DOE FOA 970

openECA — open and Extensible Control & Analytics platform for synchrophasor data

Platform and Analytics Beta Demonstration Results

Grid Protection Alliance
1206 Broad Street
Chattanooga, Tennessee 37402

Prepared for
U.S. Department of Energy

DE-OE0000778

December 2017

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Platform and Analytics Alpha Test Results

CONTENTS

1 SUMMARY

The objective of the Open and Extensible Control and Analytics (openECA) Platform for Phasor Data project is to develop an open source software platform that significantly accelerates the production, use, and ongoing development of real-time decision support tools, automated control systems, and off-line planning systems that (1) incorporate high-fidelity synchrophasor data and (2) enhance system reliability while enabling the North American Electric Reliability Corporation (NERC) operating functions of reliability coordinator, transmission operator, and/or balancing authority to be executed more effectively.

The openECA platform enables the deployment of analytics in production utility operations that incorporate both real-time (streaming) and off-line (stored) synchrophasor data. In this project, development of the openECA platform was divided into two major parts: (1) creating the openECA platform itself and (2) providing a rich library of analytics for it that serve as (a) test cases for development, production use and refinement of the platform and (b) patterns which can subsequently be used to expand the platform with additional analytics.

The openECA platform provides a Common Analytics Interface (CAI) for integration of a diverse set of platform analytics along with structured integration of platform configuration, display and storage systems. The platform includes an open-source Linear State Estimator (LSE) as a core component to enable the results from the LSE to be easily incorporated into other openECA analytical components.

The openECA platform enables the secure, high-performance exchange of synchrophasor data with external entities through publish/subscribe protocols and includes a local historian to archive openECA performance statistics. The openECA platform provides an alarming engine that can raise alarms based on high and low data set points and provides a common set of visualization displays optimized for testing and verification of analytic results that can simplify information presentation for decision support.

This project is developing and/or refining to pre-commercial status 11 analytic packages (some open source and some proprietary) that can be deployed using the openECA platform CAI. These analytic packages are divided into the three classes of real-time decision support, control, and off-line analytics as follows:

Real-Time Analytics
1. Oscillation Detection Monitor
2. Oscillation Mode Meter
3. Topology Estimator

Control Analytics

4. Local Voltage Control
5. Regional Voltage Control
6. Shadow System Simulator
7. Phasor Measurement Unit (PMU) Synchroscope

Off-Line Analytics

8. Dynamic PMU Transducer Calibration (Automated, Periodic Use Case)

9. Line Parameter Estimation (Ad-Hoc Use Case)
10. Synchronous Machine Parameter Estimation (Automated, Periodic Use Case)
11. Acceleration Trend Relay Improvement (Research Use Case)

The openECA project completed its Phase 1 Design efforts in 2016. During the first quarter of calendar year 2017, the openECA team conducted bench testing of alpha versions of the openECA platform and analytics. Beginning in June 2017, beta versions of the openECA platform and analytics were installed at Dominion Energy, Southwest Power Pool (SPP), Oklahoma Gas & Electric (OG&E), and the Bonneville Power Administration (BPA) where they were demonstrated and evaluated. This report summarizes the results of this testing.

2 OPENECA PLATFORM

The Beta Release of openECA was issued on June 5, 2017. It included metadata and data structure definitions for returning values to the openECA server components and enhanced user interface (UI) components.

Major architectural elements of the openECA platform include:

- Data Integration Services
- Common Analytics Interface
- Data Conditioning and Alarming
- Electric Network Model
- Shared Platform Services

OpenECA defines a unified environment for modeling an analytic's:

- Configuration
- Data Structures, and
- Measurement Mapping,

The Data Modeling Manager Tool allows the analytic developer to define two classes of data structures:

- The domain input, called SourceData
- The analytic product, called ResultData

The contents of these data structures are under the complete control of the analytic developer.

Beta version testing has demonstrated that the openECA platform allows developers to easily create new analytics by creating data structures and mapping to streamed data sources. Developers can select a target language for the analytic and then use the tool to create a new analytic project.

For additional demonstration results, see Appendix A.

3 Linear State Estimator (openLSE)

Improvements to the openLSE are a key deliverable for the project. The updates that were made accomplished the goals of making if far easier to deploy and maintain – substantially lowering the burden for technology adoption – most of which is made possible by integration with openECA. Being able to “try it out” without great cost in time or resources to an organization is important for delivering value through open source.

The LSE is deployable as a real-time service (https://github.com/kdjones/openLSE). It is provisioned with an easy installer package. Integration of a Topology Estimator enables use of LSE without breaker telemetry (a common hurdle to adoption). It has been integrated with Grafana for Dashboard Visualization.

Improvements were made to the Network Model Editor and Offline Analysis to semi-automate the model building process. Metadata connection to openECA assists in modeling automation. Importing models from GE-Alstom Energy Management Systems (EMS) and PSSEv33 format supports semi-automated model building. Importing measurement information supports semi-automated mapping. The LSE provides an autogenerated *.ecamap file for openECA. It is integrated with the openECA Measurement Sampler Analytic and modeling and analysis tools have been merged for better workflow.

For additional demonstration results, see Appendix D.

4 Oscillation Detection Monitor

T&D Consulting Engineers and Montana Tech developed Version 1 of the Oscillation Detection Monitor (ODM) which is in service at BPA and the Western Electricity Coordinating Council (WECC). The ODM was incorporated into the openECA architecture using the openECA API. This enables it to be productized to enable its broader commercial use. Both front-end and back-end interfaces for the oscillation detector have been re-written for compliance with the openECA API. ODM analytic results are published and oscillation alarms raised through the openECA common analytics interface for display and annunciation.

The ODM analytic is designed to process many inputs from wide geographic area, seeking oscillation energy over a broad spectrum, with minimal CPU usage and provide easy-to-use outputs scaled to engineering units, intended for operations personnel. The ODM outputs oscillation energy in four frequency bands.

The analytic was released in August 2017 and deployed with openECA beta at participant test sites. A Grafana visualization dashboard was included in the beta release.

Based on results from the demonstrations, several next steps have been identified:

- Improvements to setup utility
    - Can be part of an open source effort to create a “measurement selector widget” for openECA
- Addition of optional alarm module
    - Can be standalone, so that outputs from ODM flow to openECA and then act as inputs to another openECA analytic; or
    - Can be an integral part of the ODM analytic
- Addition of spectrum output capability
    - This feature adds many output measurements to the openECA data bus. Rather than eight new measurement channels, this could add hundreds. Need to explore whether this is worth the effort for the user community

For additional demonstration results, see Appendix C.

5 OSCILLATION MODE METER

Version 1 of the Oscillation Mode Meter (OMM), developed by Montana Tech and others, is in service at WECC, BPA and California ISO. The OMM uses the Over-Determined Modified Yule-Walker method to estimate the frequency and the damping factor for each major oscillation mode detected. The OMM processes a carefully selected set of inputs known to demonstrate "observability" to a known system oscillatory mode. Where a known system mode presents an operating challenge, this tool can be used to provide operations personnel with accurate estimates of the characteristics of the mode. Since mode damping changes with system loading conditions, it can be useful to have a "meter" that measures damping in real time.

The OMM detects oscillations with the potential to grow in amplitude and includes filtering to reduce and assure that false alarms are minimized. OMM inputs and outputs have been modified so that the analytic can be incorporated into the openECA architecture using the openECA API.

The analytic was released in October 2017 and tested at participant sites. A Grafana visualization dashboard is under development.

For additional demonstration results, see Appendix C.

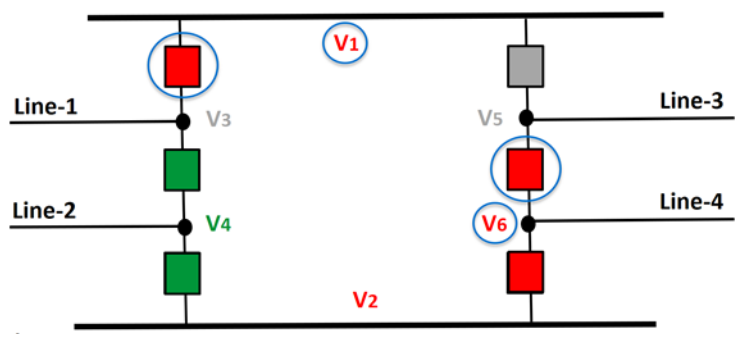
6 TOPOLOGY ESTIMATOR

Knowledge of network and substation topology is critical for correctly executing state estimation calculations. With LSE (i.e. synchrophasor-only), most implementations of today either don't have complete telemetry of breaker statuses or have no breaker status telemetry at all. These scenarios limit the usefulness of the LSE for increased observability but especially for data conditioning and model verification. In some instances, SCADA information (i.e. SCADA-sourced breaker status telemetry) can be used to supplant the absence of breaker statuses in the synchrophasor streams. However, this is ineffectual due to latency and lack of time synchronization as compared to synchrophasor data when dealing with the estimation of transient events. Additionally, telemetry of breakers that is available may contain errors.

This analytic utilizes a methodology that enables the determination of the bus-branch model used by the LSE (the standard output of a topology processor) through knowledge of phasor measurements alone. Additionally, in some instances, these measurements can be used to validate individual breaker status measurements and even supply the state estimator with virtual breaker status measurements in the absence of real telemetry. The algorithm relies upon the principle of equipotentiality among connected nodes at the same voltage levels inside a particular substation. The algorithm works first by executing a series of offline power flow analyses of various contingency scenarios inside a particular substation and subsequently repeated throughout the whole system. Thereafter, the expected value of the voltage phase angle threshold value across an open breaker is empirically determined and this pre-determined value is later used to compare phasor measurements throughout a substation topology to group the various nodes accordingly which can be either specific to a particular substation or generic to the whole system as defined by the user/utility. Several progressive levels are used to determine the topology of the bus-branch model. Each level constructs the substation clusters using different assumed input information, building upon the previous level.

- Level 0 - Each measured, energized node is its own bus
- Level 1 – Buses are grouped with breaker telemetry, where available
- Level 2 – Breaker statuses are inferred and validated using the voltage phasor deviations, further grouping buses
- Level 3 - All node groupings are determined by voltage coherency thresholds in absence of inherent substation structure.
- Level 4 – Level 3 behavior + available breaker telemetry and pseudo angle across breaker telemetry check is included to further refine the groupings.

As depicted in the following figure (two and a half breaker topology scheme), using the Level-4 methodology for topology estimation we can validate any discrepancy in breaker status telemetry and finally cluster the unobserved nodes (represented in grey) with the charged observed nodes (represented in red) and even supplement time stamped breaker status if necessary.



Evaluation: The TE was evaluated during demonstration at Dominion Energy. A cumulative distribution of angle deviations between disconnected nodes was developed. Based on this testing an angle deviation of 1.01 degrees was associated with a probability of deviation of less than 0.05, which was selected as the basis for determining whether two busses were connected.

A bus/branch model is constructed out of the nodal model, with the procedure split into steps depending on assumptions concerning the knowledge that is available. The purpose of the TE is not to establish breaker status, but rather an accurate depiction of system topology.

To determine the accuracy of the TE in reconstructing topology information, simulations were run for random substation breaker configuration under varying system conditions. The results were manually reviewed for consistency and accuracy, and it was verified that the TE works as designed. Results are accurate even under "noisy" conditions and the TE accounts for the vast majority of system cases. The TE provides an alternative and/or redundant method to estimating topology based on breaker status telemetry.

For additional demonstration results, see Appendix B.

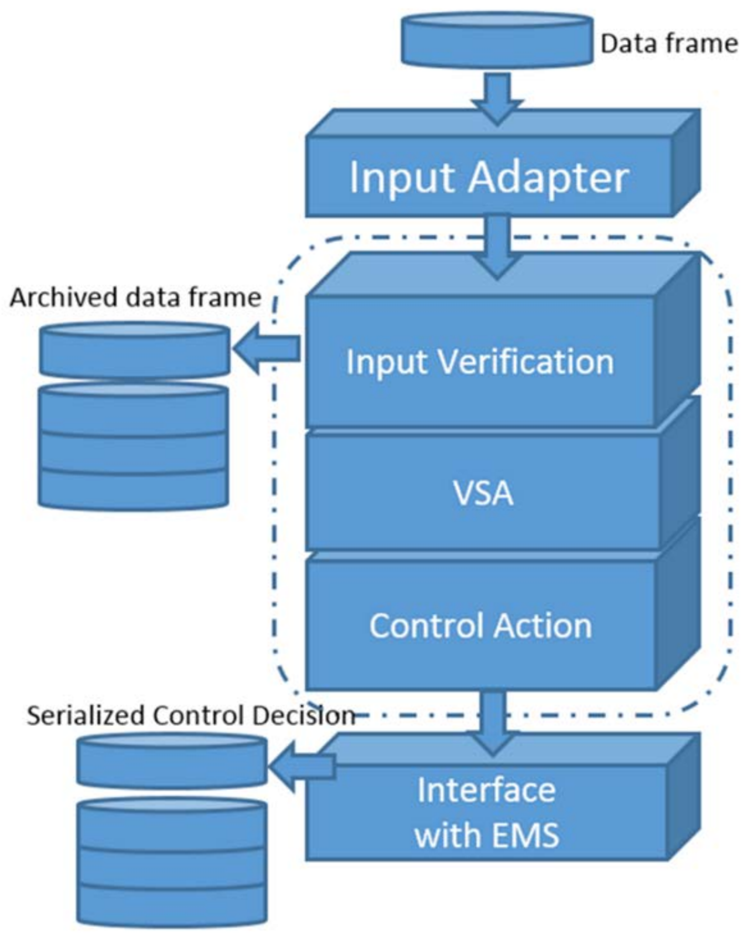
7 LOCAL VOLTAGE CONTROLLER

Most voltage controllers use SCADA data as inputs to centrally driven control algorithms for managing the load tap changers (LTC) and capacitor/reactor banks at certain individual substations. This analytic includes LTC signals and breaker statuses for capacitor/reactor banks in the synchrophasor streams to drive a control application which duplicates such control systems in the openECA environment driven by synchrophasor data instead of SCADA data.

The architecture of the local and regional voltage controllers shows a "pipes and filter structure". Three fundamental components of the voltage controller are:

1. The input adapter
2. The voltage controller adapter (i.e., the user defined application)
3. The Interface with EMS system

These components are shown in the following figure.



Voltage Controller's software architecture

The voltage control adapter is a user defined application which interface of a class into another interface. The voltage controller adapter provides functions such as input verification, voltage security assessment (VSA), and voltage control action. For the local voltage controller, the VSA is simply based on the pre-defined threshold value of target bus's voltage, reactive power flow inside control transformers, and reactive power output from the neighboring generators. When the measurement is lower or higher than the limits, the control logics will be triggered, and the control signal will be sent back to the EMS system. The VSA is conducted through evaluating PMU voltage magnitude at each bus using one decision tree.

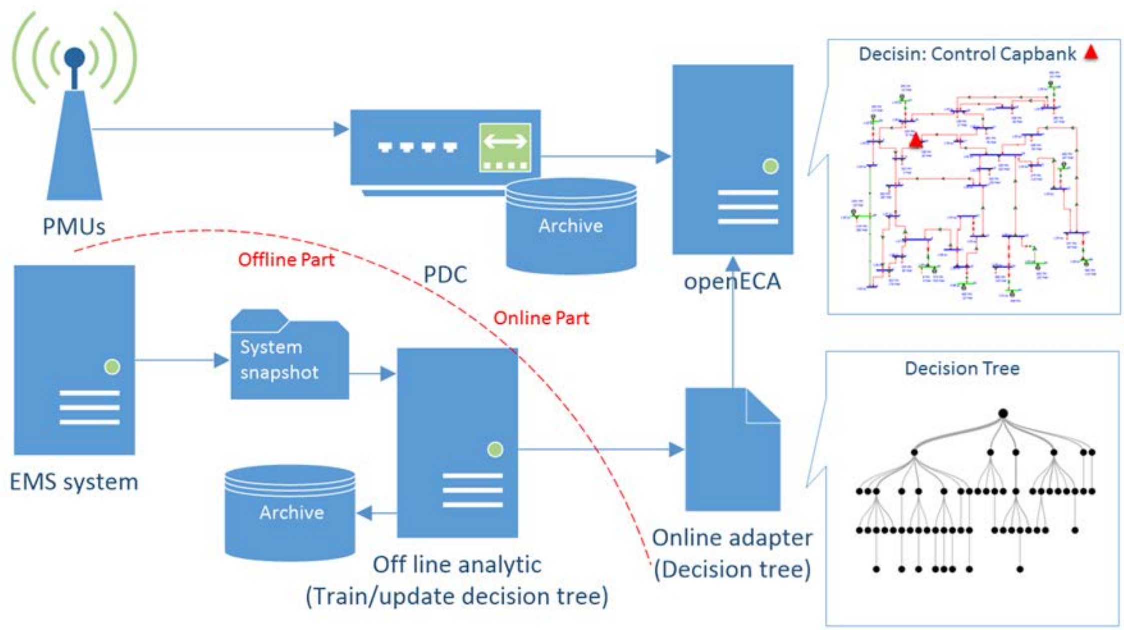
For additional demonstration results, see Appendix B.

8 Regional Voltage Controller

This analytic is an extension of the local voltage controller and provides open-loop control decisions within a control region with dense phasor measurement units (PMUs) installed. To provide more accurate and robust decisions for voltage control, a control methodology based on parallel decision trees is utilized. In this methodology, each decision tree (DT) provides voltage security assessment (VSA) for each control decision. If more than one DT provides secure control decision, the control decision with the minimum number of devices involved will be selected.

Frequent topology changes in power system result in the difference between the actual system operating conditions and the initial learning sample database. To guarantee the reliability of trained decision tree, it is necessary to incorporate new available training cases. Re-training the whole decision trees from scratch might is not as a cost-effective way. An ensemble method widely used for computer vision is used to update the classifier in an online manner.

A physical view of regional voltage controller is shown in the following figure. The analytic is divided into two parts: online and offline. The offline part includes the EMS system that archives the system snapshots. Each single system snapshot provides system topology, voltage measurements, and power flow information every five minutes. An offline adapter which creates/updates decision trees is also included in the offline section



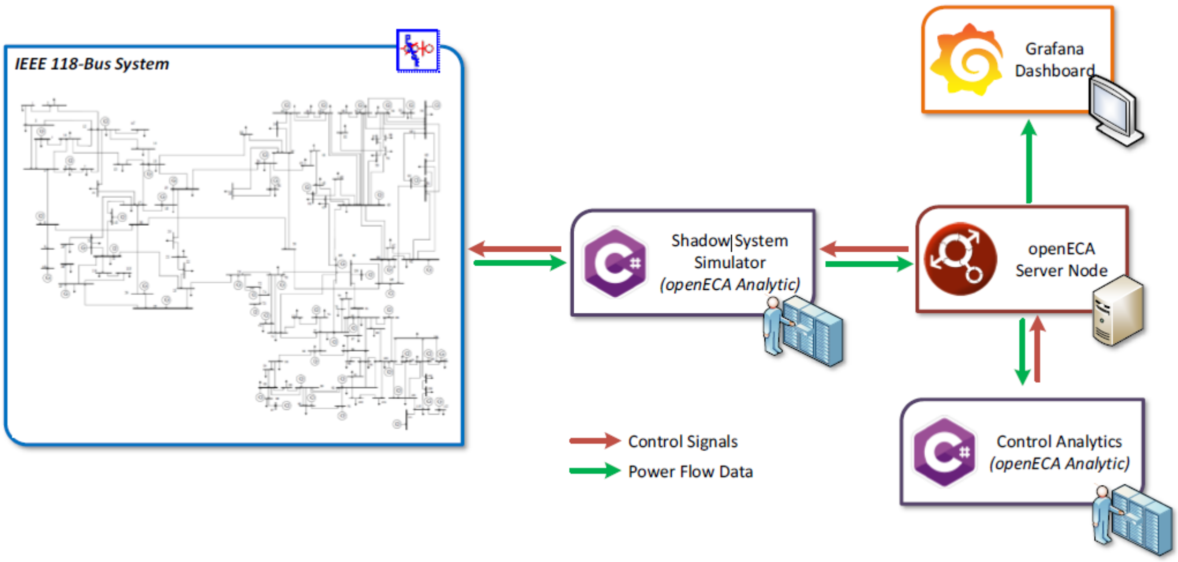
Concept of Software Framework

The online section includes multiple Phasor Measurement Units (PMU) which provide synchronized synchrophasor data with voltage/current magnitude and angle in real time. The openECA platform receives and time-synchronizes phasor data from PMUs to provide output data stream. Since the system might not be covered with PMU at each single bus, the system status still can be observed using state estimator. The Linear State Estimator developed for openECA is able to provide estimated phasor data and make the system network fully observable. In this case, the PMU estimated phasors are also considered as valid input to the voltage controller.

For additional demonstration results, see Appendix B.

9 SHADOW SYSTEM SIMULATOR

The shadow system simulator analytic interacts with the power flow engine in PSS®E and changes the system based on control signals. Before implementing any application in the field, the application should be evaluated in a simulation environment. This analytic initial testing with PSS/E based model. The PSS/E model is used to simulate the power system and to provide pseudo PMU measurements as input signals for the user Analytic implemented in openECA.



Cross-platform testing

For additional demonstration results, see Appendix B.

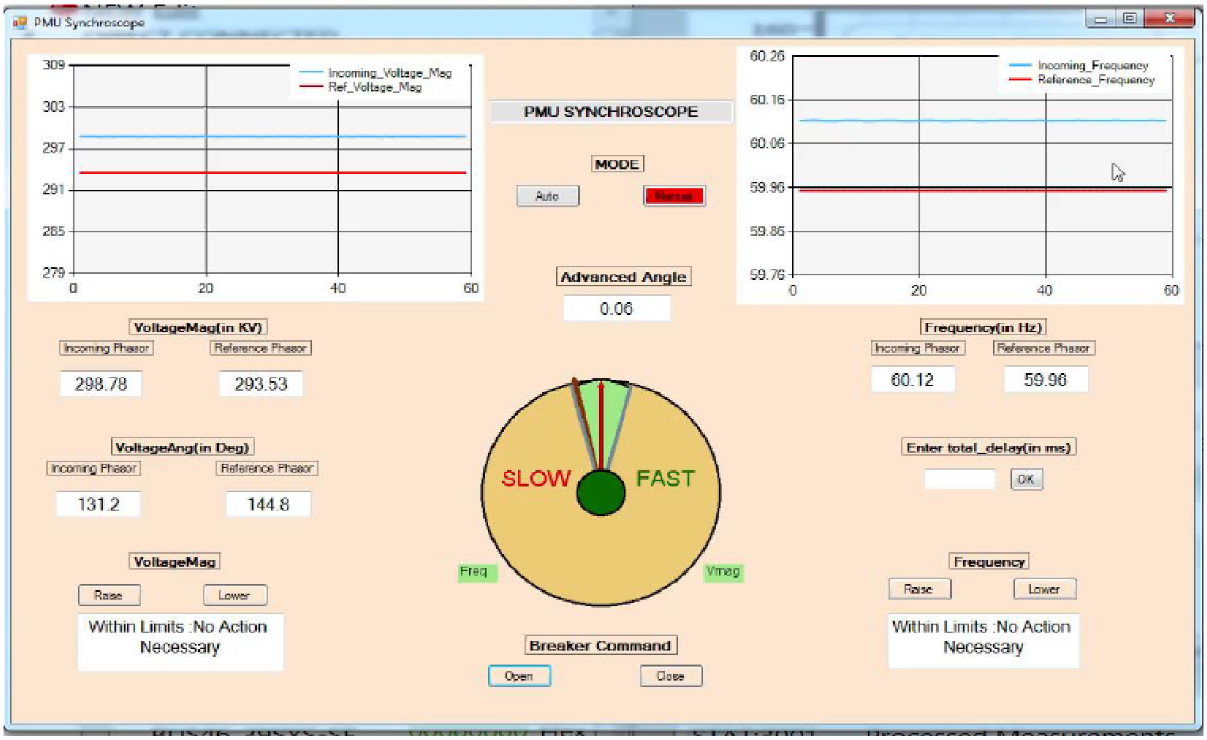
10 PMU Synchroscope

This analytic takes advantage of the high accuracy and time synchronization feature of PMU measurements. It enables synchroscope functionality at a centralized control platform thus facilitating remote synchronized breaker closing operation at any major substation. This function provides enormous flexibility and time savings for critical scenarios such as black-start system restoration and long extra high voltage (EHV) transmission line reclosing which generally requires deployment of personnel and physical synchroscope hardware at the location of synchronization

The analytic connects to a stream of synchrophasor data from the openECA platform and sends control signals back to the openECA from a centralized remote location. The significant variation included in this analytic in comparison to the existing synchroscope is the inclusion of various delays associated in the system and thus predicting an advanced or earlier instance of initiating breaker close command, thus effectively carrying out successful time synchronized breaker closing action at the substation location. The analytic can be fed with real-time estimations of the delays determined separately. All delays, generated by communication, data processing, computation and operation are overcome via estimating end-to-end delays and predicting closing time as described earlier.

All delays, generated by communication, data processing, computation and operation are overcome via estimating end-to-end delays and predicting closing time as described earlier. Operators can perform successful synchronization if the frequency and voltage phasor difference between the two buses to be synchronized displayed on the PMU synchroscope display are within acceptable user defined tolerance limits. The user has the option to perform this action either manually or automatically at an appropriate time to enable a computer algorithm to perform the breaker closing. The analytic can support estimation of angle-across-breaker to supplement system restoration activities or the closing of long transmission lines with the intention of minimizing the impact to the system.

As depicted in the following PMU Synchroscope display window, the user can provide the delays associated in the system. Depending upon these delays, an advanced angle is calculated, and a modified angle tolerance window is created within which a user/operator can initiate breaker close commands subjected to the fact that other conditions for frequency and voltage magnitude difference are satisfied. The breaker status is represented by the innermost circle (green in this figure). The plots for respective measurements can be visualized by the user along with the option to operate breaker close command in Manual or Auto mode as shown in the figure.



The primary challenge was to overcome communication (end-to-end) delays to enable accurate prediction of closing time for time-synchronized breaker closing actions.

Synchronizing Method:

To synchronize two separate islands, we need to retrieve voltage phasor measurements and frequency at Bus A and B. Cumulative delays are calculated depending upon network configuration and traffic of the path adopted. The advanced angle of operations is calculated considering delays and differences in voltage angles.

$$Adv.Ang = \left\{\frac{(Slip)cyc}{sec}\right\}\left\{\frac{sec}{60\ cyc}\right\}\left\{\frac{360}{cyc}\right\}\{(Cumulative\ Delay)cyc\}$$

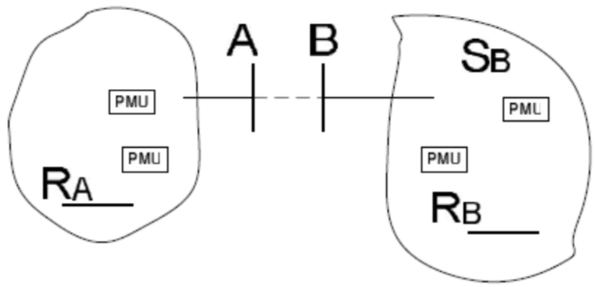


Figure 1: Two islands to be synchronized by closing breaker between Bus A and Bus B

The PMU Synchroscope analytic has the capability of synchronizing two buses from a remote centralized location. The downstream and upstream delays vary from one system to another, requiring that the advanced angle be calculated for each situation. During the demonstration, the OPAL-RT ePHASORSIM simulator was used to validate and demonstrate the comprehensive functionality of the analytic.

Load control techniques using voltage sensitivities of individual load buses inside the island are being incorporated as an add-on to the analytic.

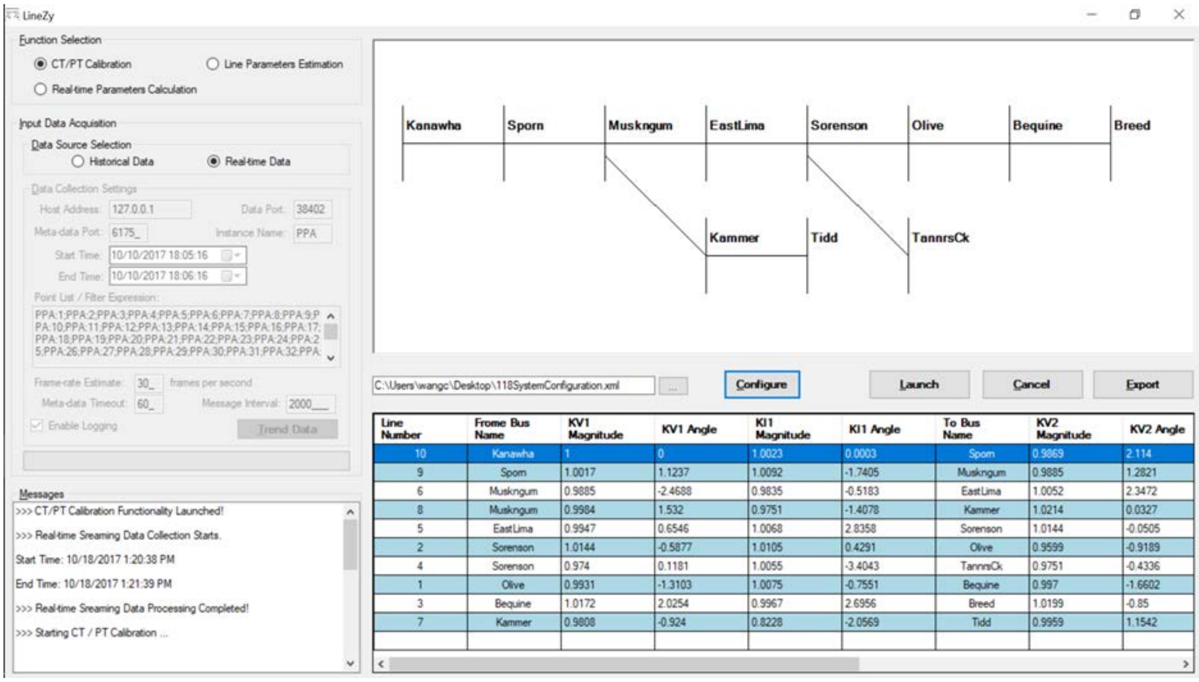
For additional demonstration results, see Appendix B.

11 Dynamic PMU Transducer Calibration

Current transformer and potential transformer bias errors can result in errors in both the magnitude of synchrophasor measurements as well as their phase angles. This analytic automatically calibrates voltage and current instrument transformers feeding PMUs. This analytic runs periodically (off-line) on the openECA platform, utilizes the configured transmission line data, and provides a set of complex ratio errors of transducers under monitoring with the Least Squares Estimation method. Real-time and historical input options are available for this analytic.

For this analytic, the system topology is analyzed to form a tree topology using an edge-based breadth-first search graph algorithm starting at the location of the high-accuracy PTs and CTs. For this project, Dominion Energy installed high-accuracy, high-precision Current Transformers (CTs) and Potential Transformers (PTs) to serve as the root point of calibration for all other phasor measurements. The estimation process is conducted for each single transmission line and the results of one line are used as the reference for the next line in the connected tree. A propagation method for the calibrated accurate measurements was also developed.

The following figure illustrates the user interface of this application. Within the application, users are provided with the options for the functionality they would like to deploy. After the selection of the function of the transducer calibration, users can choose the input data source, using historical data or collecting instantaneous data. The system diagram can be configured using XML files which contain the topology information of the concerned system. After launching the functionality, the calibration results are shown in a table. A function to export the calculation results into local CSV or XLSX files is also available.



Dynamic PMU Transducer Calibration Function User Interface

Demonstration results showed good correlation between estimated and actual CT/PT correction factors. In addition, the integrated application provides a good example for future developers of off-line and on-line analytics utilizing the openECA platform.

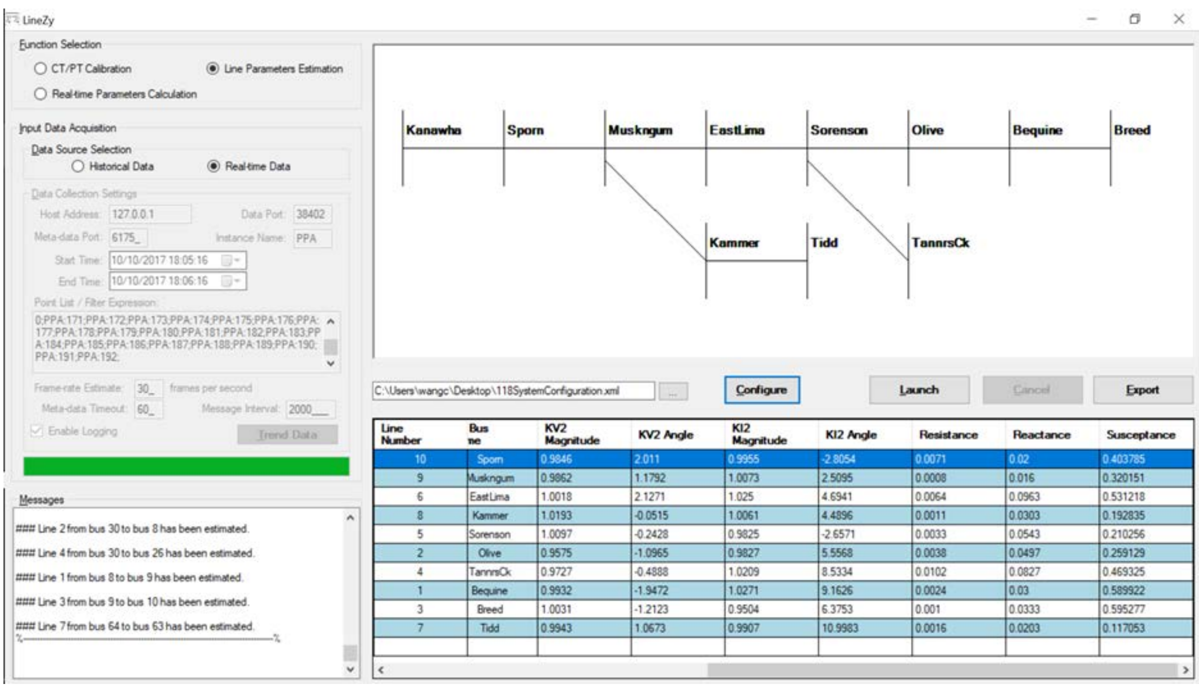
For additional demonstration results, see Appendix B.

12 LINE PARAMETER ESTIMATION

Improving the quality of the line impedance parameters directly benefits planning and real-time operations. This periodic (off-line) analytic component uses synchrophasor data to estimate line parameters across the PMU footprint. The data of the system will be provided in a xml file format. This file is used to form the propagation path as described for the Dynamic PMU Transducer Calibration analytic.

The analytic can be used with PMU measurement data from the openECA or openHistorian platforms. A least squares estimation method provides the CT/PT measurements correction factors and the line impedances.

The functionality of this analytic is demonstrated in the following user interface. After the function of line parameter estimation is selected, the PMU measurements are collected directly from openECA streaming data. With the system diagram configured, the estimation results are shown in the table. The results can then be exported into the local files.



Line Parameter Estimation Function User Interface

Demonstration results showed good correlation between actual and estimated transmission line resistance, reactance, and susceptance parameters. For additional demonstration results, see Appendix B.

13 Synchronous Machine Parameter Estimation

Improving the quality of synchronous machine model parameters will provide benefits both in planning and operations. It has recently been shown that an effective method to identify and validate synchronous machine model parameters is to compare and match event signatures captured by PMUs against simulated event signatures generated by the machine model under test. This periodic analytic component will automate the process of synchronous machine model parameter estimation and validation – a process which is currently labor-intensive, and which requires expertise from highly skilled personnel.

When given synchrophasor data from a PMU located at or near the GSU transformer of a synchronous generator, this analytic can automatically perform NERC-mandated model validation. The analytic was developed and proofed in an offline mode and tested online in openECA using the Matlab template.

Inputs to the analytic are voltage and current phasors from a PMU located as close as possible to a generator's terminals so they are accurately measuring generator output current. The parameters of the model to be validated must be stored in persistent memory and loaded to the analytic.

Next steps are to port the analytic to .NET, extend the model library to include additional generator, turbine, governor, exciter, and PSS models in PSLF, PSS/E and Powerworld formats, extend the research portion of the effort to include wind power plants models, and improve the output report template.

For additional demonstration results, see Appendix C.

.

14 Acceleration Trend Relay

A select category of synchronous generators, typically remotely located and connected to the bulk power grid over long transmission lines, are protected by an acceleration trend relay (ATR). The input to an ATR is generator shaft speed. If the shaft is accelerating beyond preset limits, the generator may be tripped offline to protect the unit.
A complication arises when the entire system is accelerating. In this case the unit may "false trip", i.e. the ATR may call for a trip when the unit is not in danger of an out-of-step condition. Incorporation of synchrophasor measurements remote from the unit may help eliminate false trips.

Inputs to the analytic are voltage phasor(s) from points representing the presumed inertia of the bulk grid to which ATR-equipped generator is connected. The output is an "ATR block" Boolean that would be used to block the trip command from the ATR relay.

Plans are for the developer to work with a host utility to build the value proposition for eliminating false trips.

For additional demonstration results, see Appendix C.

15 Utility Demonstration Observations

Dominion Energy

- openECA
    - Dominion believes that we can now easily develop/deploy its own analytics and easily work with partners to develop/test/deploy analytics using openECA
- openLSE
    - Easier to deploy and maintain an LSE at Dominion
    - Increased user population will continue to improve performance and features
- All Analytics
    - Provides a variety of development examples for the Dominion and the open source community
- Local and Regional Voltage Control Analytics
    - It has been valuable to experiment with advanced types of control with synchrophasors
    - It has also been valuable to experiment with new technologies like machine learning and cloud computing

  - It will be useful to make people aware of the capabilities of synchrophasor based control with control schemes that they are comfortable with

- Shadow System Simulator Analytic
  - Provides a basic, open source solution for steady-state simulation in pseudo-real-time with openECA – Saves time in working with proprietary tools and models
- CT-PT Calibration
  - Has the potential to provide equipment health monitoring and improve overall PMU data quality
- Line Parameter Estimation
  - Has the potential to improve knowledge of circuit parameters
- Line Z Calc
  - Simple calculator – primarily used as a development example and a troubleshooting tool for the line parameter estimation
- Topology Estimation
  - Enables adoption of LSE when utility has to breaker telemetry in their PMU streams
- PMU Synchroscope
  - To increase flexibility and speed of restoration time during severe restoration scenarios.

BPA

- After some issues getting the software installed, the openECA server was able to quickly connect to the available BPA multicast synchrophasor data feed and provide data to the openHistorian and Grafana visualizations.
- BPA is considering standing up a Grafana-based user interface for openECA-provided oscillation results for comparison to the production system that is currently operational.
- BPA plans to continue to evaluate the openLSE and compare its results to other LSE software currently used.
- Grafana and the openHistorian will be evaluated for other potential uses within BPA. Flexibility in adding displays/content a positive.
- BPA will continue to experiment with openECA generated Matlab project templates for bringing synchrophasor into Matlab and sending results back to openECA
- Detailed stream statistics provide real-time detail not available with other phasor data concentrators. These statistics will be compared with daily/weekly reports compiled by BPA.

OG&E

OG&E demonstrated opeECA, openLSE, and ODM.

openECA:

- Installation and Configuration
    - Installation and setup is easy and very similar to openPDC, which OG&E has used for many years.
    - No issues were encountered with getting data into openECA and openHistorian.
- Performance
    - 400+ PMUs with 3 Phase system requires minimal resources (3-4% CPU for each service).
- Stability
    - A crash of the services occurred one weekend. It appears to be a result of the openECA Manager being left open and crashing and taking the openECA service down with it. These should be better isolated so that cascading failures are not possible.

openLSE:

- Side-by-side results of PMU data and openLSE output show good correlation.
- Output data are delayed by 10-15 seconds compared to PMU data.

ODM:

- ODM is very sensitive to step changes, faults, and line switching.
- ODM does not include alarming or notification capabilities which are important for validation. OG&E developed a stand-alone notification service that queries openHistorian.

General:

- OG&E plans to continue to validate the ODM and tweak alarming to provide meaningful notification of events.
- OG&E also plans to test the oscillation mode meter (OMM) in the near future
- OG&E will continue to work on openLSE
- OG&E would like to test the line impedance calculator in the future.

SPP

openECA Platform:

- Installation and Configuration -

  - As with all GPA tools, installation and data flow configuration was painless
  - Attaching a dedicated Historian was straightforward
- User Interface
  - The openECA Manager thick client has a lot of functionality, but moving to web-based is the right direction
  - Bulk operations have to be done in the database
  - Power Calculations are difficult to maintain
- Analytics Integration
  - Both analytics SPP tested were easily able to connect to openECA, receive metadata and real-time data
  - Process of creating input/output measurement mappings is a challenge. Mappings are static, not easily maintainable as models and device availability changes. Both analytic developers tried to streamline.
- Suggested Improvements
  - Support of additional metadata fields for devices and signals, specifically for LSE:
    - Substation
    - Voltage Level
    - Measured Device Type
    - Measured Device Name
    - Opposite Substation (for lines)
- Continue web-based manager development for all the GPA platforms

<u>openLSE:</u>

- SPP and OGE had several iterations through the network model build process
- openLSE is still evolving - Dr. Jones worked hard to add functionality to assist SPP and OGE during testing
- Large time investment required to map PMU info to model
- openLSE is very dependent on accurate measurement mapping
- With latest improvements, a ground-up model build should be easy
- Suggested Improvements
  - Support cleanup of unused openLSE output measurements - each station has many unused node voltage groups
  - Add a Network Topology Viewer
    - A user interface with a graphical layout of the topology would make the LSE results much easier to use.
    - Could use node-graph layout to visualize topology from the network model or bring in measurements and estimates from openECA.
    - Add support for measurement residual streaming to openECA which could be useful for quickly identifying mismatches

General:

- SPP sees great value in openECA and the potential for new analytics. We hope that the framework continues to grow and be adopted by industry.

- SPP will continue to run ODM and openLSE in a test environment.
- SPP plans to test OMM and Line Parameter Estimation as time permits.

<u>ODM:</u>

- ODM-reported Percent Bad Data is always > 0%. For most PMUs, closer to 30-50%. The data appears good on the input side.
- Easy to visually recognize trends in large amounts of data with ODM + openHistorian + Grafana
- ODM is flexible - accepts and processes signals of any type
- Adding measurements, a bit tedious at first. SPP added a filter to the ODMapper datagrid to streamline signal selection
- Suggested Improvements
    - Built-in time-based alarming (e.g. >5MW oscillation for >30 seconds)
    - An output signal indicating frequency of largest oscillation present in the band for each PMU
    - Configurable frequency ranges for each band

Complete presentations of site demonstration results are included in Appendixes D-G.

APPENDIX A – OPENECA PLATFORM

Insert Carroll – openECA – November 7 R1

APPENDIX B – VIRGINIA TECH ANALYTICS

Insert VT Nov7th_Compiled_version

APPENDIX C – T&D ANALYTICS

Insert Donnelly – openECA – November 7.v2

APPENDIX D – DOMINION ENERGY DEMONSTRATION RESULTS

Insert Jones Nov 7 Dominion Findings

APPENDIX E – BPA DEMONSTRATION RESULTS

Insert BPA demo results – openECA R1

APPENDIX F – SPP DEMONSTRATION RESULTS

Insert SPP_openECA_Results

APPENDIX G – OG&E DEMONSTRATION RESULTS

Insert OGE_openECA_Results

J. Ritchie Carroll
Grid Protection Alliance

Dominion Energy Offices
Richmond, VA
November 7, 2017

The openECA Platform Design

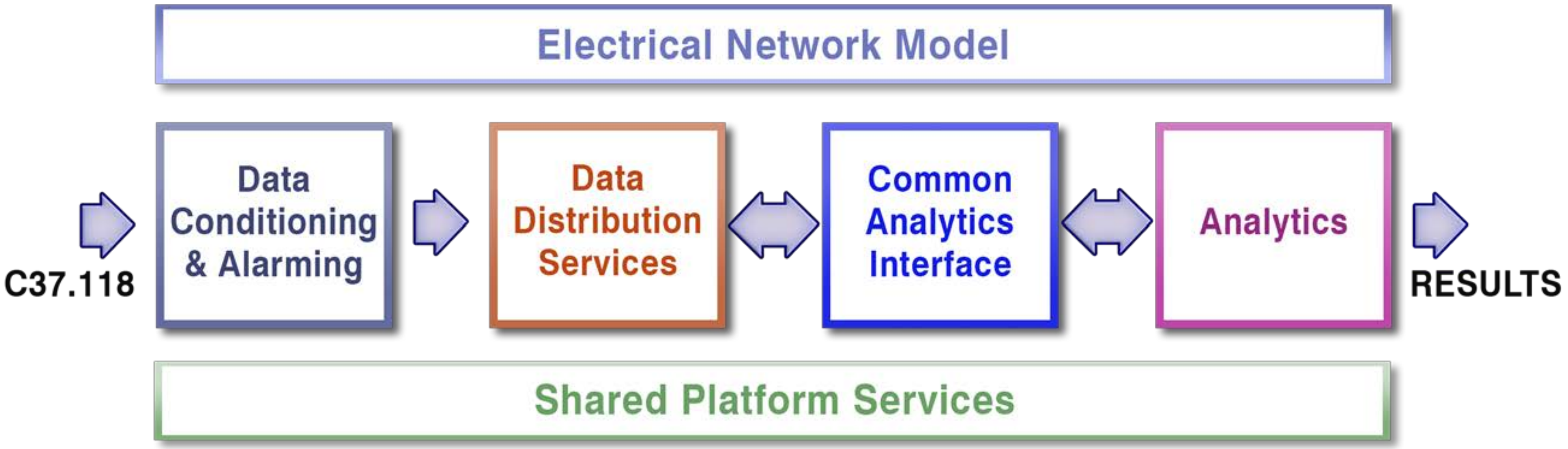
Development Approach

- **Build upon existing open source solutions**- Leverage GPA's production-grade open-source code base to create an open source application suite under a permissive license
- **Develop a standard interface** - Provide a "Common Analytics Interface" (CAI) where "data structures" are made available for subscription
- **Detect Bad Data Early** - Create a multi-tier bad data detection and correction system with alarming services
- **Create "3rd Generation" Data Exchange Methods** - Provide secure phasor data exchange using a next-generation version of the Gateway Exchange Protocol and the forthcoming Secure Telemetry Transport Protocol
- **Include Visualization Tools** - Develop a visualization tool optimized for testing and verification of analytic results
- **Test and Refine** - Test the CAI with 10 provided analytics at four utility partner locations
- **Create an Analytics Storefront:**

  https://github.com/GridProtectionAlliance/openECA/tree/master/Source/Analytics

Architectural Elements

- Data Conditioning / Alarming
- Data Distribution Service
- Common Analytics Interface (CAI)
- Electrical Network Model
- Shared Platform Services
- Analytics



Data Conditioning and Alarming Summary

- Data conditioning is provisioned through the Linear State Estimator (LSE):

  https://github.com/kdjones/lse

- The LSE engine is hosted in an openECA analytic that integrates with the Data Distribution Service for input and output:

  https://github.com/kdjones/openlse

- Conditioned result data from the LSE is made available along with real-time data to other analytics
- Alarms can be configured based on residuals, i.e., differences between measured and estimated data

Data Distribution Service Summary

- Real-time and historical data acquisition
  - Adapter-driven data providers will include:
    - All common synchrophasor protocols
    - Various common RDBM systems for configuration, including:
      - MS SQL Server, Oracle, MySQL, PostgreSQL and SQLite
    - OSI-PI, openHistorian and other historians
    - Other protocols, e.g., Modbus, DNP3, Kafka, COMTRADE
- Device management
  - Automated connectivity
  - Data quality reporting
- Time-series data management

Time-series Data Management Functions

- Measurement Definition
  - Flexible Data Types
  - Automatic Creation from Phasor Sources
- Measurement Validation
  - Data Quality Testing
  - Flat-line Detection
- Data Acquisition and Routing
- Adapter Configuration Management
- Linear State Estimation Value Integration
  - Available via secondary install and configuration
  - Allows grouping of actual and estimated values
- Archiving of System Performance Metrics

Common Analytics Interface (CAI) Summary

- Server API (targets .NET)
  - Authorizes client data source connectivity
  - Provisions time-series data and metadata
- Client API (targets multiple platforms)
  - Manages server connectivity
  - Executes data filtering, organization, aggregation and time-alignment functionality over user defined time-intervals
- Data Modeling Manager Tool
  - Defines data filtering, organization and aggregation
  - Trends incoming data sources

CAI – Server API

Configuration Serialization

- Data class structure definitions
- Labeling and identification of class instances with full measurement mapping

Security Management

- Validating clients and connections
- Validating access to needed measurements

Open API for Multivendor Support

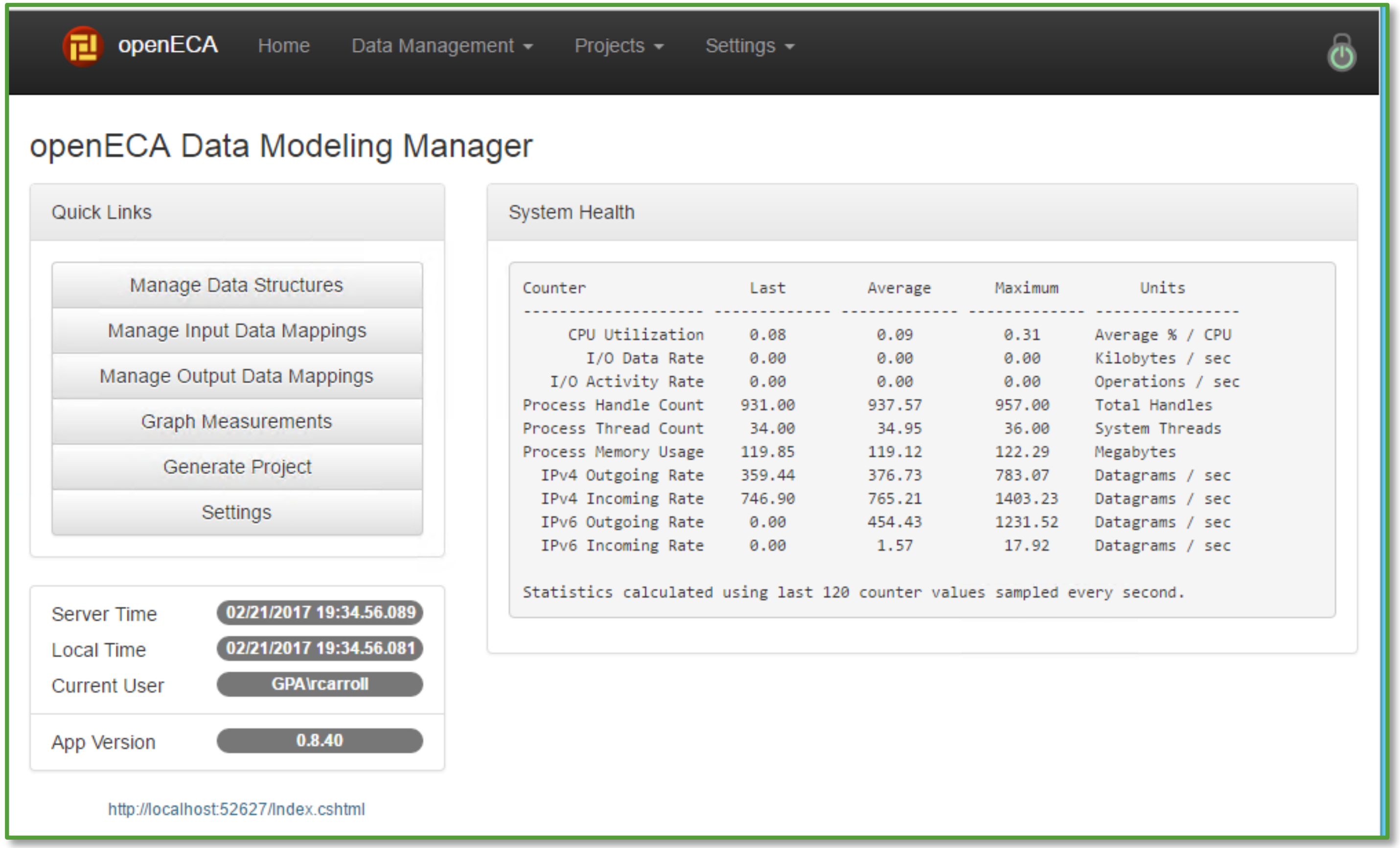
- Server API will define minimum requirements needed to implement a server side solution that will allow vendors to support analytics written using the openECA Client API

CAI – Client API

- User Defined Data Structure Definitions and Mappings
  - Automatically referenced within analytic tools, the Client API will exist as a set of base services used to retrieve sets of user defined data for input and outputs – i.e., user defined complex input and output data structures that are mapped to time-series data values
- User Defined Data Collection Windows and Arrays
  - The Client API will handle time-alignment of incoming data and support creation of “windows” of data to an analytic, e.g., providing a one second window of data to an analytic every second

CAI – Data Modeling Manager Tool

- Organizes data into logical groupings
- Creates data structures that directly map to time-aligned measurement values
- Provides a visual representation of these user defined data structures
- Displays source data as simple trends



Shared Platform Services Summary

- Security and authentication services
- Time-series data transport with support for multiple data types
- Metadata distribution and synchronization services
- Management of data structure definitions and associated point mappings
- RDBMS connectivity, as required
  - *Includes support for SQL Server, Oracle, MySQL, PostgreSQL and SQLite*
- Logging services

Analytic Development

- Using the Data Modeling Manager tool, the openECA defines a unified environment for modeling analytic:
  - Configuration
  - Data Structures and
  - Measurement Mapping
- The openECA project templates target modern development tools
- Project templates also include tools to deploy analytics, e.g., setup packages that include registration of the analytic as a Windows service application.

