Open Framework for Integration Data Operations (OpenFIDO)

CEC EPC-17-047
Final Product Release Report
SLAC National Accelerator Laboratory
13 January 2023

Table of Contents

SLAC National Accelerator Laboratory is operated by Stanford for the US Department of Energy under Contract DE-AC02-76SF00515.

Stakeholder Inputs, Benefits Analysis and Technical Merits

This section provides a summary of the results of the requirements analysis for the joint GLOW, HiPAS, and OpenFIDO projects presented to the project team in March 2019. The analysis included a review of current tool usage, includign a review of the DOE version of GridLAB-D, a synopsis of user interviews, and general requirements identification from the current usage, code reviews, and interviews. The reviews included the ARPA-E VADER project, the ARPA/CEC PowerNET with Markets projects, and the DOE GRIP projects.

Individual source for this section include Prof. Leigh Tesfatsion (Iowa State University), GridLAB-D Issues (GitHub), Mayank Malik (ARPA-E VADER project system architect), Alyona Teyber (DOE GRIP project PI), Claudio Rivetta (DOE PowerNET and CEC PowerNET with Markets projects). Other sources included input from CPUC, utility, CCA, and vendor staff.

Projects Reviewed

The Visual Analytics for Distributed Energy Resources (VADER) uses a simulation engine for scenario modeling, and relies on utility models, AMI and SCADA data for supported analytics. The analytics supported by VADER include solar disaggregation, network switch status detection, and machine learning-based power flow.

PowerNET (DOE) and PowerNET With Markets (CEC) is used to simulate and estimate control system impacts using price-based dispatch of distributed energy resources. Data and model sources include Cyme network models, tariff/DR/DER control models, and emi-aggregated AMI and SCADA data (AWS MySQL/Mariah RDS). Analytics implemente include advanced load control performance evaluation, demand response performance evaluation, distributed energy resource performance impacts, and cost and revenue impacts.

Grid Resilience Intelligence Project (GRIP) provides utility resilience anticipation/absorption tools. Data and model sources include Cyme network and DMS control models, AMI, and SCADA data. Analytics supported include pole failure impacts, absorption strategy performance evaluation, and optimal recovery control strategies.

In addition the DOE Advanced Load Modeling (ALM) project provides NERC load composition for planning studies. The ALM tools rely on data collected from utilities, including feeder load measurements and AMI data. The analysis of data to validate load composition delivers weather sensitivity, filling in gaps in data coverage, verification of default value, and managing datasets used in planning studies

Code Review

A code review of the DOE simulation tools was completed covering the following topic areas:

Area 1: Data formats: Support for new/modern data formats, e.g., JSON (widely used by python-based software), CIM (new standard for network modeling), OpenADR (new standard for DR/DER controls), emerging standards for tariff modeling.

Area 2: Core structure: Separate core from data, UX and module APIs, including main platform, module and solver management. Use APIs for user interface (UI/UX), module, solver and host access libraries. Support reentrancy for scenarios analysis, including ability to run multiple cores in a single process and running multiple session streams and the ability to copy needed data for scenario processing.

Area 3: Platform-level dependencies: Provide a standardized deployment for multiple hosts including Docker for workstations and locally-hosted servers, Amazon Web Services (AWS), Google Cloud, and Microsoft Azure. Embedding in open-source languages is necessary for broad develop support, including Python 3 module and perhaps other open-source languages such as Julia and R.

Area 4: Data-handling: Standardize data handling APIs, including support for credential management for cloud-based data access, download caches for using remote data, data pipelines used in analysis methods, uploading, reviewing, and publishing data & results, and scenario-based data management. This should also include tools for providing general access to data based on scenario parameters to meet common data processing needs such as data cleaning, preview, and generating common output formats.

User Interviews

User Interviews were conducted to gain insights into emerging needs and requirements. Interviews included government (state and federal) users, utilities (TSO/DSO, public/private), service providers (data analytics), and researchers (academic, government).

Government users focused on tariff analysis, with particular attention to customer revenue and resource cost impacts at the distribution level, as well as tariff parameter sensitivity analysis on revenue and costs. ICA and LNBA analysis were noted, with specific attention to the implementation of a standard integrated capacity analysis method. Resilience analysis was also noted with regard to asset vulnerability assessment and large-scale rare-event impacts analysis.

Utility users focused on ICA and LNBA analysis, tariff design, resilience planning, and resource adequacy modeling. ICA, LNBA, and tariff design analysis concerns were centered on validation and standardization of the tools with particular attention to the use of utility test models. Resilience planning needs focus on demonstrating new planning and analysis methods on asset vulnerability scenarios for extreme weather events such as storms and extreme heat or cold. Resource adequacy modeling requires development and broad deployment of advanced load modeling and load forecasting tools that incorporate distributed energy resources, demand response, and impacts of end-use electrification.

