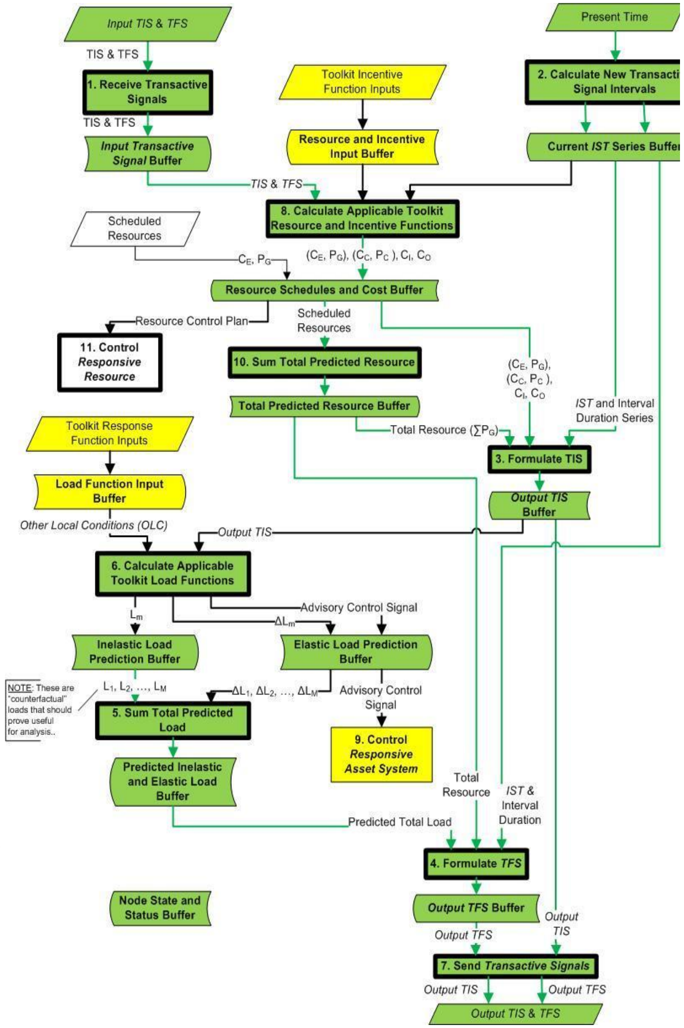


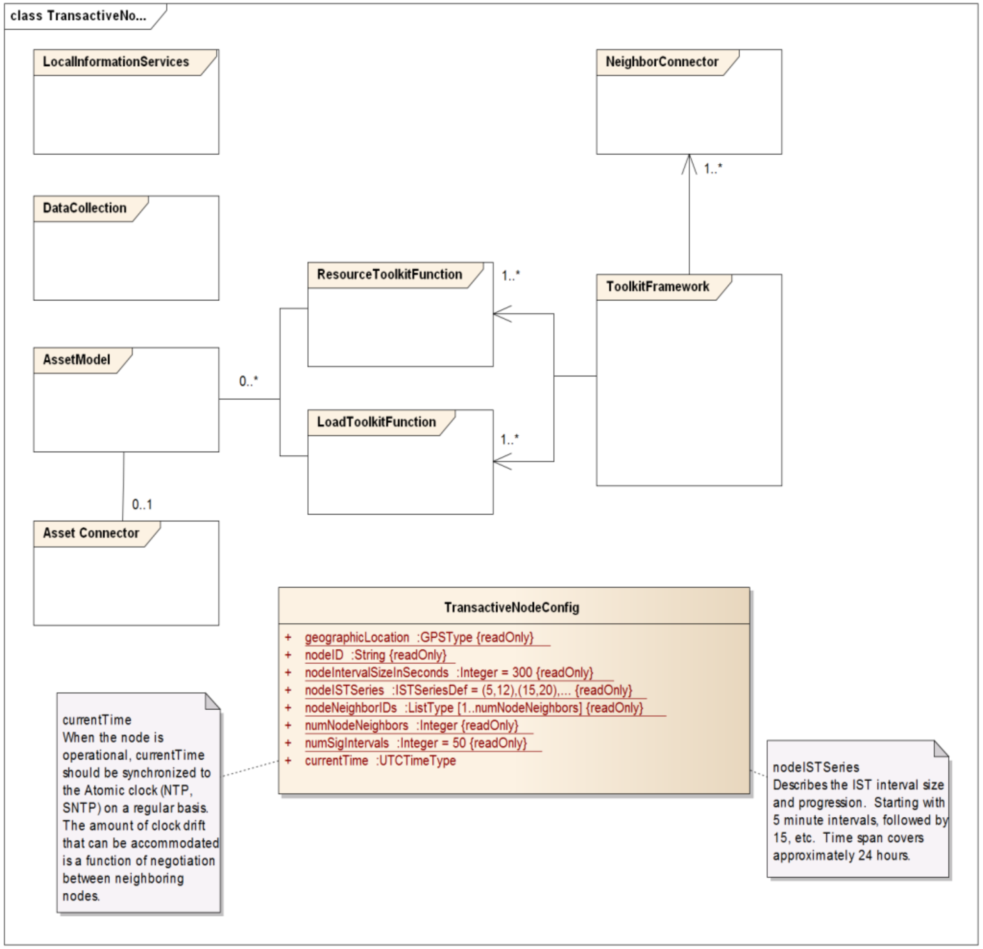
Figure 1. *Toolkit Framework* of Functions and Processes at a *Transactive node*

PNW-SGDP: Software Objects

Pacific Northwest SMART GRID DEMONSTRATION PROJECT

- TC Node objects, configuration and intra-node interactions defined and documented using UML
- Implementation agnostic

Node object diagram from <u>Transactive Control Node: Interactions, Interfaces & Class Structures, v0.90</u>



Reference TC Node Design

- A reference node based on the specifications:
  - PNW-SGDP Transactive Node Toolkit Framework, v1.0
  - PNW-SGDP Transactive Control Node: Interactions, Interfaces and Class Structures v 0.90
- Node Goals and Guidelines
  - Based on open source, available to research community
  - Configurable (TC-related structures, update frequency)
  - Allow for integration of different communication protocols for signals and assets
  - Add asset control and feedback interface
  - Provide a real-time visualization capability
  - Incorporate other learnings from project where feasible
  - Able to pass signaling conformance tests developed for the project

VOLTTRON Provided Complementary Platform and Asset Services for TC

PNW-SGDP TC Node Definition

Event Signaling
(support for power grid events)

Decision Logic (creating local schedules, asset cost/load models)

Services (platform mgmt, weather services, logging and data collection)

Device Interface (Local asset control and bus interfaces)