Analytic Data Structure Definition

- The Data Modeling Manager Tool allows the analytic developer to define two classes of data structures:
  - The first is the domain input, e.g., SourceData*
  - The second is the analytic product, e.g., ResultData*

    The contents of these data structures are under the complete control of the analytic developer

- The SourceData is automatically populated with time-aligned data over the desired data window
  - Auto-generated code that populates the SourceData structure is available for the analytic writer to review and validate
  - Note that a parallel metadata structure that includes timestamp and quality information is also provided as function input
- The ResultData is populated by the analytic developer and returned at the end of processing

* *Not a fixed name, developer names these data structures*

Analytic Entry Point

Either by using a template for a quick start or starting a new analytic from scratch, the analytic writer will be implementing analysis code from a single point of entry, i.e., a single function call into their analysis code called on an interval of their choosing:

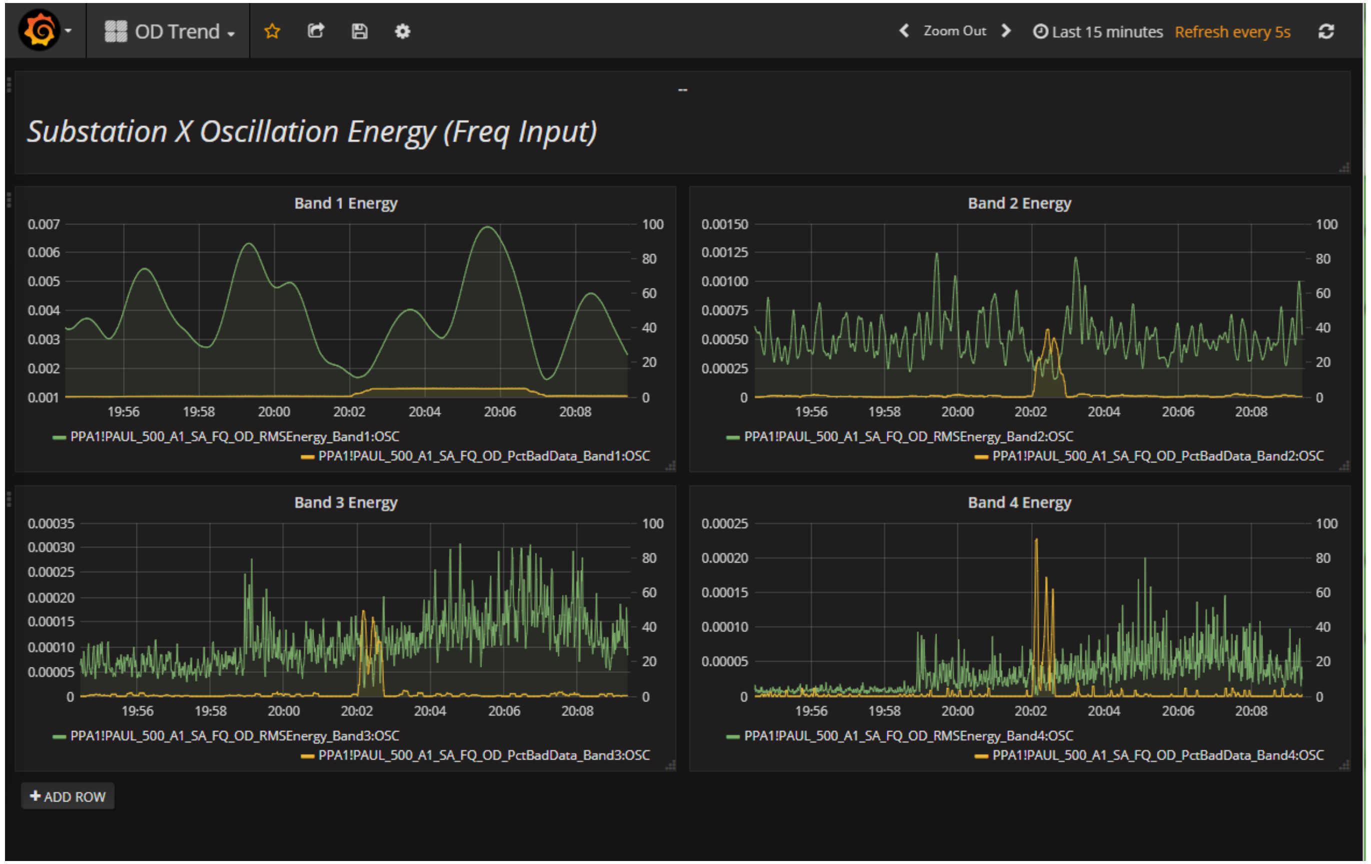
ResultData Mapping | SourceData Mapping | SourceMeta Mapping

```
MyOutputType Execute(MyInputType inputData, MyInputTypeMeta inputMeta)
{
    // Analytic code goes here…
}
```

Analytic Results and Visualization

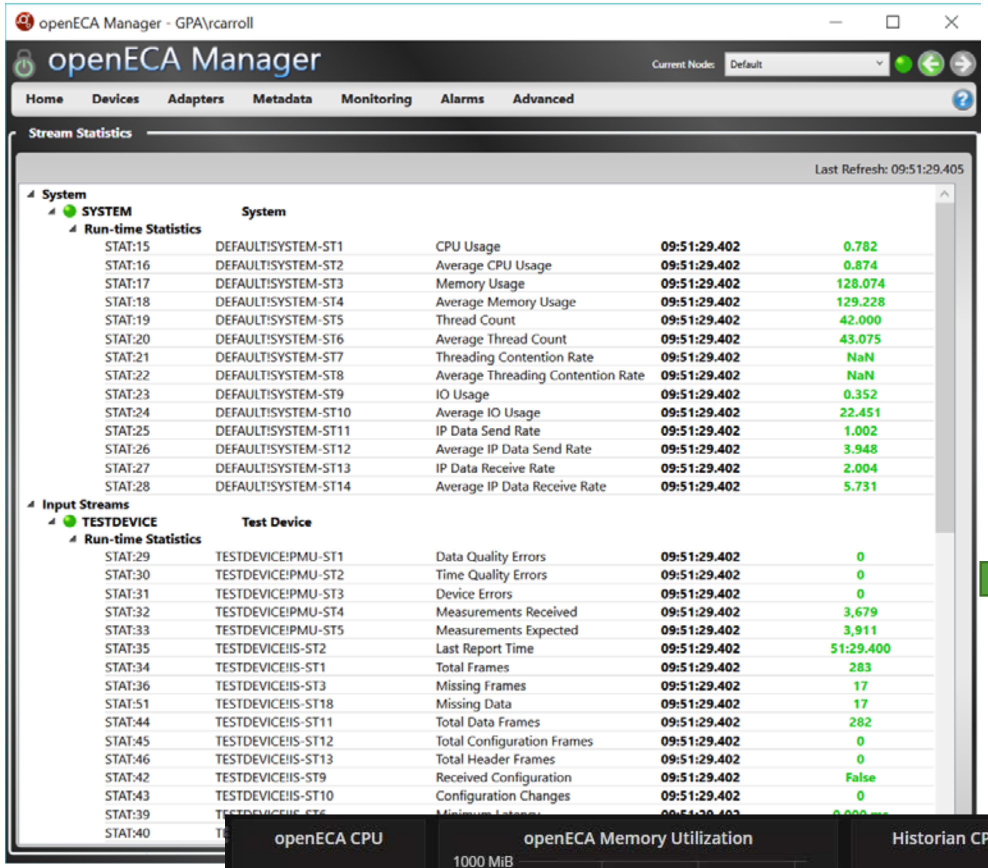
- The ResultData structure is used by the analytic developer to hold the calculated results
- The analytic result data are defined as a data structure so that results:

  1) flow back to the Server API as a unit and can easily disseminated to historians and other adapter-based outputs
  2) are directly available as a complete input into other analytics

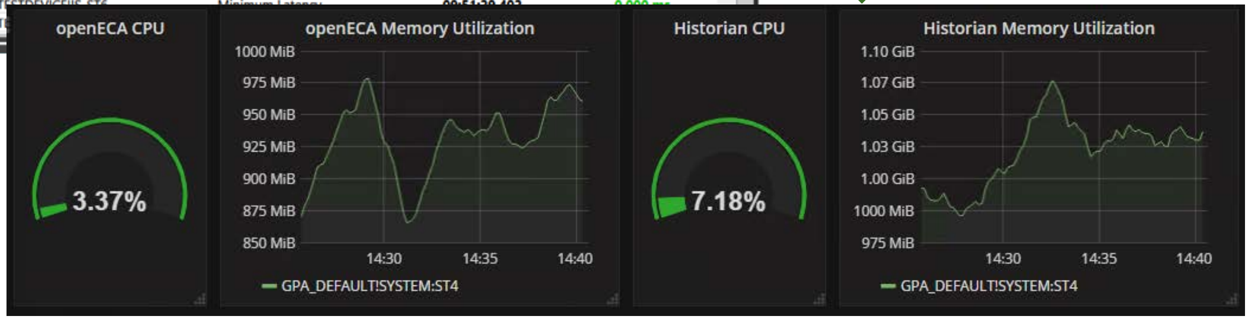


System Performance Metrics

- Performance of the openECA is continually monitored and archived – metrics include, among others:
  - *CPU Utilization*
  - *Memory Usage*
  - *Thread Count*
  - *Lock Contention*
  - *Bandwidth Utilization*

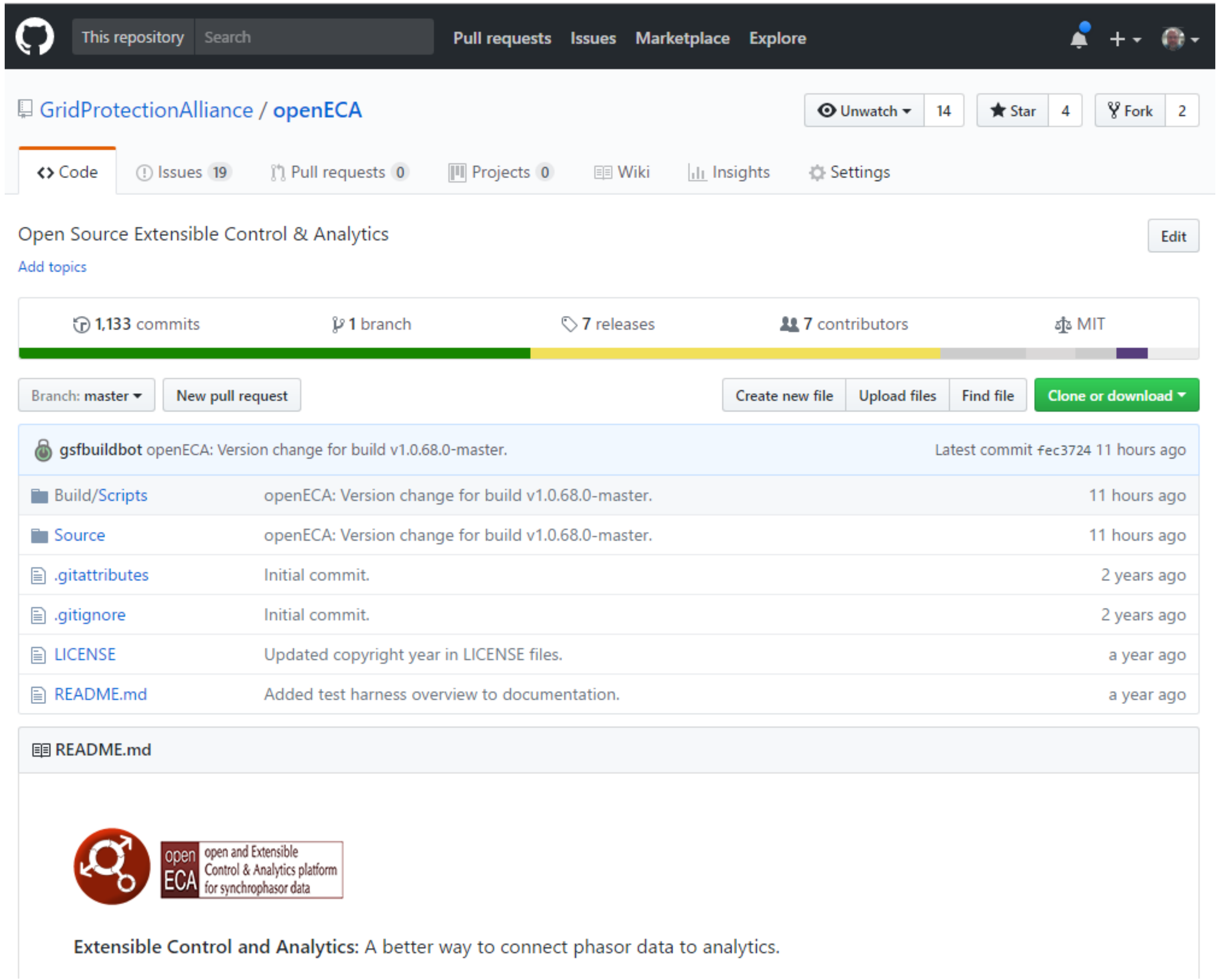


Metrics Visualized in Grafana



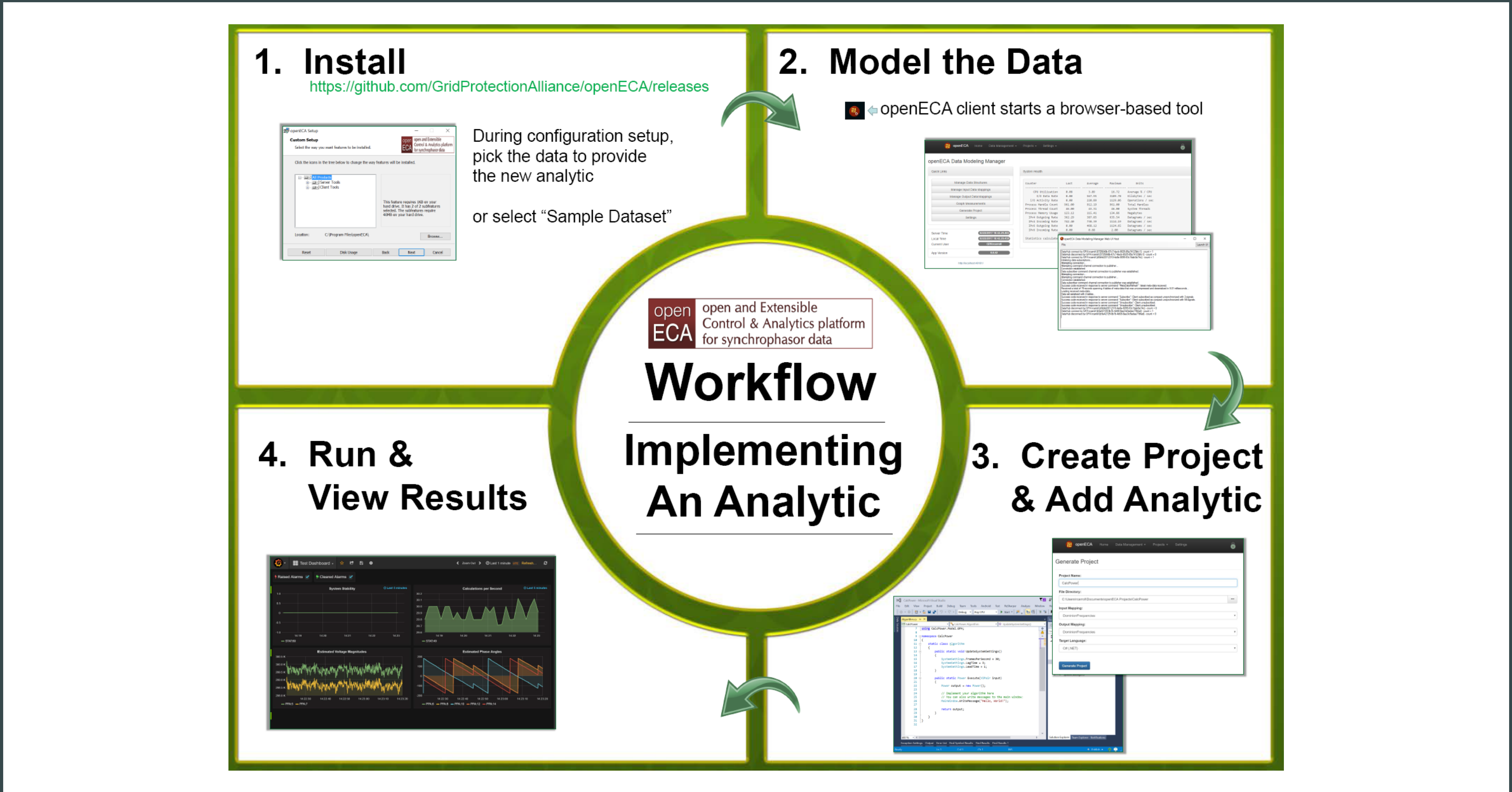
The openECA platform is available on GitHub

https://github.com/GridProtectionAlliance/openECA



Releases

- Beta – June 2017
- Pre-Production – Sept 2017
- Project Final – Jan 2018


1. Install
https://github.com/GridProtectionAlliance/openECA/releases
During configuration setup, pick the data to provide the new analytic
or select "Sample Dataset"
2. Model the Data
openECA client starts a browser-based tool
open ECA open and Extensible Control & Analytics platform for synchrophasor data
Workflow
Implementing An Analytic
3. Create Project & Add Analytic
4. Run & View Results

Virginia Tech/Dominion Energy Demonstration Results Report

Real-time Line Impedance Calculator,

CTPT Calibration, **and**

Transmission Line Parameters Estimation

Prepared by: Chen Wang

October 20, 2017

DOE FOA 970
DE-OE0000778

Dominion Energy

VirginiaTech

GRID PROTECTION ALLIANCE

openECA open and Extensible Control & Analytics platform for synchrophasor data

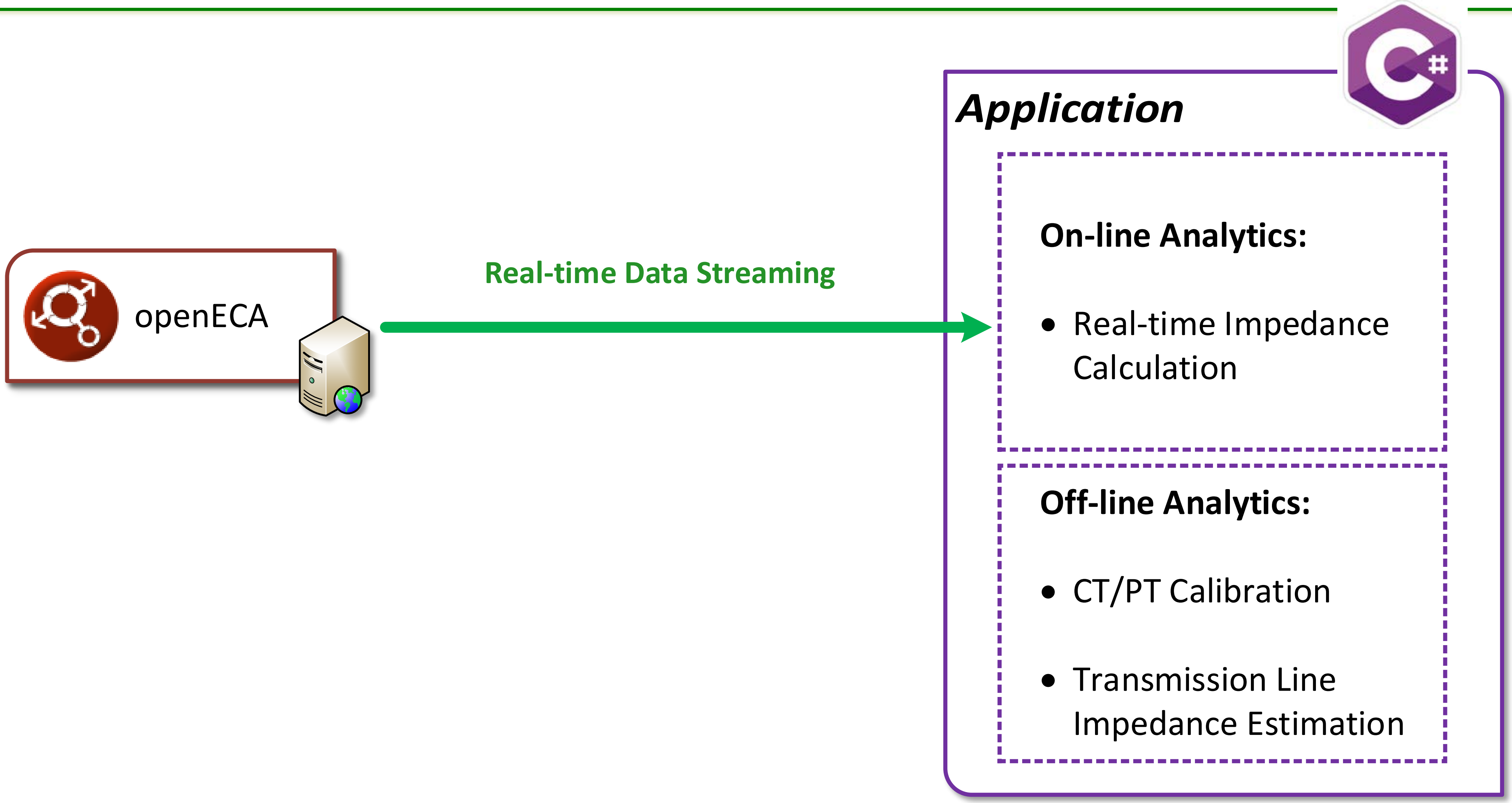
Deployment Results Report

Real-time Line Impedance Calculator

Virginia Tech Real-time Line Impedance Calculator Observations

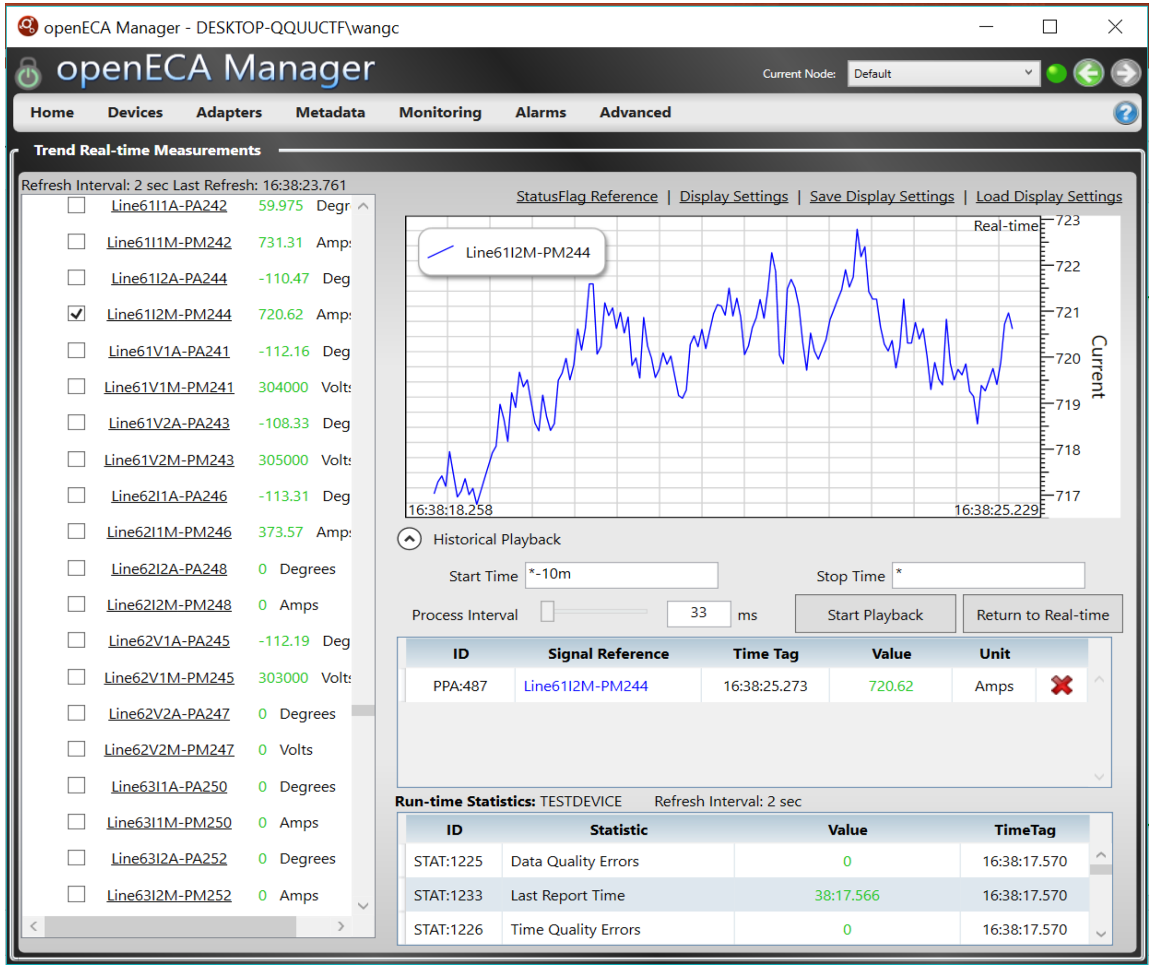
- Analytic Installation and Configuration Observations
  - The installation was convenient and quick. Cooperating with the configured openECA platform, the application can properly acquire the real-time data streaming. The system diagram can be demonstrated by the application with the provided XML file containing system topology information.
- Issues List
  - Real PMU measurements data quality need to be improved.
- Stability
  - The application can continuously and stably provide transmission lines parameters calculation results. No stability issues have been noted so far.
- Results display / presentation
  - Test based on both the IEEE 118-bus standard system and the Dominion Energy Virginia system are conducted. The line parameters calculated are demonstrated based on the system diagram generated by the windows application. The results are refreshed every 0.3 seconds currently for results observation. The reporting rate can be increased as needed.

Application Data Flow

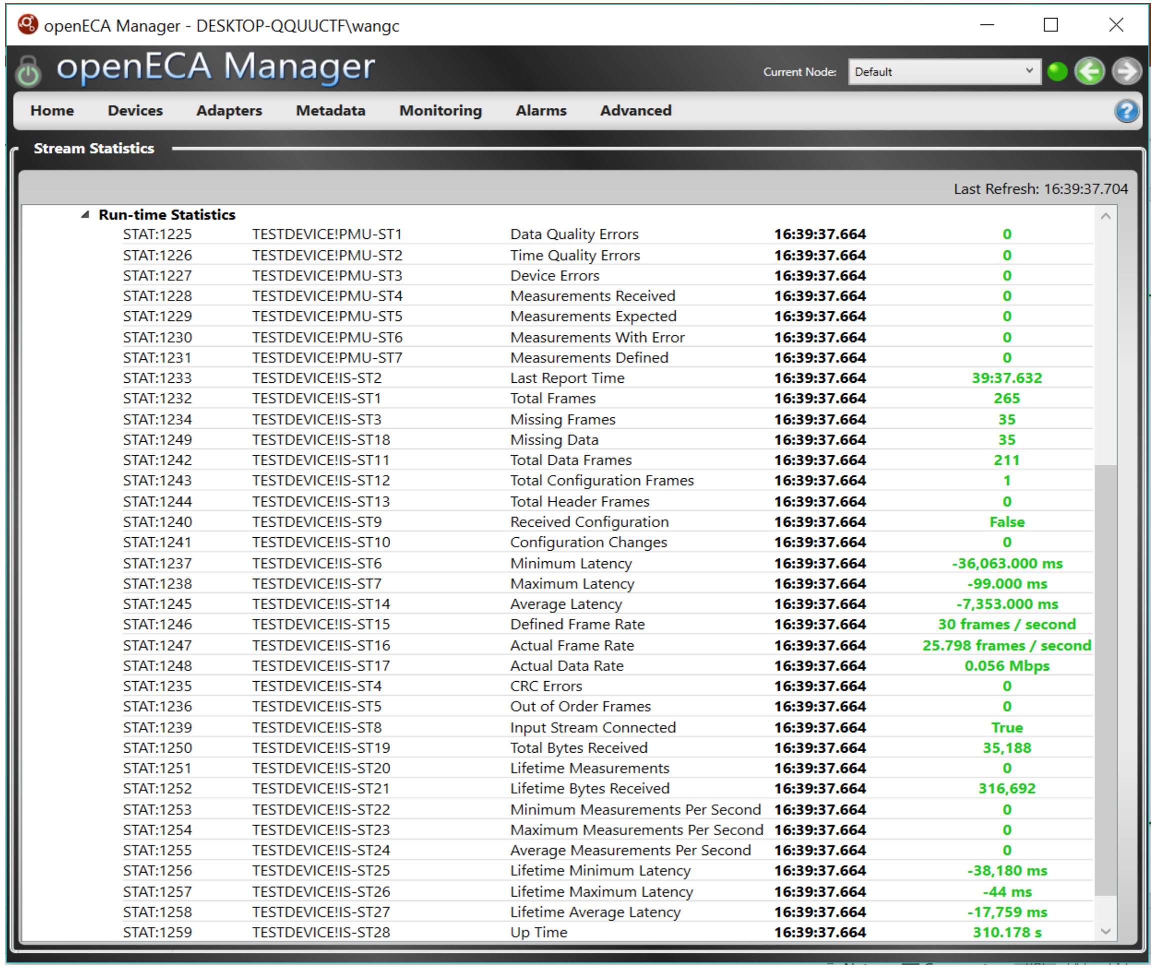


openECA Streaming Measurements

Real-time Streaming Measurements



Run-time Statistics

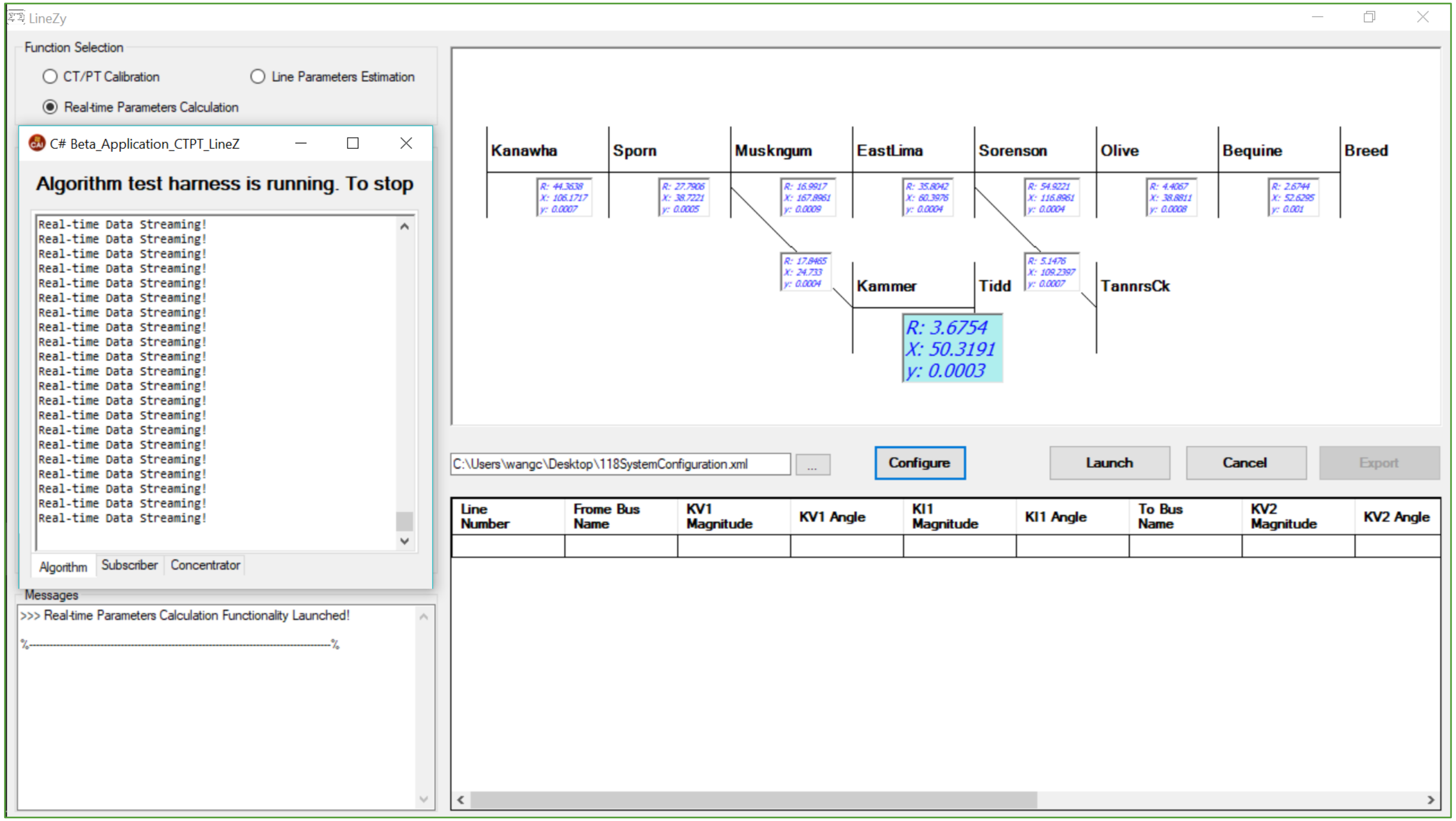


IEEE 118-bus Standard System Topology Information


```xml
<?xml version="1.0" encoding="utf-8"?>
<StudyCase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" BaseMVA="100" BaseKV="345">
  <Branches>
    <Branch LineNumber="1" FromBusNumber="8" ToBusNumber="9" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="2" FromBusNumber="8" ToBusNumber="30" Resistance="0.00431" Reactance="0.050399998" Susceptance="0.256999996" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="3" FromBusNumber="9" ToBusNumber="10" Resistance="0.00258" Reactance="0.032200001" Susceptance="0.615000005" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="4" FromBusNumber="26" ToBusNumber="30" Resistance="0.00799" Reactance="0.086000005" Susceptance="0.454000003" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="5" FromBusNumber="30" ToBusNumber="38" Resistance="0.00464" Reactance="0.054" Susceptance="0.210999996" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="6" FromBusNumber="38" ToBusNumber="65" Resistance="0.00901" Reactance="0.098600002" Susceptance="0.522999992" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="7" FromBusNumber="63" ToBusNumber="64" Resistance="0.00172" Reactance="0.02" Susceptance="0.108000002" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="8" FromBusNumber="64" ToBusNumber="65" Resistance="0.00269" Reactance="0.0302" Susceptance="0.189999998" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="9" FromBusNumber="65" ToBusNumber="68" Resistance="0.00138" Reactance="0.015999995" Susceptance="0.319000001" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="10" FromBusNumber="68" ToBusNumber="81" Resistance="0.00175" Reactance="0.020200001" Susceptance="0.404000015" HighVoltageLineFlag="true" ReferenceFlag="true" />
    <Branch LineNumber="11" FromBusNumber="68" ToBusNumber="116" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="12" FromBusNumber="5" ToBusNumber="8" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="13" FromBusNumber="17" ToBusNumber="30" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="14" FromBusNumber="25" ToBusNumber="26" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="15" FromBusNumber="37" ToBusNumber="38" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="16" FromBusNumber="59" ToBusNumber="63" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="17" FromBusNumber="61" ToBusNumber="64" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="18" FromBusNumber="65" ToBusNumber="66" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="19" FromBusNumber="68" ToBusNumber="69" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="20" FromBusNumber="80" ToBusNumber="81" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="21" FromBusNumber="8" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="22" FromBusNumber="10" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="23" FromBusNumber="26" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="24" FromBusNumber="65" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
  </Branches>
  <Buses>
    <Bus BusNumber="8" BusName="Olive" ReferenceFlag="false" />
    <Bus BusNumber="9" BusName="Bequine" ReferenceFlag="false" />
    <Bus BusNumber="10" BusName="Breed" ReferenceFlag="false" />
    <Bus BusNumber="26" BusName="TannrsCk" ReferenceFlag="false" />
    <Bus BusNumber="30" BusName="Sorenson" ReferenceFlag="false" />
    <Bus BusNumber="38" BusName="EastLima" ReferenceFlag="false" />
    <Bus BusNumber="63" BusName="Tidd" ReferenceFlag="false" />
    <Bus BusNumber="64" BusName="Kammer" ReferenceFlag="false" />
    <Bus BusNumber="65" BusName="Muskngum" ReferenceFlag="false" />
    <Bus BusNumber="68" BusName="Sporn" ReferenceFlag="false" />
    <Bus BusNumber="81" BusName="Kanawha" ReferenceFlag="true" />
  </Buses>
</StudyCase>
```

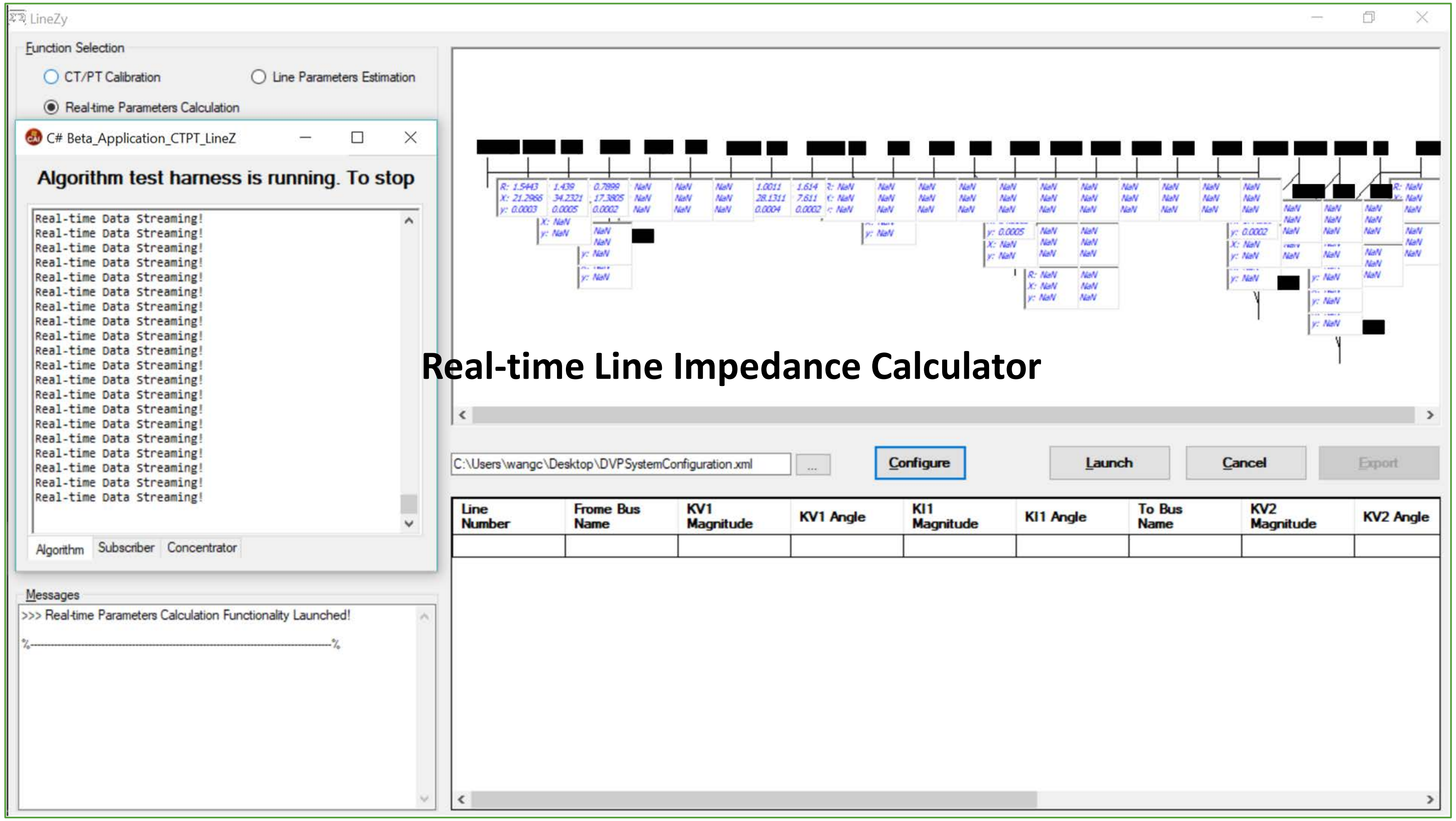
Application Calculation Results Example Demonstration

IEEE 118-bus System



Application Calculation Results Example Demonstration

Dominion Energy Virginia Field Data



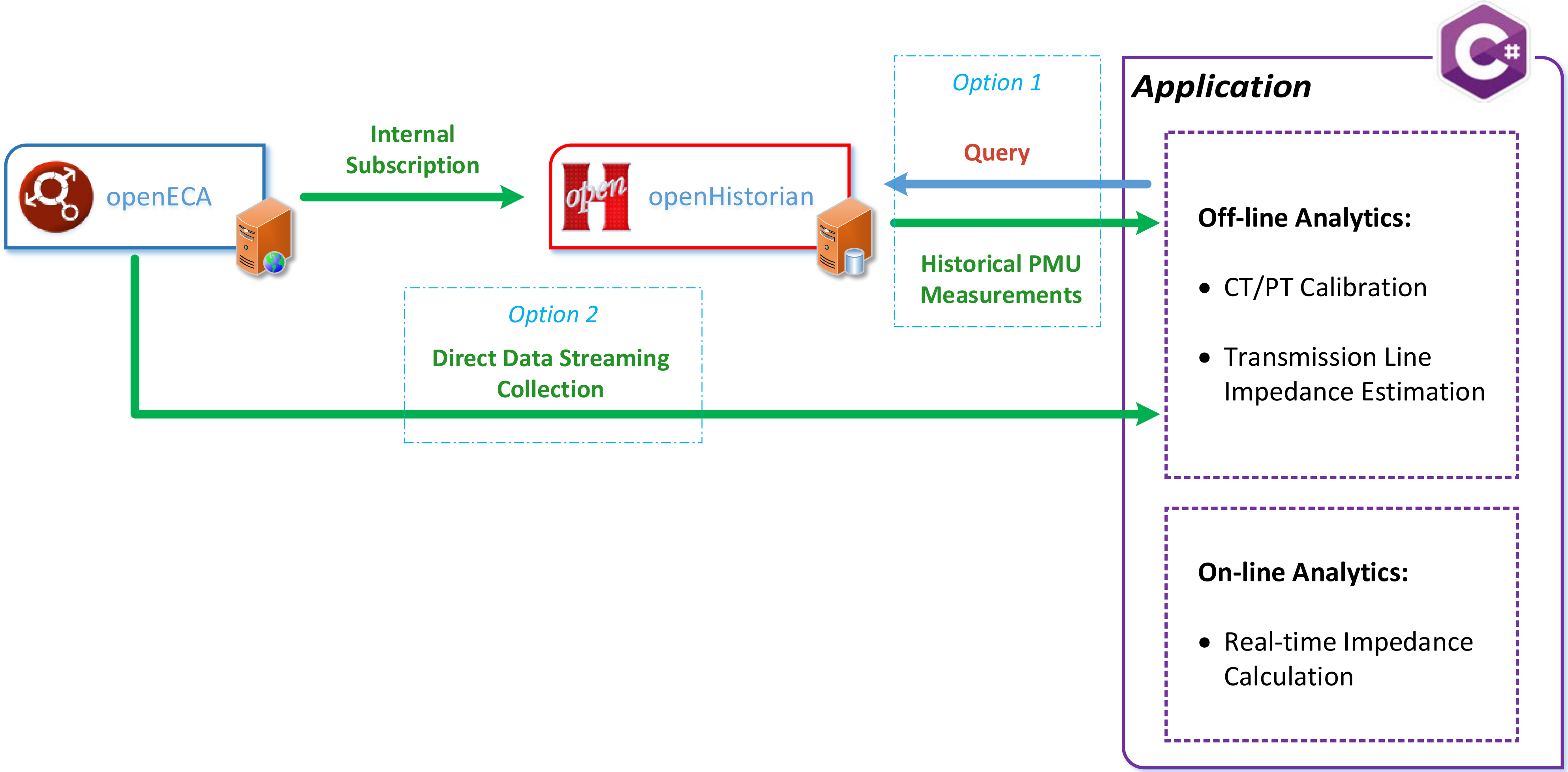
Deployment Results Report

CT/PT Calibration

Virginia Tech CT/PT Calibration Observations

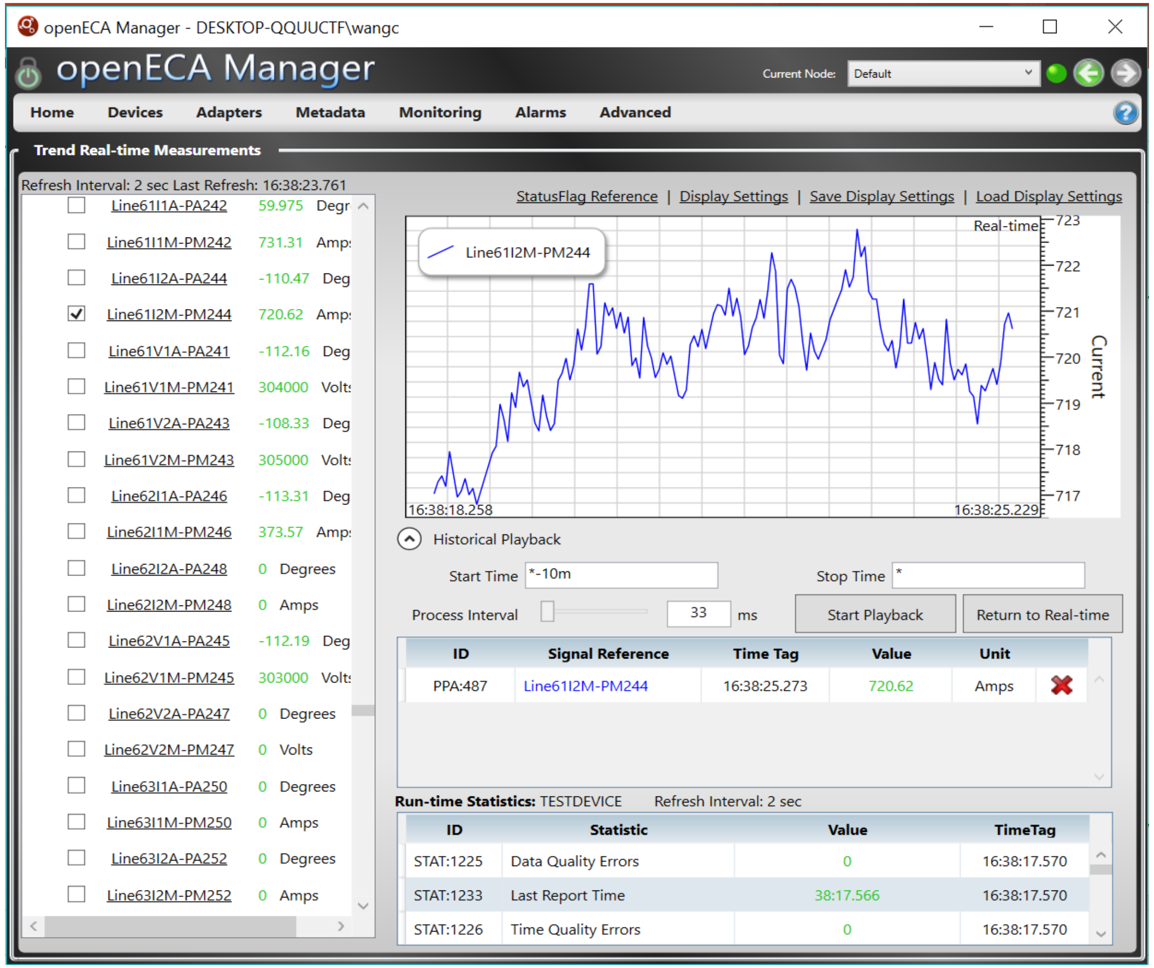
- Analytic Installation and Configuration Observations
  - The installation was convenient and quick. Cooperating with the configured openECA platform, the application can properly acquire and collect the real-time data streaming. The real-time data is also archived in openHistorian and accessible for the application to query. The system diagram can be demonstrated by the application with the provided XML file containing system topology information.
- Issues List
  - Real PMU measurements data quality need to be improved.
- Stability
  - The application can continuously and stably provide transmission lines parameters calculation results. No stability issues have been noted so far.
- Results display / presentation
  - Test based on both the IEEE 118-bus standard system and the Dominion Energy Virginia system are conducted. The ratio errors (correction factors) of the CTs and PTs are demonstrated by the table in the application.