Service/analytics providers focused their attention on custom analysis implementation and deployment, including automate access to public data repositories, weather data collection/processing/delivery, and access to standard grid, building, and tariff model and libraries. They also draw attention to the importance of broad support for data import and export, including automatic data format recognition and handling, data cleaning, time-standardization, component identification and linkage between models.

Researchers draw attention to the challenges of maintaining accurate and up-to-date documentation, especially in regards to current internal/version-specific documentation, which is needed for all analytic methods. Training resources are lacking in many new tools, and user cases with step-by-step instruction online as videos are needed. Valid examples for all system components, with useful code snippets, standardized units, parameter ranges, good defaults are often needed to improve tool usability. This includes more taxonomies of distribution grids, households, buildings, renewable resources, as well as data-driven building models. These latter are in particular lacking as the CEUS/RBSA data for residential and commercial building models are become outdated and need to be updated or at least made more current as building end-use compositions and occupancies change. Improved optimization solvers are also needed, including updates to existing optimization modules to support new methods.

Identified Requirements

The following categories of requirements were identified.

- **Usability**: Improve GridLAB-D user experience and support GLOW UX design. Although GLOW will be widely available to most users through Hitachi, there is still a need for direct access to GridLAB-D without using the command-line interface. A well-design UI/UX needs to include data access and simulation hosting capabilities, including plug-and-play input, cleaning, viewing, and export tools. Simulations must be hosted using either individual or organizational credential management, with local, organization and cloud-based simulation dispatch.

- **Speed**: Enable multicore, multihost, and cloud scale simulations and analysis without manually programming it in scripts. Parallelization for internal loops is necessary to improve overall speed of simulation because there are many potentially parallel internal loops that can be optimized at every level of simulation. This will require access and control of external hosts, configuration of multiple host pools, scalable job control, shared resource management (e.g., input data, intermediate results), and automatic dispatch of processing pipeline.

- **Flexibility**: Enable user-defined data pipelines, scenarios, and modules to facilitate extensibility and customization of workflows. User-defined scenario design is needed, with model selection/customization and scenario-based data selection. Scenario execution, scalable hosting of processing, and resource usage/cost tracking/management are needed. Scenario result analysis including collection of

scenario output results, preview, and output formatting (e.g., collation, tabulation, plotting) are also needed.

- **Validation**: Enable user-defined model validation tools. Validation of methods and tools are a big challenge for new analysis tools and methods that do not have pre-existing examples. Integrated capacity analysis (ICA) and Locational Net-Benefit Analysis (LNBA) are not yet fully standardized. Tariff design lacks standard models for tariffs. Resilience analysis (asset vulnerability, rare large event impacts) lacks generally accepted metrics for resilience performance. Focus should be on canonical test and results, including standard test cases, results matched to a predetermined error margin, and incorporated validation into the main product build process.

- **Reproducibility**: Enable high-reproducibility simulation capabilities. Stochastic model reproducibility should be used whenever possible, including the use of local entropy sources and saving/restoring entropy state in model save/load to allow simulation and analysis continuation from intermediate/scenario branch states. Variables and properties need to include latent states, cross-correlated properties, and uncertainty propagation.

Summary of Data Exchange Formats Implemented

OpenFIDO implements data conversion and processing using pipelines that are distributed on GitHub at https://github.com/openfido.

The following pipelines are currently implemented and publicly available:

Pipeline name	**Description**
Address	Obtains addresses from locations, or locations from addresses
Loadshape	Performs loadshape cluster analysis based on AMI data
Weather	Obtains historical weather data
CYME extract	Converts CYME models to GLM models
Resilience	Performs system resilience analysis
Integration capacity	Performs integration capacity analysis
Electrification	Performs end-use electrification analysis
Vegetation	Obtains vegetation data
Powerline	Performs powerline sag, sway, and contact analysis
Elevation	Obtains ground elevation data
Utility	Obtains data about utilities
Census	Obtains census data
Forecast	Performs load forecasting based on historical data
Tariff design	Performs tariff design analysis

Each pipeline supports one or more of the following input and output file formats and semantics:

Pipeline name	**Inputs files**	**Output files**
Address	CSV (geodata)	CSV (geodata)
Loadshape	CSV (amidata)	CSV (loadshapes and groups), GLM (models), PNG (loadshapes)
Weather	N/A	CSV (weather data), GLM (models)

Pipeline name	Inputs files	Output files
CYME extract	MDB, CSV	CSV (CYME data), GLM (models)
Resilience	CSV, GLM	CSV (results), PNG (results)
Integration capacity	CSV, GLM	CSV (results), PNG (results)
Electrification	CSV, GLM	CSV (results), PNG (results)
Vegetation	CSV (geodata)	CSV (geodata)
Powerline	CSV (geodata)	CSV (geodata)
Elevation	CSV (geodata)	CSV (geodata)
Utility	CSV (geodata)	CSV (geodata)
Census	CSV (geodata)	CSV (geodata)
Forecast	CSV (amidata)	CSV (weather), GLM (model)
Tariff design	CSV, GLM	CSV (results), PNG (results)

In addition, a pipeline for general GridLAB-D support is included which allows access to all the built-in data format and semantic converters that are part of GridLAB-D. For details see the Converters section of the GridLAB-D documentation at https://docs.gridlabd.us/.