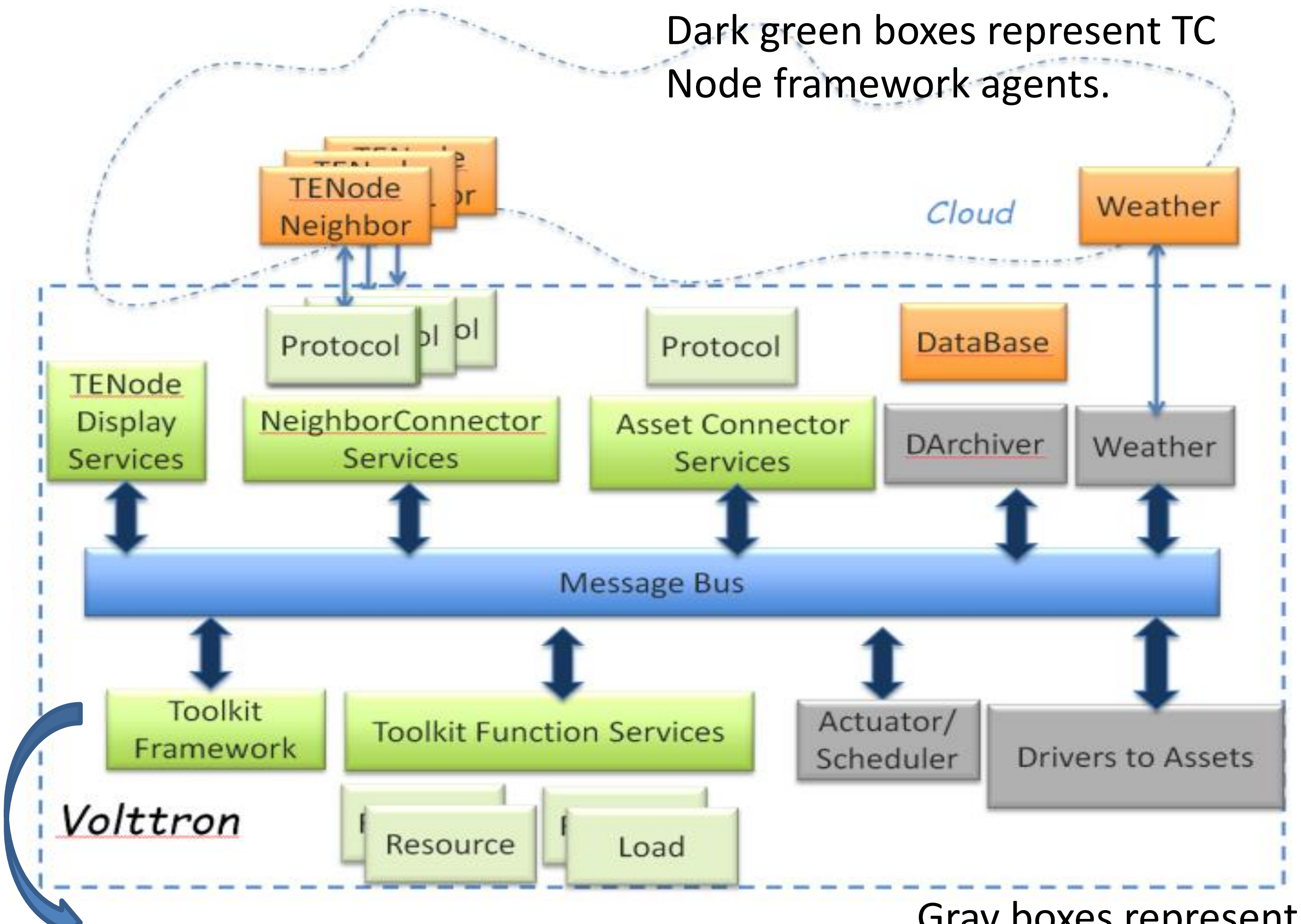
VOLTTRON

TC Node: An implementation of Transactive Energy that uses exchange of incentive/feedback schedules, along with local information to make decisions about controlling local assets.

VOLTTRON: Integration platform for devices (RTUs, HVACs), with external resources, services and applications. Usage model is typically receiving events from power grid, controlling devices thru standard interfaces (BACNET, MODBUS). Linux/Python implementation.

VOLTTRON TC Node

Dark green boxes represent TC Node framework agents.



Gray boxes represent existing VOLTTRON services.