Application Data Flow

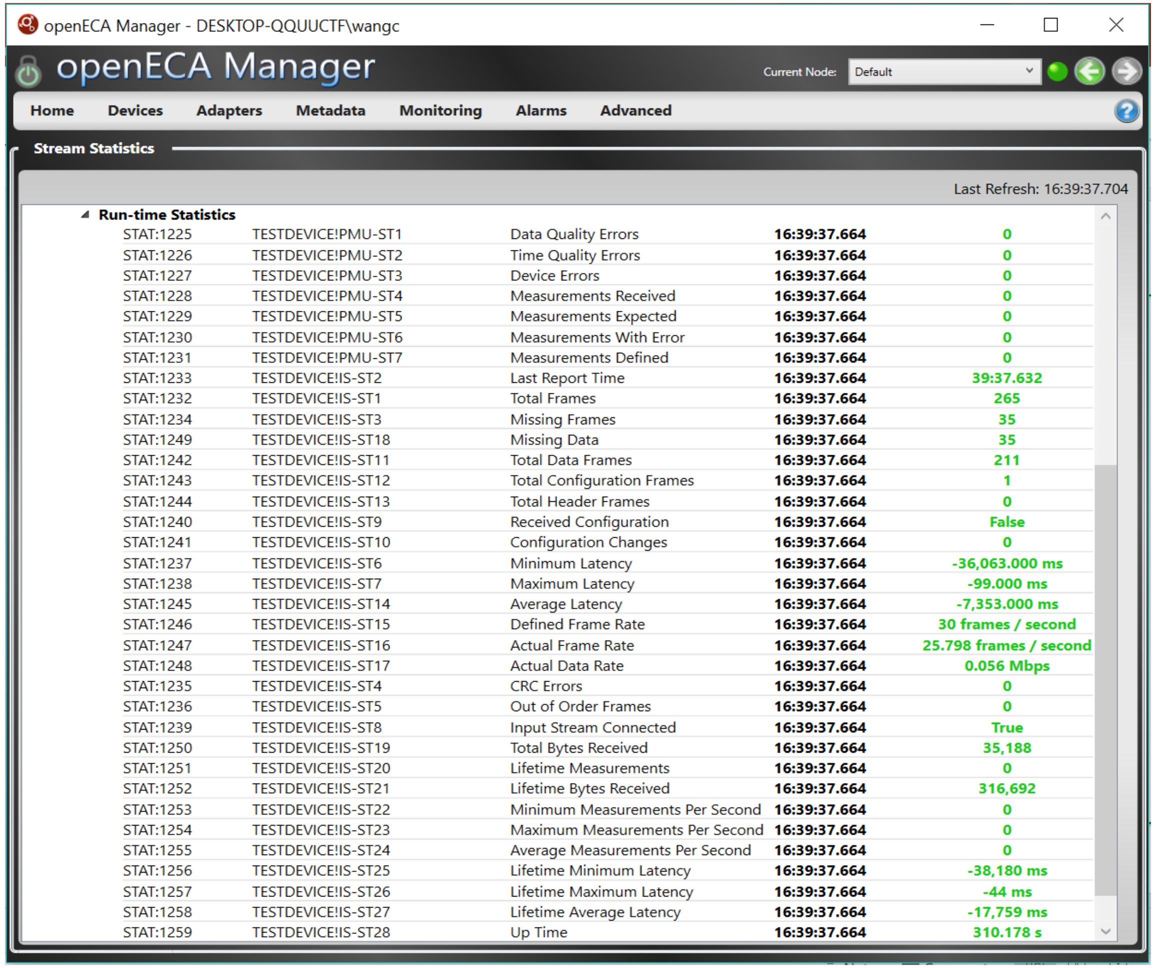


openECA Streaming Measurements

Real-time Streaming Measurements

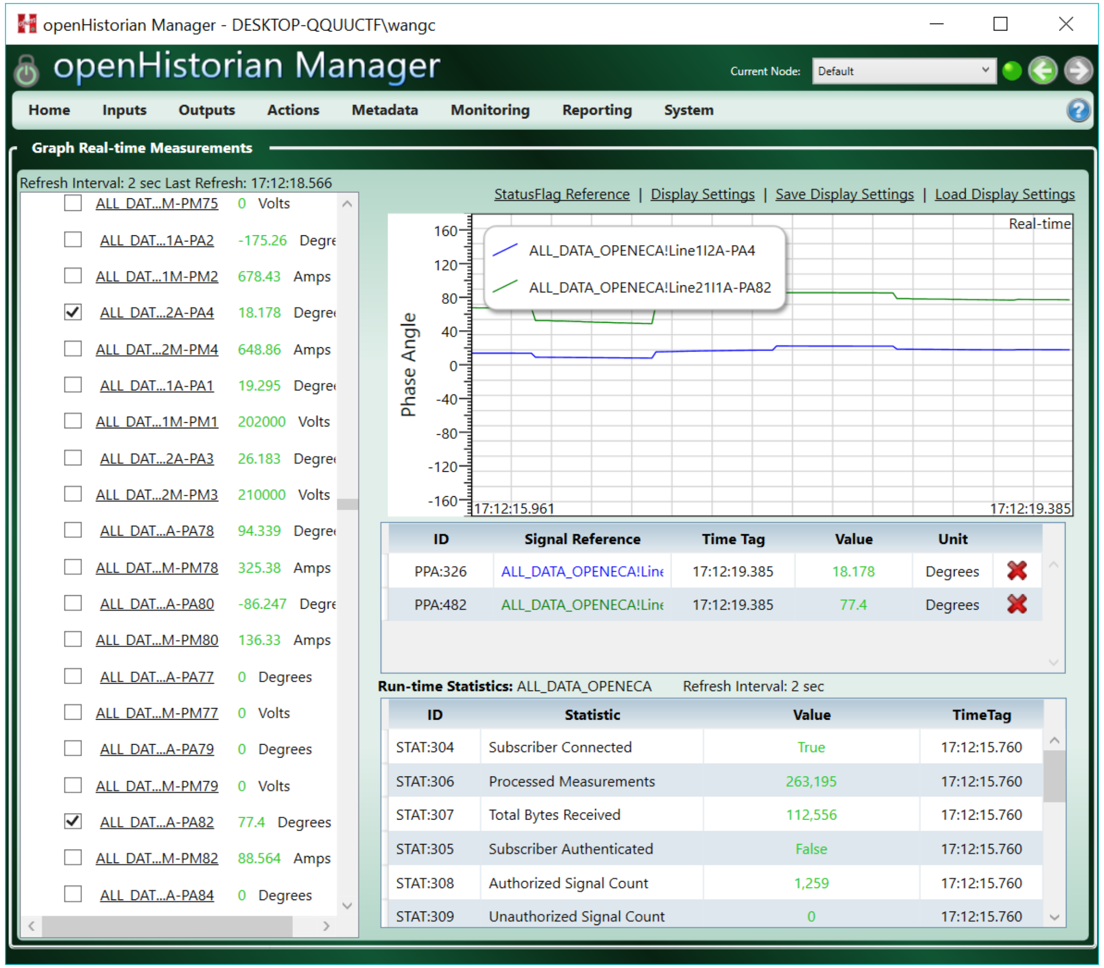


Run-time Statistics

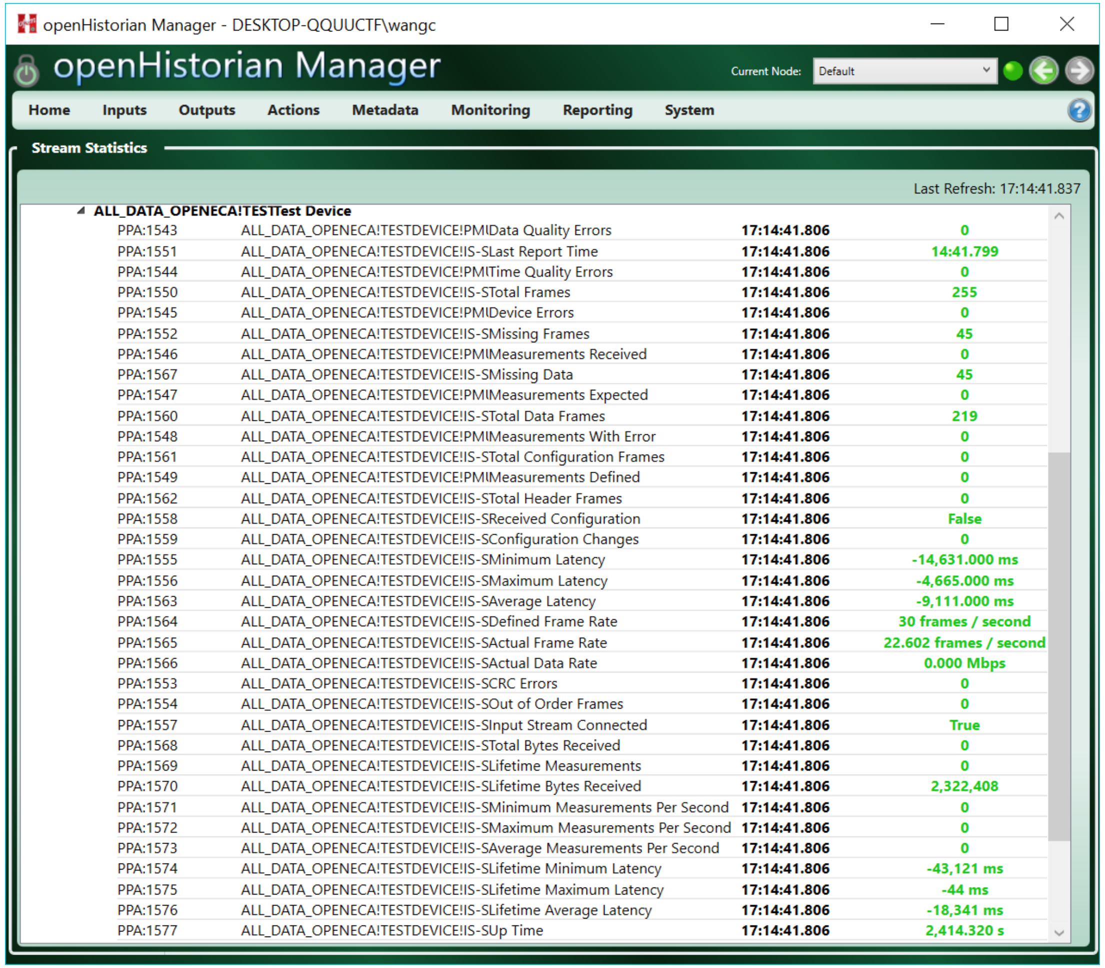


openHistorian Archiving Measurements

Real-time openECA Feeding



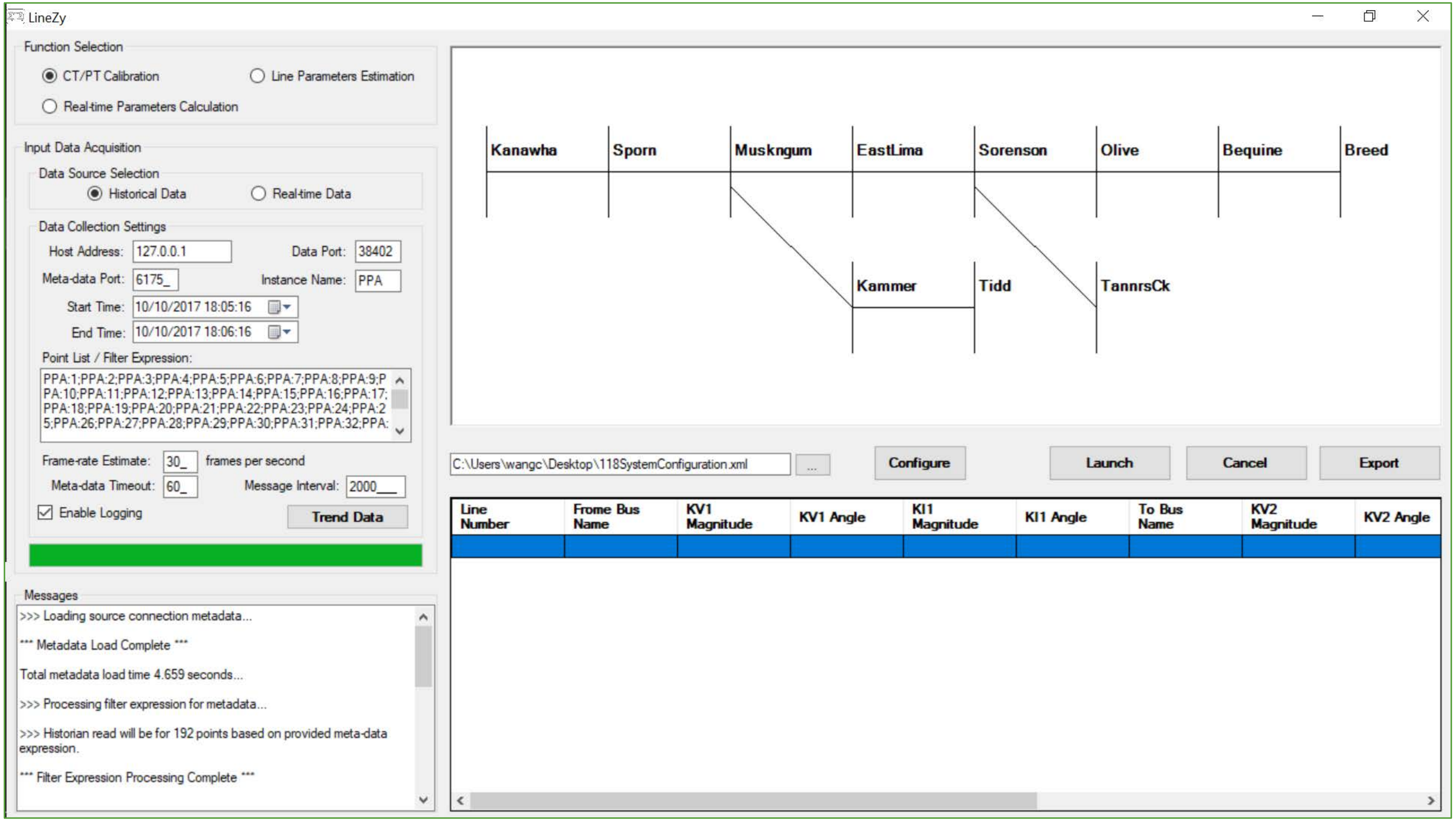
Run-time Statistics



IEEE 118-bus Standard System Topology Information


```xml
<?xml version="1.0" encoding="utf-8"?>
<StudyCase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" BaseMVA="100" BaseKV="345">
  <Branches>
    <Branch LineNumber="1" FromBusNumber="8" ToBusNumber="9" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="2" FromBusNumber="8" ToBusNumber="30" Resistance="0.00431" Reactance="0.050399998" Susceptance="0.256999996" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="3" FromBusNumber="9" ToBusNumber="10" Resistance="0.00258" Reactance="0.032200001" Susceptance="0.615000005" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="4" FromBusNumber="26" ToBusNumber="30" Resistance="0.00799" Reactance="0.086000005" Susceptance="0.454000003" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="5" FromBusNumber="30" ToBusNumber="38" Resistance="0.00464" Reactance="0.054" Susceptance="0.210999996" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="6" FromBusNumber="38" ToBusNumber="65" Resistance="0.00901" Reactance="0.098600002" Susceptance="0.522999992" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="7" FromBusNumber="63" ToBusNumber="64" Resistance="0.00172" Reactance="0.02" Susceptance="0.108000002" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="8" FromBusNumber="64" ToBusNumber="65" Resistance="0.00269" Reactance="0.0302" Susceptance="0.189999998" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="9" FromBusNumber="65" ToBusNumber="68" Resistance="0.00138" Reactance="0.015999995" Susceptance="0.319000001" HighVoltageLineFlag="true" ReferenceFlag="false" />
    <Branch LineNumber="10" FromBusNumber="68" ToBusNumber="81" Resistance="0.00175" Reactance="0.020200001" Susceptance="0.404000015" HighVoltageLineFlag="true" ReferenceFlag="true" />
    <Branch LineNumber="11" FromBusNumber="68" ToBusNumber="116" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="12" FromBusNumber="5" ToBusNumber="8" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="13" FromBusNumber="17" ToBusNumber="30" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="14" FromBusNumber="25" ToBusNumber="26" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="15" FromBusNumber="37" ToBusNumber="38" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="16" FromBusNumber="59" ToBusNumber="63" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="17" FromBusNumber="61" ToBusNumber="64" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="18" FromBusNumber="65" ToBusNumber="66" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="19" FromBusNumber="68" ToBusNumber="69" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="20" FromBusNumber="80" ToBusNumber="81" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="21" FromBusNumber="8" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="22" FromBusNumber="10" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="23" FromBusNumber="26" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
    <Branch LineNumber="24" FromBusNumber="65" ToBusNumber="-1" Resistance="0.00244" Reactance="0.030499999" Susceptance="0.580999961" HighVoltageLineFlag="false" ReferenceFlag="false" />
  </Branches>
  <Buses>
    <Bus BusNumber="8" BusName="Olive" ReferenceFlag="false" />
    <Bus BusNumber="9" BusName="Bequine" ReferenceFlag="false" />
    <Bus BusNumber="10" BusName="Breed" ReferenceFlag="false" />
    <Bus BusNumber="26" BusName="TannrsCk" ReferenceFlag="false" />
    <Bus BusNumber="30" BusName="Sorenson" ReferenceFlag="false" />
    <Bus BusNumber="38" BusName="EastLima" ReferenceFlag="false" />
    <Bus BusNumber="63" BusName="Tidd" ReferenceFlag="false" />
    <Bus BusNumber="64" BusName="Kammer" ReferenceFlag="false" />
    <Bus BusNumber="65" BusName="Muskngum" ReferenceFlag="false" />
    <Bus BusNumber="68" BusName="Sporn" ReferenceFlag="false" />
    <Bus BusNumber="81" BusName="Kanawha" ReferenceFlag="true" />
  </Buses>
</StudyCase>
```

Application Data Collection from openHistorian Demonstration



Application Results Demonstration

LineZy

Function Selection

- (•) CT/PT Calibration
- ( ) Line Parameters Estimation
- ( ) Real-time Parameters Calculation

Input Data Acquisition

Data Source Selection

- (•) Historical Data
- ( ) Real-time Data

Data Collection Settings

Host Address: 127.0.0.1
Data Port: 38402
Meta-data Port: 6175_
Instance Name: PPA
Start Time: 10/10/2017 18:05:16
End Time: 10/10/2017 18:06:16

Point List / Filter Expression:

PPA:1;PPA:2;PPA:3;PPA:4;PPA:5;PPA:6;PPA:7;PPA:8;PPA:9;PPA:10;PPA:11;PPA:12;PPA:13;PPA:14;PPA:15;PPA:16;PPA:17;PPA:18;PPA:19;PPA:20;PPA:21;PPA:22;PPA:23;PPA:24;PPA:25;PPA:26;PPA:27;PPA:28;PPA:29;PPA:30;PPA:31;PPA:32;PPA:

Frame-rate Estimate: 30_ frames per second
Meta-data Timeout: 60_
Message Interval: 2000__

☑ Enable Logging

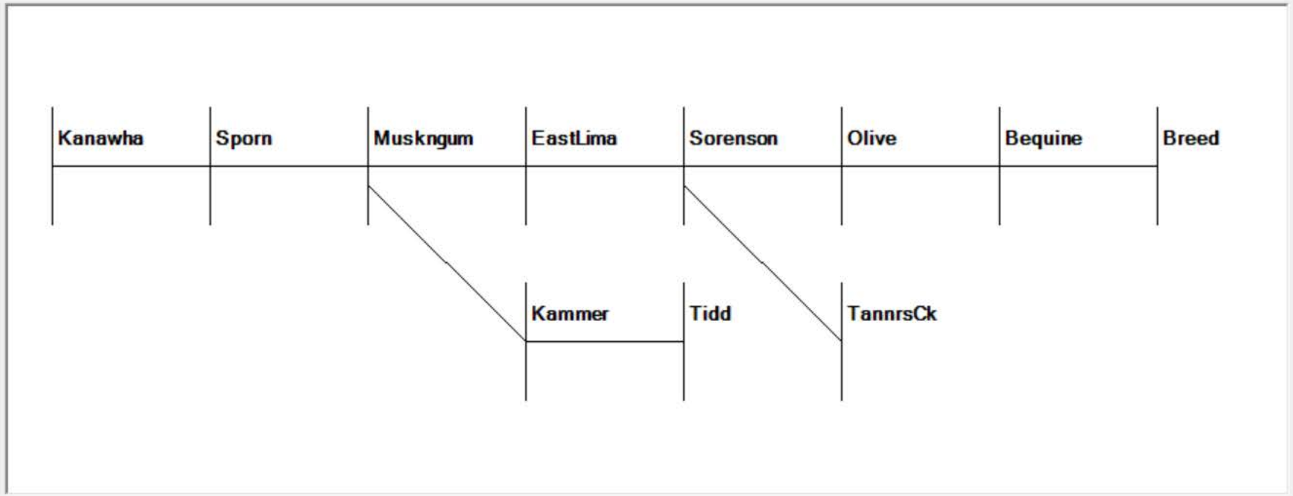
Trend Data

Messages

Meta-data points: 1342
Time-span covered: 60 seconds: 1 minute
Processed timestamps: 1,292
Expected points: 345,600 @ 30 samples per second
Received points: 248,064
Duplicate points: 0
Data completeness: 71.778%

>>> Historical Data Collection Completed!

>>> CT/PT Calibration Functionality Launched!



C:\Users\wangc\Desktop\118SystemConfiguration.xml

Configure | Launch | Cancel | Export

Line Number	Frome Bus Name	KV1 Magnitude	KV1 Angle	KI1 Magnitude	KI1 Angle	To Bus Name	KV2 Magnitude	KV2 Angle
10	Kanawha	1	0	1.0006	0.016	Sporn	0.9847	2.1373
9	Sporn	0.9994	1.1471	1.0064	-1.7162	Muskngum	0.9863	1.3061
6	Muskngum	0.9863	-2.4448	0.9841	-0.4055	EastLima	1.0025	2.3944
8	Muskngum	0.9961	1.556	0.9828	-1.513	Kammer	1.0193	0.0591
5	EastLima	0.9919	0.7017	1.0046	2.8735	Sorenson	1.0104	0.0228
2	Sorenson	1.0106	-0.5144	1.0085	0.4749	Olive	0.9569	-0.8579
4	Sorenson	0.9704	0.1915	1.0028	-3.334	TannrsCk	0.9728	-0.361
1	Olive	0.9901	-1.2493	1.0062	-0.7118	Bequine	0.9925	-1.5878
3	Bequine	1.0123	2.0978	0.978	2.5363	Breed	1.001	-0.915
7	Kammer	0.9787	-0.8976	0.8765	-2.6133	Tidd	0.994	1.1798

Application Results Based on Real-time Data Demonstration

LineZy

Function Selection

- (●) CT/PT Calibration
- ( ) Line Parameters Estimation
- ( ) Real-time Parameters Calculation

Input Data Acquisition

Data Source Selection

- ( ) Historical Data
- (●) Real-time Data

Data Collection Settings

Host Address: 127.0.0.1 Data Port: 38402

Meta-data Port: 6175_ Instance Name: PPA

Start Time: 10/10/2017 18:05:16

End Time: 10/10/2017 18:06:16

Point List / Filter Expression:

PPA:1;PPA:2;PPA:3;PPA:4;PPA:5;PPA:6;PPA:7;PPA:8;PPA:9;PPA:10;PPA:11;PPA:12;PPA:13;PPA:14;PPA:15;PPA:16;PPA:17;PPA:18;PPA:19;PPA:20;PPA:21;PPA:22;PPA:23;PPA:24;PPA:25;PPA:26;PPA:27;PPA:28;PPA:29;PPA:30;PPA:31;PPA:32;PPA:

Frame-rate Estimate: 30_ frames per second

Meta-data Timeout: 60_ Message Interval: 2000__

☑ Enable Logging Trend Data

Messages

>>> CT/PT Calibration Functionality Launched!

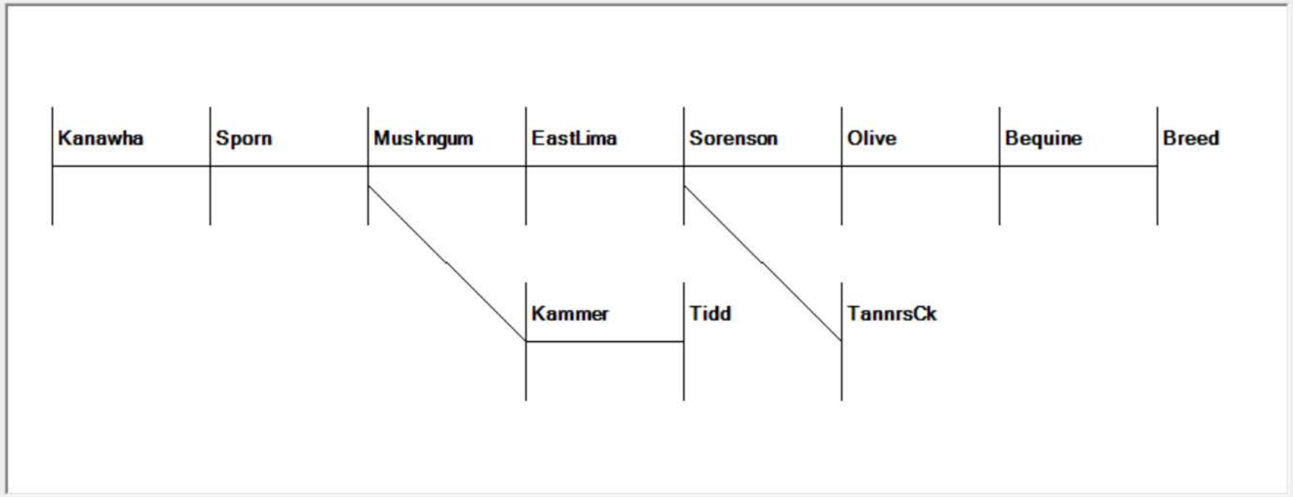
>>> Real-time Sreaming Data Collection Starts.

Start Time: 10/18/2017 1:20:38 PM

End Time: 10/18/2017 1:21:39 PM

>>> Real-time Sreaming Data Processing Completed!

>>> Starting CT / PT Calibration ...

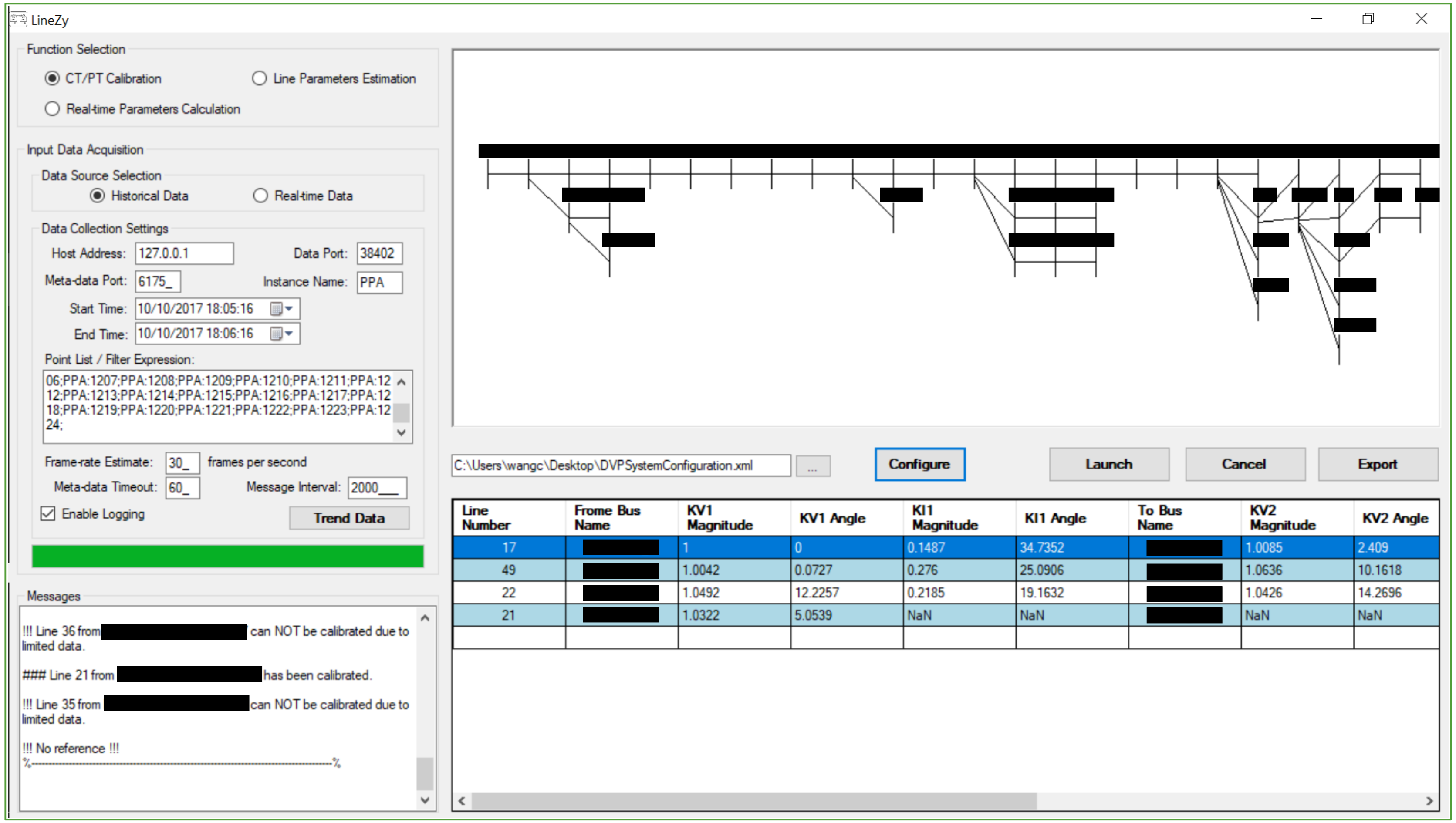


C:\Users\wangc\Desktop\118SystemConfiguration.xml ... Configure Launch Cancel Export

Line Number	Frome Bus Name	KV1 Magnitude	KV1 Angle	KI1 Magnitude	KI1 Angle	To Bus Name	KV2 Magnitude	KV2 Angle
10	Kanawha	1	0	1.0023	0.0003	Sporn	0.9869	2.114
9	Sporn	1.0017	1.1237	1.0092	-1.7405	Muskngum	0.9885	1.2821
6	Muskngum	0.9885	-2.4688	0.9835	-0.5183	EastLima	1.0052	2.3472
8	Muskngum	0.9984	1.532	0.9751	-1.4078	Kammer	1.0214	0.0327
5	EastLima	0.9947	0.6546	1.0068	2.8358	Sorenson	1.0144	-0.0505
2	Sorenson	1.0144	-0.5877	1.0105	0.4291	Olive	0.9599	-0.9189
4	Sorenson	0.974	0.1181	1.0055	-3.4043	TannrsCk	0.9751	-0.4336
1	Olive	0.9931	-1.3103	1.0075	-0.7551	Bequine	0.997	-1.6602
3	Bequine	1.0172	2.0254	0.9967	2.6956	Breed	1.0199	-0.85
7	Kammer	0.9808	-0.924	0.8228	-2.0569	Tidd	0.9959	1.1542

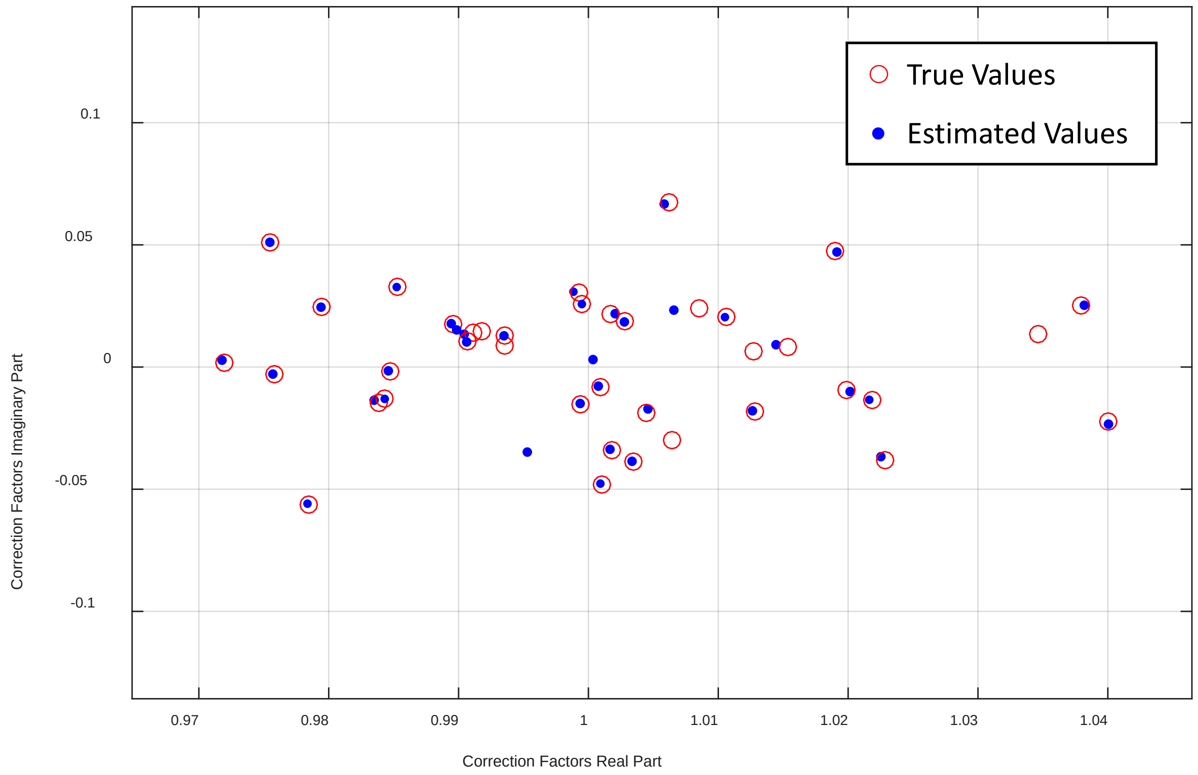
Application Results Based on Field Data Demonstration

- Dominion Energy Virginia Field Data



Demonstration Results

- Numerical Results for IEEE 118-bus System



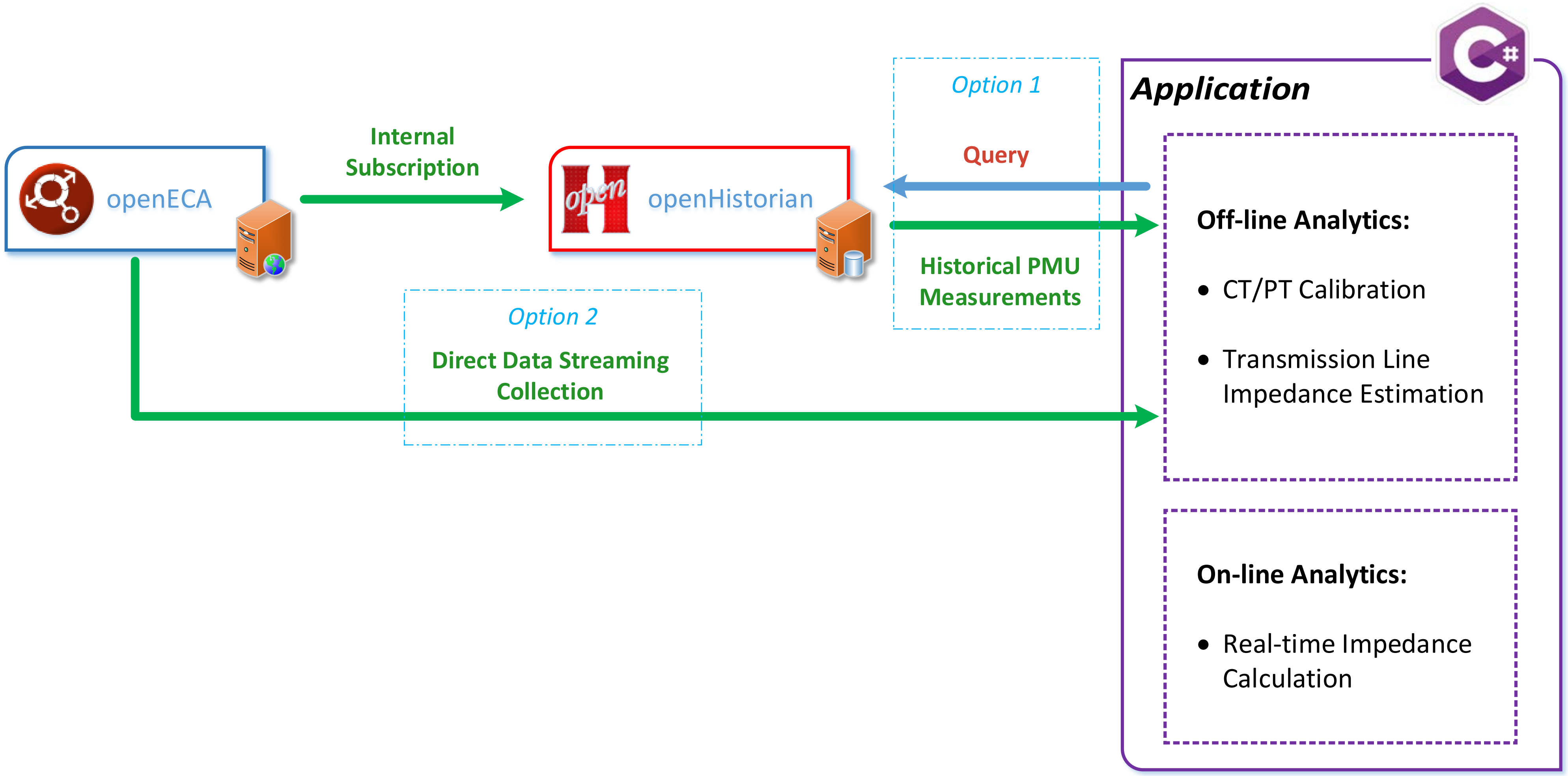
Deployment Results Report

Transmission Line Parameters Estimation

Virginia Tech Transmission Line Parameters Estimation Observations

- Analytic Installation and Configuration Observations
  - The installation was convenient and quick. Cooperating with the configured openECA platform, the application can properly acquire and collect the real-time data streaming. The real-time data is also archived in openHistorian and accessible for the application to query. The system diagram can be demonstrated by the application with the provided XML file containing system topology information.
- Issues List
  - Real PMU measurements data quality need to be improved.
- Stability
  - The application can continuously and stably provide transmission lines parameters calculation results. No stability issues have been noted so far.
- Results display / presentation
  - Test based on both the IEEE 118-bus standard system and the Dominion Energy Virginia system are conducted. The ratio errors (correction factors) of the CTs and PTs and the concerned transmission lines' parameters are demonstrated by the table in the application.

Application Data Flow



Application Results Demonstration

LineZy

Function Selection

- ○ CT/PT Calibration
- ◉ Line Parameters Estimation
- ○ Real-time Parameters Calculation

Input Data Acquisition

Data Source Selection

- ◉ Historical Data
- ○ Real-time Data

Data Collection Settings

Host Address: 127.0.0.1 Data Port: 38402

Meta-data Port: 6175_ Instance Name: PPA

Start Time: 10/10/2017 18:05:16

End Time: 10/10/2017 18:06:16

Point List / Filter Expression:

0;PPA:171;PPA:172;PPA:173;PPA:174;PPA:175;PPA:176;PPA:177;PPA:178;PPA:179;PPA:180;PPA:181;PPA:182;PPA:183;PPA:184;PPA:185;PPA:186;PPA:187;PPA:188;PPA:189;PPA:190;PPA:191;PPA:192;

Frame-rate Estimate: 30_ frames per second

Meta-data Timeout: 60_ Message Interval: 2000__

☑ Enable Logging

Trend Data

Messages

Line 2 from bus 30 to bus 8 has been estimated.

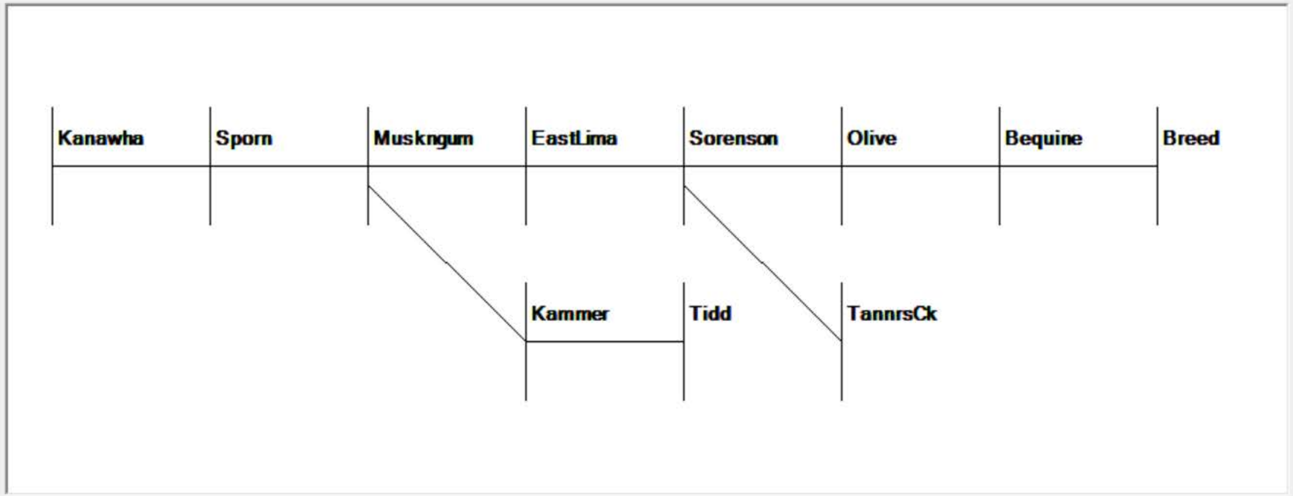
Line 4 from bus 30 to bus 26 has been estimated.

Line 1 from bus 8 to bus 9 has been estimated.

Line 3 from bus 9 to bus 10 has been estimated.

Line 7 from bus 64 to bus 63 has been estimated.

%--%

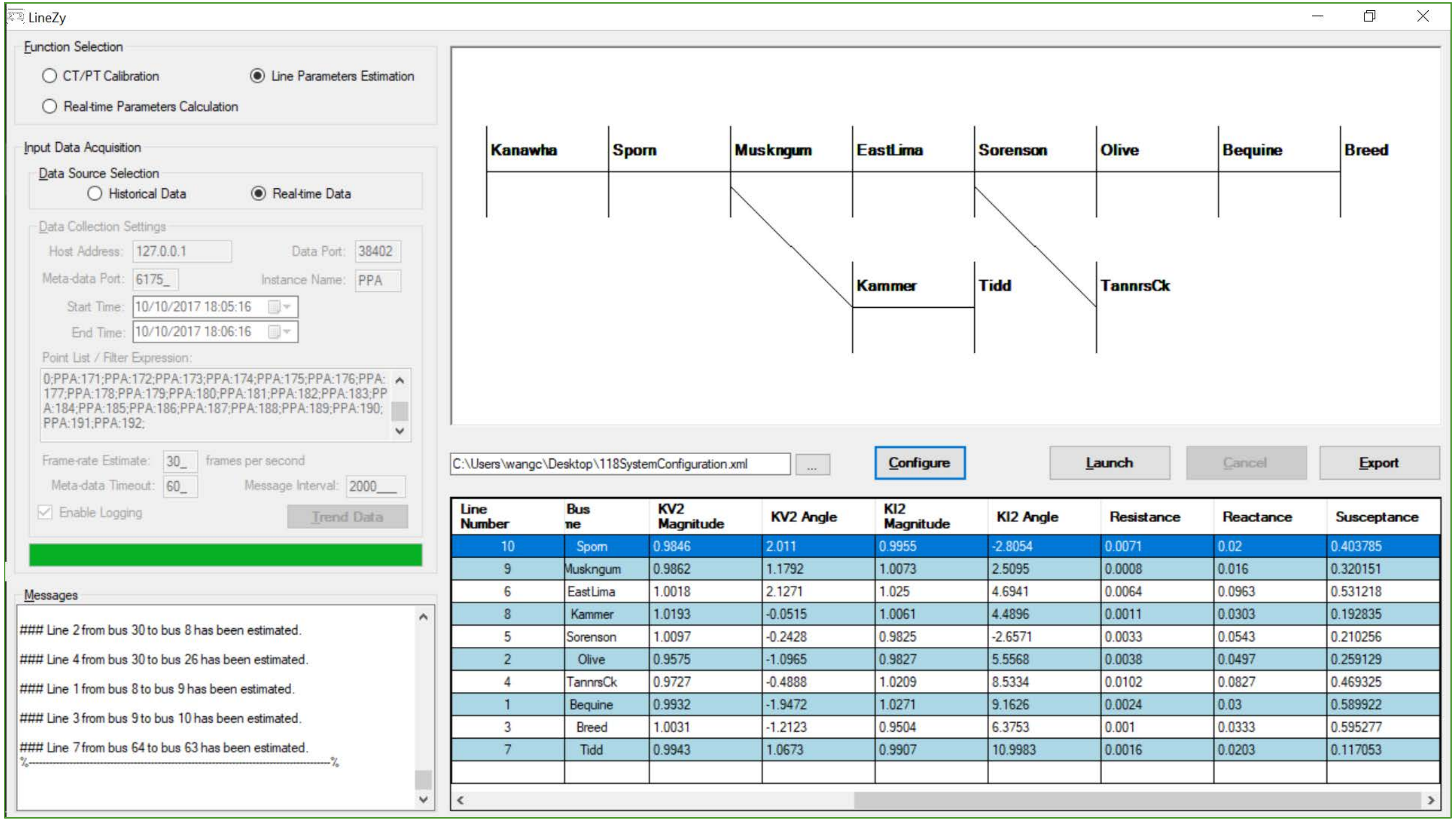


C:\Users\wangc\Desktop\118SystemConfiguration.xml ...

Configure Launch Cancel Export

Line Number	Bus ne	KV2 Magnitude	KV2 Angle	KI2 Magnitude	KI2 Angle	Resistance	Reactance	Susceptance
10	Sporn	0.9846	2.0062	0.9951	-2.8083	0.0074	0.02	0.403767
9	Muskngum	0.9862	1.1743	1.009	0.8112	0.0013	0.0159	0.320935
6	EastLima	1.0017	2.1273	1.0283	7.1306	0.0024	0.0961	0.531118
8	Kammer	1.0193	-0.0546	1.0056	5.628	0.0013	0.0302	0.192989
5	Sorenson	1.0096	-0.2435	0.9874	1.7713	0.0018	0.0541	0.21095
2	Olive	0.9578	-1.1159	0.9777	2.6387	0.0008	0.0502	0.258881
4	TannrsCk	0.9726	-0.4923	1.0207	8.5474	0.0102	0.0827	0.469563
1	Bequine	0.9933	-1.9629	1.0264	9.1641	0.0023	0.0301	0.589271
3	Breed	1.0019	-1.2287	0.9504	6.3861	0.0014	0.0333	0.595884
7	Tidd	0.9943	1.0662	0.9685	11.0617	0.0019	0.0207	0.115562

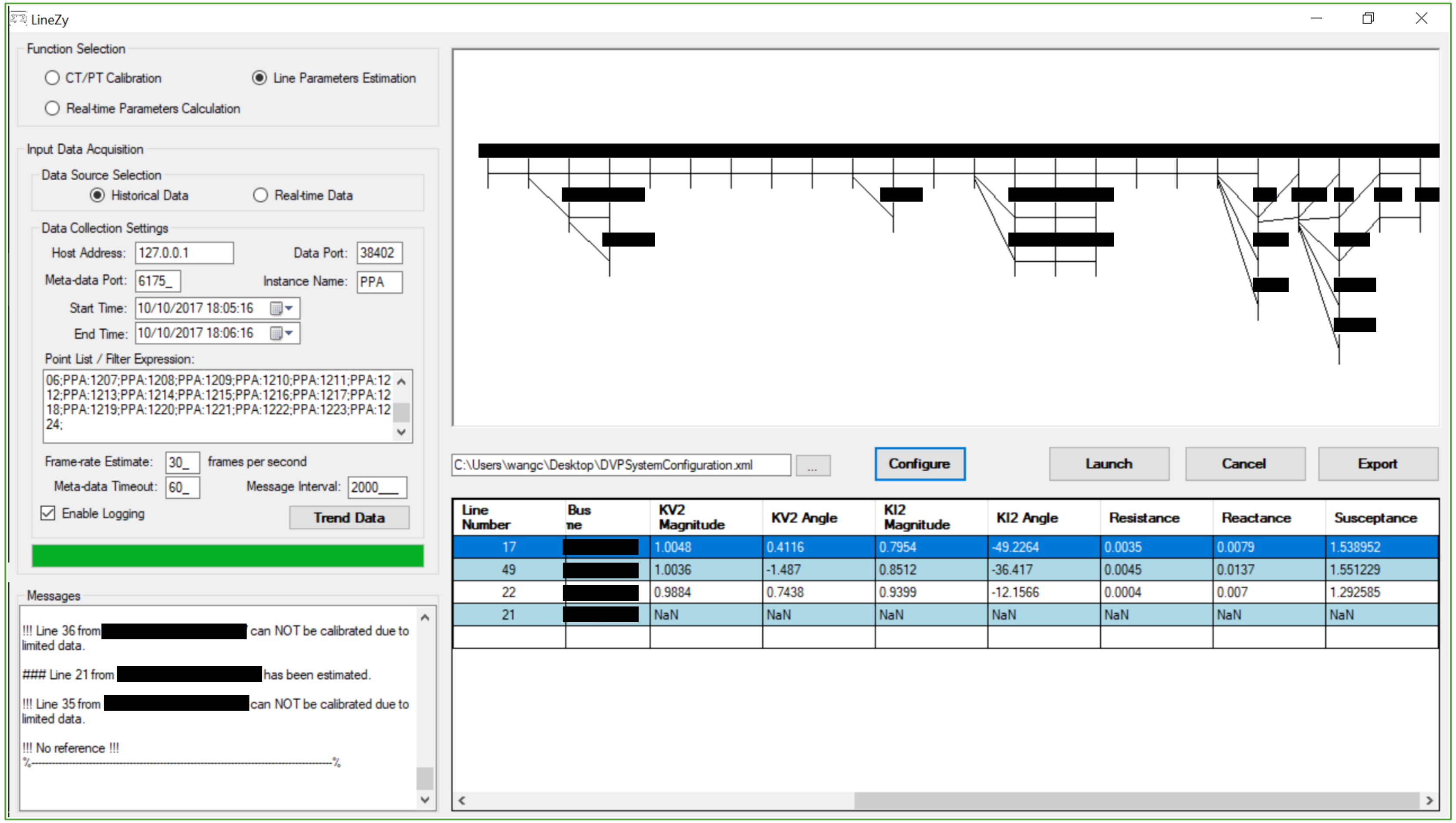
Application Results Based on Real-time Data Demonstration



Line Number	Bus ne	KV2 Magnitude	KV2 Angle	KI2 Magnitude	KI2 Angle	Resistance	Reactance	Susceptance
10	Sporn	0.9846	2.011	0.9955	-2.8054	0.0071	0.02	0.403785
9	Muskngum	0.9862	1.1792	1.0073	2.5095	0.0008	0.016	0.320151
6	EastLima	1.0018	2.1271	1.025	4.6941	0.0064	0.0963	0.531218
8	Kammer	1.0193	-0.0515	1.0061	4.4896	0.0011	0.0303	0.192835
5	Sorenson	1.0097	-0.2428	0.9825	-2.6571	0.0033	0.0543	0.210256
2	Olive	0.9575	-1.0965	0.9827	5.5568	0.0038	0.0497	0.259129
4	TannrsCk	0.9727	-0.4888	1.0209	8.5334	0.0102	0.0827	0.469325
1	Bequine	0.9932	-1.9472	1.0271	9.1626	0.0024	0.03	0.589922
3	Breed	1.0031	-1.2123	0.9504	6.3753	0.001	0.0333	0.595277
7	Tidd	0.9943	1.0673	0.9907	10.9983	0.0016	0.0203	0.117053

Application Results Based on Field Data Demonstration

- Dominion Energy Virginia Field Data



Demonstration Results

- Numerical Results for IEEE 118-bus System

From Bus	KV1 Error	KI1 Error	To Bus	KV2 Error	KI2 Error	R		X		y	
						Error	Error Rate (%)	Error	Error Rate (%)	Error	Error Rate (%)
81	N/A	N/A	68	-4.3574e-07 -2.1523e-06i	8.9373e-06 +2.3307e-04i	2.73E-06	0.1558	4.02E-07	0.0020	-5.30E-05	-0.0131
68	-6.0682e-07 -1.5022e-06i	-3.2204e-03 -2.2736e-02i	65	1.7384e-06 -1.4741e-06i	-4.4428e-03 -2.1793e-02i	3.55E-06	0.2570	6.85E-05	0.4278	-0.00144	-0.4508
65	2.3563e-06 -2.0403e-06i	-7.4500e-03 -3.2999e-02i	38	-3.1054e-05 +9.1608e-05i	-9.2774e-03 -3.5116e-02i	9.76E-05	1.0833	7.47E-04	0.7577	-0.00287	-0.5485
65	1.4714e-06 -1.7005e-06i	7.7091e-03 -2.0367e-02i	64	4.8234e-06 -3.0588e-06i	8.3469e-03 -2.3036e-02i	2.52E-05	0.9365	-2.40E-04	-0.7935	0.003803	2.0016
38	-3.5161e-05 +8.9137e-05i	-9.6383e-03 -3.4871e-02i	30	-2.9807e-05 +8.6343e-05i	-8.6824e-03 -3.5108e-02i	4.79E-05	1.0327	4.25E-04	0.7869	-0.00161	-0.7634
64	4.9404e-06 -3.3658e-06i	8.9270e-03 -2.2739e-02i	63	-9.6428e-06 +8.1608e-05i	1.5750e-02 -3.8739e-02i	-1.99E-04	-11.5762	-2.71E-04	-1.3573	0.017419	16.1283
30	-3.1998e-05 +8.0769e-05i	8.7203e-03 -5.1042e-02i	8	-6.7690e-05 +1.0684e-04i	1.1632e-02 -5.0322e-02i	-9.40E-05	-2.1806	-5.01E-04	-0.9944	0.002828	1.1003
30	-2.8248e-05 +8.2693e-05i	-3.6838e-02 -1.0731e-02i	26	-1.4855e-04 +1.6522e-05i	-3.6589e-02 -8.7539e-03i	3.93E-04	4.9192	3.23E-03	3.7564	-0.01661	-3.6576
8	-6.5116e-05 +1.1153e-04i	-1.9191e-02 +1.1096e-02i	9	-5.5538e-05 +9.7252e-05i	-1.8980e-02 +1.0538e-02i	5.93E-05	2.4303	6.09E-04	1.9954	-0.01073	-1.8460
9	-5.5518e-05 +9.9123e-05i	-1.9866e-02 +9.7138e-03i	10	-4.4170e-05 +1.0676e-04i	-2.0304e-02 +9.3947e-03i	4.02E-05	1.5574	6.51E-04	2.0229	-0.01191	-1.9363

Virginia Tech/Dominion Energy Demonstration Results Report

PMU Synchroscope Analytic Demonstration Results

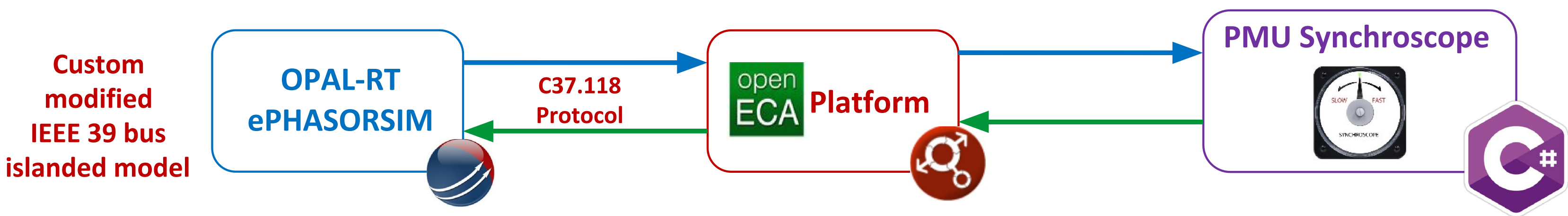
Tapas Kumar Barik

October 20, 2017

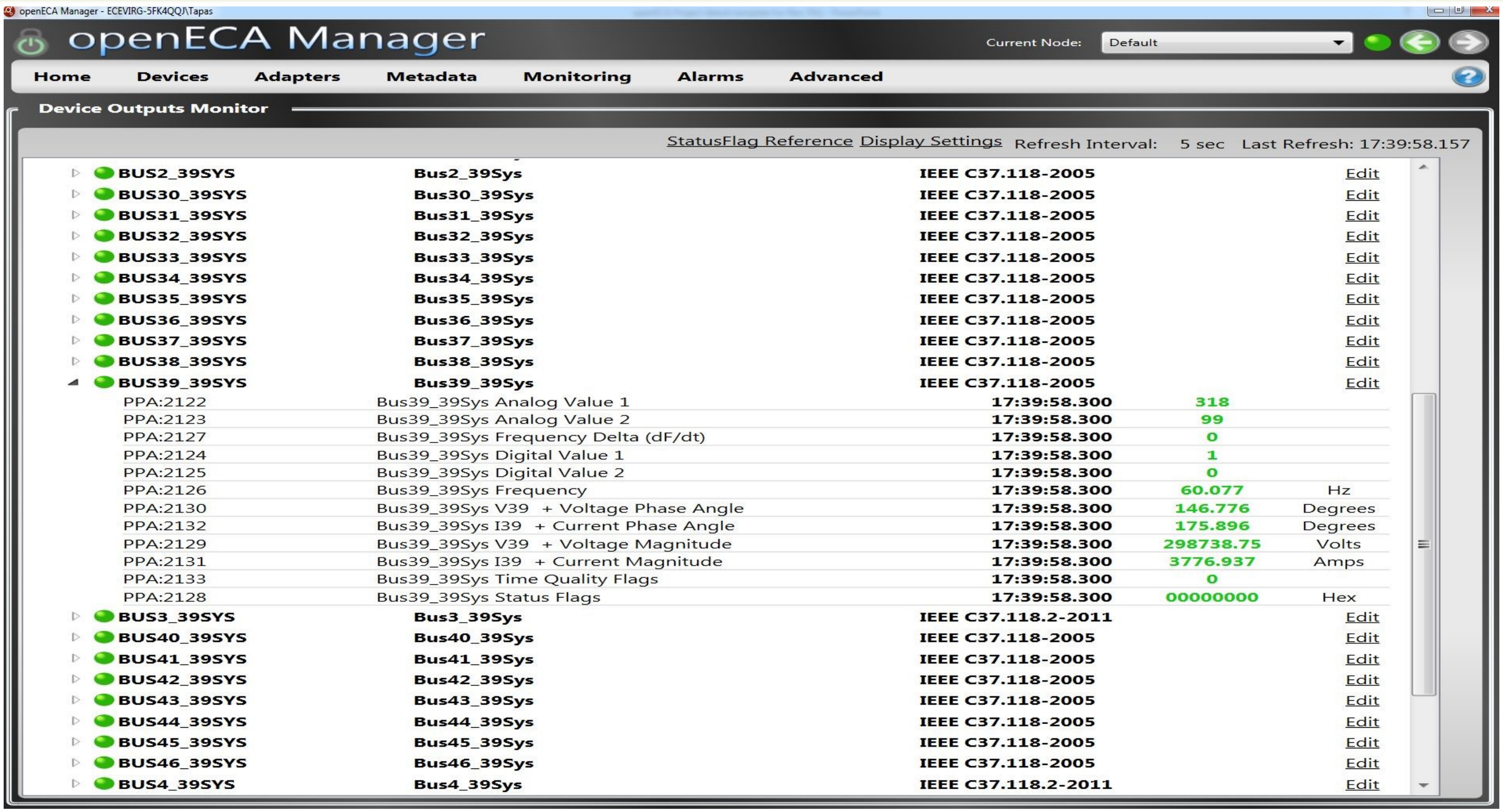
DOE FOA 970
DE-OE0000778

Installation Overview

- The final executable PMU Synchroscope.exe file (built using openECA) is installed on the client/user system.
- For testing and demonstration purpose, real time simulated PMU data are streamed in using the OPAL-RT ePHASORSIM simulator procured by Dominion Energy.
- A custom modified two islanded model for the 39 bus IEEE system is generated and executed on the RT-LAB target simulator and the simulated synchrophasor data is streamed over the C37.118 protocol to the openECA instance on the client's location.
- The PMU Synchroscope installed application receives and generates signals from and to the openECA platform as described in the following slides.



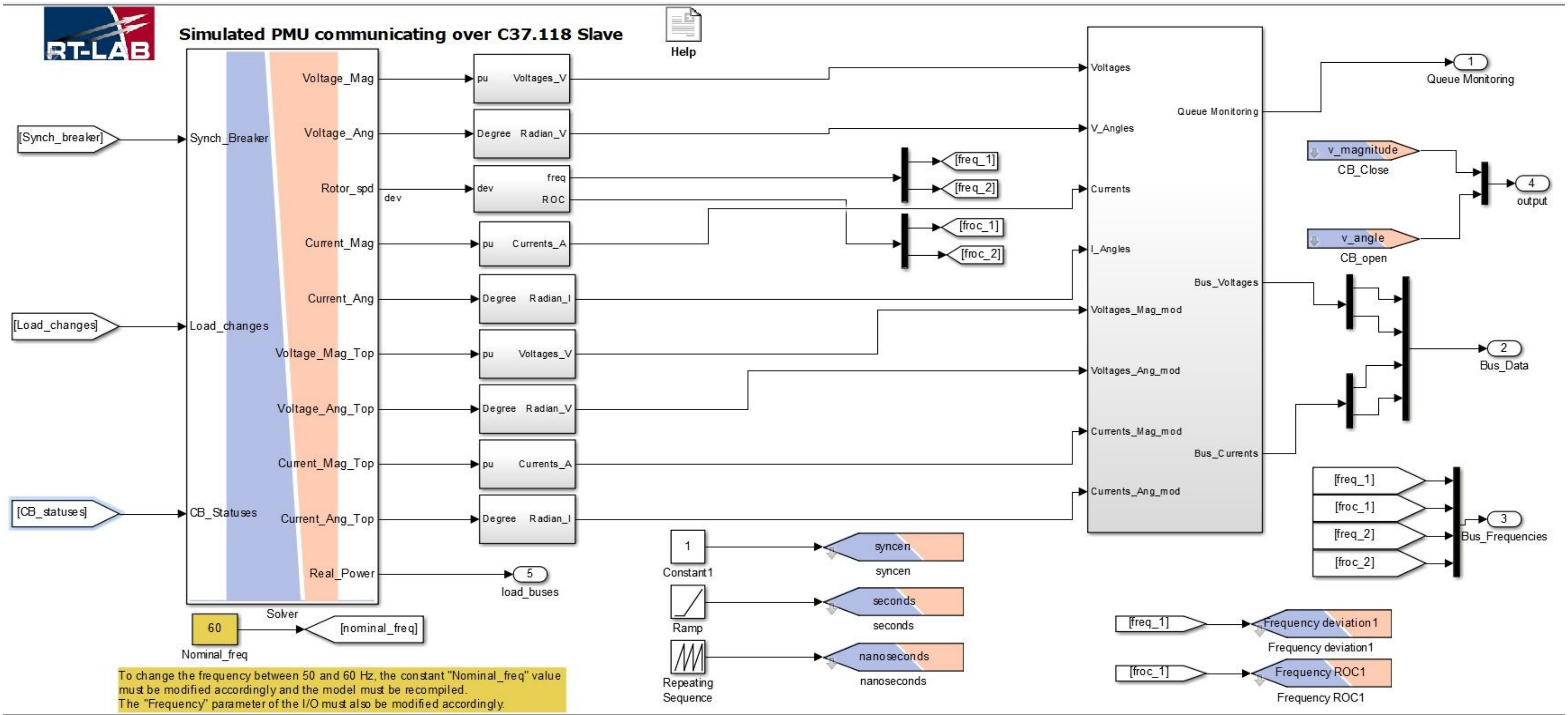
Data Stream Connection to openECA Platform



PMU Synchroscope Execution

- Once the installed application is executed as an administrator, an Input screen pop up window appears.
- The user can select the two remote buses on each island to be synchronized using the signal reference as depicted on the openECA platform.
- For the custom modified model Bus 39 (Island-1) and Bus 46(Island-2) are to be synchronized and thus respective fields are filled .
- Synchronizing breaker close and open commands are sent in form of voltages (0 or 5 Volts)to predefined measurement channels (required fields in the input screen) which are configured into an output stream of openECA and ultimately sent back to the simulator via C37.118 protocol.
- Once all the signal references are filled up, the framework is created by clicking ENTER button and data starts streaming from openECA platform into the Synchroscope analytic.
- Tolerance limits (which can be user defined as opposed to physical synchroscope hardware) are also provided and finally the Synchroscope analytic is launched

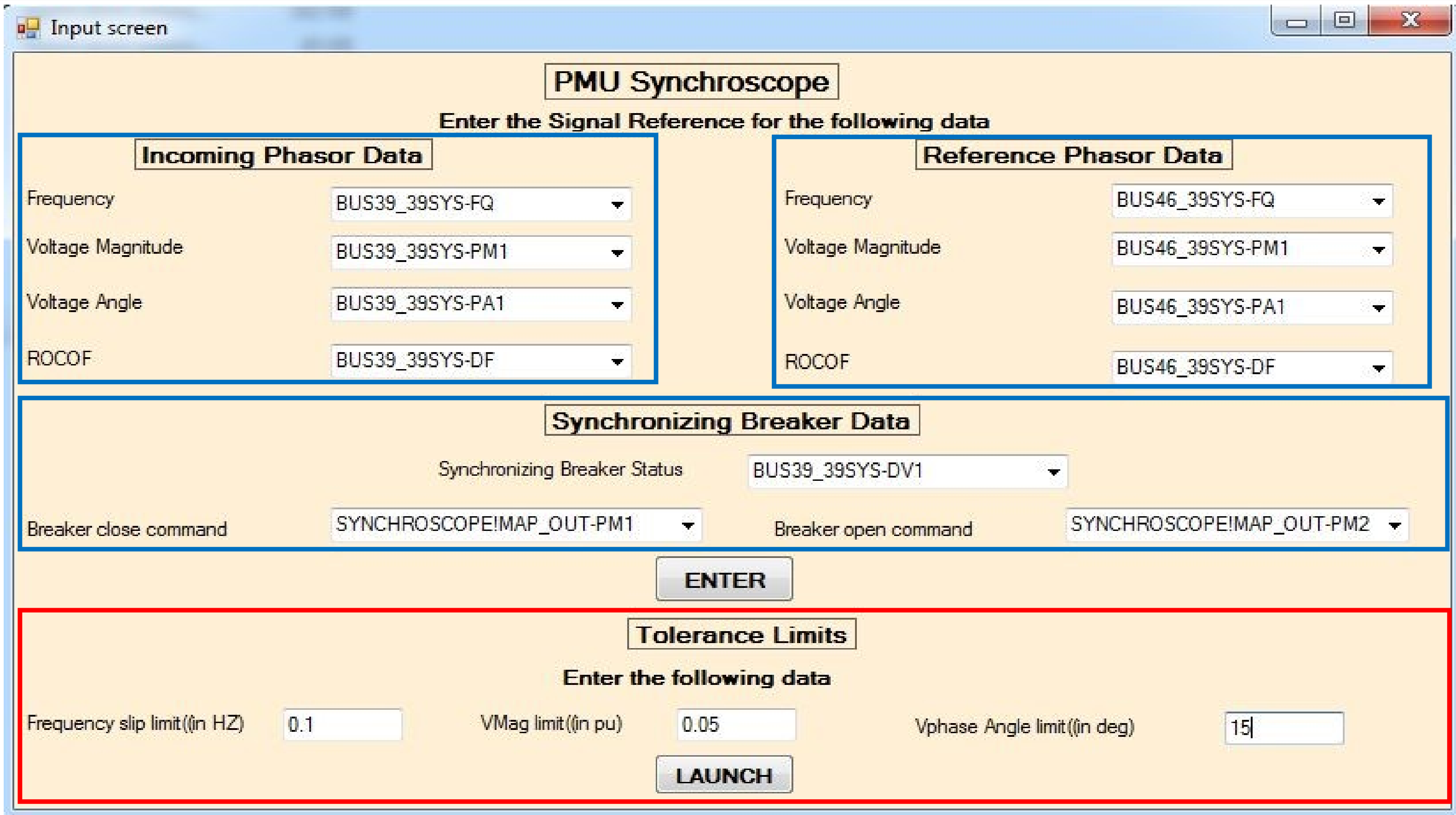
RT-LAB Simulator Custom Modified IEEE-39 Bus Model



DOE FOA 970

DE-OE-778

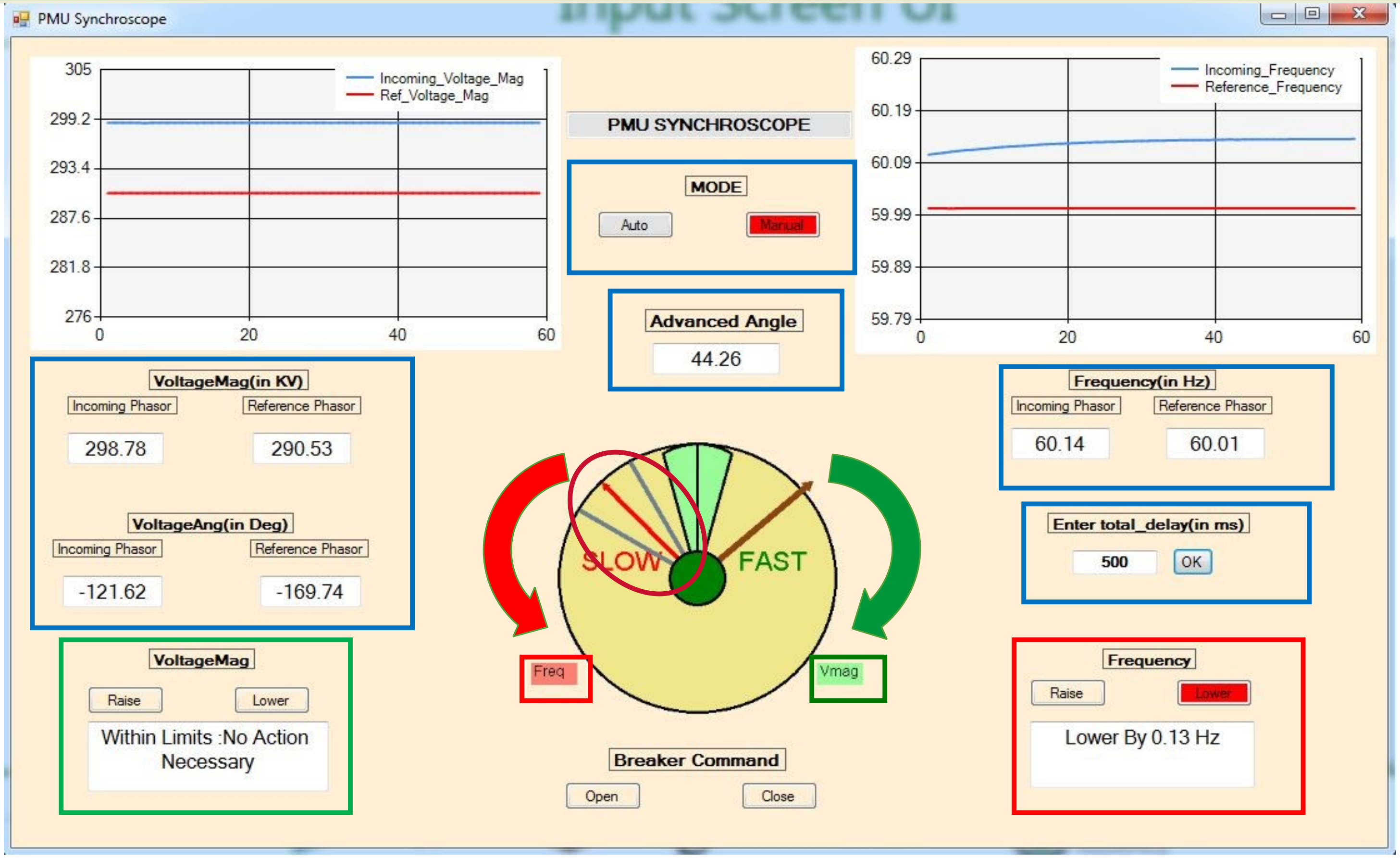
Input Screen UI



PMU Synchroscope Execution

- Depending upon whether the incoming phasor has a frequency greater or smaller than the reference phasor, the rotating hand will move clockwise or counter-clockwise respectively as shown in the figure.
- If the difference in values of frequency or voltage magnitude between both phasors are beyond limit, the checkboxes would show up in red (green if within limits).
- The user can provide the downstream delay (predefined upon study of system which includes communication delay and breaker closing time delay) according to which an advanced angle(function of frequency slip and delays) is calculated and a modified tolerance angle window is generated.
- Once all the necessary requirements are fulfilled, the user can initiate the breaker close command when the rotating phasor hand lies within this modified tolerance angle window.
- AUTO mode is also provided for the ease of the user which senses the advanced angle at which the breaker close command should be initiated such that the actual final breaker closing is carried out with 0 degree phase angle difference(12 o'clock position)

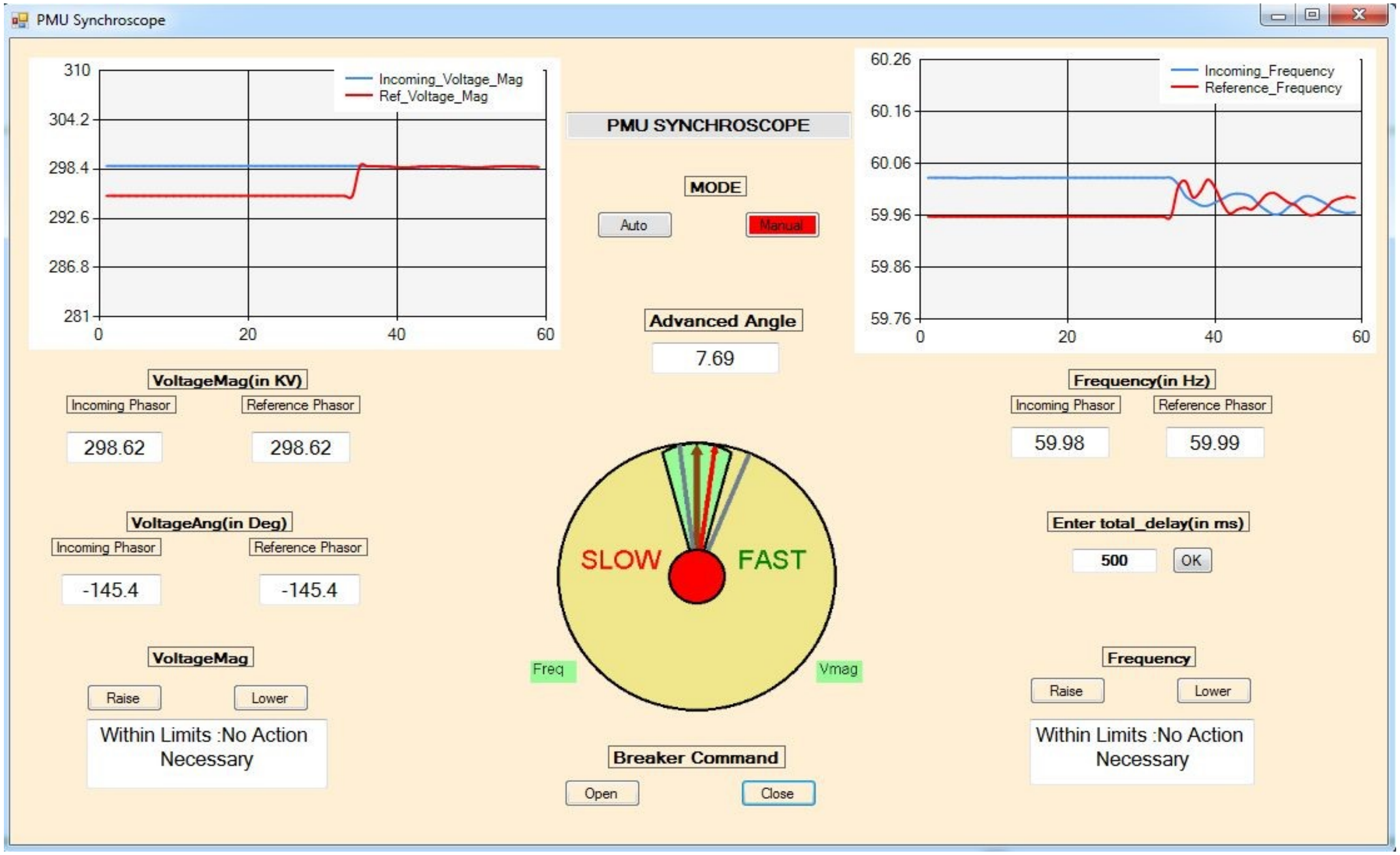
PMU Synchroscope UI



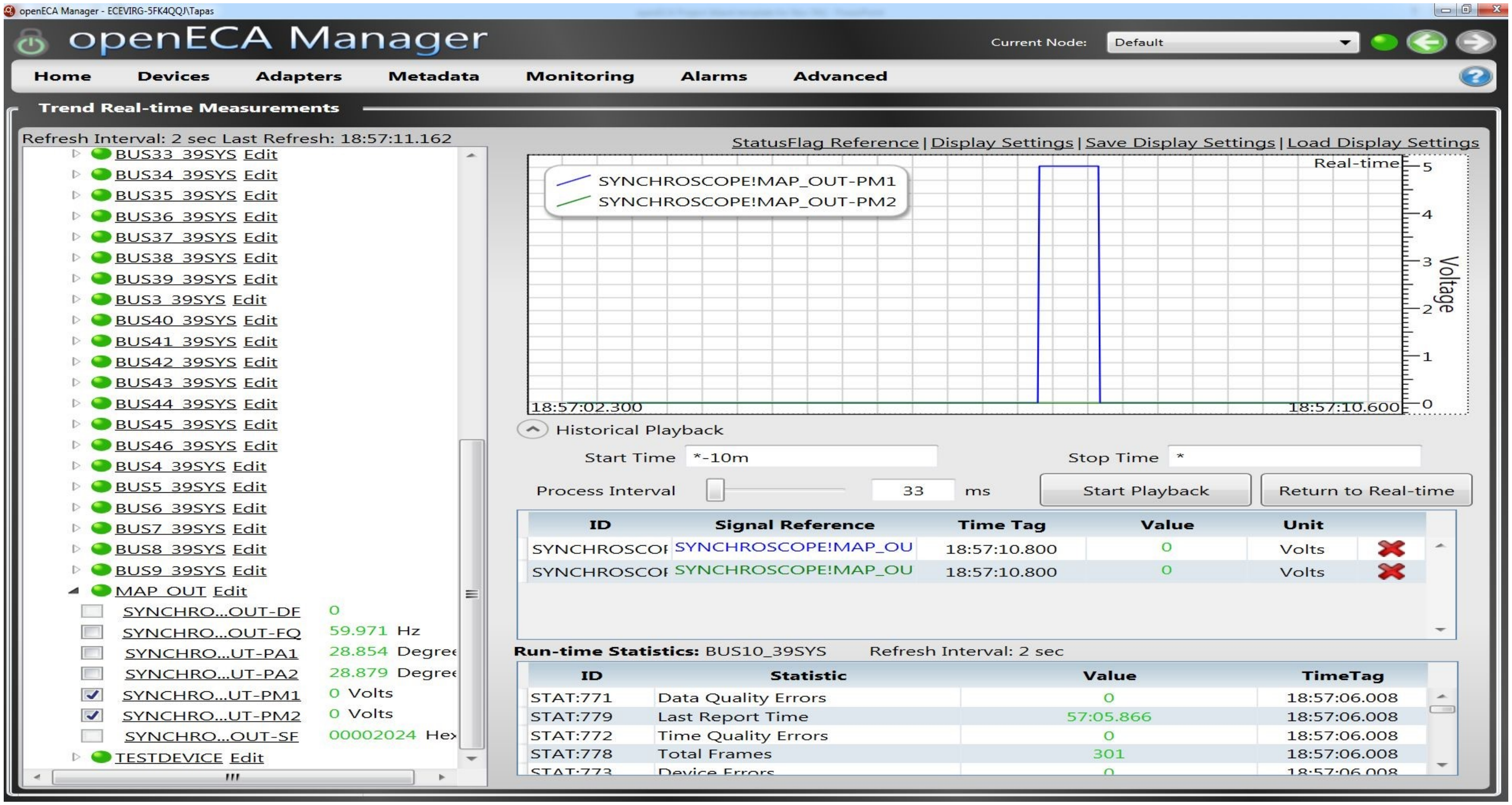
Demonstration Results

- The application was executed with variable downstream delays and variable frequencies and successful and satisfactory response of synchronization between the two islands was observed.
- The Synchronizing breaker status can be viewed as the circle in the middle of the synchroscope (Green-Open ,Red-Close)
- The transients involved after the synchronization were minimal which is a indication of proper and successful synchronization.
- The Synchronization breaker close and open signals generated using the analytic can be visualized on the openECA platform.
- These signals for the demonstration purpose are sent back to the OPAL-RT simulator as output streams from openECA via C37.118 protocol.

Demonstration Results



Circuit Breaker Close Signal on openECA



DOE FOA 970
DE-OE-778

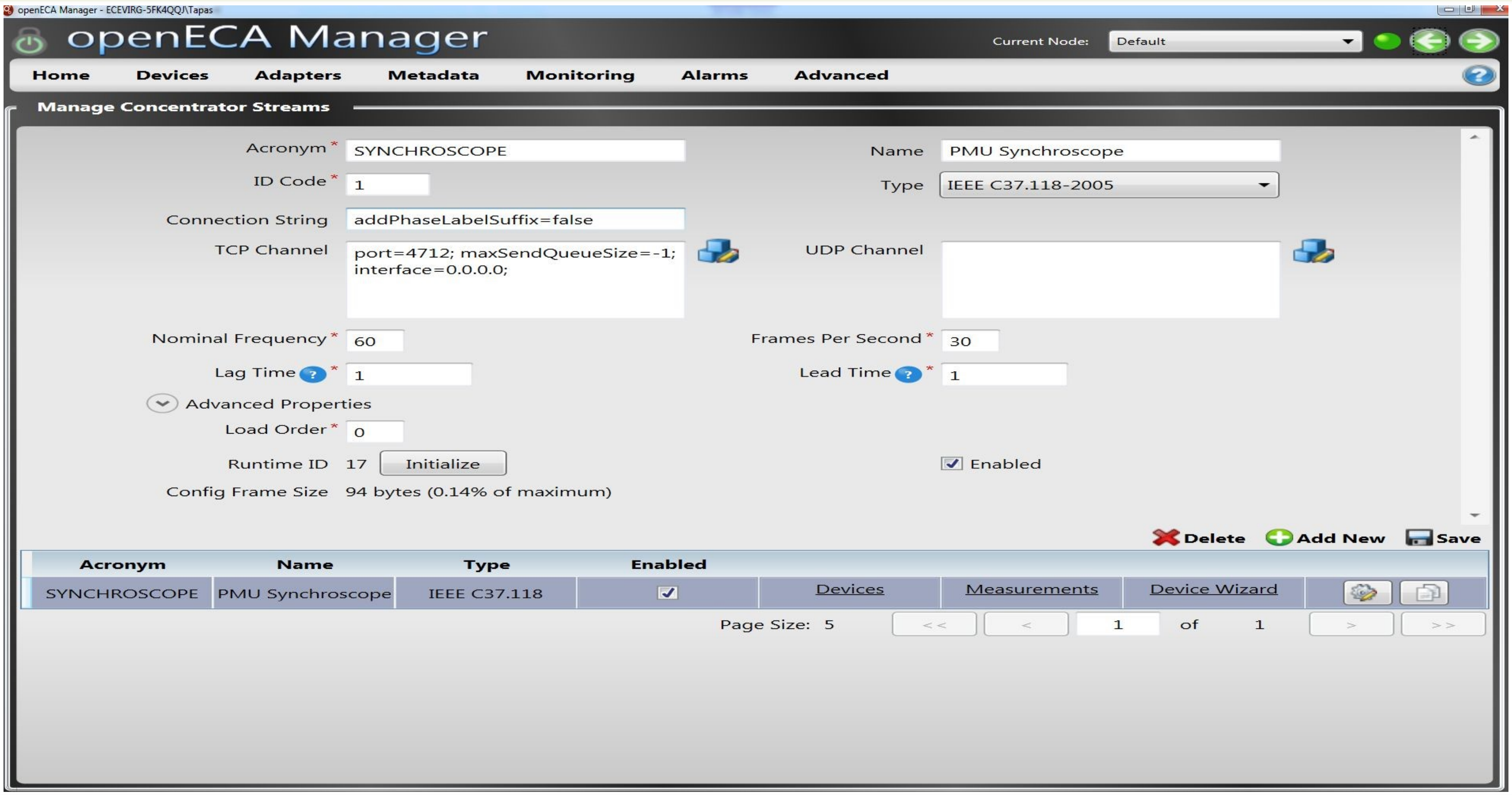
Dominion Energy

VirginiaTech

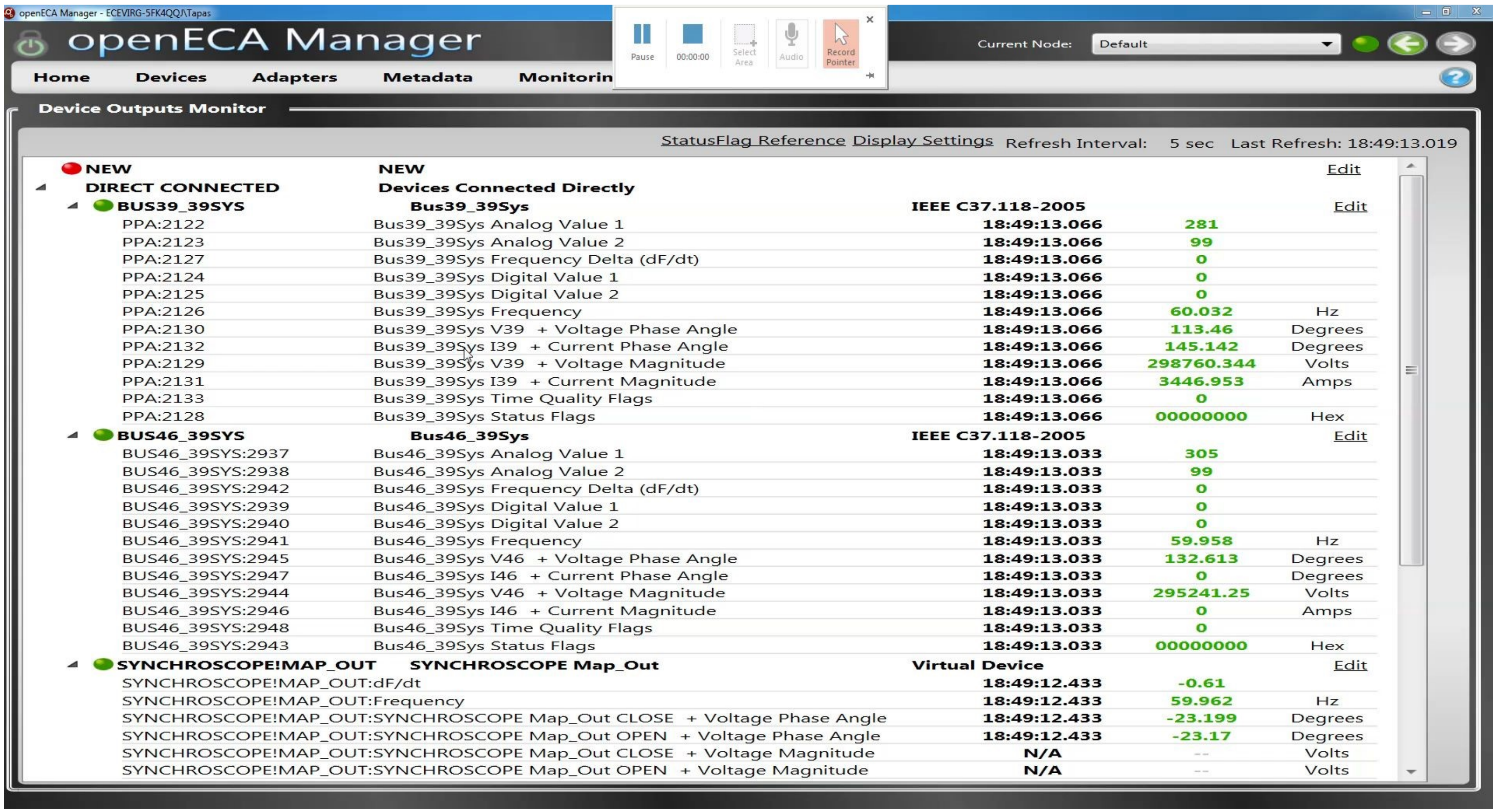
GRID PROTECTION ALLIANCE

open ECA

openECA Output Stream



Testing Video



Matt Donnelly
T&D Consulting/Montana Tech

openECA: Analytics for Power System Dynamics

Dominion Energy Offices
Richmond, VA
November 7, 2017

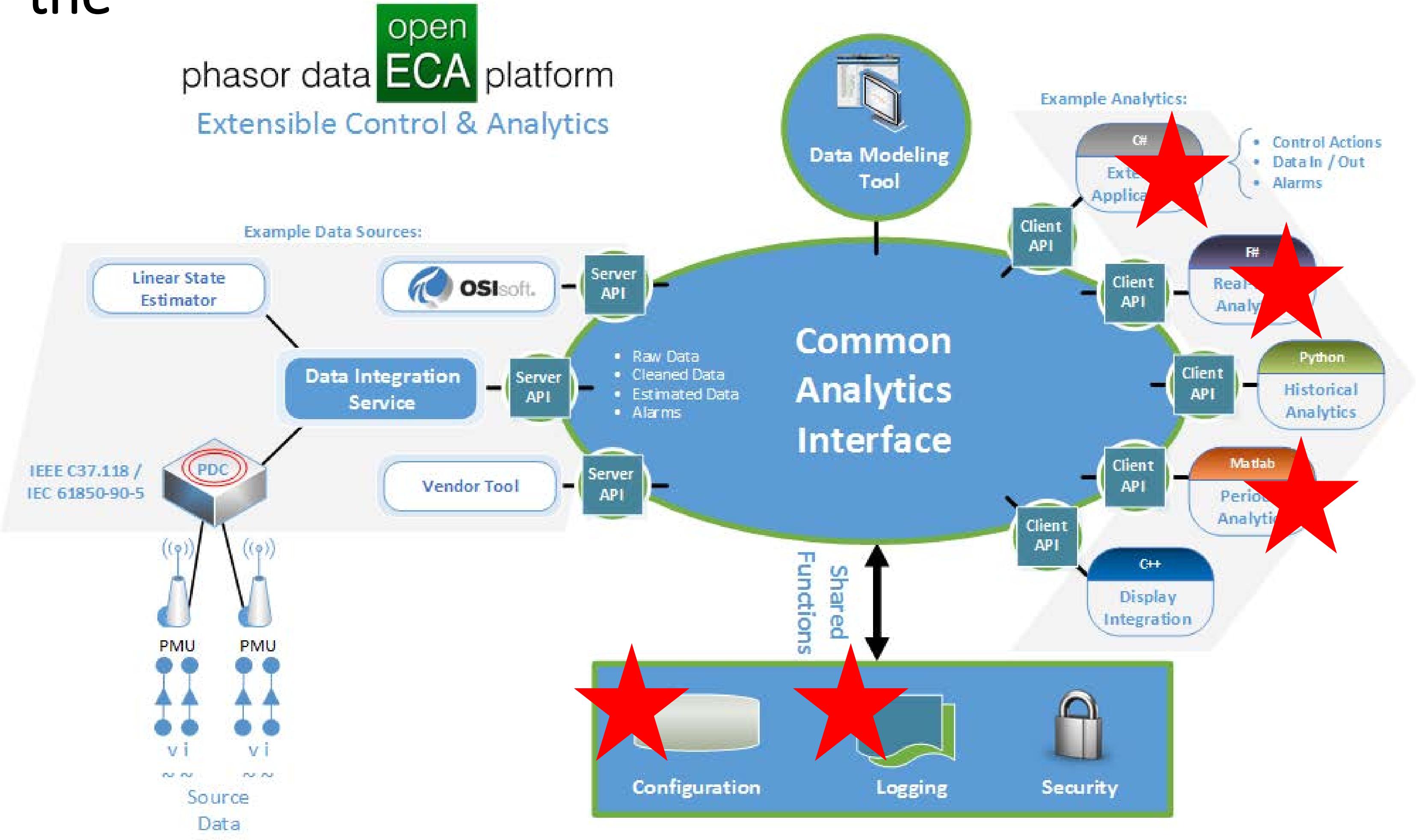
DOE FOA 970
DE-OE0000778

Objectives

- To deploy analytics within the openECA framework with special focus on power system dynamics;
- To contribute to openECA interface standards and specifications;
- To test and provide feedback to GPA on openECA strengths/weaknesses for our use case.