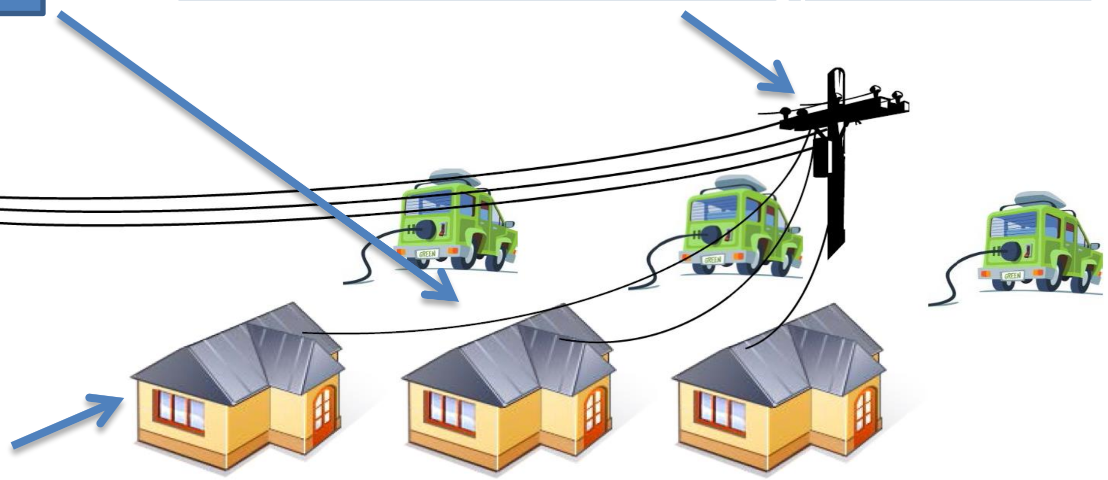
Toolkit Framework Agent is the sequential point of control for Transactive Control functions.
All other agents are event-based, asynchronous

Constrained Feeder 4-Node Demo

Feeder/Home Communications: TIS/TFS using 0mq

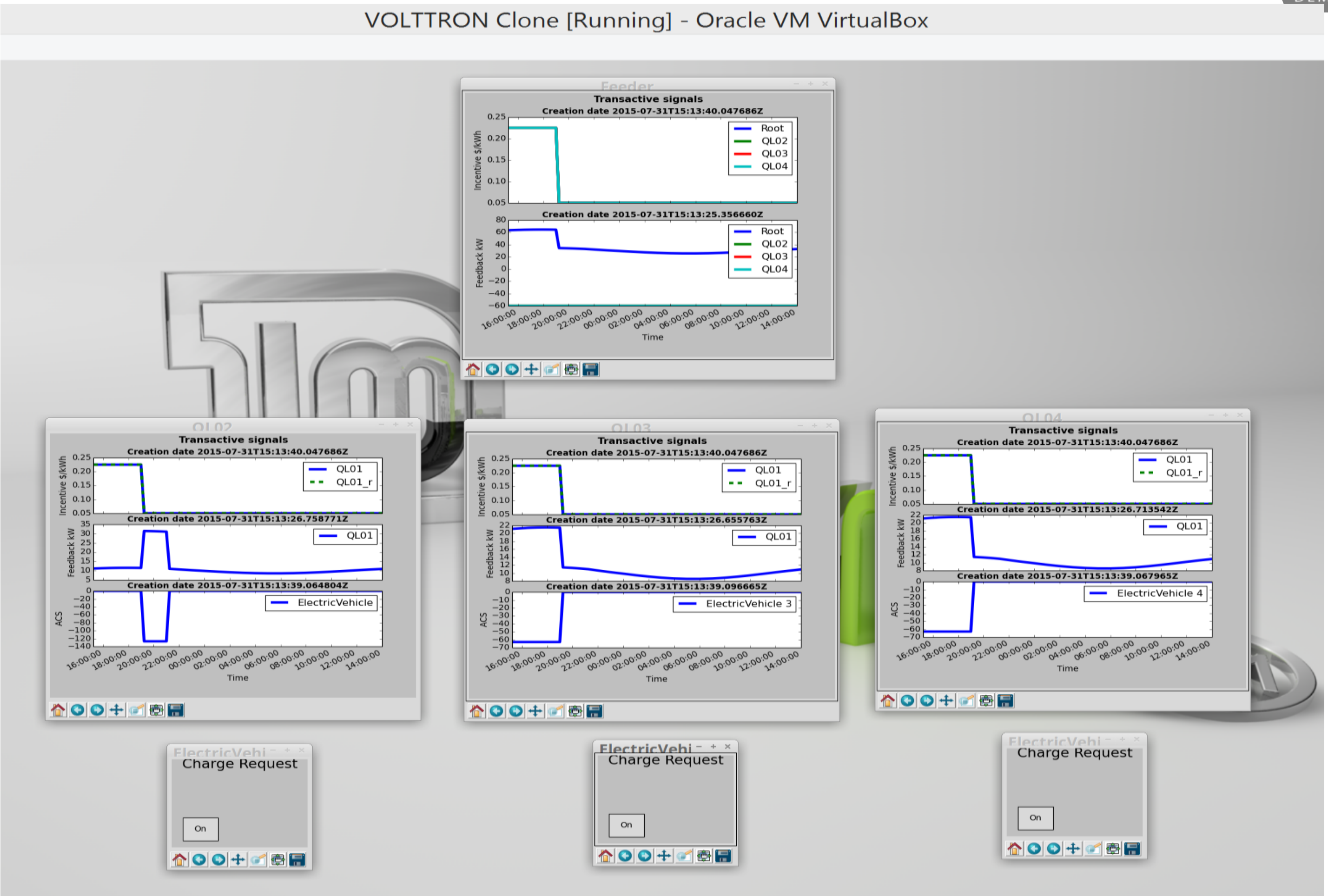
<u>Constrained Feeder Node</u>
Resource Toolkit Functions:
Transactive Energy, Feeder Limit
Load Toolkit Functions:
Transactive Energy