System Data Flow Diagram

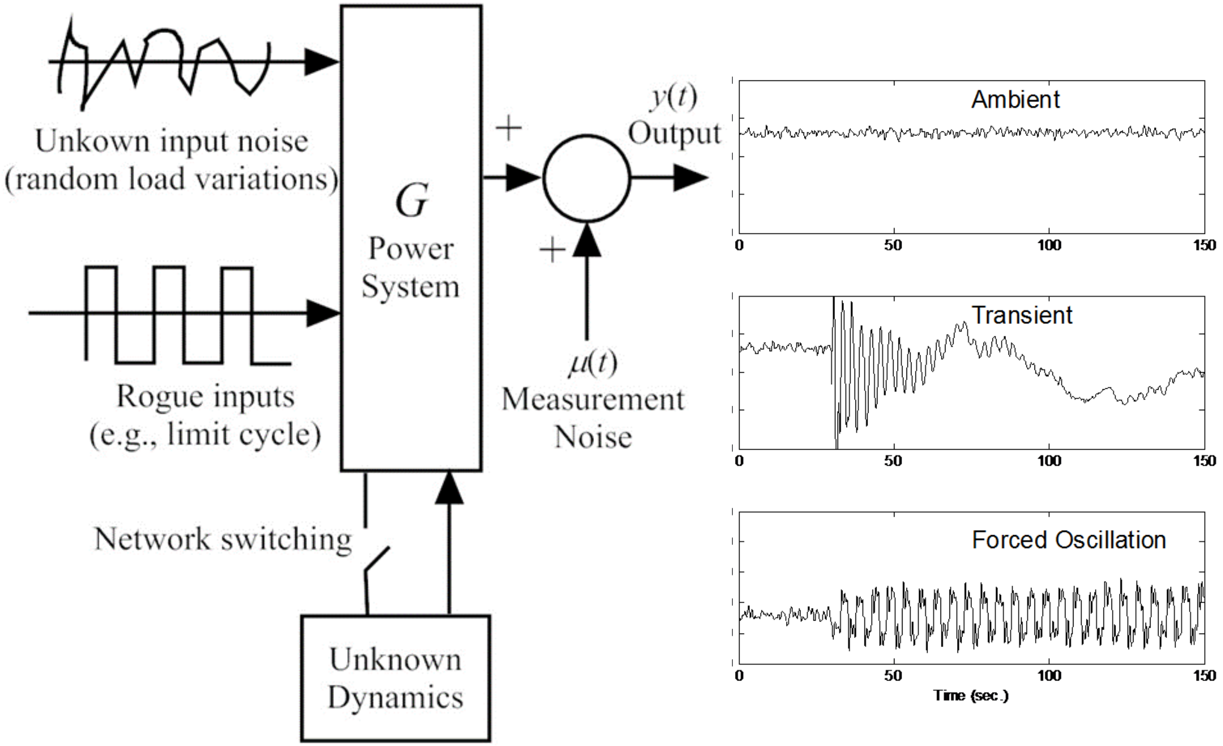
Our analytics “exercised” the openECA here…



Power System Dynamics

The power system is a dynamic electro-mechanical system. As such, disturbances or perturbations to the system elicit a dynamic response.

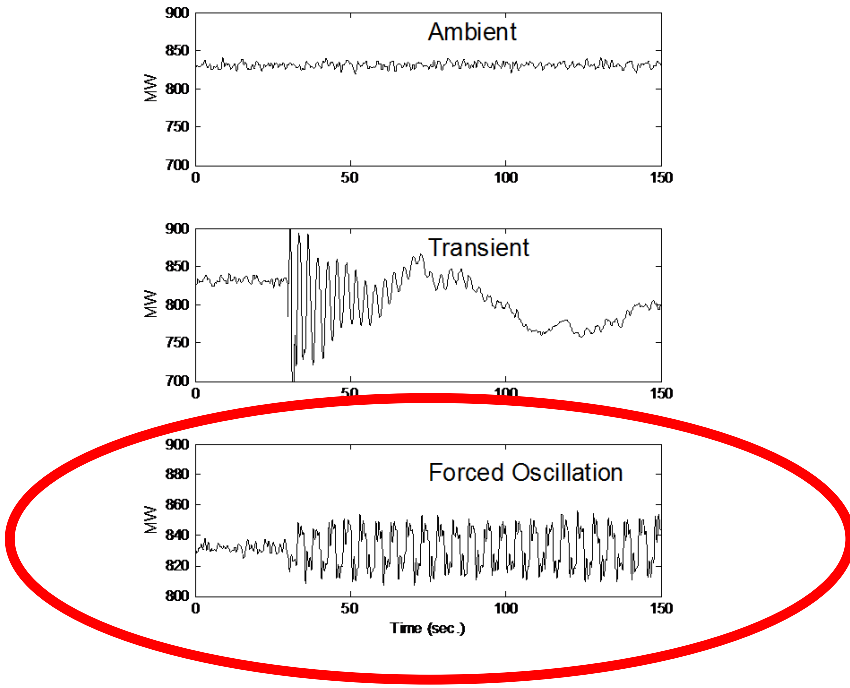
- If we build a physics-based model of the power system, “G”, we can experiment by simulating different perturbations.
  - Noise inputs, e.g. turning on a light switch;
  - Oscillatory inputs, e.g. broken controller;
  - Switching actions, e.g. generator trip
- Measurement noise also exists, interfering with our ability to accurately process the measurements.



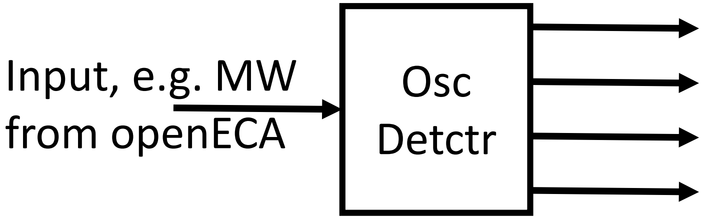
Oscillation Detector

Oscillation Detector

- Intended TRL: openECA "Real-Time Analytic"
- Intended Use:
  - Process many inputs from wide geographic area, seeking oscillation energy over broad spectrum, with minimal CPU usage
  - Easy-to-use outputs scaled to engineering units, intended for operations personnel
- Status:
  - Released August 2017
  - Deployed with openECA beta at participant test sites
  - Grafana visualization dashboard in beta release

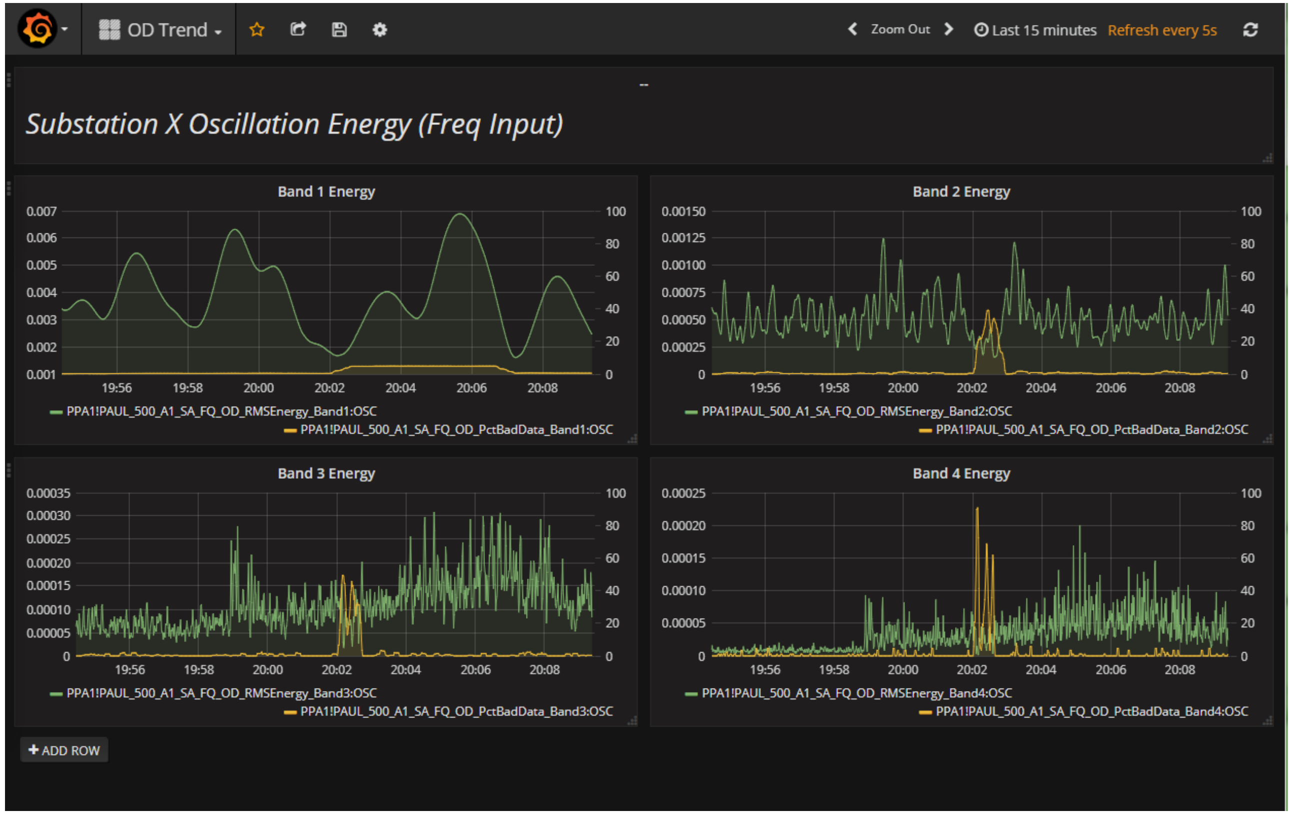


Oscillation Detector



Outputs to openECA

- Band1 Energy (e.g. governor)
- Band2 Energy (e.g. system mode)
- Band3 Energy (e.g. local mode)
- Band4 Energy (e.g. voltage controller)



- One input at, for example, 30 samples per second
- Four outputs at one sample per second
- There are an additional four outputs to indicate trustworthiness of energy estimates

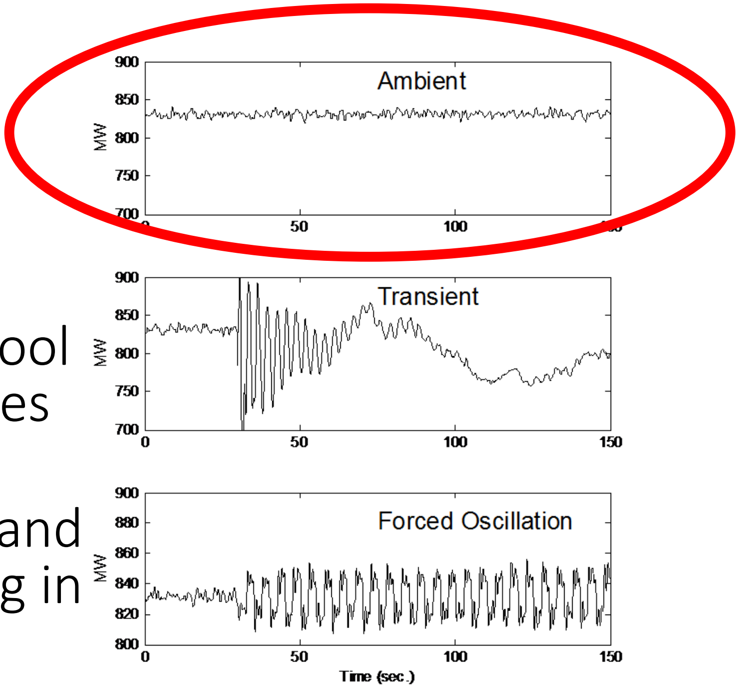
Oscillation Detector

- Next steps
  - Improvements to setup utility
    - Can be part of open source effort to create a "measurement selector widget" for openECA and for STTP
  - Addition of optional alarm module
    - Can be standalone, so that outputs from OD flow to openECA and then act as inputs to another openECA analytic; or
    - Can be an integral part of the OD analytic
  - Addition of spectrum output capability
    - This feature adds many output measurements to the openECA data bus. Rather than eight new measurement channels, we would add hundreds. Need to explore whether this is worth the effort for the user community.

Mode Meter

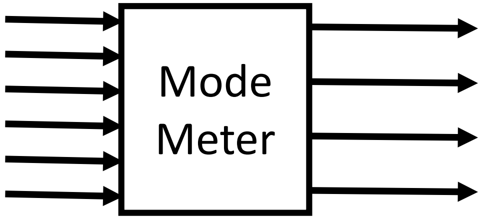
Mode Meter

- Intended TRL: openECA "Real-Time Analytic"
- Intended Use:
  - Processes a carefully selected set of inputs known to demonstrate "observability" to a known system oscillatory mode
  - Where known system mode presents an operating challenge, this tool can be used to provide operations personnel with accurate estimates of the characteristics of the mode
  - Note that mode damping changes with system loading conditions, and therefore it can be useful to have a "meter" that measures damping in real time
- Status:
  - Released October 2017
  - Currently being deployed at participant sites
  - Grafana visualization dashboard under development



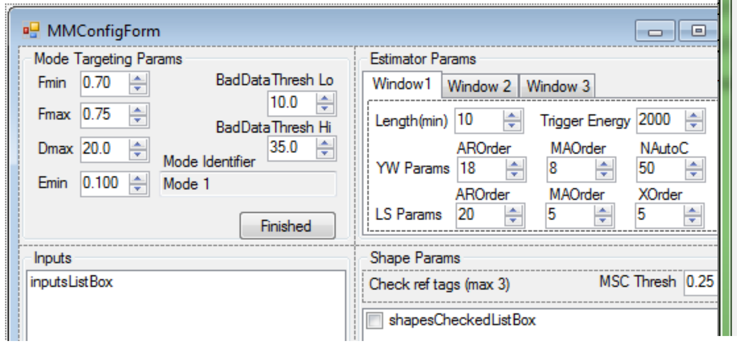
Mode Meter

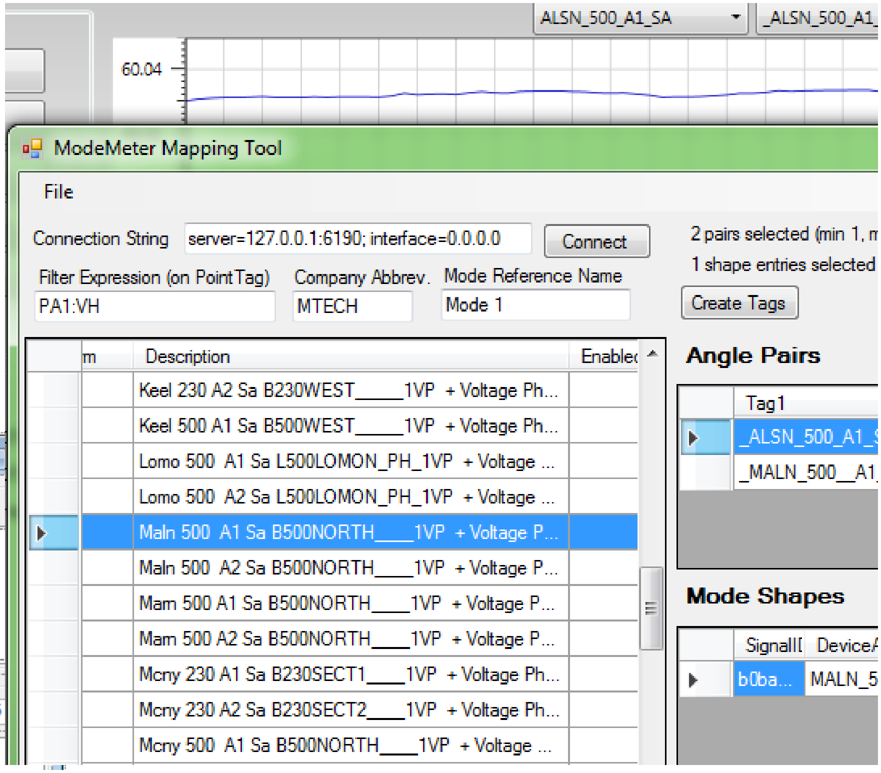
Inputs from openECA
- Must be voltage angles
- Up to 20 inputs
- Selected to offer max visibility of modal activity



Outputs to openECA
- Mode Frequency
- Mode Damping
- Energy captured by signal processing algorithms
- Mode shape, consisting of up to 20 magnitudes and angles

- Multiple inputs from openECA
- Each input is required to be a *voltage angle*
- Extensive setup and configuration required – this analytic really tested openECA's configuration mechanism





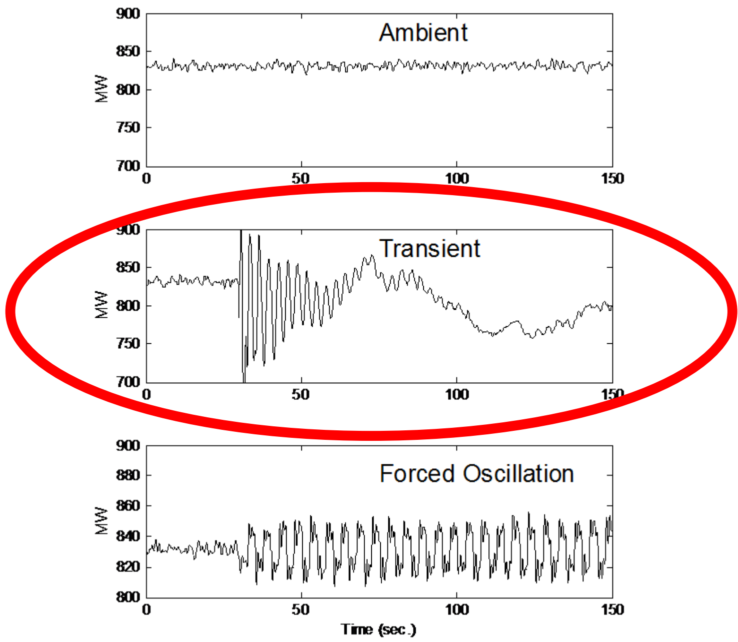
Mode Meter

- Next steps
  - Improvements to setup utility
    - Can be part of open source effort to create a “measurement selector widget” for openECA and for STTP
  - Grafana visualization dashboard
    - Mode frequency and damping are fairly straightforward
    - Best way to visualize shape is with polar plot, and this “widget” is not readily available in Grafana
  - Improvements to flexibility of configuration
    - Allow for inputs other than voltage angles
    - Offer guidance and help files for signal processing parameter entries

Generator Model Validator

Synchronous Machine Parameter Validation

- Intended TRL: openECA “Off-Line Analytic, Research Use Case”
- Intended Use:
  - When given synchrophasors from a PMU located at or near the GSU transformer of a synchronous generator, this analytic can automatically perform NERC-mandated model validation
- Status:
  - Analytic was developed and proofed in offline mode
  - Tested online in openECA using the Matlab template



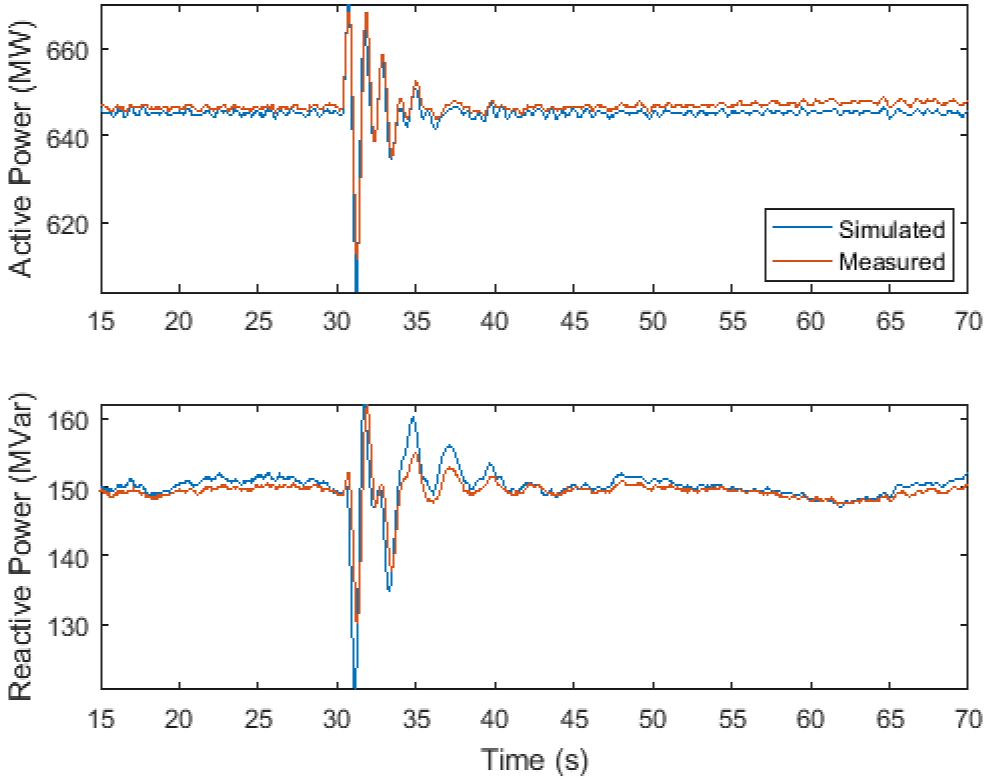
Synchronous Machine Parameter Validation

Inputs from openECA
- Voltage phasor
- Current phasor

→ **Model Vlidatr** →

Outputs to openECA
- Boolean indicating new report generated
- (also stores report to local or remote storage media)

- Two inputs to the analytic from openECA
- Inputs must be from PMU as close as possible to generator terminals, i.e. measuring generator output current
- The parameters of the model to be validated must be stored in persistent memory and loaded to the analytic (much like the LSE analytic)



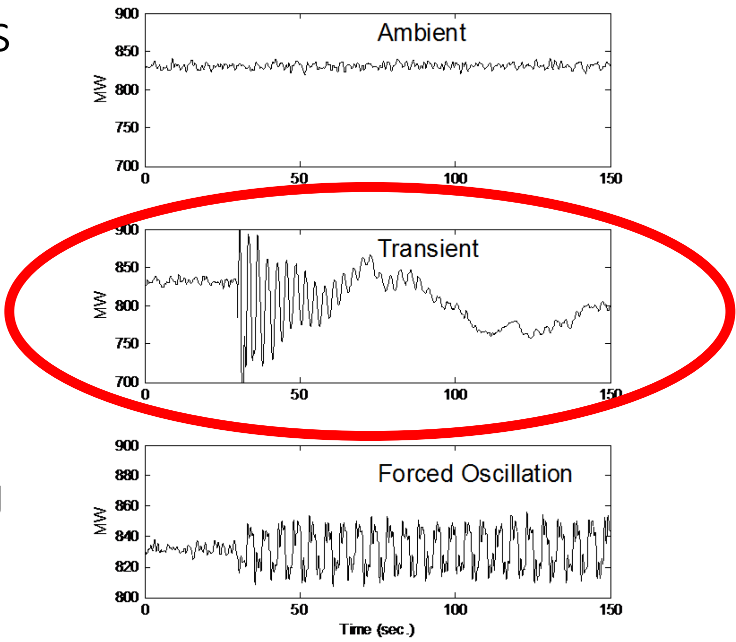
Synchronous Machine Parameter Validation

- Next steps
  - Port to .NET
  - Extend the model library to include additional generator, turbine, governor, exciter and PSS models in PSLF, PSS/E and Powerworld formats
  - Extend the research portion of the effort to include wind power plant models
  - Improve the output report template

Acceleration Trend Relay Enhancements

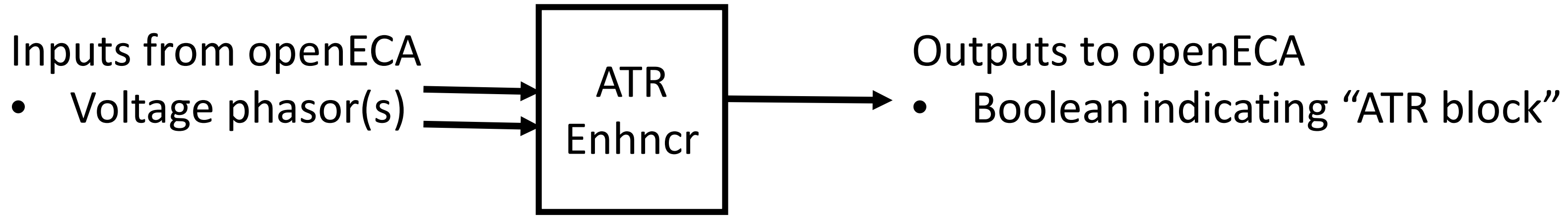
Acceleration Trend Relay Enhancements

- Intended TRL: openECA “Off-Line Analytic, Research Use Case”
- Intended Use:
  - A select category of synchronous generators, typically remotely located and connected to the bulk power grid over long transmission lines, are protected by an acceleration trend relay (ATR). The input to an ATR is generator shaft speed. If the shaft is accelerating beyond preset limits, the generator may be tripped offline to protect the unit.
  - A complication arises when the entire system is accelerating. In this case the unit may “false trip”, i.e. the ATR may call for a trip when the unit is not in danger of an out-of-step condition.
  - Incorporation of synchrophasor measurements remote from the unit may help eliminate false trips.

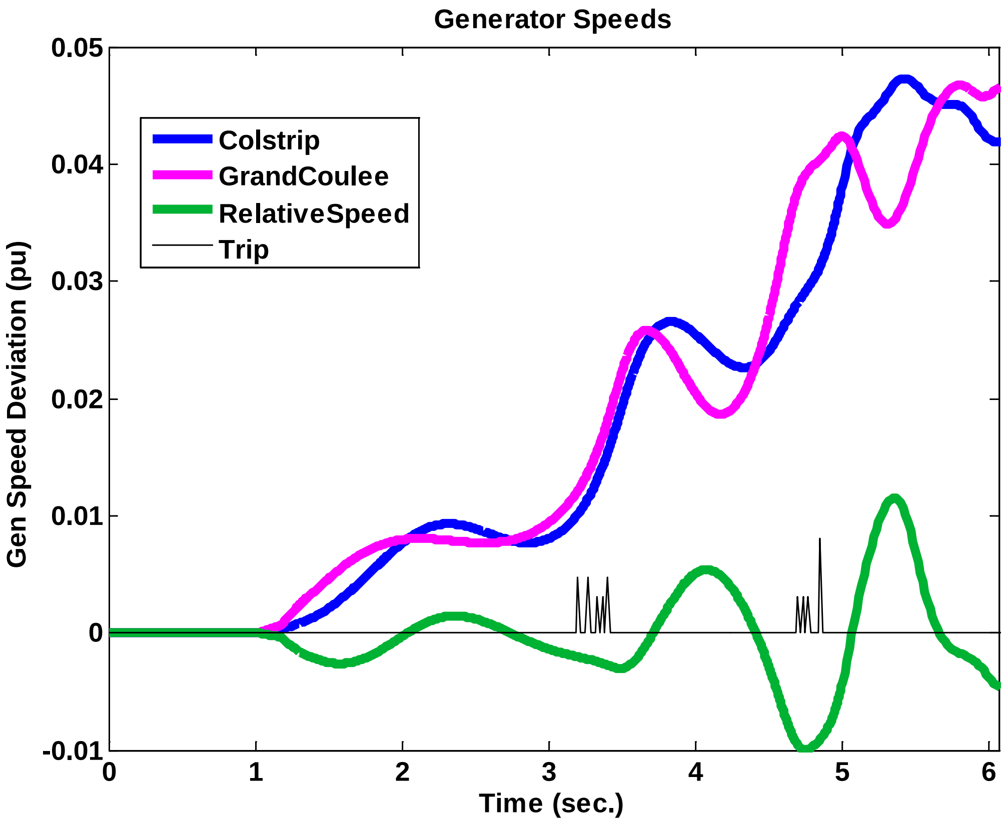


- Status:
  - Concept proven, to be written up in MS thesis and in a paper
  - openECA could host the entire ATR logic, or could process the PMU input and pass to ATR

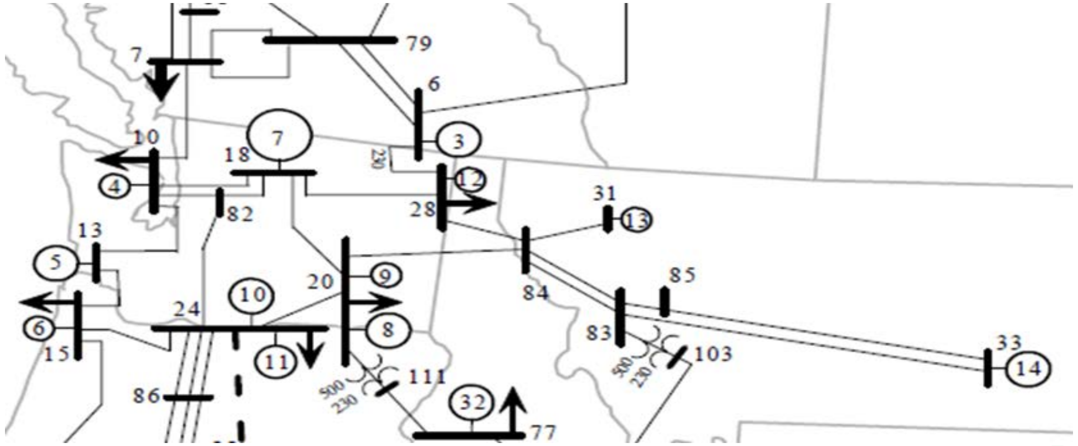
Acceleration Trend Relay Enhancements



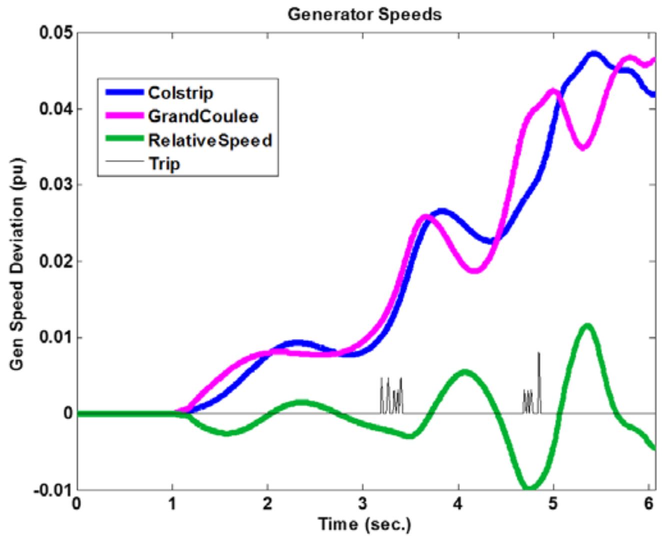
- Inputs are voltage phasor(s) from points representing the presumed inertia of the bulk grid to which ATR-equipped generator is connected
- Output is an "ATR block" Boolean that would be used to block the trip command from the ATR relay



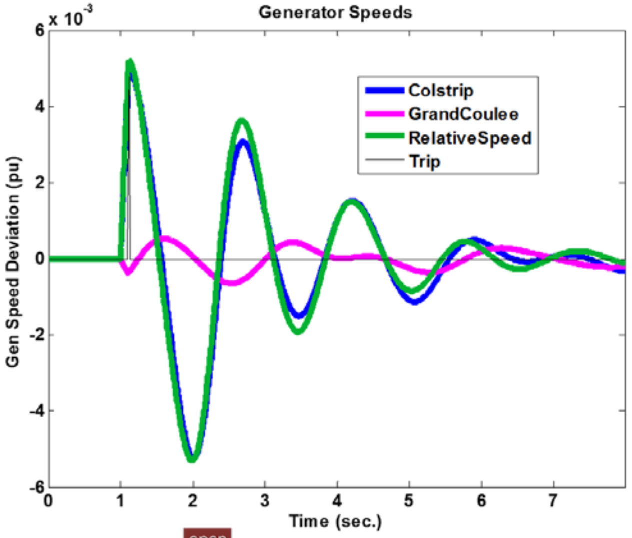
Acceleration Trend Relay Enhancements

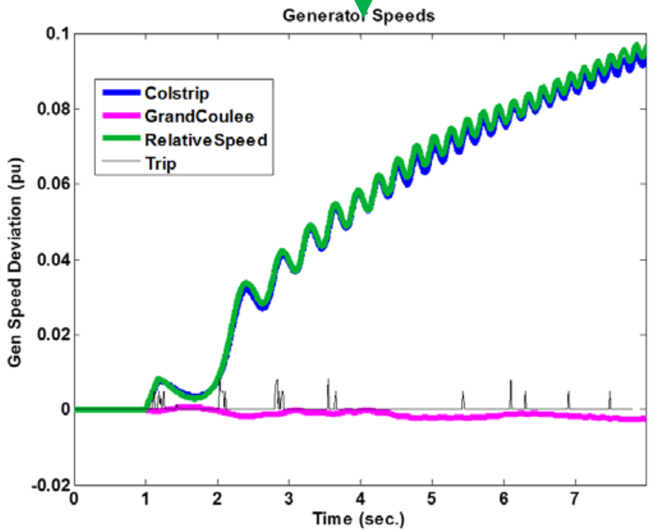


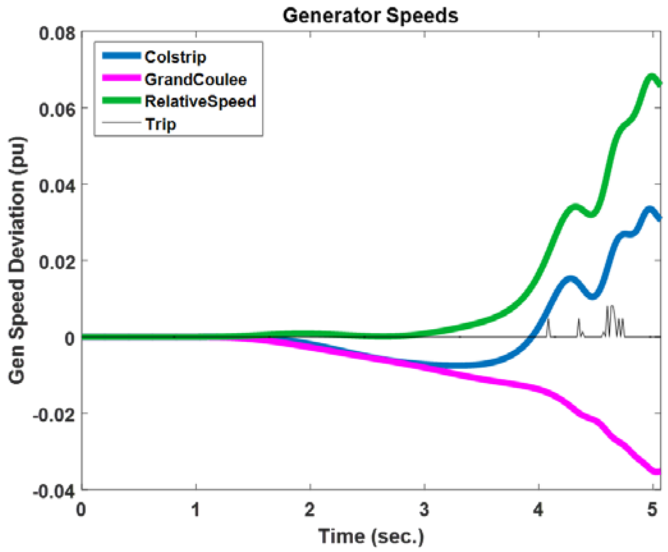
BLOCK



DO NOT BLOCK





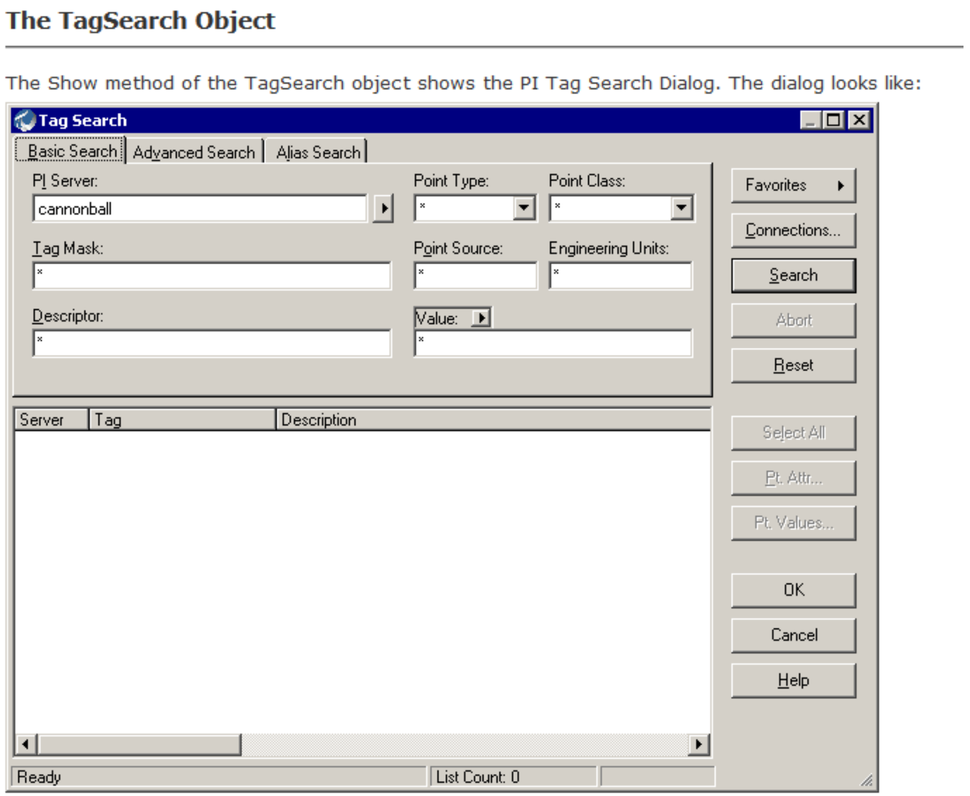


Acceleration Trend Relay Enhancements

- Next steps
  - Work with host utility to build value proposition for eliminating false trips

Lessons Learned

- Standardized mappings allow flexibility
  - Build analytic one time, then deploy with different mapping files for different openECA users and instances
- Common suite of utilities could be developed to assist with setup and configuration… Perhaps like PI AF SDK
- With advent of STTP, this effort to establish common interface methods for analytics and a suite of core convenience utilities should continue



Dominion Energy/Virginia Tech Demonstration Results Report

Local/Regional Voltage Controller

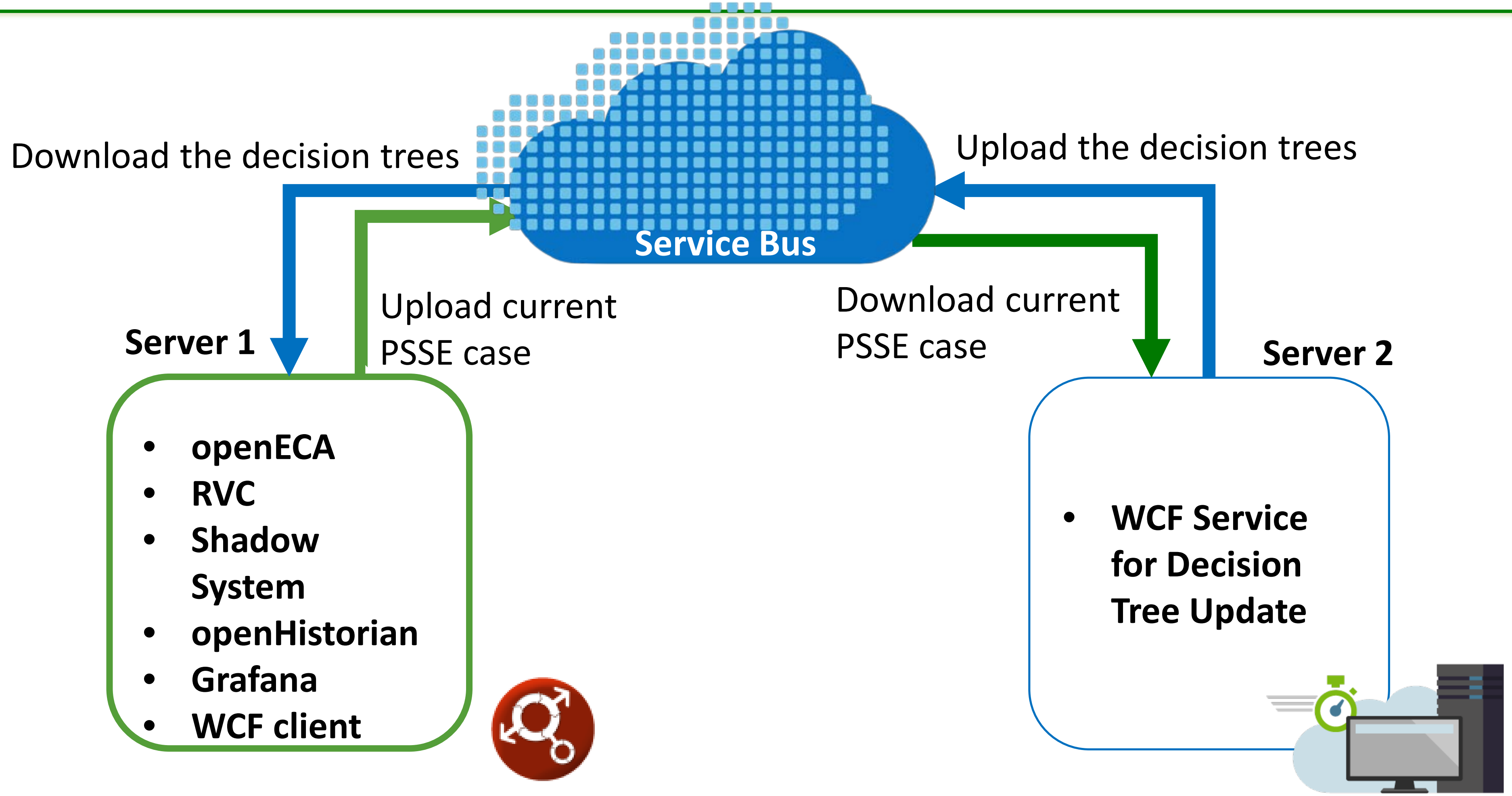
Prepared by: Duotong Yang, Ph.D

Oct 20, 2017

Demonstration Summary

- In Summer 2017, Virginia Tech graduate students completed the demonstration of Local/ Regional Voltage Controller using Bench Mark model
- Dominion Energy carried on the test by adding a server for Regional Voltage Controller (RVC)'s periodic decision tree update
- Two servers are set up for the openECA project
- Server 1 (2 cores, 8 GB RAM) is hosting:
  - Shadow System
  - Control Analytics
  - Grafana Dashboard
  - WCF Client
- Server 2 (4 cores, 20 GB RAM) is hosting:
  - WCF Server for tree training
- A Cloud Service Bus is implemented to provide Windows Communication Foundation (WCF) relay between two servers

Test Architecture



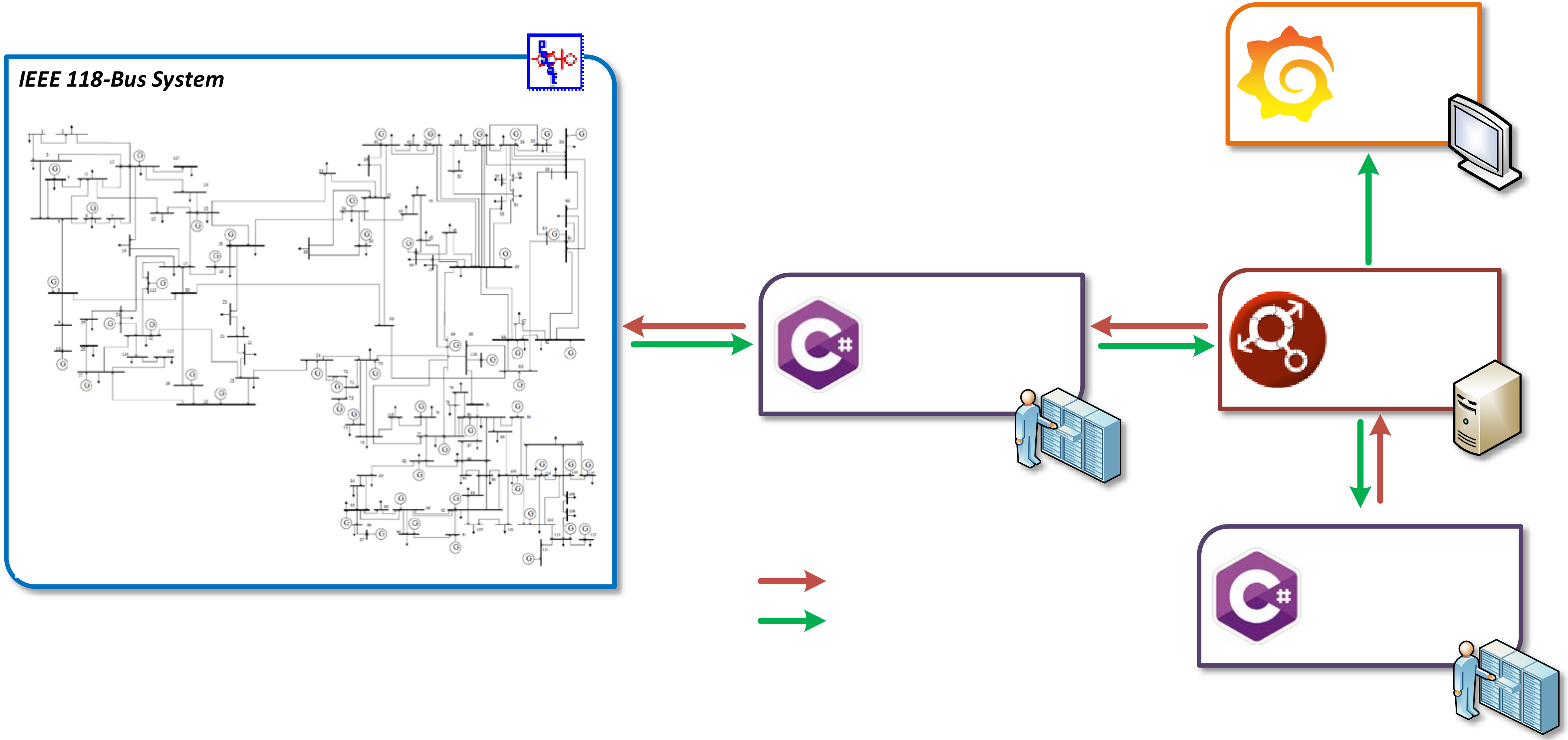
Demonstration Results Report

Shadow System

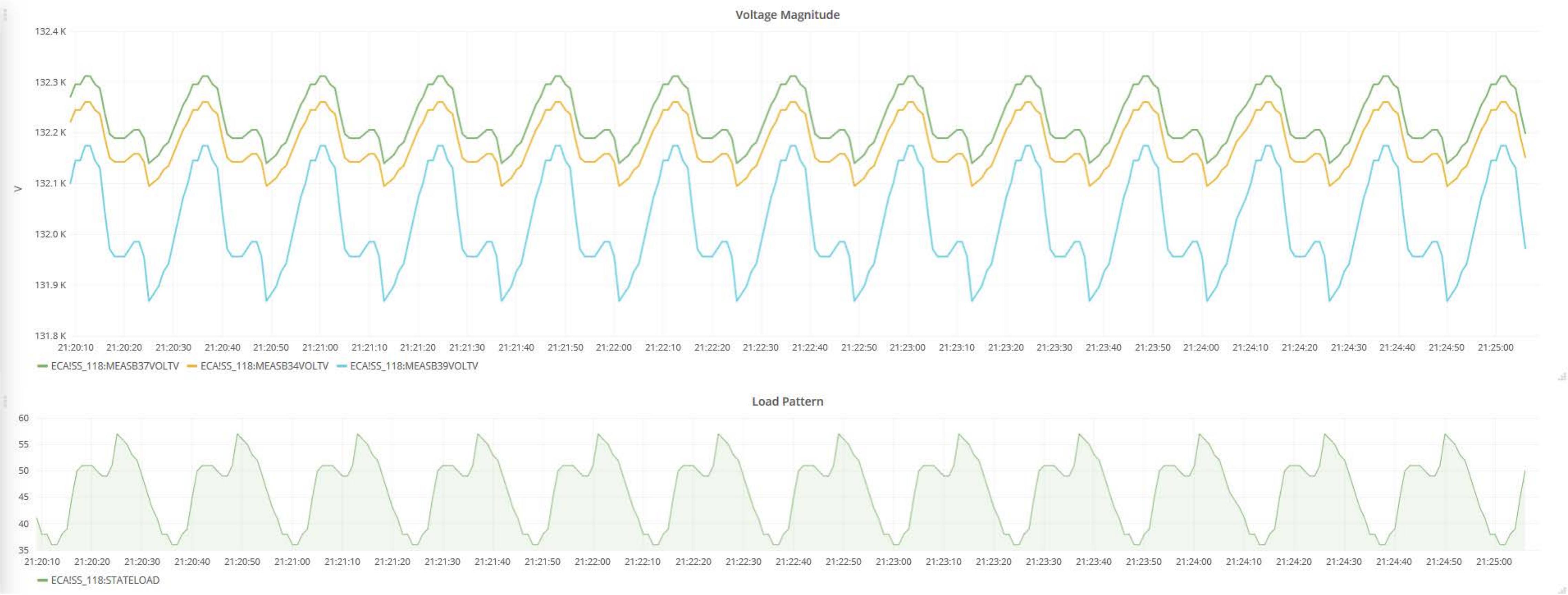
Shadow System Overview

- ***Shadow System Simulator*** is an openECA analytic deployed as a Windows service. It interacts with the power flow engine in PSS®E to provide an 1 frame/sec set of signals of all of the values collected from the static power flow calculation
- ***Control Analytics*** refers to the openECA analytics, such as Local Voltage Controller and Regional Voltage Controller, that send out control signals like raising/lowering LTCs, opening/closing circuit breakers, etc., to optimize the system operation
- ***Grafana Dashboard*** serves as the monitor of the historical measurement data collected from openECA server node

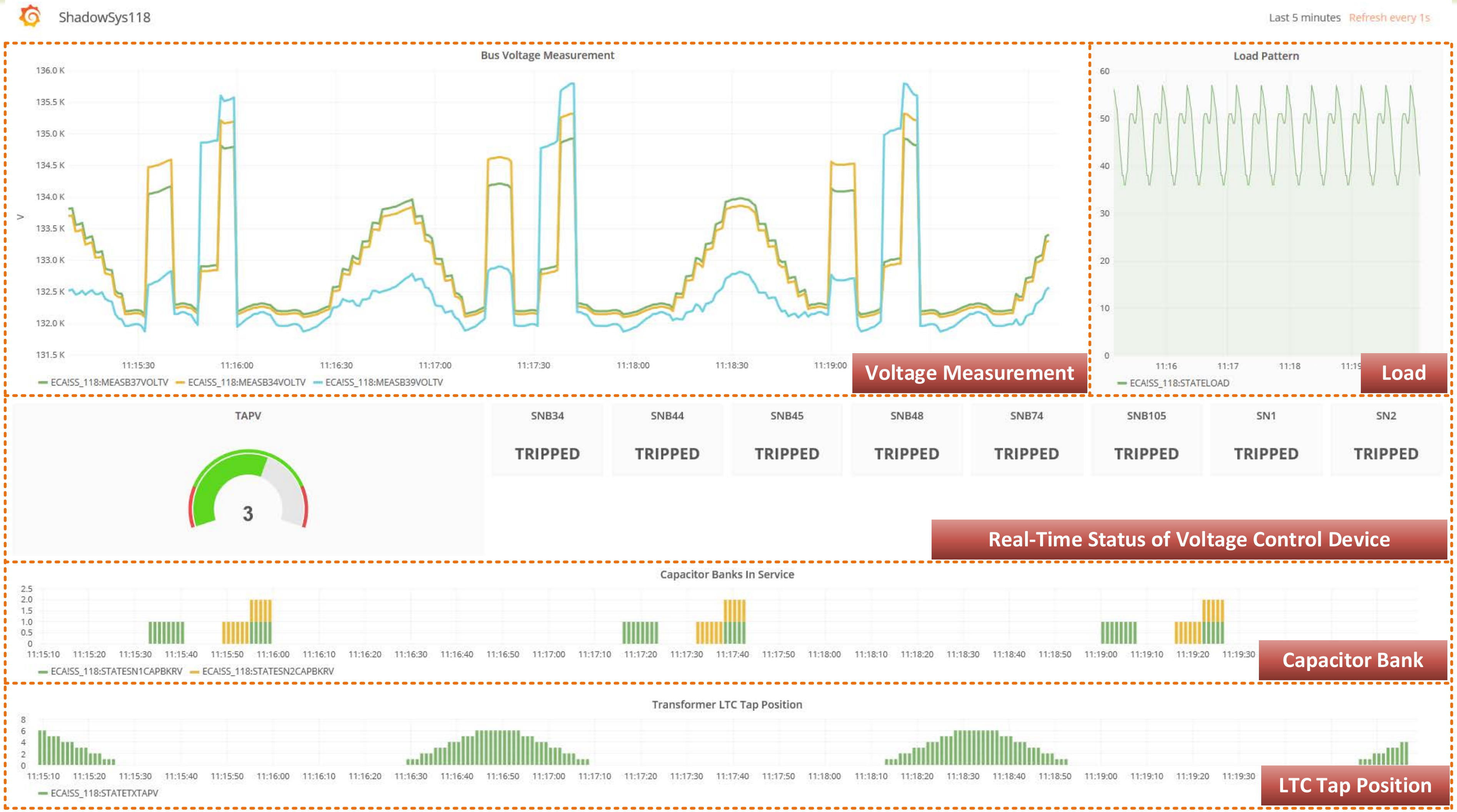
Shadow System Architecture



Shadow System Test Result (Load Scaling)



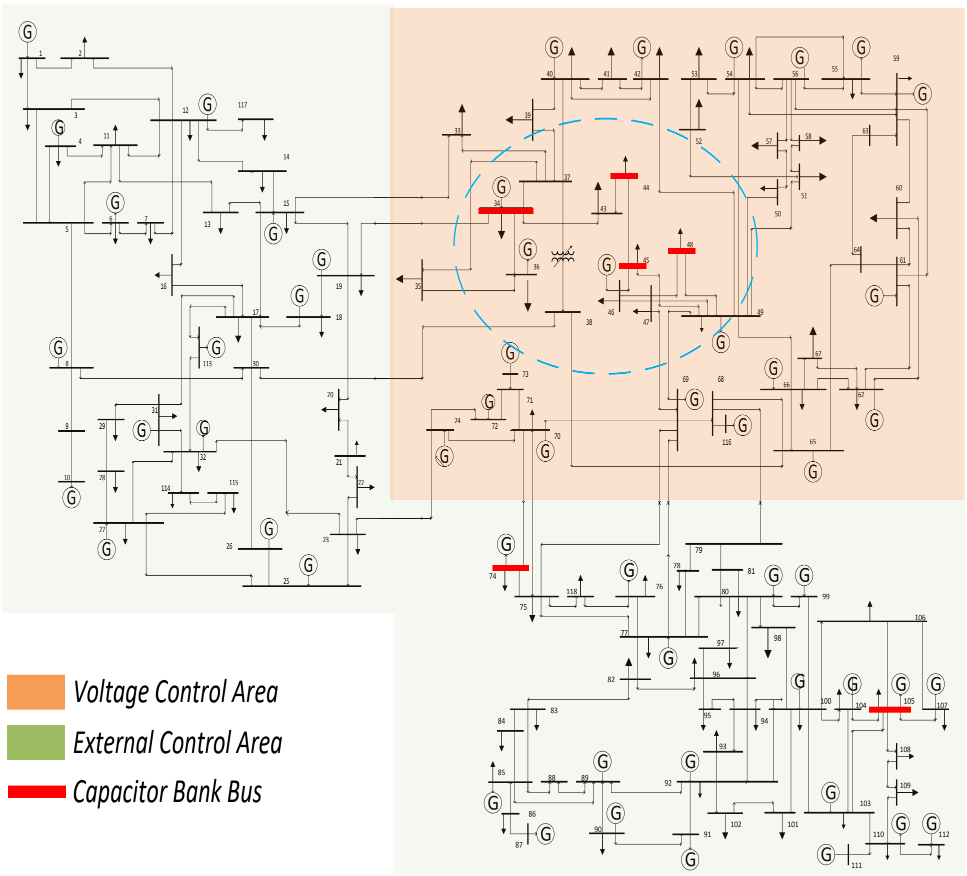
Shadow System Test Result (Controls)

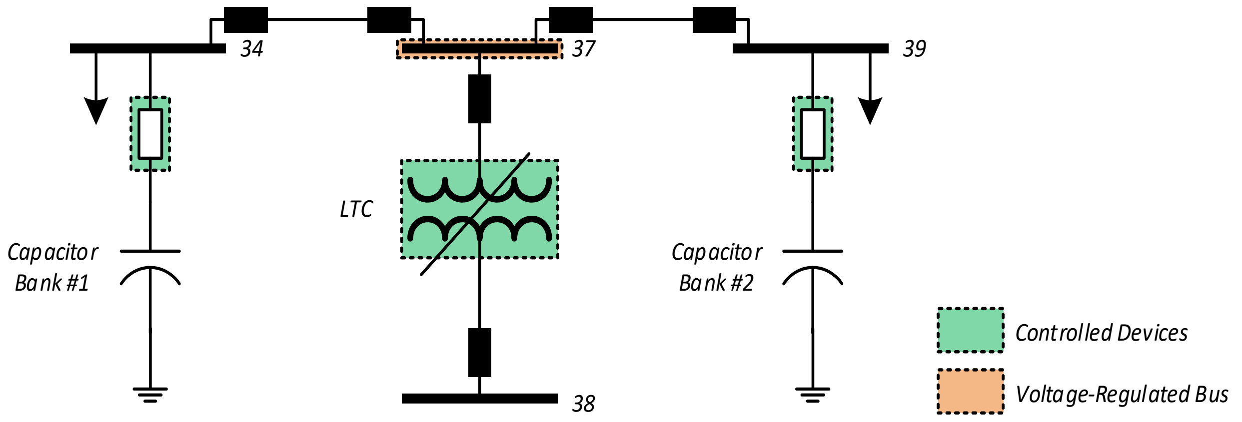


Demonstration Results Report

Local Voltage Controller

Local Voltage Controller Overview

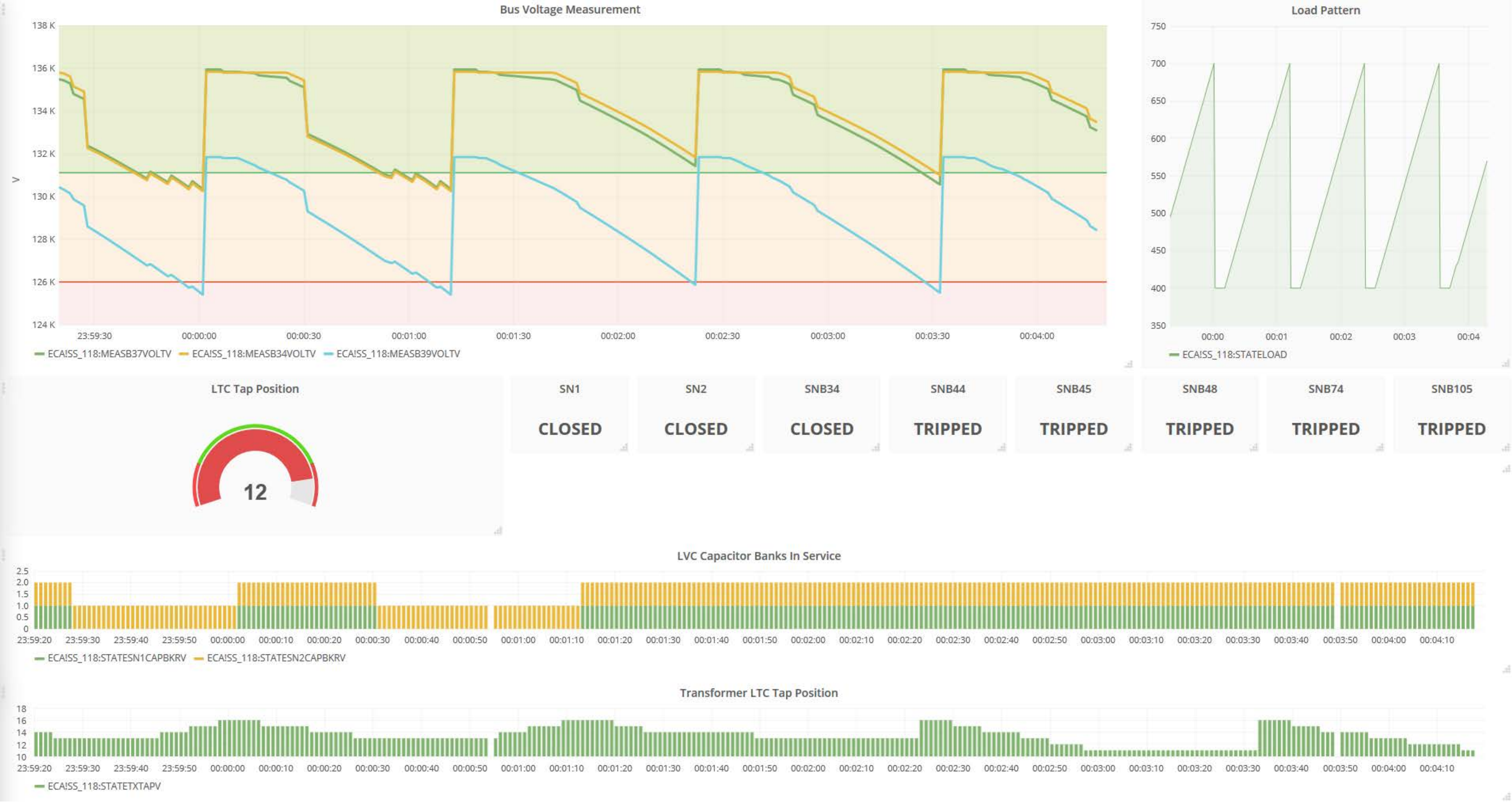




Local Voltage Controller (LVC) Employs three standard control logics as programmed in the Dominion EMS to execute control decisions regarding to substation capacitor banks and LTCs based on localized synchrophasor measurements. The “local” term refers to the scope of the measurement footprint.

LVC Test Result (Cap Control)

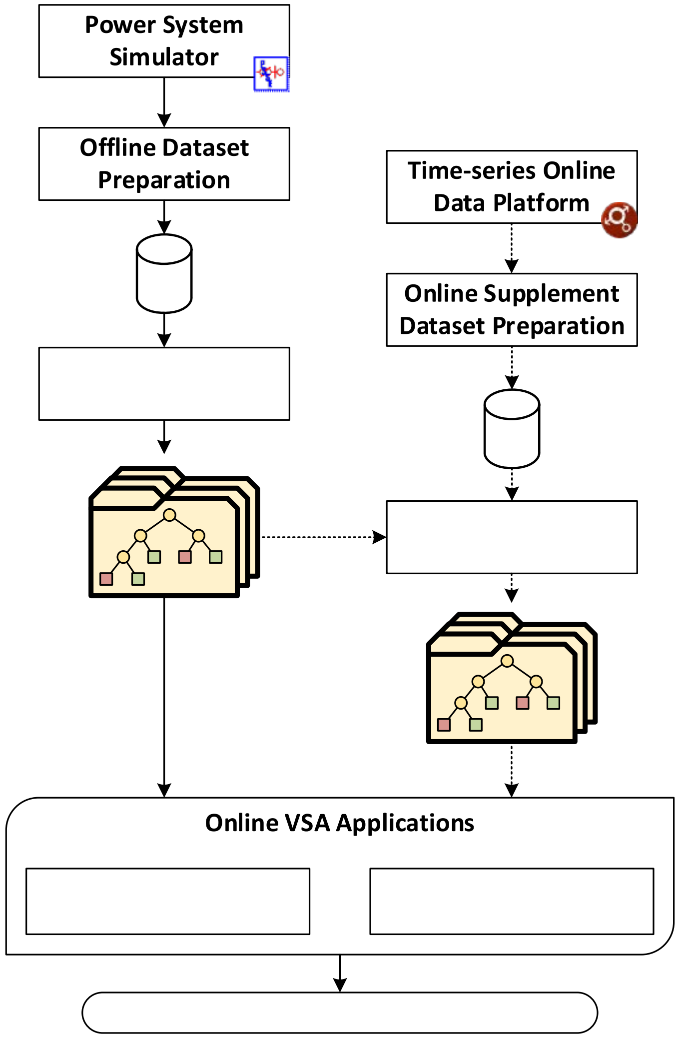
LVC Test Result (Tap Control)



Demonstration Results Report

Regional Voltage Controller

Regional Voltage Controller Architecture



Focuses on a wider range of characteristics in the large power systems. For RVC, the objective should be capable of regulating voltage profiles when the system is subjected to impending voltage insecure problems.

Decision-Tree-Based Classification Models

- **Fast responsive**, capable of real-time use for online VSA applications
- **Periodic Update** of models is required
- Classification Models trained using ***Accord.NET Framework***

Voltage Security Assessment

Voltage Security Assessment

- ***Preventive VSA*** reads each data frames of voltage magnitude measurements published by openECA and returns a security assessment result of “Secure” or “Insecure” based on the trained/updated decision-tree model
- ***Remedial VSA*** responses when the preventive VSA returns a “Insecure” result for current openECA input frame, and starts searching for feasible voltage control strategies that bring the system back to a “Secure” operating condition

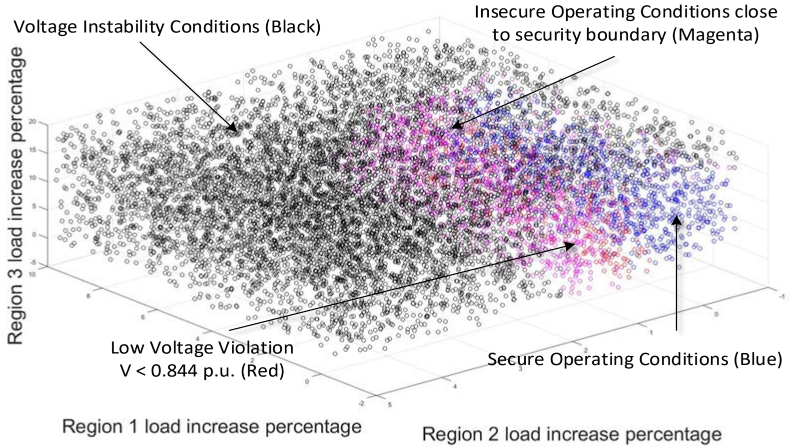


Figure (a): Voltage Security Assessment – Before Control

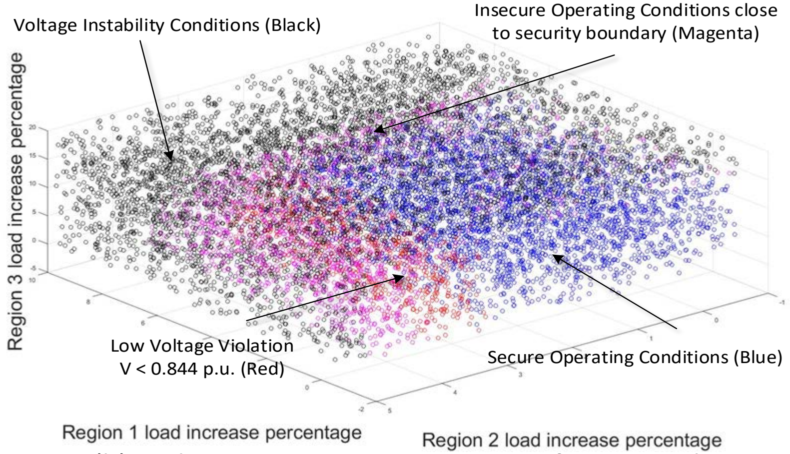


Figure (b): Voltage Security Assessment – After Control

RVC Test Result (Cap Control)

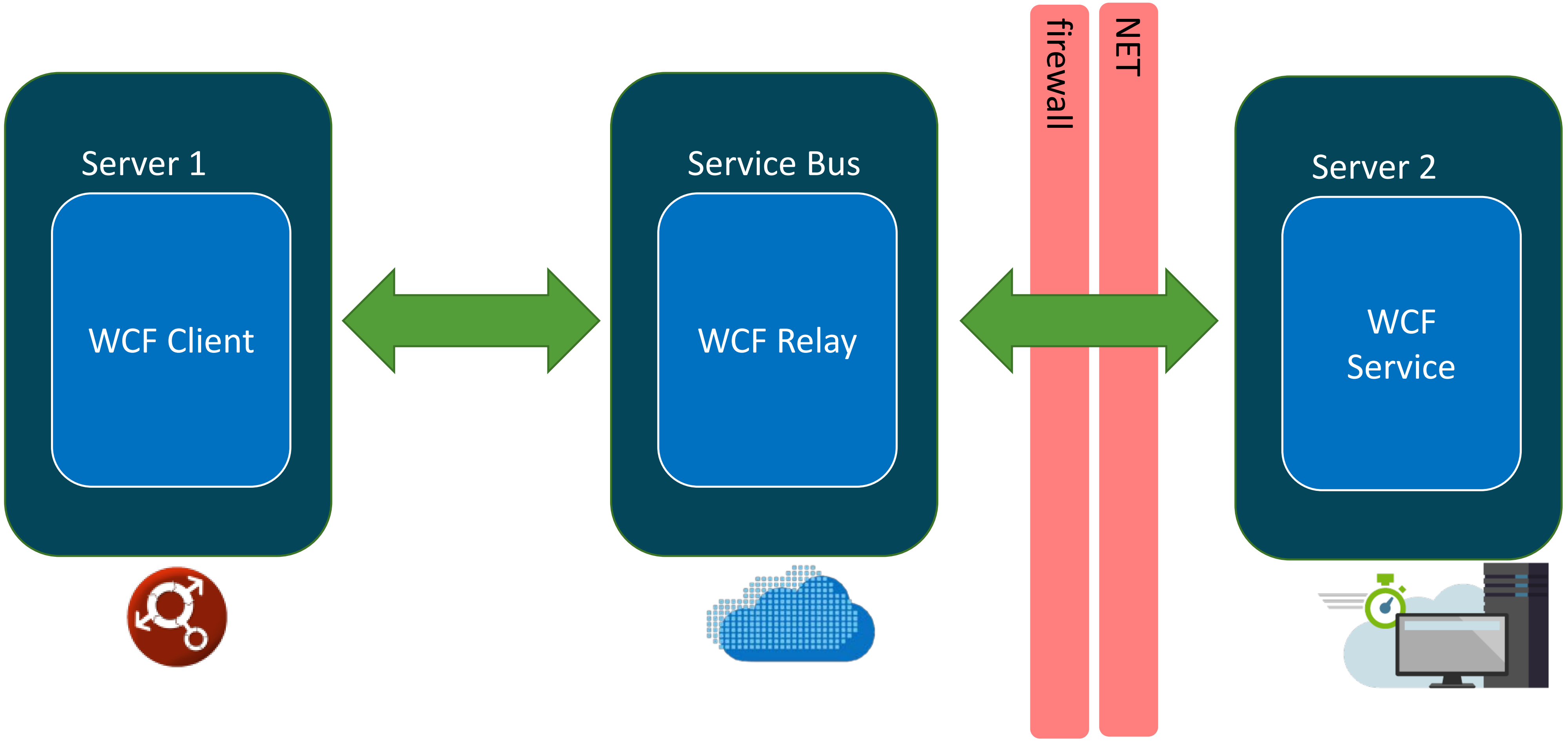
Demonstration Results Report

Cloud WCF Relay

Cloud WCF Relay Overview

- ***Cloud Service Bus:*** is a messaging/storage service between applications and services. The service bus is hosted on MS Azure
- ***WCF Relay:*** enables engineers to build applications to run on multiple servers
- ***WCF Service:*** An endpoint that hosting on-premises applications
- ***WCF Client***: A client that can invoke the operations of the WCF Service
- ***Blob trigger***: Download the decision trees when tree training is completed

Cloud WCF Relay Architecture

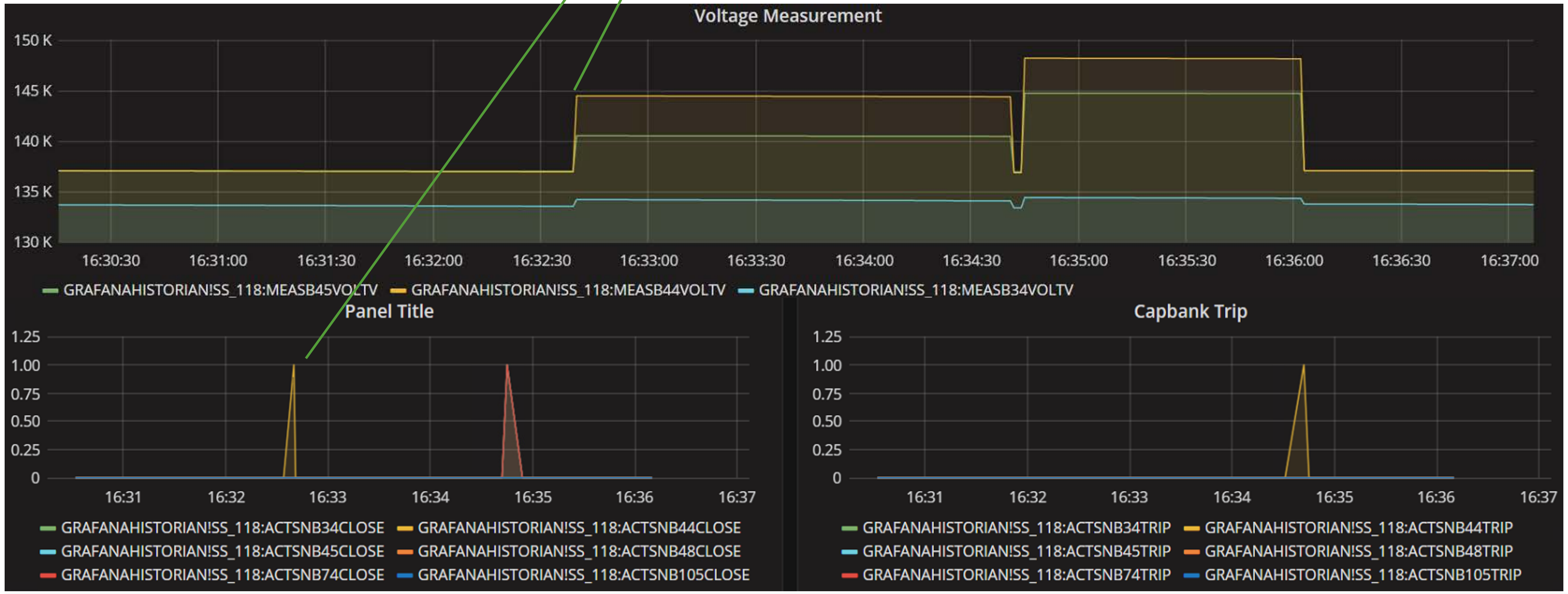


WCF Relay Test Results (Control triggers the Decision Tree Update)

```
Service address: sb://openecatest.servicebus.windows.net/EchoService/
Press [Enter] to exit
Control is activated
Echoing: current load incremental percentage ieee118bus_divided_basecase.sav
```

Control Decision Triggers the Decision Tree Update at Server 2

Control Decision at Server 1



WCF Relay Test Results (Upload Decision Trees to the Cloud)

Upload | Download | Open | New Folder | Copy URL | Select All | Copy | Paste | Rename | Delete | Make Snapshot | Manage Snapshot | Folder Statistics | Refresh

treecontainer

Search by prefix (case-sensitive)

Name	Last Modified	Blob Type	Content Type	Size	Lease State	Disk Name	VM Name	Disk Type	Resource Group Name
DecisionTreeModel_ControlForLowVoltage_0_train	Thu, 19 Oct 2017 19:32:03 GMT	Block Blob	application/octet-stream	7.6 KB					
DecisionTreeModel_ControlForLowVoltage_10_train	Thu, 19 Oct 2017 19:30:57 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_11_train	Thu, 19 Oct 2017 19:30:58 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_12_train	Thu, 19 Oct 2017 19:30:58 GMT	Block Blob	application/octet-stream	6.9 KB					
DecisionTreeModel_ControlForLowVoltage_13_train	Thu, 19 Oct 2017 19:30:58 GMT	Block Blob	application/octet-stream	6.9 KB					
DecisionTreeModel_ControlForLowVoltage_14_train	Thu, 19 Oct 2017 19:30:59 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_15_train	Thu, 19 Oct 2017 19:30:59 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_16_train	Thu, 19 Oct 2017 19:30:59 GMT	Block Blob	application/octet-stream	6.9 KB					
DecisionTreeModel_ControlForLowVoltage_17_train	Thu, 19 Oct 2017 19:31:00 GMT	Block Blob	application/octet-stream	6.9 KB					
DecisionTreeModel_ControlForLowVoltage_18_train	Thu, 19 Oct 2017 19:31:00 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_19_train	Thu, 19 Oct 2017 19:31:00 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_1_train	Thu, 19 Oct 2017 19:30:54 GMT	Block Blob	application/octet-stream	7.6 KB					
DecisionTreeModel_ControlForLowVoltage_20_train	Thu, 19 Oct 2017 19:31:01 GMT	Block Blob	application/octet-stream	6.9 KB					
DecisionTreeModel_ControlForLowVoltage_21_train	Thu, 19 Oct 2017 19:31:01 GMT	Block Blob	application/octet-stream	6.9 KB					
DecisionTreeModel_ControlForLowVoltage_22_train	Thu, 19 Oct 2017 19:31:01 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_23_train	Thu, 19 Oct 2017 19:31:02 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_24_train	Thu, 19 Oct 2017 19:31:02 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_25_train	Thu, 19 Oct 2017 19:31:02 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_26_train	Thu, 19 Oct 2017 19:31:03 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_27_train	Thu, 19 Oct 2017 19:31:04 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_28_train	Thu, 19 Oct 2017 19:31:04 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_29_train	Thu, 19 Oct 2017 19:31:08 GMT	Block Blob	application/octet-stream	7.2 KB					
DecisionTreeModel_ControlForLowVoltage_2_train	Thu, 19 Oct 2017 19:30:55 GMT	Block Blob	application/octet-stream	6.9 KB					

Showing 1 to 64 of 64 cached items

The updated Decision Trees are saved in the Cloud and ready for download

WCF Relay Test Results (Decision Trees Training on the Cloud)

For a richer monitoring experience, including live metrics and custom queries, enable Application Insights for your Function app

Success count since Oct 1st
41

Error count since Oct 1st
0

Invocation log Refresh — live event stream

Function	Status	Details: Last ran	(duration)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(6,937 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(5,484 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(8,281 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(9,578 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(8,359 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(8,937 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(9,547 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(8,515 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(7,625 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(2,703 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(5,641 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(2,625 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(3,937 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(1,250 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(1,234 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(2,422 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(2,953 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(2,437 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(2,531 ms)
blobTrigger (blobcontainer/trai ...)	✔	4 days ago	(5,844 ms)

Invocation details

Parameter	
myBlob	blobcontainer/train_63
name	train_63
log	

Logs

```
C# Blob trigger function Processing blob Name:train_63
 size: 507265
BlockBlob destination name: tree_train_63
C# Blob trigger function trained tree completed. successfully train decision tree_train_63
 total time consumption in Milliseconds: 6169
```

The decision tree can be also trained in the Cloud

Advantages:

1. It is a server-less function (easy to use)
2. The computation can be scaled up automatically (far more than 4 cores)

Summary

- The openECA, Grafana, openHistorian 2.0's installation and operation is quick and simple
- openECA is an easy to use platform that enable engineers to quickly transfer research results to pre-commercial products
  - ***Shadow system:*** windows service that hosting PSSE case
  - ***Local Voltage Controller:*** Voltage controller with Dominion standard control logic
  - ***Regional Voltage Controller:*** Voltage controller using Machine Learning techniques
- openECA is demonstrated to have a capability of integrating multiple powerfull frameworks
  - ***Accord.NET :*** A Machine learning framework
  - ***Microsoft.Azure :*** A framework to configure the cloud based storage, service bus, and server-less function

Dominion Energy Virginia Demonstration Results Report

openECA, openHistorian, Grafana Results

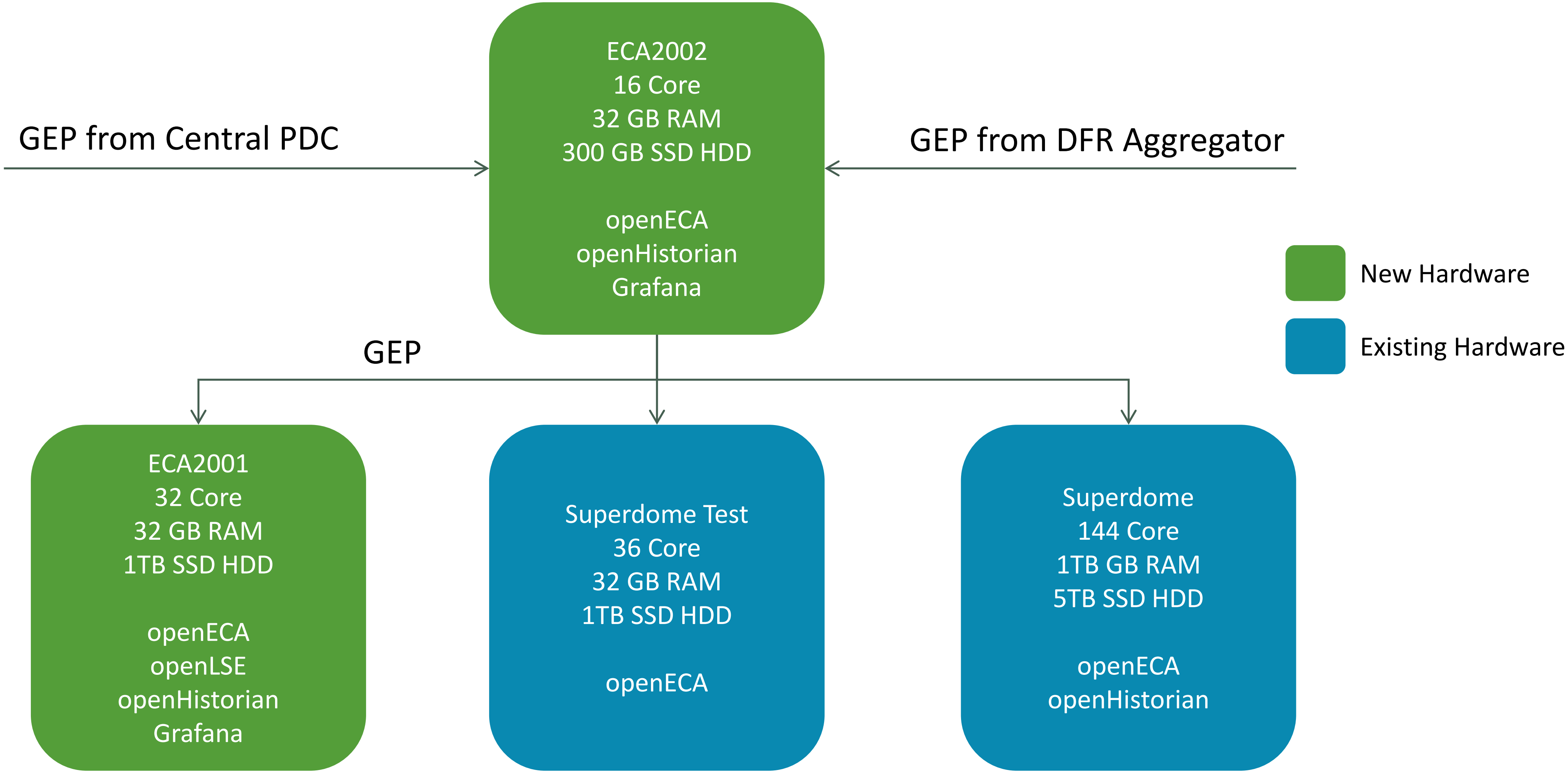
Prepared by: Kevin D. Jones, Ph.D.

October 20, 2017

Deployment Summary

- Dominion deployed an additional two physical servers in August 2017 as part of the openECA project.
- The role of the two servers are:
  - (16 Core, 32 GB RAM, 300 GB SSD HDD) – To host openECA (server node)
  - (32 Core, 32 GB RAM, 1TB SSD HDD) – To host the openLSE
- In early September, the openECA was installed into several environments for testing, including the two new servers as depicted on the Deployment Architecture.
- The 16 Core server node subscribed to two GEP connections from two upstream synchrophasor data systems and is publishing to the 32 Core node and other environments (Superdome).
- openHistorian 2.0 was deployed into a subset of the servers for testing and integration of Grafana as depicted in the Deployment Architecture.
- Grafana was deployed into a subset of the servers for results and performance visualization as depicted in the Deployment Architecture.

Deployment Architecture

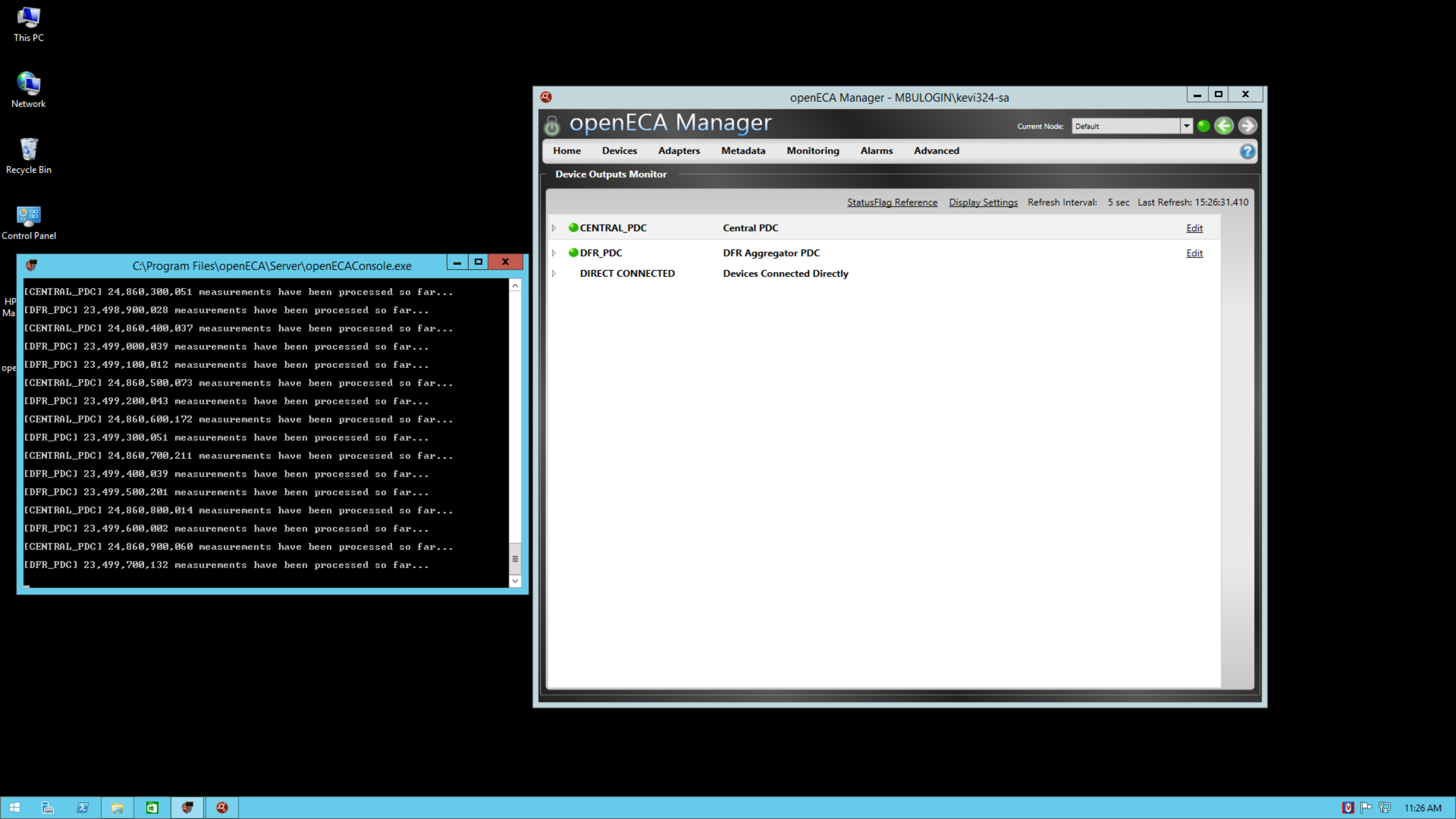


Deployment Results Report

openECA

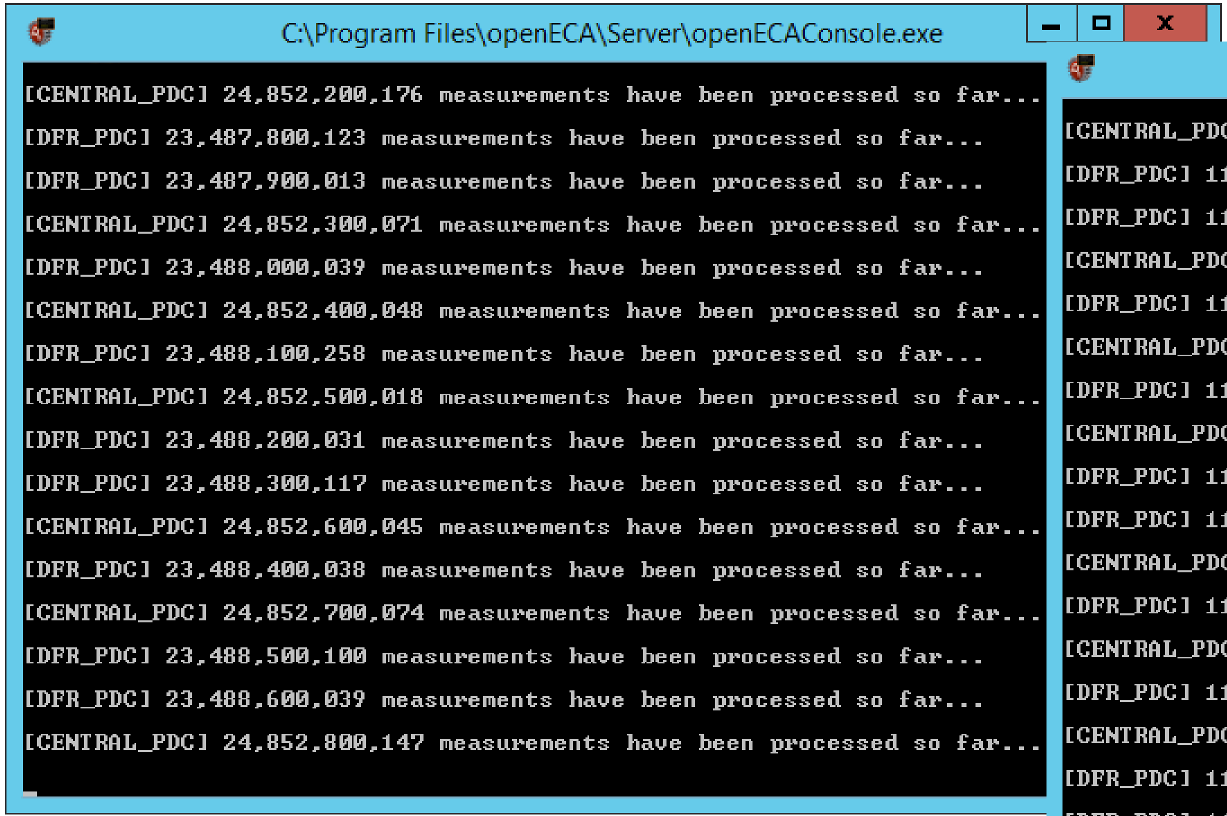
openECA Deployment Observations

- Installation was easy and caused no issues
- The two GEP subscriptions to the openECA Server node were not able to establish a connection at first. Ultimately, the subscriber device had to have compression disabled in order for the stream to work. The team suspects this has to do with compatibility of the versions of openPDC or .NET that we have running on our upstream systems.
- Besides this, the subscription process was very easy.
- The openECA was left to run for over a month with no issues.
- CPU utilization for unloaded (no analytics) openECA was negligible.