<u>Home Nodes</u>
Resource Toolkit Functions:
Transactive Energy
Load Toolkit Functions:
Transactive Energy, Home Base Load,
EV Charger Asset Model
Asset Connector:
EV Charger



Demonstrates how price and predicted load are negotiated between home and feeder nodes.

Screen Shot of Demo: 4 TC Nodes



Feeder/Home Nodes TIS/TFS and charging profiles after negotiation

Observations: TC Node Modular Design

- Message definition choices
  - Used “topic/op/type/qualifier” constructs
  - Ops were send/receive, update/publish, request/publish
- Application distributed across agents
  - Designed and implemented configuration discovery
  - Common functions and message definitions in global location
  - Ability to create node types using agent configuration options
- Cons
  - Common functions, signal definitions and scripts placed within platform code
  - Extensive logging used for debug impacted thruput (needed better controls)
- Pros
  - Modular architecture allowed parallel development (4 months to develop!)
  - VOLTTRON agent-based architecture allows mix-n-match of comms interfaces, asset connectors, toolkit functions, etc.
  - Clear APIs defined for developers

VOLTTRON Observations

- Terrific platform for fast development
  - Python/Linux are effective and efficient
  - VOLTTRON messaging and services allowed focus on application
  - Mix/Match of agents and configurations allowed for complex node design. Installation, configuration, packaging straightforward.
  - Open source examples accelerated development
  - Git, GitHub as revision control environment supports collaborative, parallel development
- High value in adding VOLTTRON functions
  - 3.0 Peer-to-Peer comms would be used for TIS/TFS exchange
  - 3.0 System Management would be used for deploying nodes on small form factors
    - Remote Debug?
  - Security: adding/removing agents in trusted manner for commercial applications

Growing the User Community

- Develop interoperability guidelines
  - Message construction guidelines
  - Requirements for packaging and installation scripts
  - Version compatibility and use of external libraries
  - API guidelines for built-in services
- Self-certification tool
  - Users can use to verify that code and changes meet interoperability guidelines
  - Can be used to manage VOLTTRON code updates

Thank You

For further questions and comments contact me at:

linda.j.rankin@gmail.com