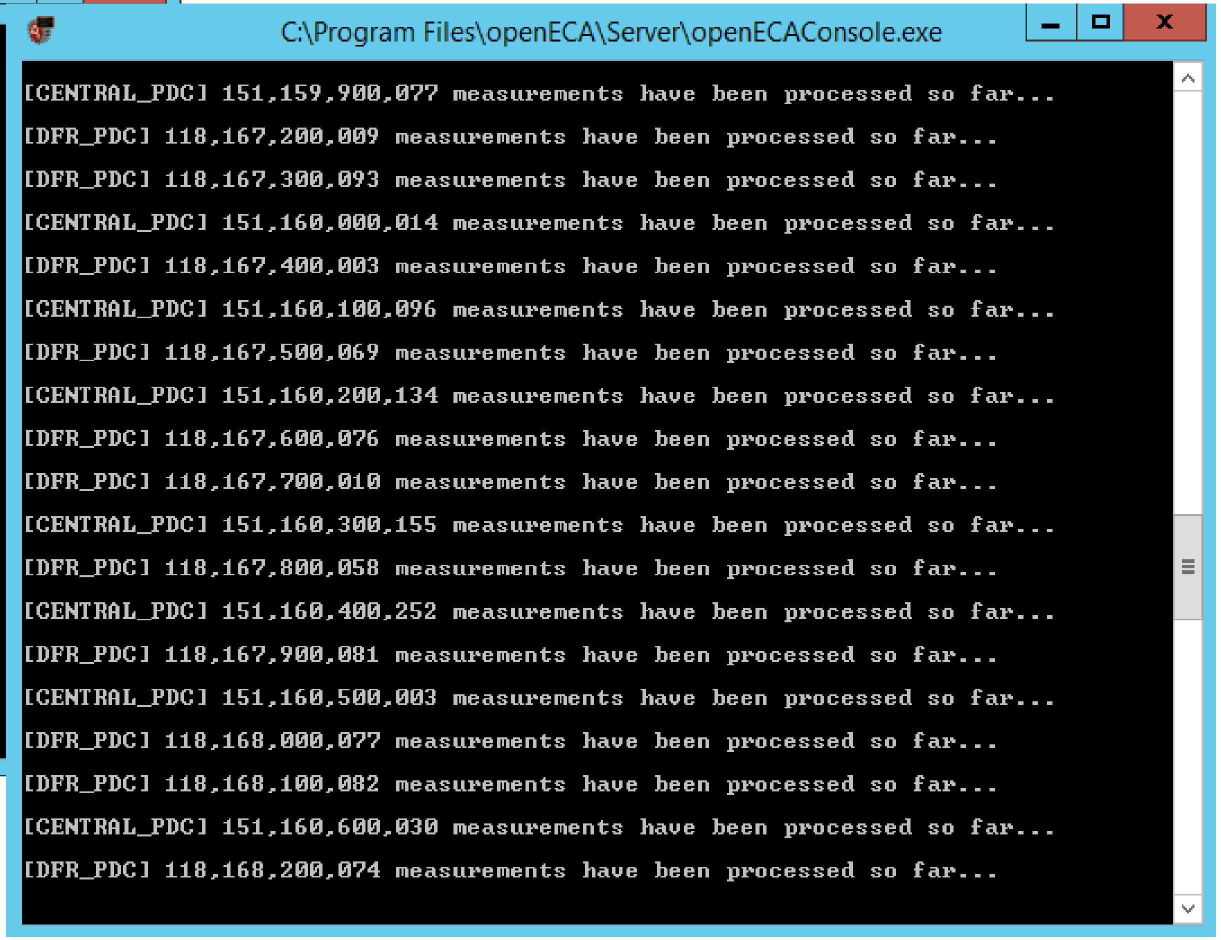

This PC
Network
Recycle Bin
Control Panel
C:\Program Files\openECA\Server\openECAConsole.exe
[CENTRAL_PDC] 24,860,300,051 measurements have been processed so far...
[DFR_PDC] 23,498,900,028 measurements have been processed so far...
[CENTRAL_PDC] 24,860,400,037 measurements have been processed so far...
[DFR_PDC] 23,499,000,039 measurements have been processed so far...
[DFR_PDC] 23,499,100,012 measurements have been processed so far...
[CENTRAL_PDC] 24,860,500,073 measurements have been processed so far...
[DFR_PDC] 23,499,200,043 measurements have been processed so far...
[CENTRAL_PDC] 24,860,600,172 measurements have been processed so far...
[DFR_PDC] 23,499,300,051 measurements have been processed so far...
[CENTRAL_PDC] 24,860,700,211 measurements have been processed so far...
[DFR_PDC] 23,499,400,039 measurements have been processed so far...
[DFR_PDC] 23,499,500,201 measurements have been processed so far...
[CENTRAL_PDC] 24,860,800,014 measurements have been processed so far...
[DFR_PDC] 23,499,600,002 measurements have been processed so far...
[CENTRAL_PDC] 24,860,900,060 measurements have been processed so far...
[DFR_PDC] 23,499,700,132 measurements have been processed so far...
openECA Manager - MBULOGIN\kevi324-sa
openECA Manager
Current Node: Default
Home
Devices
Adapters
Metadata
Monitoring
Alarms
Advanced
Device Outputs Monitor
StatusFlag Reference
Display Settings
Refresh Interval: 5 sec
Last Refresh: 15:26:31.410
CENTRAL_PDC
Central PDC
Edit
DFR_PDC
DFR Aggregator PDC
Edit
DIRECT CONNECTED
Devices Connected Directly
11:26 AM

openECA Processed Measurements

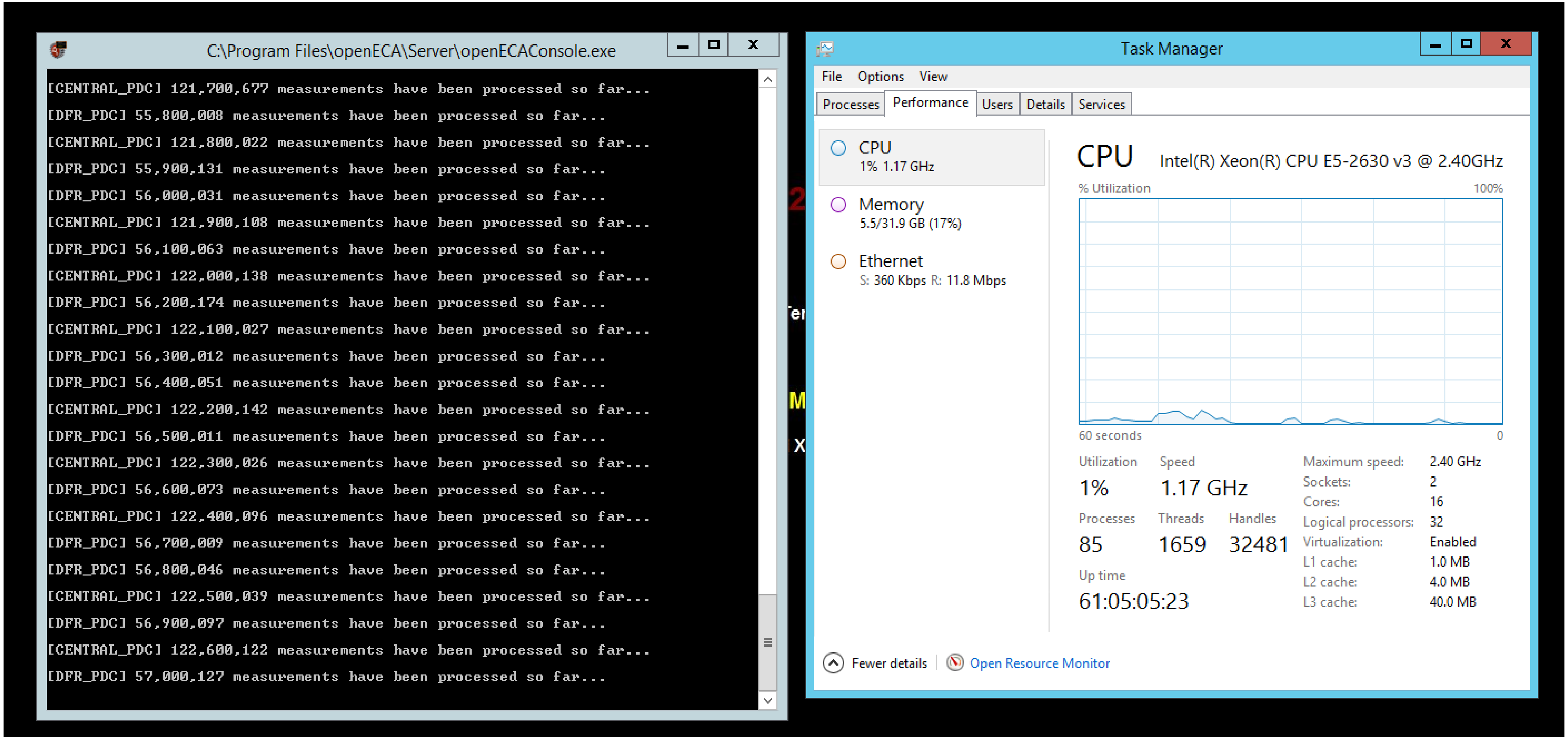
ECA2001



Superdome Test



openECA CPU Utilization

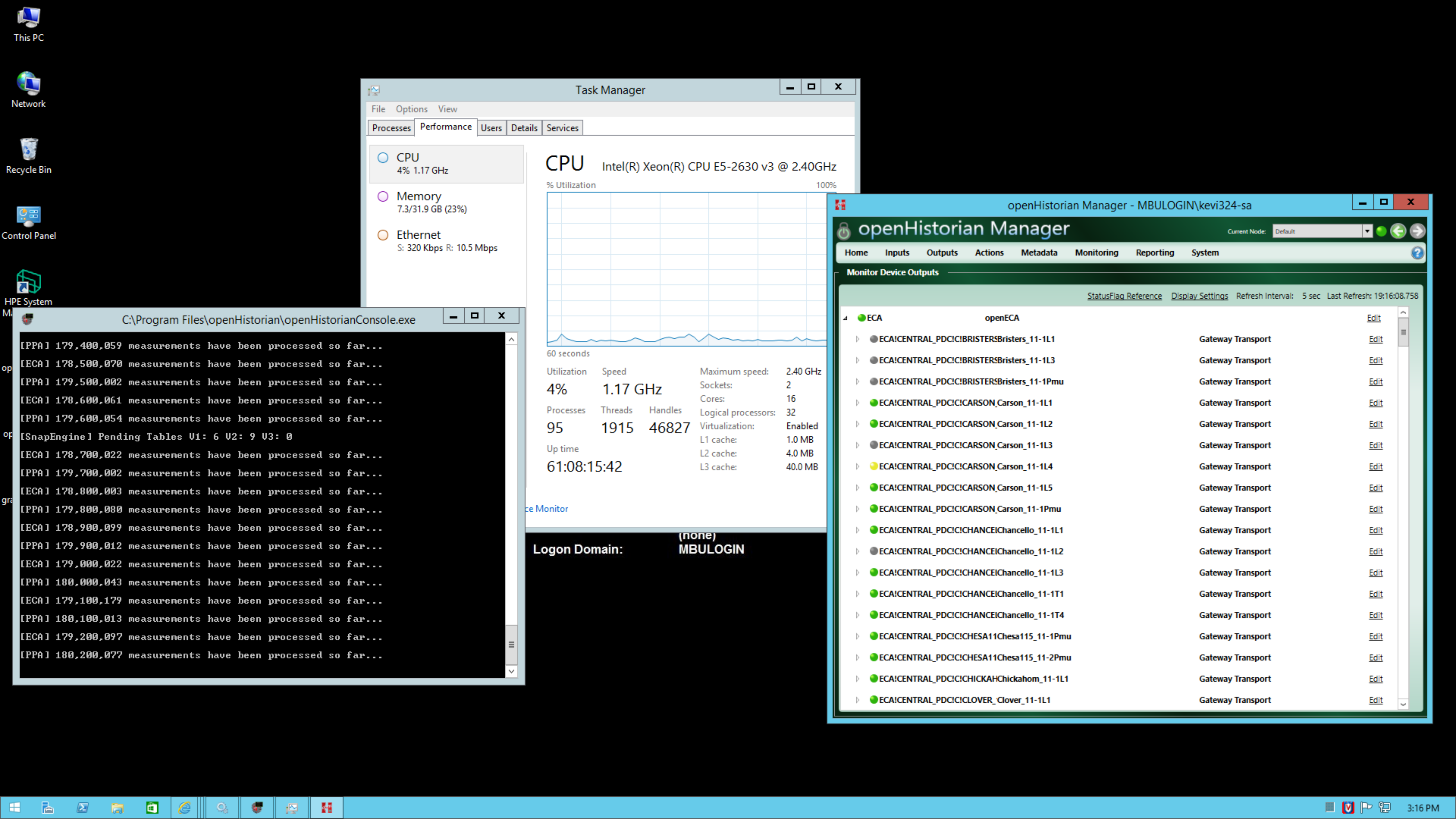


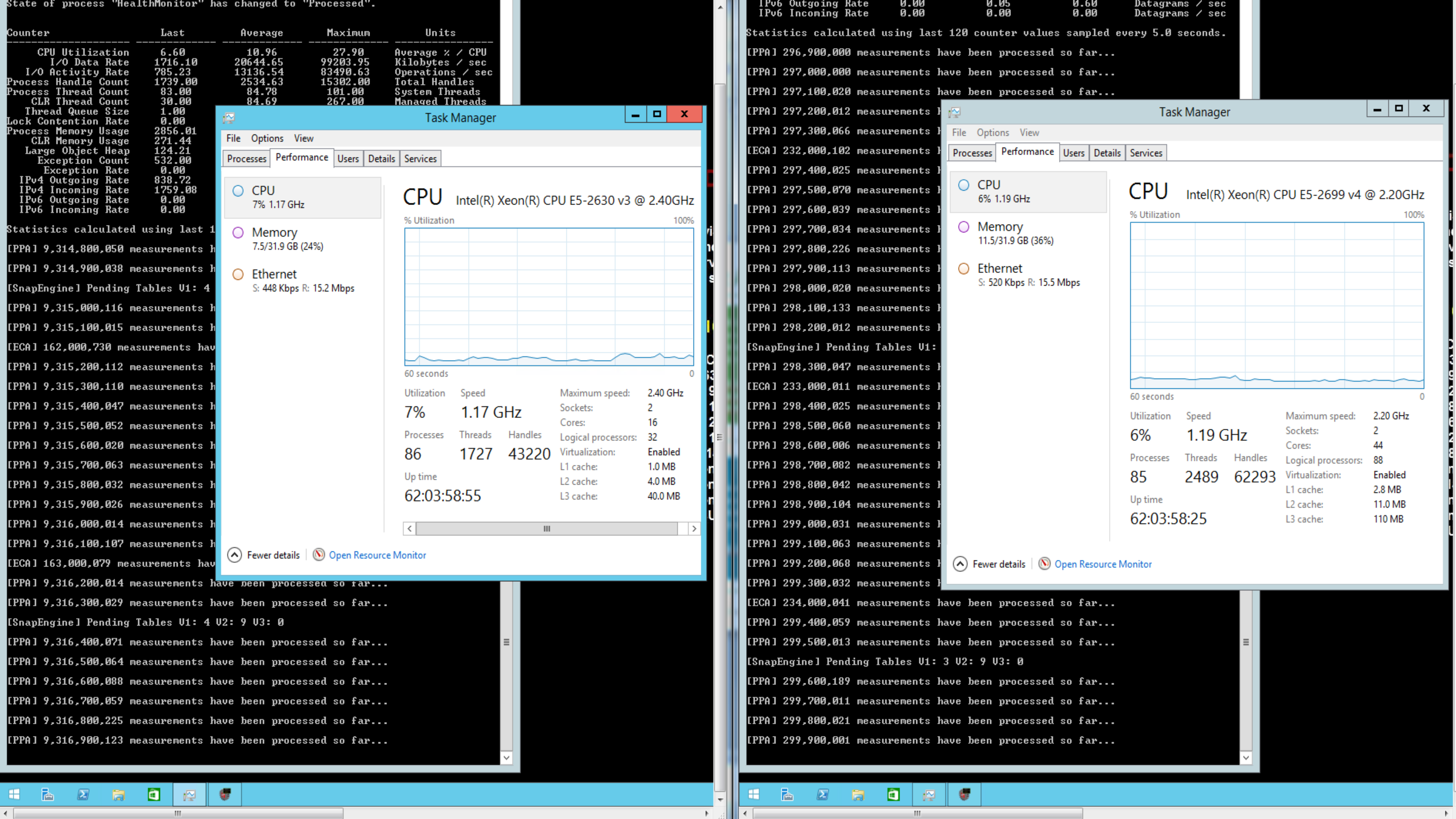
Deployment Results Report

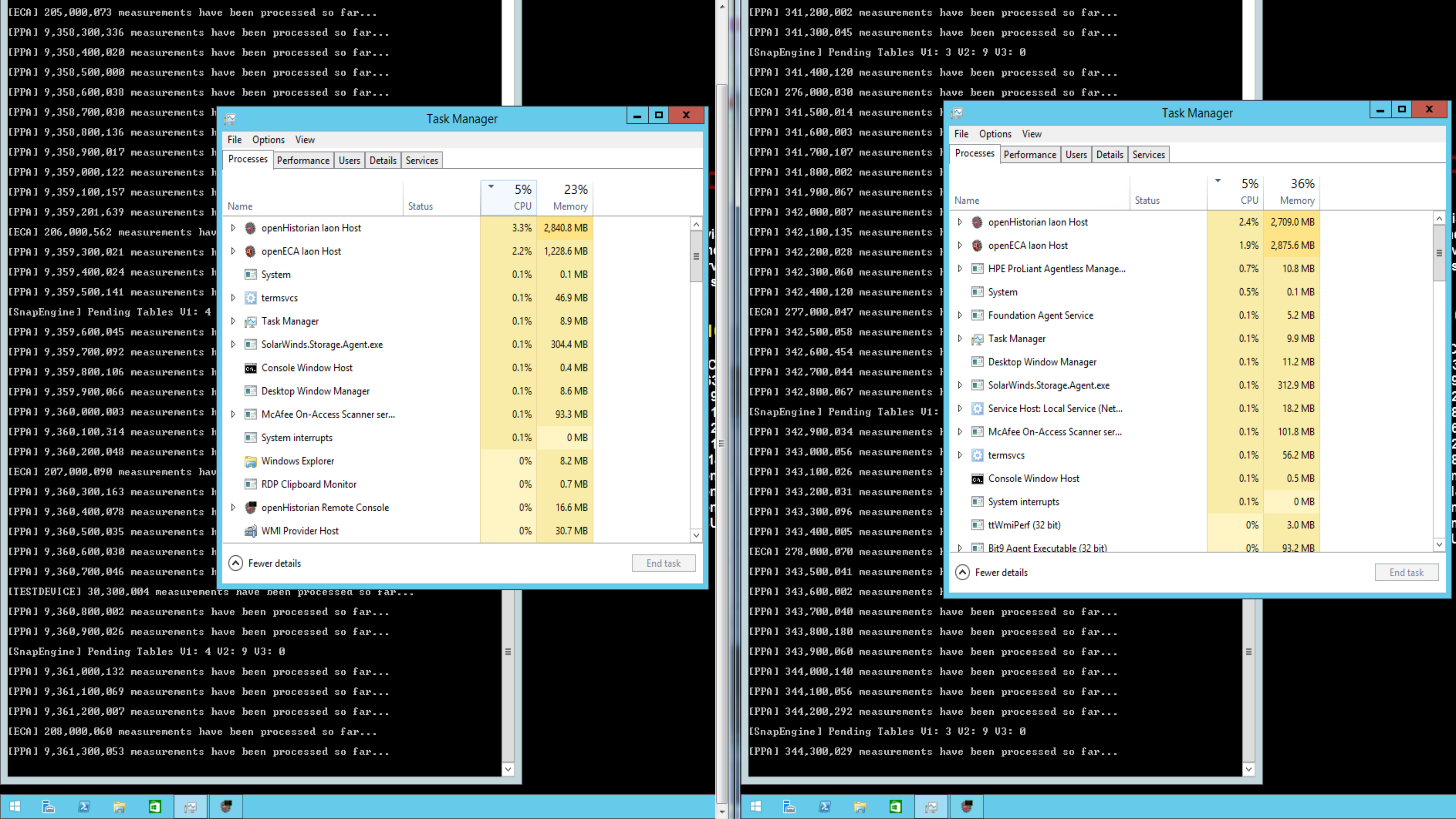
openHistorian

openHistorian Deployment Summary

- openHistorian was installed into three environments as noted on the Deployment Architecture.
- Installation was easy and caused no issues
- GEP Subscription was easy to establish. Default settings were able to connect to openECA.
- CPU Utilization for ECA2002 with openECA running in parallel
  - ~100,000 events per second
    - 3-5% average CPU utilization @ 1.15GHz (2.4GHz max)
    - 8% CPU utilization max
  - ~175,000 events per second
    - 4-6% average CPU utilization @ 1.15 GHz (2.4GHz max)
    - 12% CPU utilization max


This PC
Network
Recycle Bin
Control Panel
HPE System
Task Manager
File Options View
Processes Performance Users Details Services
CPU
4% 1.17 GHz
Memory
7.3/31.9 GB (23%)
Ethernet
S: 320 Kbps R: 10.5 Mbps
CPU Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
% Utilization 100%
60 seconds
Utilization 4%
Speed 1.17 GHz
Processes 95
Threads 1915
Handles 46827
Up time 61:08:15:42
Maximum speed: 2.40 GHz
Sockets: 2
Cores: 16
Logical processors: 32
Virtualization: Enabled
L1 cache: 1.0 MB
L2 cache: 4.0 MB
L3 cache: 40.0 MB
C:\Program Files\openHistorian\openHistorianConsole.exe
[PPA] 179,400,059 measurements have been processed so far...
[ECA] 178,500,070 measurements have been processed so far...
[PPA] 179,500,002 measurements have been processed so far...
[ECA] 178,600,061 measurements have been processed so far...
[PPA] 179,600,054 measurements have been processed so far...
[SnapEngine] Pending Tables V1: 6 V2: 9 V3: 0
[ECA] 178,700,022 measurements have been processed so far...
[PPA] 179,700,002 measurements have been processed so far...
[ECA] 178,800,003 measurements have been processed so far...
[PPA] 179,800,080 measurements have been processed so far...
[ECA] 178,900,099 measurements have been processed so far...
[PPA] 179,900,012 measurements have been processed so far...
[ECA] 179,000,022 measurements have been processed so far...
[PPA] 180,000,043 measurements have been processed so far...
[ECA] 179,100,179 measurements have been processed so far...
[PPA] 180,100,013 measurements have been processed so far...
[ECA] 179,200,097 measurements have been processed so far...
[PPA] 180,200,077 measurements have been processed so far...
Logon Domain: MBULOGIN
openHistorian Manager - MBULOGIN\kevi324-sa
openHistorian Manager
Current Node: Default
Home Inputs Outputs Actions Metadata Monitoring Reporting System
Monitor Device Outputs
StatusFlag Reference Display Settings Refresh Interval: 5 sec Last Refresh: 19:16:08.758
ECA openECA Edit
ECA!CENTRAL_PDC!C!BRISTERSBristers_11-1L1 Gateway Transport Edit
ECA!CENTRAL_PDC!C!BRISTERSBristers_11-1L3 Gateway Transport Edit
ECA!CENTRAL_PDC!C!BRISTERSBristers_11-1Pmu Gateway Transport Edit
ECA!CENTRAL_PDC!C!CARSON_Carson_11-1L1 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CARSON_Carson_11-1L2 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CARSON_Carson_11-1L3 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CARSON_Carson_11-1L4 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CARSON_Carson_11-1L5 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CARSON_Carson_11-1Pmu Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHANCEIChancello_11-1L1 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHANCEIChancello_11-1L2 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHANCEIChancello_11-1L3 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHANCEIChancello_11-1T1 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHANCEIChancello_11-1T4 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHESA11Chesa115_11-1Pmu Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHESA11Chesa115_11-2Pmu Gateway Transport Edit
ECA!CENTRAL_PDC!C!CHICKAHChickahom_11-1L1 Gateway Transport Edit
ECA!CENTRAL_PDC!C!CLOVER_Clover_11-1L1 Gateway Transport Edit
3:16 PM


State of process "HealthMonitor" has changed to "Processed".
Counter Last Average Maximum Units
CPU Utilization 6.60 10.96 27.90 Average % / CPU
I/O Data Rate 1716.10 20644.65 99203.95 Kilobytes / sec
I/O Activity Rate 785.23 13136.54 83490.63 Operations / sec
Process Handle Count 1739.00 2534.63 15302.00 Total Handles
Process Thread Count 83.00 84.78 101.00 System Threads
CLR Thread Count 30.00 84.69 267.00 Managed Threads
Thread Queue Size 1.00
Lock Contention Rate 0.00
Process Memory Usage 2856.01
CLR Memory Usage 271.44
Large Object Heap 124.21
Exception Count 532.00
Exception Rate 0.00
IPv4 Outgoing Rate 838.72
IPv4 Incoming Rate 1759.08
IPv6 Outgoing Rate 0.00
IPv6 Incoming Rate 0.00
Statistics calculated using last 1
[PPA] 9,314,800,050 measurements h
[PPA] 9,314,900,038 measurements h
[SnapEngine] Pending Tables V1: 4
[PPA] 9,315,000,116 measurements h
[PPA] 9,315,100,015 measurements h
[ECA] 162,000,730 measurements hav
[PPA] 9,315,200,112 measurements h
[PPA] 9,315,300,110 measurements h
[PPA] 9,315,400,047 measurements h
[PPA] 9,315,500,052 measurements h
[PPA] 9,315,600,020 measurements h
[PPA] 9,315,700,063 measurements h
[PPA] 9,315,800,032 measurements h
[PPA] 9,315,900,026 measurements h
[PPA] 9,316,000,014 measurements h
[PPA] 9,316,100,107 measurements h
[ECA] 163,000,079 measurements hav
[PPA] 9,316,200,014 measurements have been processed so far...
[PPA] 9,316,300,029 measurements have been processed so far...
[SnapEngine] Pending Tables V1: 4 V2: 9 V3: 0
[PPA] 9,316,400,071 measurements have been processed so far...
[PPA] 9,316,500,064 measurements have been processed so far...
[PPA] 9,316,600,088 measurements have been processed so far...
[PPA] 9,316,700,059 measurements have been processed so far...
[PPA] 9,316,800,225 measurements have been processed so far...
[PPA] 9,316,900,123 measurements have been processed so far...
Task Manager
File Options View
Processes Performance Users Details Services
CPU
7% 1.17 GHz
Memory
7.5/31.9 GB (24%)
Ethernet
S: 448 Kbps R: 15.2 Mbps
CPU Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
% Utilization 100%
60 seconds 0
Utilization Speed
7% 1.17 GHz
Processes Threads Handles
86 1727 43220
Up time
62:03:58:55
Maximum speed: 2.40 GHz
Sockets: 2
Cores: 16
Logical processors: 32
Virtualization: Enabled
L1 cache: 1.0 MB
L2 cache: 4.0 MB
L3 cache: 40.0 MB
Fewer details
Open Resource Monitor
IPv6 Outgoing Rate 0.00 0.05 0.60 Datagrams / sec
IPv6 Incoming Rate 0.00 0.00 0.00 Datagrams / sec
Statistics calculated using last 120 counter values sampled every 5.0 seconds.
[PPA] 296,900,000 measurements have been processed so far...
[PPA] 297,000,000 measurements have been processed so far...
[PPA] 297,100,020 measurements have been processed so far...
[PPA] 297,200,012 measurements
[PPA] 297,300,066 measurements
[ECA] 232,000,102 measurements
[PPA] 297,400,025 measurements
[PPA] 297,500,070 measurements
[PPA] 297,600,039 measurements
[PPA] 297,700,034 measurements
[PPA] 297,800,226 measurements
[PPA] 297,900,113 measurements
[PPA] 298,000,020 measurements
[PPA] 298,100,133 measurements
[PPA] 298,200,012 measurements
[SnapEngine] Pending Tables V1:
[PPA] 298,300,047 measurements
[ECA] 233,000,011 measurements
[PPA] 298,400,025 measurements
[PPA] 298,500,060 measurements
[PPA] 298,600,006 measurements
[PPA] 298,700,082 measurements
[PPA] 298,800,042 measurements
[PPA] 298,900,104 measurements
[PPA] 299,000,031 measurements
[PPA] 299,100,063 measurements
[PPA] 299,200,068 measurements
[PPA] 299,300,032 measurements
[ECA] 234,000,041 measurements have been processed so far...
[PPA] 299,400,059 measurements have been processed so far...
[PPA] 299,500,013 measurements have been processed so far...
[SnapEngine] Pending Tables V1: 3 V2: 9 V3: 0
[PPA] 299,600,189 measurements have been processed so far...
[PPA] 299,700,011 measurements have been processed so far...
[PPA] 299,800,021 measurements have been processed so far...
[PPA] 299,900,001 measurements have been processed so far...
Task Manager
File Options View
Processes Performance Users Details Services
CPU
6% 1.19 GHz
Memory
11.5/31.9 GB (36%)
Ethernet
S: 520 Kbps R: 15.5 Mbps
CPU Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
% Utilization 100%
60 seconds 0
Utilization Speed
6% 1.19 GHz
Processes Threads Handles
85 2489 62293
Up time
62:03:58:25
Maximum speed: 2.20 GHz
Sockets: 2
Cores: 44
Logical processors: 88
Virtualization: Enabled
L1 cache: 2.8 MB
L2 cache: 11.0 MB
L3 cache: 110 MB
Fewer details
Open Resource Monitor


[ECA] 205,000,073 measurements have been processed so far...
[PPA] 9,358,300,336 measurements have been processed so far...
[PPA] 9,358,400,020 measurements have been processed so far...
[PPA] 9,358,500,000 measurements have been processed so far...
[PPA] 9,358,600,038 measurements have been processed so far...
[PPA] 9,358,700,030 measurements h
[PPA] 9,358,800,136 measurements h
[PPA] 9,358,900,017 measurements h
[PPA] 9,359,000,122 measurements h
[PPA] 9,359,100,157 measurements h
[PPA] 9,359,201,639 measurements h
[ECA] 206,000,562 measurements hav
[PPA] 9,359,300,021 measurements h
[PPA] 9,359,400,024 measurements h
[PPA] 9,359,500,141 measurements h
[SnapEngine] Pending Tables V1: 4
[PPA] 9,359,600,045 measurements h
[PPA] 9,359,700,092 measurements h
[PPA] 9,359,800,106 measurements h
[PPA] 9,359,900,066 measurements h
[PPA] 9,360,000,003 measurements h
[PPA] 9,360,100,314 measurements h
[PPA] 9,360,200,048 measurements h
[ECA] 207,000,090 measurements hav
[PPA] 9,360,300,163 measurements h
[PPA] 9,360,400,078 measurements h
[PPA] 9,360,500,035 measurements h
[PPA] 9,360,600,030 measurements h
[PPA] 9,360,700,046 measurements h
[TESTDEVICE] 30,300,004 measurements have been processed so far...
[PPA] 9,360,800,002 measurements have been processed so far...
[PPA] 9,360,900,026 measurements have been processed so far...
[SnapEngine] Pending Tables V1: 4 V2: 9 V3: 0
[PPA] 9,361,000,132 measurements have been processed so far...
[PPA] 9,361,100,069 measurements have been processed so far...
[PPA] 9,361,200,007 measurements have been processed so far...
[ECA] 208,000,060 measurements have been processed so far...
[PPA] 9,361,300,053 measurements have been processed so far...
Task Manager
File Options View
Processes Performance Users Details Services
Name Status 5% CPU 23% Memory
openHistorian Iaon Host 3.3% 2,840.8 MB
openECA Iaon Host 2.2% 1,228.6 MB
System 0.1% 0.1 MB
termsvcs 0.1% 46.9 MB
Task Manager 0.1% 8.9 MB
SolarWinds.Storage.Agent.exe 0.1% 304.4 MB
Console Window Host 0.1% 0.4 MB
Desktop Window Manager 0.1% 8.6 MB
McAfee On-Access Scanner ser... 0.1% 93.3 MB
System interrupts 0.1% 0 MB
Windows Explorer 0% 8.2 MB
RDP Clipboard Monitor 0% 0.7 MB
openHistorian Remote Console 0% 16.6 MB
WMI Provider Host 0% 30.7 MB
Fewer details
End task
[PPA] 341,200,002 measurements have been processed so far...
[PPA] 341,300,045 measurements have been processed so far...
[SnapEngine] Pending Tables V1: 3 V2: 9 V3: 0
[PPA] 341,400,120 measurements have been processed so far...
[ECA] 276,000,030 measurements have been processed so far...
[PPA] 341,500,014 measurements
[PPA] 341,600,003 measurements
[PPA] 341,700,107 measurements
[PPA] 341,800,002 measurements
[PPA] 341,900,067 measurements
[PPA] 342,000,087 measurements
[PPA] 342,100,135 measurements
[PPA] 342,200,028 measurements
[PPA] 342,300,060 measurements
[PPA] 342,400,120 measurements
[ECA] 277,000,047 measurements
[PPA] 342,500,058 measurements
[PPA] 342,600,454 measurements
[PPA] 342,700,044 measurements
[PPA] 342,800,067 measurements
[SnapEngine] Pending Tables V1:
[PPA] 342,900,034 measurements
[PPA] 343,000,056 measurements
[PPA] 343,100,026 measurements
[PPA] 343,200,031 measurements
[PPA] 343,300,096 measurements
[PPA] 343,400,005 measurements
[ECA] 278,000,070 measurements
[PPA] 343,500,041 measurements
[PPA] 343,600,002 measurements
[PPA] 343,700,040 measurements have been processed so far...
[PPA] 343,800,180 measurements have been processed so far...
[PPA] 343,900,060 measurements have been processed so far...
[PPA] 344,000,140 measurements have been processed so far...
[PPA] 344,100,056 measurements have been processed so far...
[PPA] 344,200,292 measurements have been processed so far...
[SnapEngine] Pending Tables V1: 3 V2: 9 V3: 0
[PPA] 344,300,029 measurements have been processed so far...
Task Manager
File Options View
Processes Performance Users Details Services
Name Status 5% CPU 36% Memory
openHistorian Iaon Host 2.4% 2,709.0 MB
openECA Iaon Host 1.9% 2,875.6 MB
HPE ProLiant Agentless Manage... 0.7% 10.8 MB
System 0.5% 0.1 MB
Foundation Agent Service 0.1% 5.2 MB
Task Manager 0.1% 9.9 MB
Desktop Window Manager 0.1% 11.2 MB
SolarWinds.Storage.Agent.exe 0.1% 312.9 MB
Service Host: Local Service (Net... 0.1% 18.2 MB
McAfee On-Access Scanner ser... 0.1% 101.8 MB
termsvcs 0.1% 56.2 MB
Console Window Host 0.1% 0.5 MB
System interrupts 0.1% 0 MB
ttWmiPerf (32 bit) 0% 3.0 MB
Bit9 Agent Executable (32 bit) 0% 93.2 MB
Fewer details
End task

Grafana Integration

Grafana Deployment Summary

- Grafana was installed in two environments as depicted in the Deployment Architecture.
- Grafana installation and (particularly) configuration required the use of documentation provided by VT graduate students but an easy to find walkthrough should be available on the openECA and openHistorian Wiki pages. Contact Kevin Jones for more information.
- Once properly configured, the use of Grafana for quick dashboard prototyping is very easy.
- It should be noted that it is best to know in advance which points you want to display because the drop down menu is very difficult to navigate for individual points.
- Charts and graphs are better made with robust FILTER statements.
- GPA may also consider a sample dashboard that can be imported to monitor openECA, openHistorian, openPDC, etc. performance including a demo dashboard for the sample data in TESTDEVICE.

The Demonstration Phase

- Topology Estimation
- Linear State Estimator

- Dynamic PMU Transducer Calibration
- Line Parameter Estimation

- Local VAR Control
- Region VAR Control
- (Shadow System Simulator – One frame per second)

- PMU Synchroscope

Analytics

Linear State Estimator

Bonneville
POWER ADMINISTRATION

openECA Tests

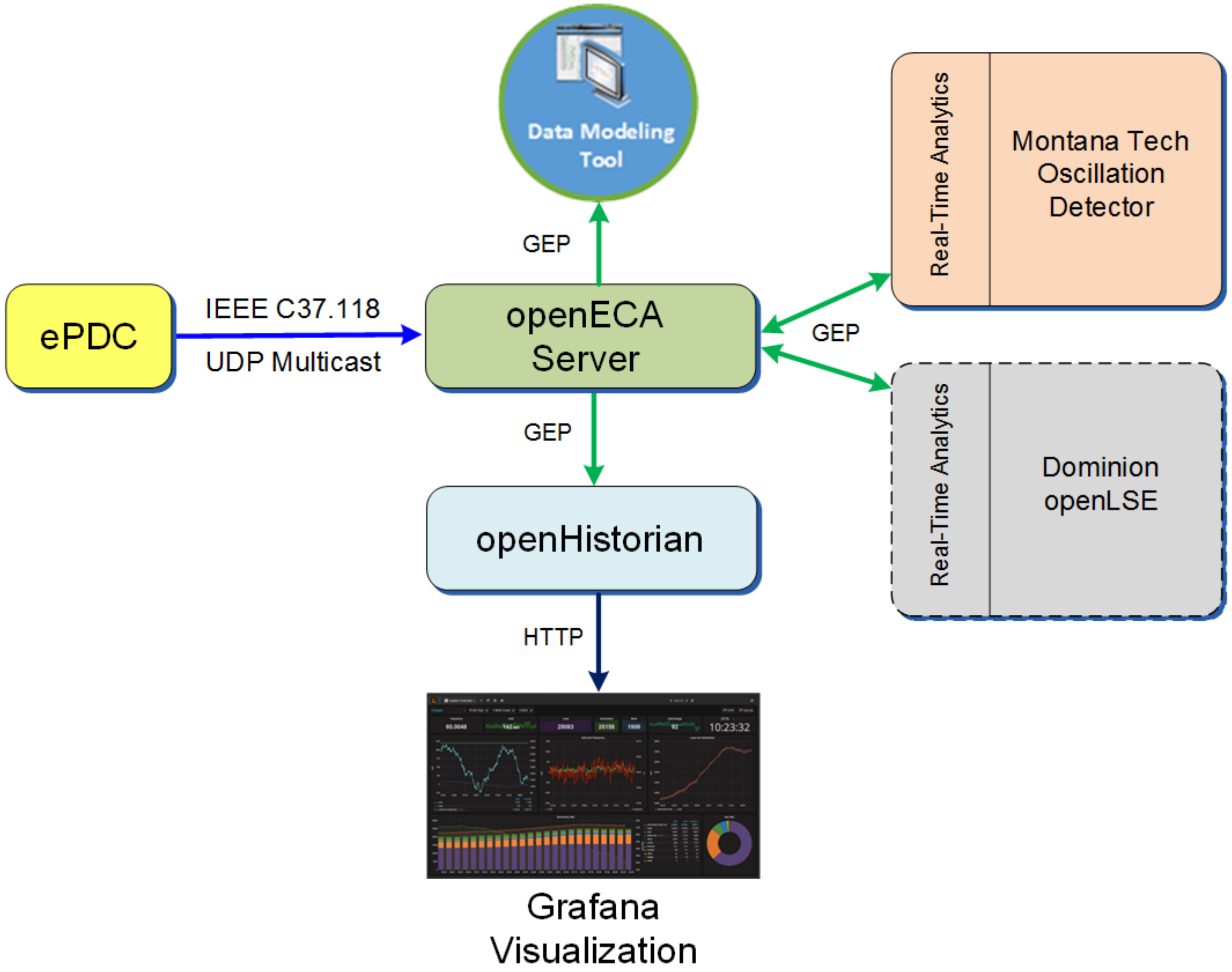
phasor data open ECA platform

Ritchie Carroll
November 7, 2017

Installation overview

- BPA PMU Data Flows in from the substations and is concentrated into a UDP Multicast stream in the BPA synchrophasor lab using ePDC
- The openECA Server subscribes to the multicast stream to receive PMU data
- The PMU data is then distributed using the Gateway Exchange Protocol to the openECA Client Data Modeling Tool (as needed), the openHistorian for short-term data archival as well as to the installed analytics
- Data from the historian is served up to the Grafana visualization tool over HTTP

BPA openECA Demo Data Flow Diagram



Platform Observations

- Platform Installation and Configuration Observations
  - After some issues getting the software installed (see Issues List below), the openECA server was able to quickly connect to the available BPA multicast synchrophasor data feed and provide data to the openHistorian and Grafana visualizations.
- Issues List
  - Installation on Windows Server 2008 was problematic due to security issues. The openECA system was able to be installed by manually disabling security, however, it is expected that if the operating system was brought to up to current patch levels that security could be reenabled. GPA noted that this had not been observed on any other test system installations to date.
  - Very old version of Internet Explorer was unable to properly render HTML 5 based web pages. We were able to move another machine and effectively access the openECA server with more recent browser technology.
  - Reinstalled OpenECA on Windows Server 2012 with no security issues. The software is stable, and web pages are accessible with updated browser. All further testing will be performed on this server.
- Stability Observations
  - The system will be left running and observed for stability for the remainder of the project. No stability issues have been noted thus far.
- Platform Component Observations
  - Server components
    - The server components seemed to pick up the BPA data feed well and quickly reported data anomalies where some of the PMU devices were offline.
  - Displaying data/results
    - The openECA Manager, openHistorian and Grafana were all tested – each system was able to display both real-time data and calculated results, e.g., MW values calculated by openECA server.
  - Creating a project (optional)
    - BPA will attempt to implement a simple proof-of-concept MATLAB-based project, and assess potntial of further projects using MATLAB.

Montana Tech Oscillation Detector Observations

- Analytic Installation and Configuration Observations
  - Installation was very quick. Once installed the provided configuration tool was used to select several Megawatt values and generate new signal mappings for the OD service. This generated a file which we copied to the "Model" folder where the OD service was installed. We then ran the OD Service Console and invoked the "Initialize" command which promptly cycled the OD analytic with the new mappings. Overall, this took very little time and was easy to configure.
- Issues List
  - None noted.
- Stability
  - The system will be left running and observed for stability for the remainder of the project. No stability issues have been noted thus far.
- Results display / presentation
  - We imported a sample Grafana dashboard provided with the installation and were able to quickly visualize results coming from the analytic. The results seemed reasonable for the input.

openLSE Observations

- Issues related to the Windows 2008 host system prevented proper installation of openLSE. BPA hopes to test this in the coming weeks on the Windows 2012 machine, starting small with a couple of substations – Kevin has offered to assist.
- Analytic Installation and Configuration Observations
- Issues List
- Stability
- Results display / presentation

Demo Summary

- Considering standing up a Grafana based user interface for openECA provided oscillation results for comparison to the production system that is currently operational
- We will continue to evaluate the openLSE and compare its results to other LSE software currently use
- Grafana and the openHistorian will be evaluated for other potential uses within BPA. Flexibility in adding displays/content a positive.
- BPA will continue to experiment with openECA generated Matlab project templates for bringing synchrophasor into Matlab and sending results back to openECA
- Detailed stream statistics provide real-time detail not available with other PDCs. These statistics will be compared with daily/weekly reports compiled by BPA.

Southwest Power Pool

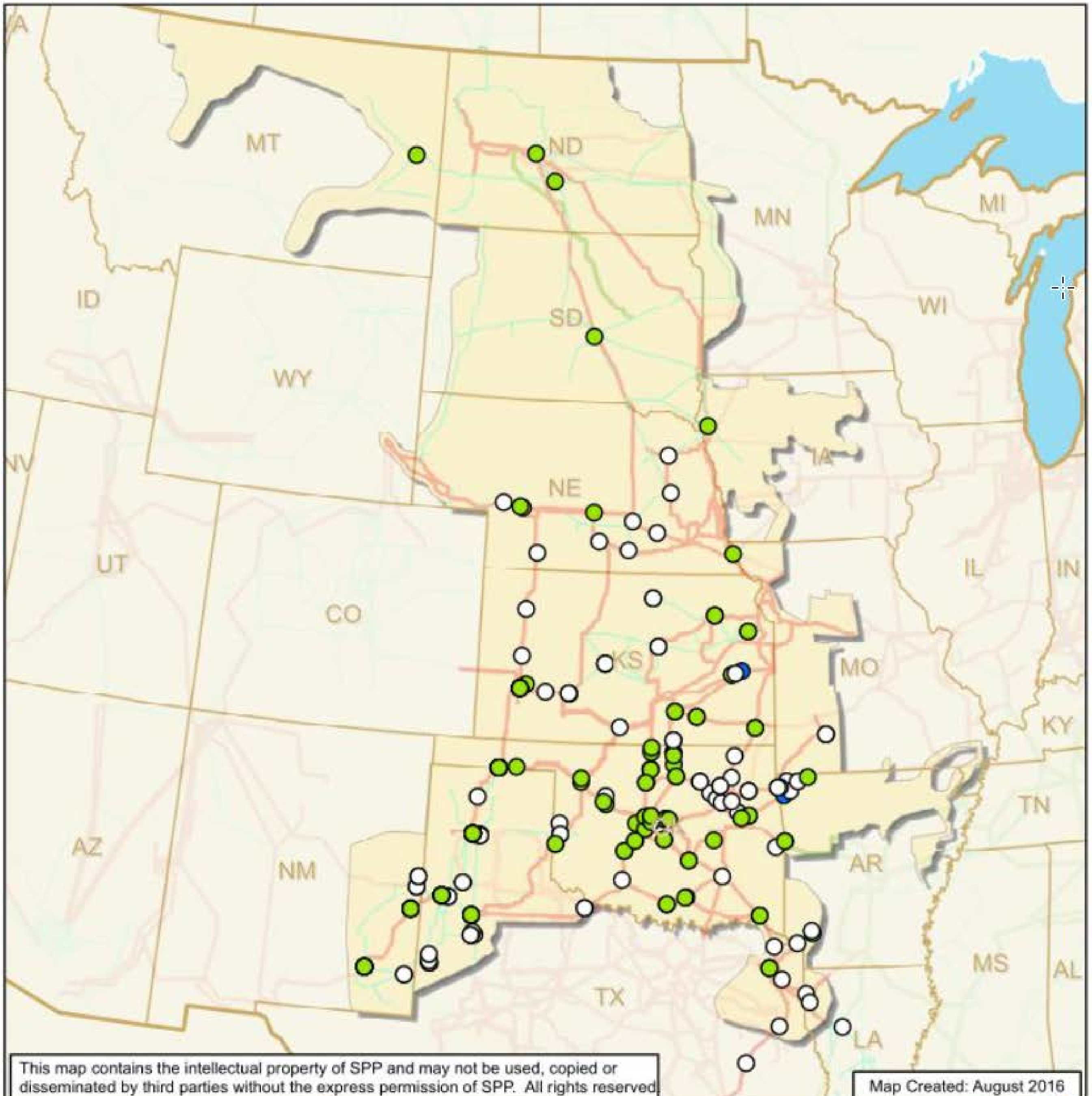
SPP openECA Beta Demonstration

Results of Testing Oscillation Detection and Linear State Estimator Analytics

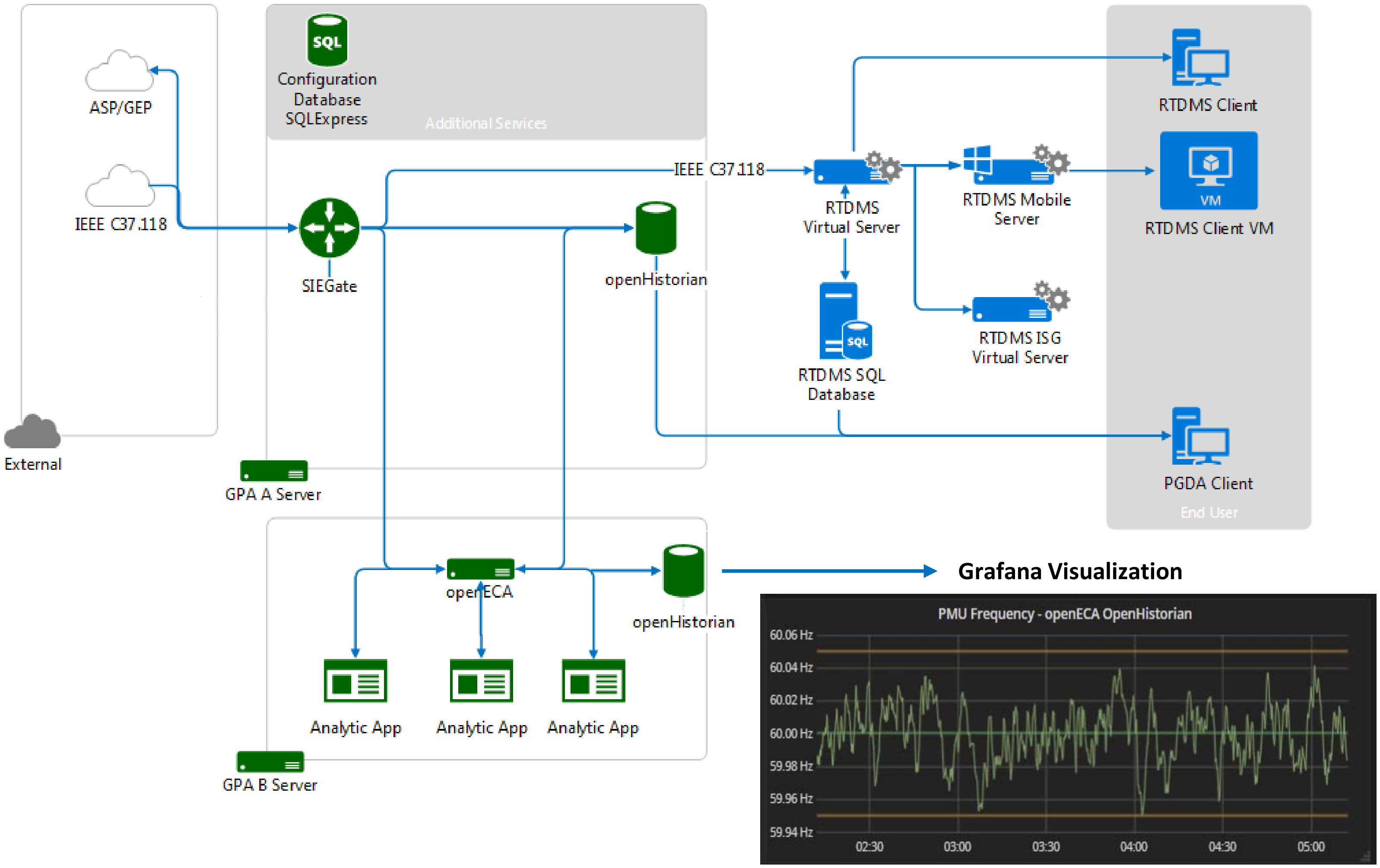
SPP PMU Project Overview

- Project began in early 2016
- Tested EPG and GPA tools in a development environment for over a year
- Plan to deploy RTDMS as an informational system for Shift Engineers this year
- Beta-testing openECA analytics in a development environment. Possibly looking to integrate with existing tools in the future.

Existing/Potential PMU Locations



SPP Development Environment System



SPP

open ECA

open and Extensible
Control & Analytics platform
for synchrophasor data

DOE FOA 970
DE-OE0000778

openECA Testing Results

SPP

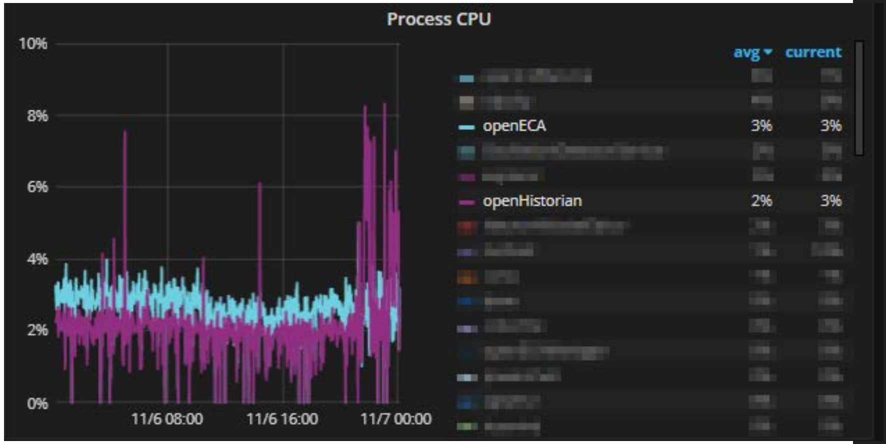
OpenECA Platform Observations

- Installation and Configuration
  - As with all GPA tools, installation and data flow configuration was painless
  - Attaching a dedicated Historian was straightforward
- User Interface
  - The openECA Manager thick client has a lot of functionality, but moving to web-based is the right direction
  - Bulk operations have to be done in the database
  - Power Calculations are difficult to maintain
- Analytics Integration
  - Both analytics SPP tested were easily able to connect to openECA, receive metadata and real-time data
  - Process of creating input/output measurement mappings is a challenge. Mappings are static, not easily maintainable as models and device availability changes. Both analytic developers tried to streamline.

SPP

openECA Test Results

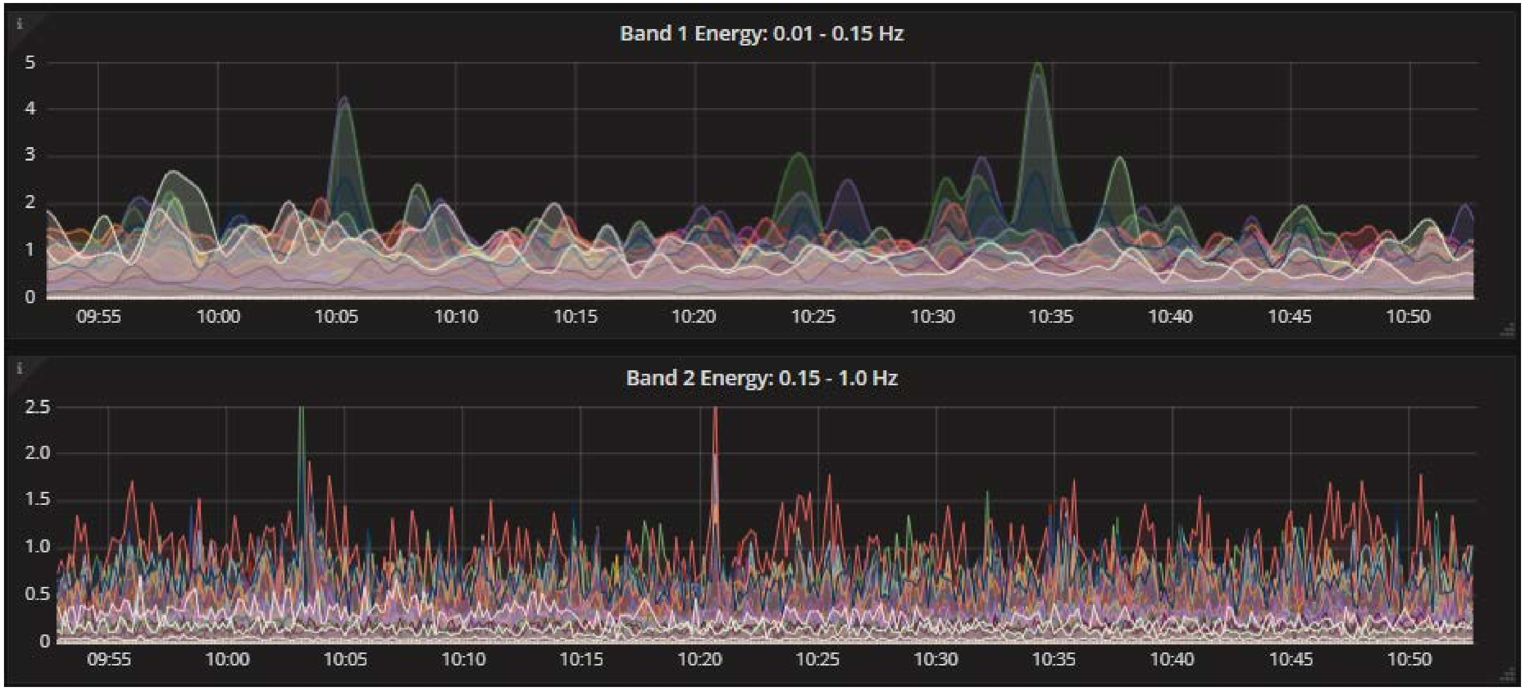
- Server Specs
  - 128GB RAM
  - Intel Xeon 16 core 3.5GHz
  - Windows Server 2012 R2
- Performance
  - 1600 PMU signals, 445 power calculations, 13000 LSE outputs
  - Avg. 3% CPU Utilization
  - Avg. ~2GB RAM Usage
- Stability
  - After initial setup, 100% uptime
  - openECA web client periodically loses connection and is slow with large model (LSE)



OpenECA Suggested Improvements

- Support of additional metadata fields for devices and signals, specifically for LSE:
  - Substation
  - Voltage Level
  - Measured Device Type
  - Measured Device Name
  - Opposite Substation (for lines)
- Continue web-based manager development for all the GPA platforms

Oscillation Detection Monitor (ODM) Testing Results

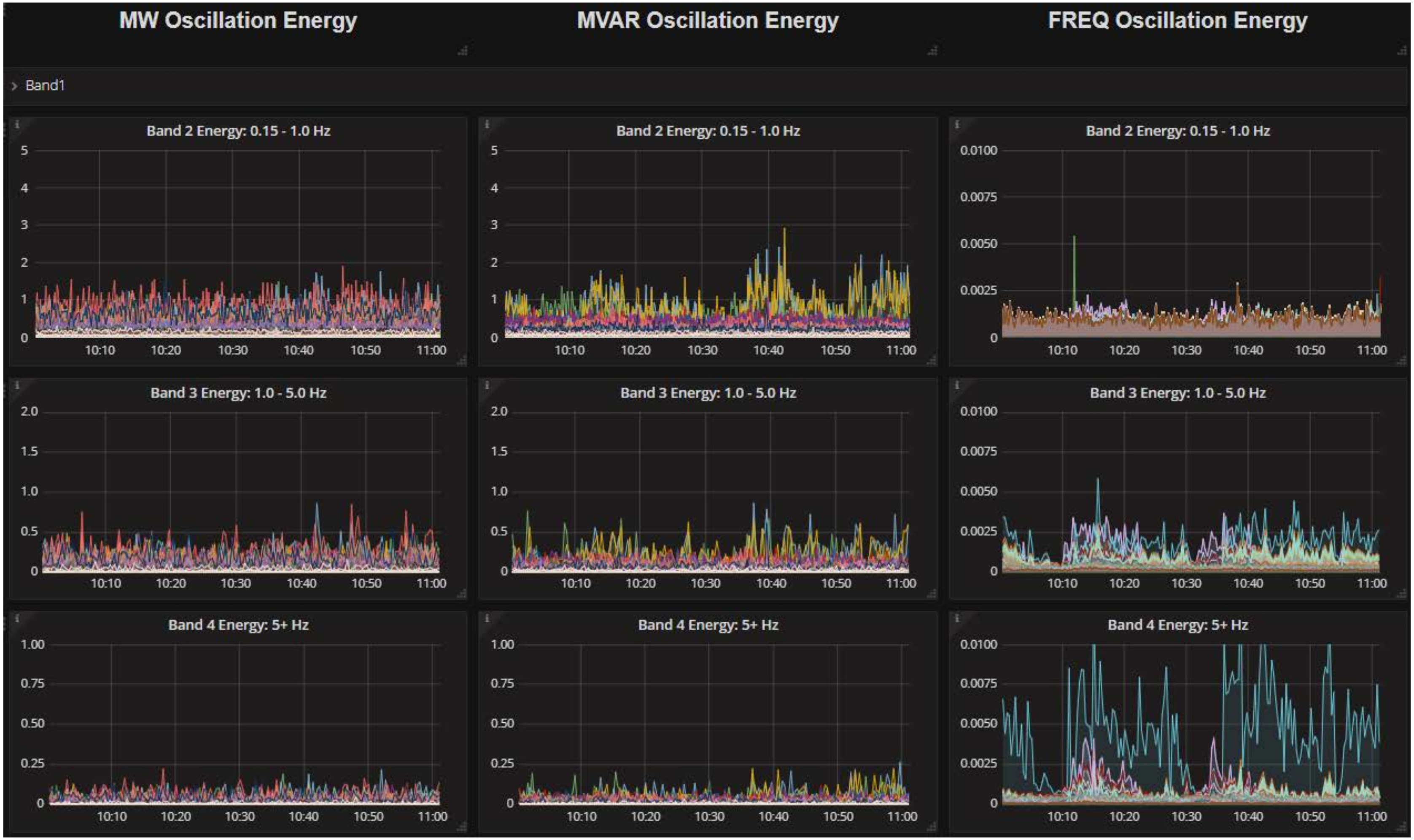


Oscillation Detection Monitor (ODM) Test Approach

- Create power calculations for all PMUs in openECA database
- Set up ODM monitoring of MW, MVAR, and Frequency signals.
- Create dashboards in Grafana to visualize results
- Validate ODM results against other tools (RTDMS and offline results)
- Monitor analytic stability and performance

SPP

Oscillation Detection Monitor (ODM) Test Displays



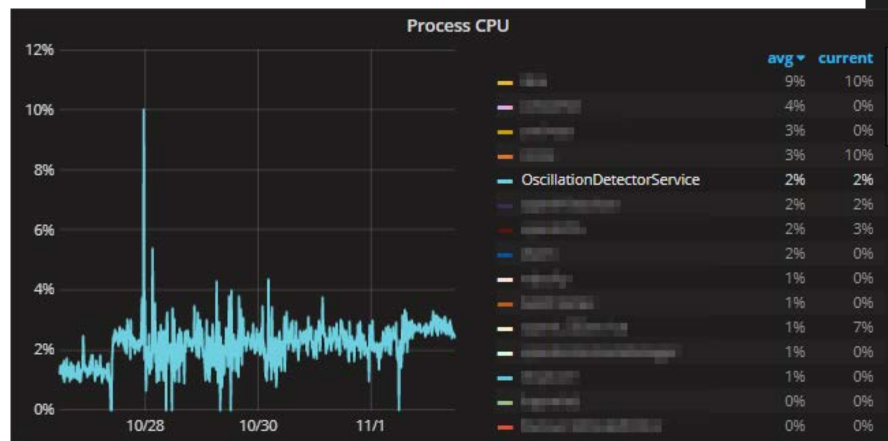
Oscillation Detection Monitor (ODM) Observations

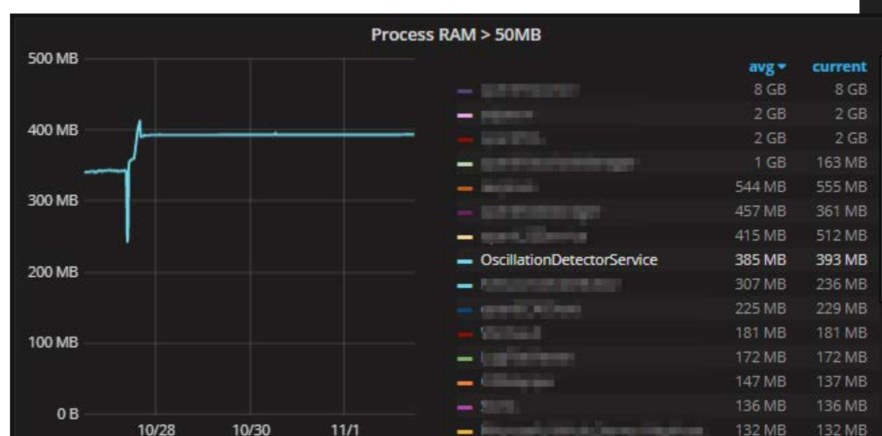
- ODM-reported Percent Bad Data is always > 0%. For most PMUs, closer to 30-50%. The data appears good on the input side.
- Easy to visually recognize trends in large amounts of data with ODM + openHistorian + Grafana
- ODM is flexible.. accepts and processes signals of any type
- Adding measurements a bit tedious at first. We added a filter to the ODMapper datagrid to streamline signal selection
- Dr. Donnelly and Ritchie were very helpful with troubleshooting and support

SPP

Oscillation Detection Monitor (ODM) Test Results

- Performance
  - 53 MW, 53 MVAR, 156 Freq Inputs
  - Running 1 frame per second
  - Avg. 2% CPU Utilization
  - Avg. ~400MB RAM Usage
- Stability
  - After initial setup, 100% uptime
- Accuracy of Results
  - Calculations verified through after-the-fact analysis
  - Calculations match up with RTDMS results (uses same backend)





SPP

Oscillation Detection Monitor (ODM) Suggested Improvements

- Built-in time-based alarming (e.g. >5MW oscillation for >30 seconds)
- An output signal indicating frequency of largest oscillation present in the band for each PMU
- Configurable frequency ranges for each band

openLSE Linear State Estimator Testing Results

Linear State Estimator (LSE) Test Approach

- Create LSE model based on SPP's GE/Alstom EMS Model
- Create mappings between PMU devices/signals and EMS equipment, including CBs, LNs, and XFMRs
- Import model into offline model tool, prune/reduce based on measurement availability, share model information with OG&E for their testing
- Get model solving in offline LSE environment with PMU snapshot data
- Get model solving in real-time with LSE service
- Document LSE's enhanced observability
- Monitor estimated residuals, compare estimated values with SCADA or EMS State Estimator

SPP

Linear State Estimator (LSE) Model

- OG&E has ~600 substations in EMS
- Live data available from 152 PMUs
- 55 Directly-Observed substation-KV buses in LSE
- 33 Indirectly-Observed substation-KV buses in LSE
- LSE solves with 9 islands
  - Island 1:
    - 208 substations
    - 41 observable
  - Islands 2-9:
    - 1-3 substations each

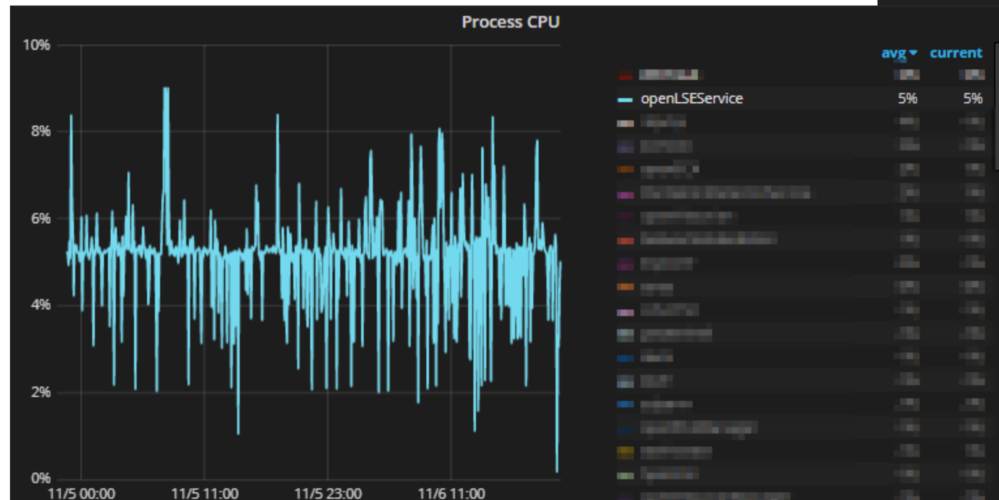
Linear State Estimator (LSE) Observations

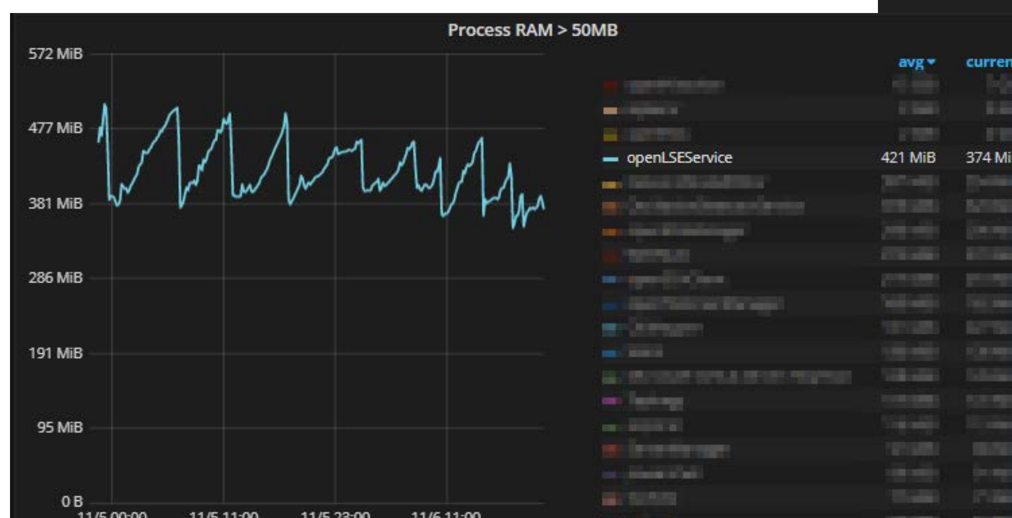
- SPP and OGE had several iterations through the network model build process
- openLSE is still evolving.. Dr. Jones worked hard to add functionality to assist SPP and OGE during testing
- Large time investment required to map PMU info to model
- LSE is very dependent on accurate measurement mapping
- With latest improvements, a ground-up model build should be easy

SPP

Linear State Estimator (LSE) Test Results

- Performance
  - Running 5 frames per second
  - Avg. 5% CPU Utilization
  - Avg. ~420MB RAM Usage
  - Occasional CPU spikes when topology is rebuilt
- Stability
  - After initial setup, 100% uptime





Linear State Estimator (LSE) Accuracy

- At most locations, LSE tracks PMU measurements very well.
- Additional mapping validation is needed in certain locations

Linear State Estimator (LSE) Suggested Improvements

- Support clean up of unused LSE output measurements
  - Each station has many unused node voltage groups
- Add a Network Topology Viewer
  - A user interface with a graphical layout of the topology would make the LSE results much easier to use.
  - Could use node-graph layout to visualize topology from the network model. Could bring in measurements and estimates from openECA.
- Add support for measurement residual streaming to openECA
  - Useful for quickly identifying mismatches

Demonstration Summary

- SPP sees great value in openECA and the potential for new analytics. We hope that the framework continues to grow and be adopted by industry.
- SPP will continue to run ODM and LSE in a testing environment.
- Plan to test Mode Meter (OMM) and Line Parameter Estimation as time permits.

SPP

SPP Potential Future Uses for openECA

- Incorporate Grafana-based PMU data with existing SPP dashboard views
- In the future, use LSE to expand PMU visibility
- Integrate ODM results and potential alerts with existing geospatial tools

SPP

phasor data open ECA platform

OG&E Demonstration Results

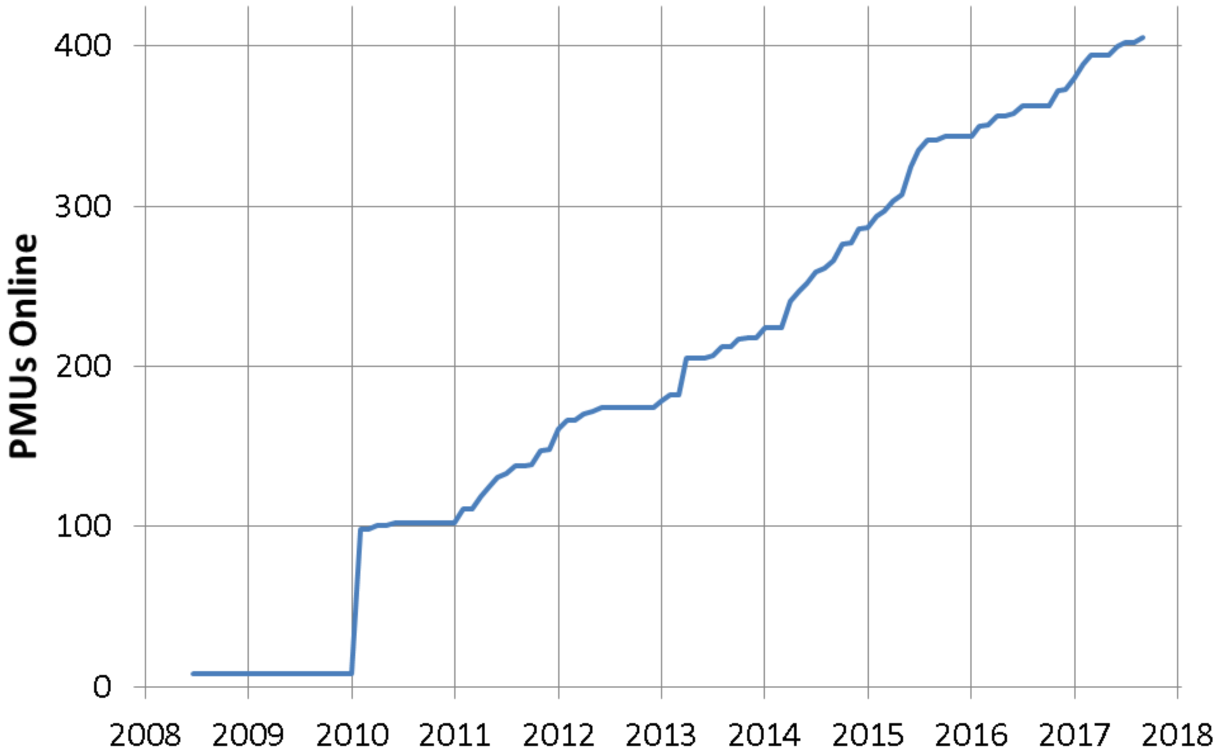
Nov 7, 2017

DOE FOA 970

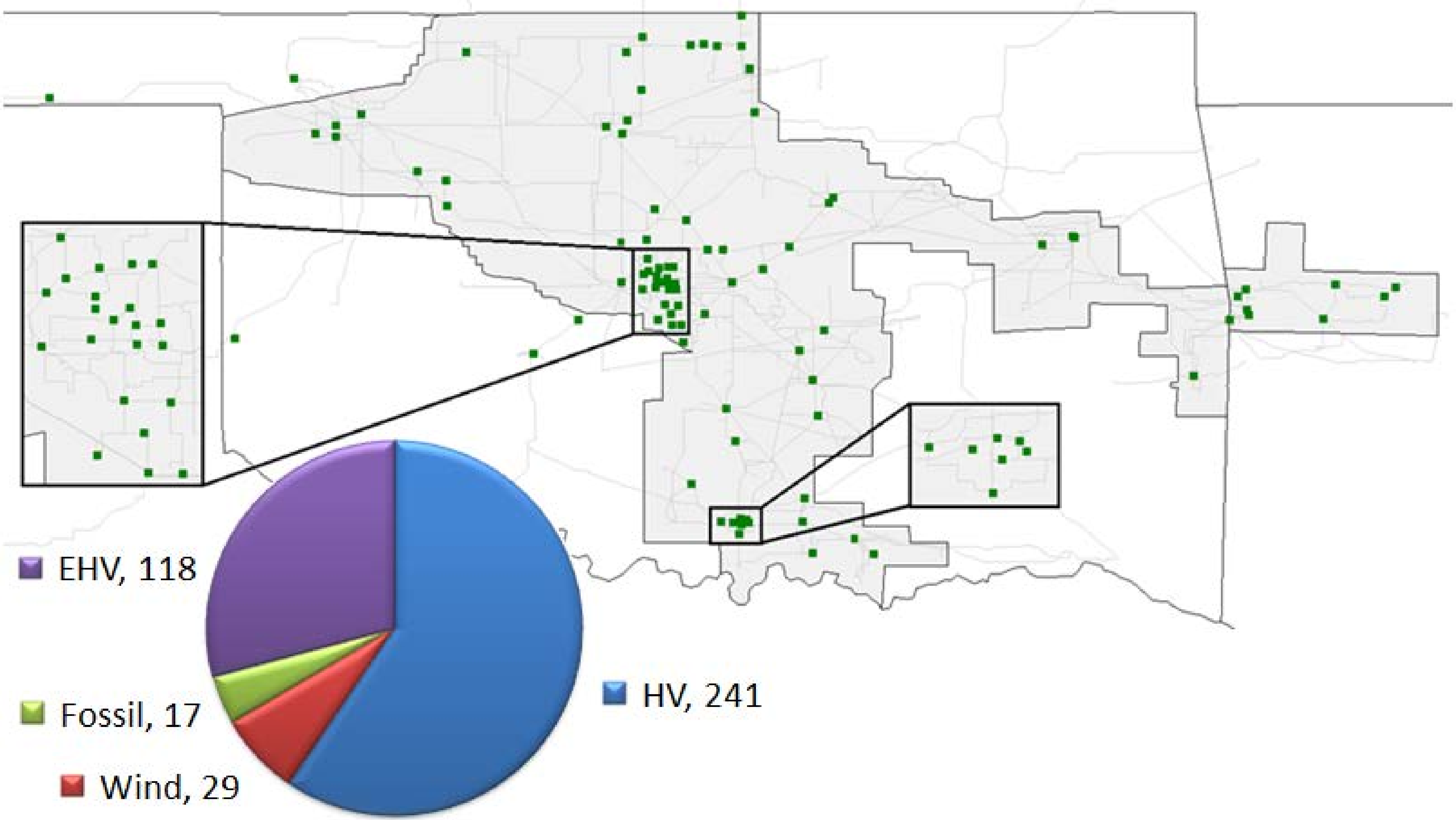
DE-OE-778

OG&E Background

- 2008 - Synchrophasor Pilot
- Followed by several standalone projects to network existing PMUs
- 2014 - Standardized SP&C & telecom design to network PMUs

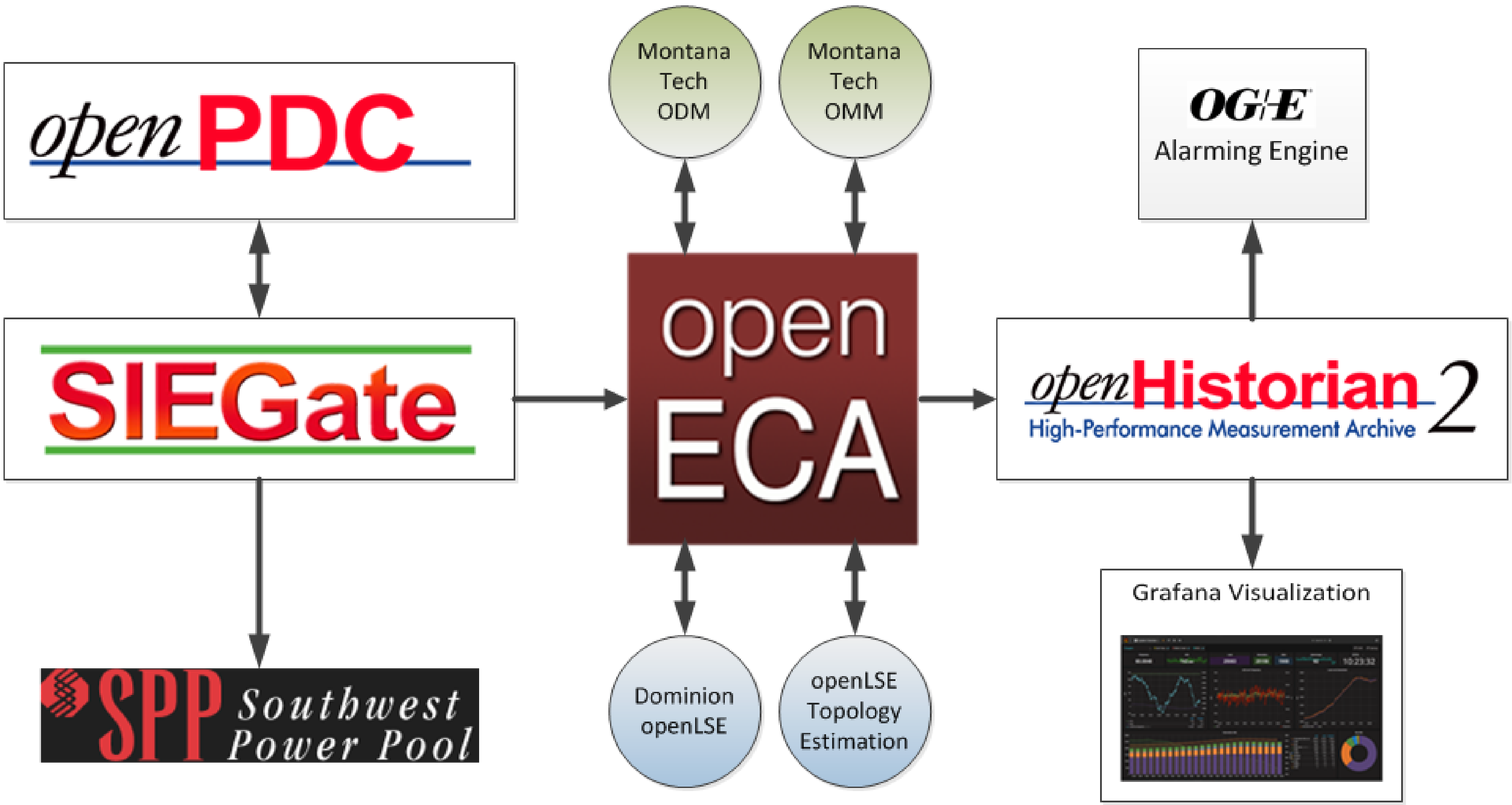


PMU Locations



405 PMUs Total

openECA Data Flow



openECA

DEMONSTRATION / TESTING RESULTS

openECA Platform Observations

- Installation and Configuration
  - Installation and setup is easy and very similar to openPDC, which we have used for many years
  - No issues were encountered with getting data into openECA and openHistorian
- Performance
  - 400+ PMUs with 3 Phase system requires minimal resources (3-4% CPU for each service)
- Stability
  - We had a crash of the services one weekend. It appears to be a result of the openECA Manager being left open and crashing and taking the openECA service down with it. These should be better isolated so that cascading failures are not possible.

Oscillation Detection Monitor (ODM) Test Approach

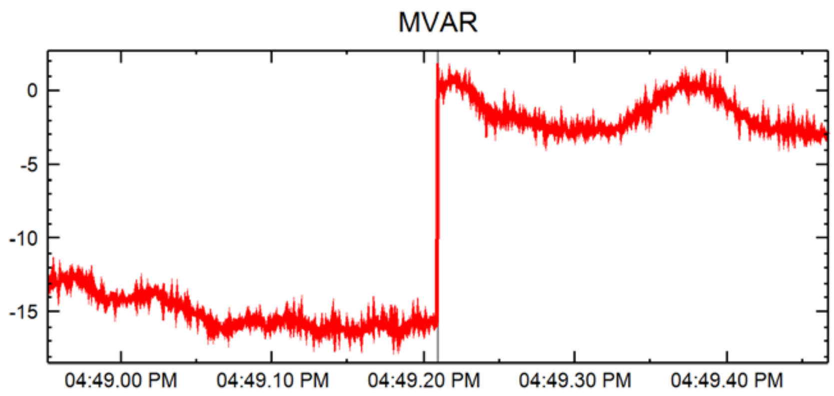
- Create power calculations for all generator PMUs in openECA
- Set up ODM monitoring of MW and MVAR signals
- Create dashboards in Grafana to visualize results
- Validate ODM results against other tools (in house FFT oscillation detection)
- Monitor analytic stability and performance

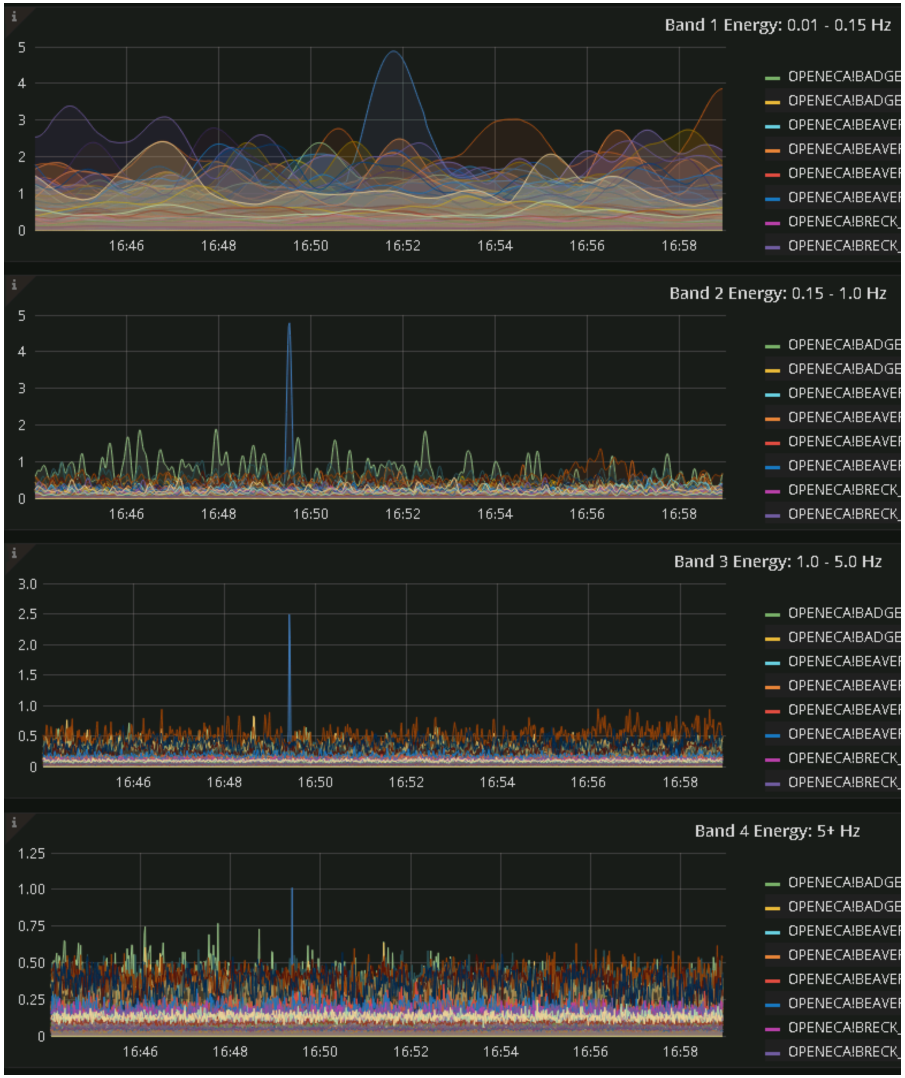
Oscillation Detection Monitor (ODM) Observations

- Installation and Configuration
  - Installation and setup is relatively easy. The OD mapper tool could be improved by adding a text filter and better means of selecting many signals.
  - Generating the power calcs in openECA is not easy and was the most time consuming part.
- Performance
  - We are monitoring 78 signals from all of our generators (MW and MVAR). The service uses 5-6% CPU resources
- Stability
  - After the openECA service crash was resolved, the ODM was producing erroneous results. We restarted the ODM service and it returned to normal. Otherwise we have not had any problems with stability.

Oscillation Detection Monitor (ODM) Test Results

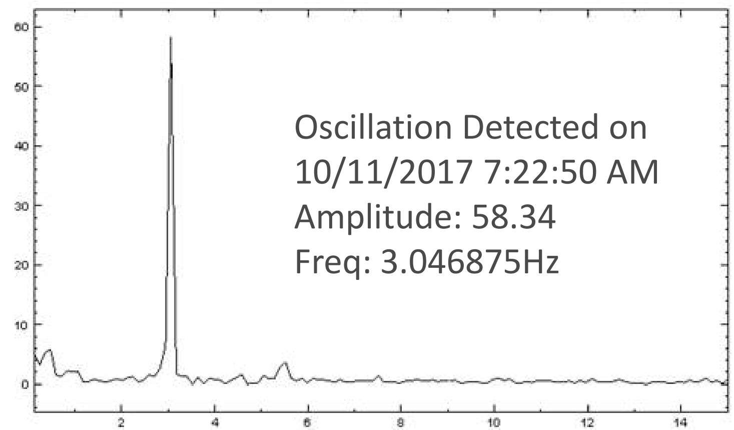
- ODM is very sensitive to step changes, faults and line switching
- ODM response to a 15 MVAR step

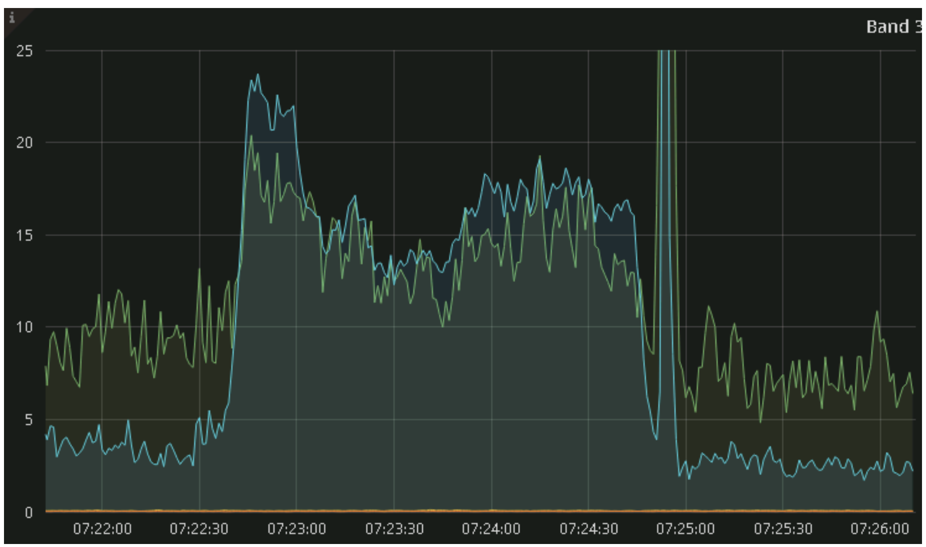




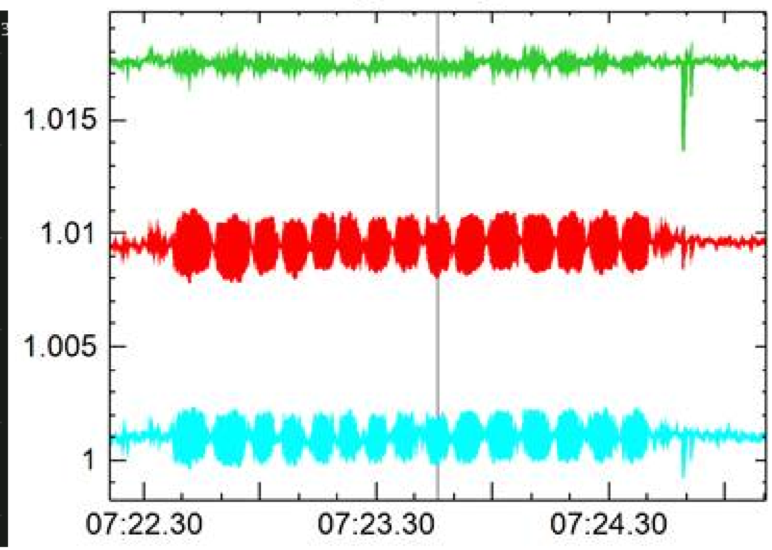
Oscillation Detection Monitor (ODM) Test Results

- ODM comparison to in house FFT for actual oscillation event on 10/11/2017



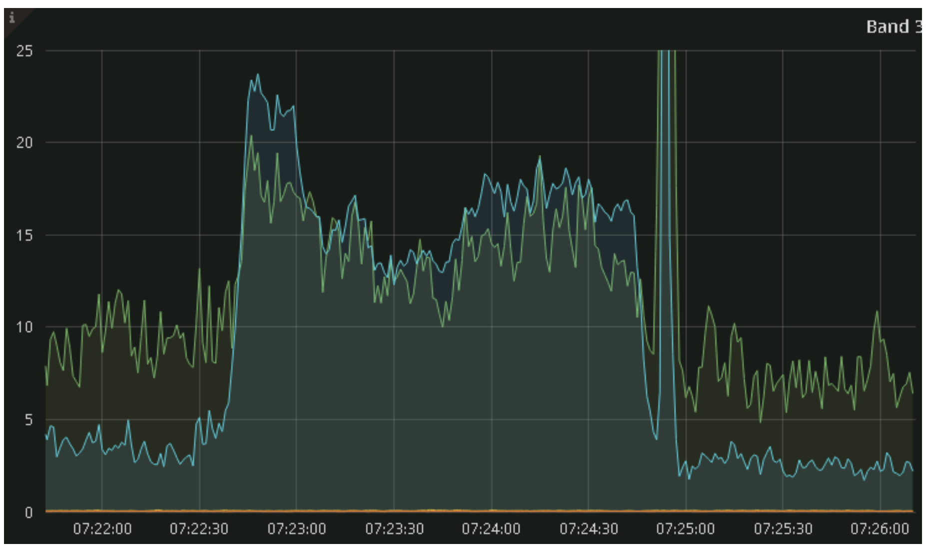


Voltage Magnitude



Oscillation Detection Monitor (ODM) Test Results

- ODM does not include alarming or notification capabilities, which is important for validation
- OG&E has developed a stand alone notification service that queries the Open Historian 2.0



Band	Mag	Window	PU	DO
1	50	900	50%	10%
2	25	60	50%	10%
3	10	30	50%	10%
4	10	20	50%	10%

Linear State Estimator (LSE) Test Approach

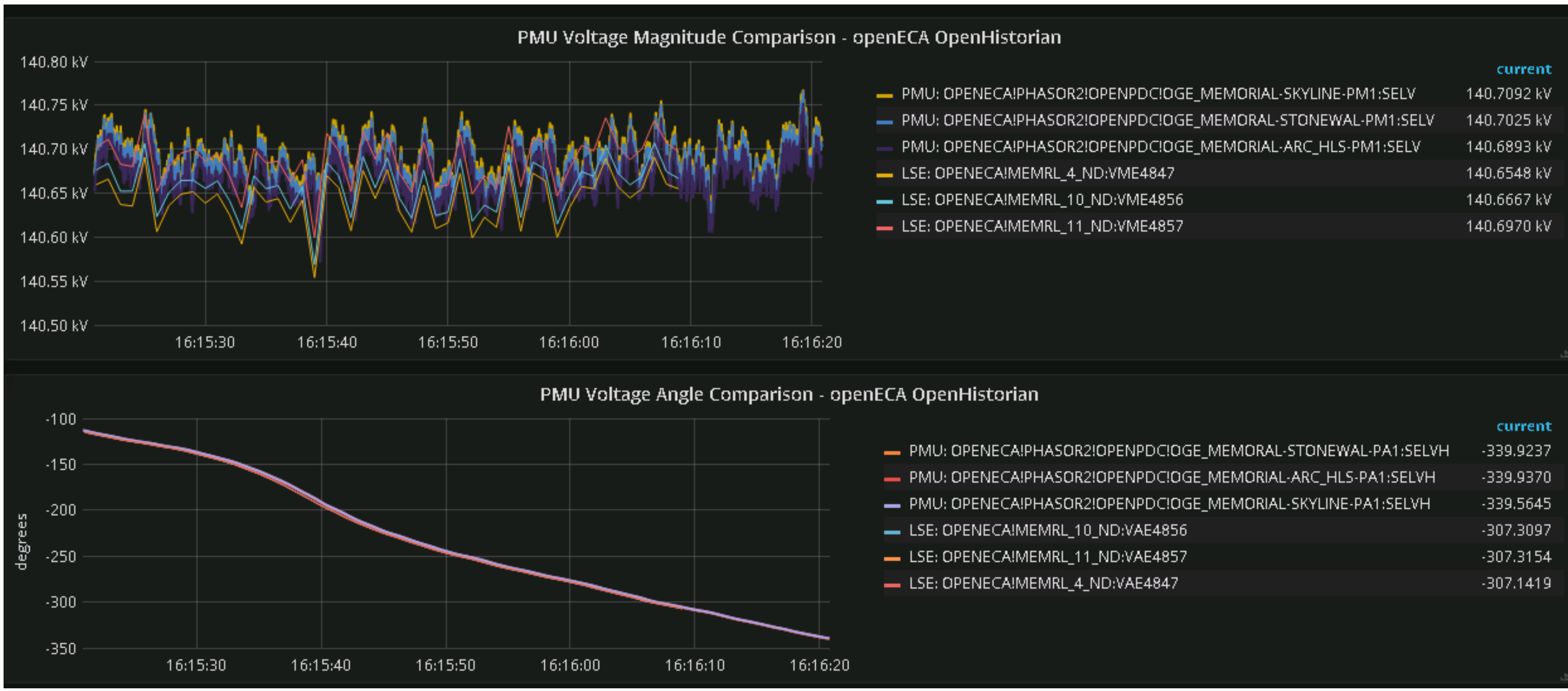
- SPP was generous to let us use their model since the PMU data is coming from OG&E anyway
- With the model from SPP, import model into offline model tool, prune/reduce based on measurement availability
- Get model solving in offline LSE environment with PMU snapshot data
- Get model solving in real-time with LSE service
- Document LSE's enhanced observability
- Monitor estimated residuals, compare estimated values with SCADA or EMS State Estimator

Linear State Estimator (LSE) Observations

- Installation and Configuration
  - Installation is very easy. There were several iterations before we were able to get an offline solution to run in the network model editor. Recent improvements to the NME reduce the configuration burden significantly.
- Performance
  - The openLSE service uses 4-5% CPU resources, but RAM usage continues to grow if measurements are queued. We dropped the interval down to 1 sample per second and our server was able to keep up without queueing. This also reduced the CPU burden to 1-2%.
- Stability
  - We were not able to run the LSE in real time at 30 samples per second, but it runs very well at 1 sample per second.

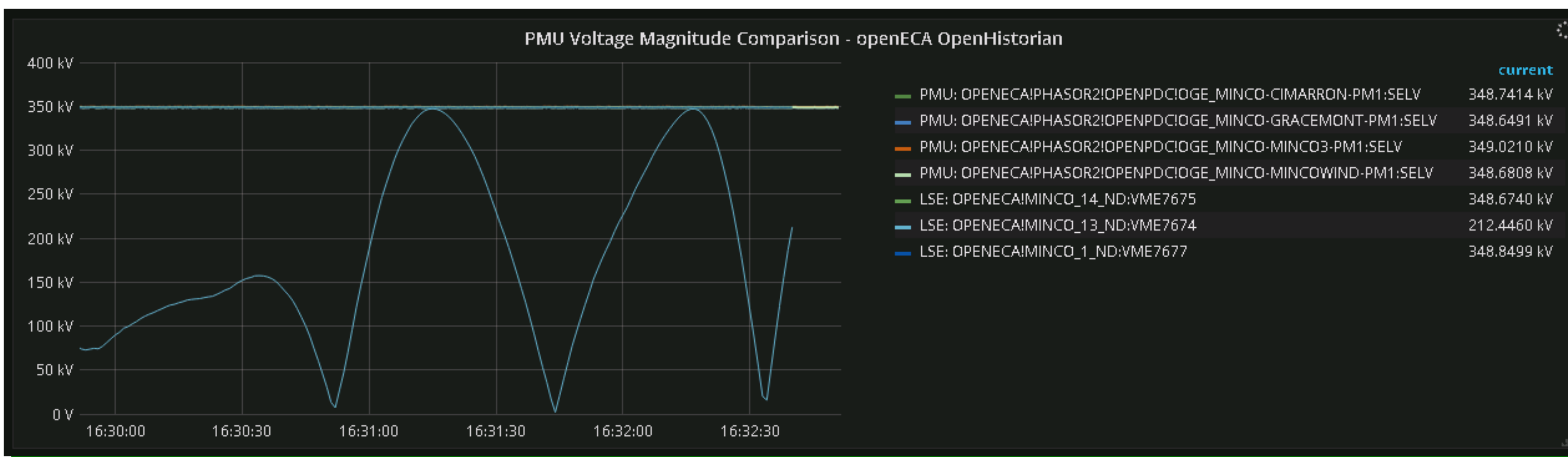
Linear State Estimator (LSE) Test Results

- Side by side results of PMU data and LSE output show good correlation

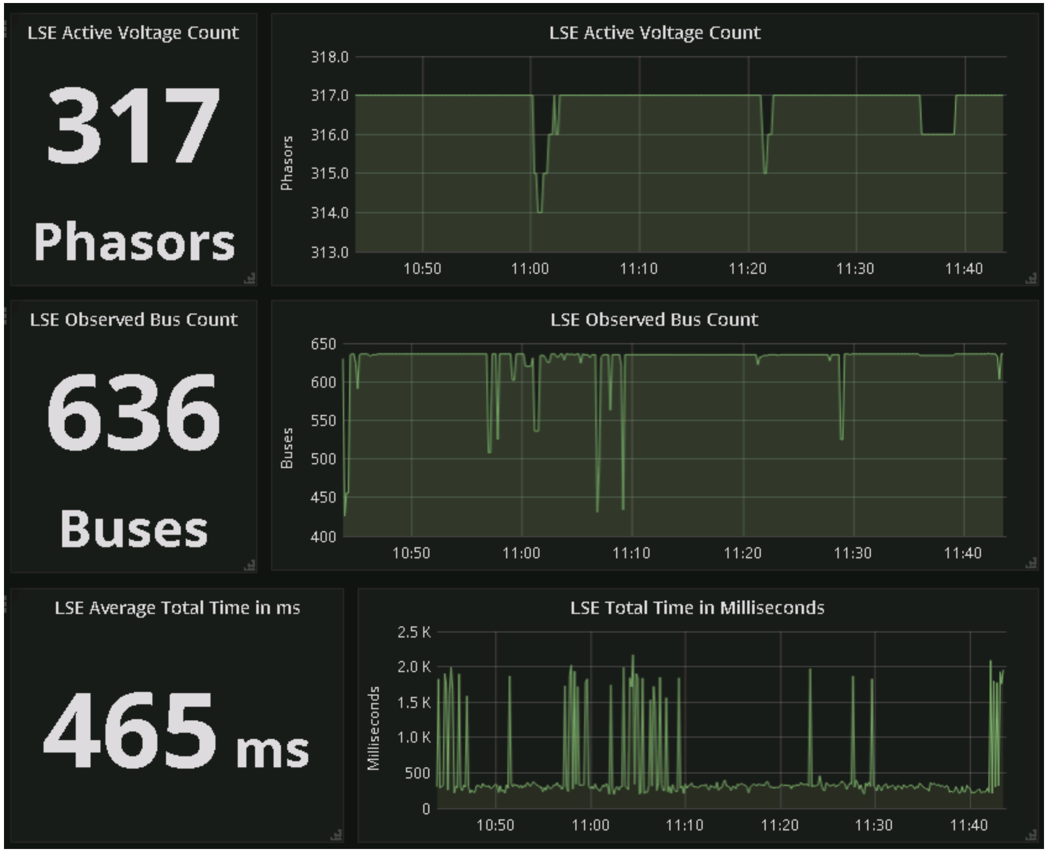


Linear State Estimator (LSE) Test Results

- Some output signals appear to drift
- The output data is delayed by 10-15 seconds compared with the PMU data
- More time needed for studying results



Linear State Estimator (LSE) Test Results



- Performance metrics show active phasors, bus observability and time delays for various aspects of the LSE
- Our total time is usually around 350ms, but often spikes up to the 1.5-2 second range due to network reconstruction

Summary

- We will continue to validate the ODM and tweak our alarming to provide meaningful notification of events
- We also plan to test the oscillation mode meter (OMM) in the near future
- We will continue to work on the LSE progress
- We would also like to test the line impedance calculator in the future