Results from Beta Testing Corrective Measures

During testing and validation, a total of 28 issues were identified with the OpenFIDO system (not including the individual pipelines). Of these 8 are completely resolved. The remaining 20 are pending resolutions in a future release of OpenFIDO. The follow lists include links to the original GitHub issues where additional details may be found.

Resolved Issues

1. Update deployment process documents
2. No output generated sometimes
3. Change the set of initial pipelines
4. Bar chart does not label x-axis correctly
5. Add ability stop a pipeline run
6. Optimize git fetches on worker processes
7. Chart Preview is not correctly showing the dataset
8. Bar Chart display is not showing all of the x-axis items

Open Issues

1. Openfido CLI should use docker-compose as baseimages
2. Pipeline killed with no error message
3. Deprecated MINIO_SECRET_KEY and MINIO_ACCESS_KEY
4. Invitations not being received by email recipient
5. Pipeline run occasionally continues to run "in progress" even when completed
6. Have stdout include the GitHub repo commit id to verify that the user is getting the right version
7. Link new jobs to GitHub projects
8. Workers continue running after deleting pipeline/pipeline run while "in progress"
9. Naming images
10. Open artifacts in browser instead of download
11. Charts should be available to reuse from previous pipelines
12. Parameterize the initial set of pipelines created in a new docker image
13. Email invitation doesn't expire in 14 days
14. Add information about who started a run
15. Add ability to copy files from last job
16. Allow varied capitalization in user emails
17. Increase Auth Service code coverage to reach 100%
18. bug: user was removed from an organization, then re-invited, but was unable to accept invitation once more
19. Add "v1" prefix to auth service urls to match other services
20. Add API key requirements to the Auth Service

Summary of Testing and Validation Results

The Testing and Validation Report (Task 4.2) includes details on the methodology and implementation of the Testing and Validation procedure for OpenFIDO. An automated continuous integration/continuous deployment (CI/CD) system is employed by the GitHub platform that hosts the source code and deployment tools. The current status of the CI/CD is available at https://github.com/openfido and reports both the application status, as well as the status of the individual pipelines that are available for public use.

This section summarizes the results of the deployment testing process at the time of the initial release of OpenFIDO. These results are also presented in Appendix A of that report. Note that these results are expected to change with future releases of OpenFIDO.

Appendix B of the Testing and Validation Report contains additional information about the testing and validation result of the individual pipelines described in the previous section(s). Pipeline testing and validation results are expected to change more frequently than those for OpenFIDO itself because pipelines are delivered individually and independently of OpenFIDO.

GitHub Actions Testing Results

Component	Test	Result	Disposition
App	211 unit tests	>97% coverage	100% tests passing. See Table A.2
	style	100/100 ok	
	lint	>90% score	See Table A.3
Auth	136 unit tests	>92% coverage	100% tests passing. See Table A.4
	style	100% ok	
	lint	>90% score	See Table A.5
Workflow	160 unit tests	>99% coverage	100% tests passing. See Table A.6
	style	100% ok	
	lint	>90% score	See Table A.7
Client	jest-puppeteer	No issues	

App Missing Coverage Testing Results

File location	Missed Statements	Proposed Remedy
app/pipelines/services.py	18/254 uncovered	Add additional tests for uncovered statements.
app/workflows/routes.py	4/134 uncovered	Add additional tests for

		uncovered statements.
app/workflows/services.py	7/279 uncovered	Add additional tests for uncovered statements.

App Lint Issues Testing Results

Test Result	**Issue Description**	**Proposed Remedy**
92.6% lint score	Code Style and Formatting	Run the *lint* command, and review each potential style or formatting issue listed against the linter's recommended fix.

Auth Missing Coverage Testing Results

File location	**Missed Statements**	**Proposed Remedy**
app/auth.py	32/260 uncovered	Add additional tests for uncovered statements.
app/mail.py	13/52 uncovered	Add additional tests for uncovered statements.
app/org.py	14/221 uncovered	Add additional tests for uncovered statements.
app/services.py	3/211 uncovered	Add additional tests for uncovered statements.

Auth Lint Issues Testing Results

Test Result	**Issue Description**	**Proposed Remedy**
90.4% lint score	Code Style and Formatting	Run the *lint* command, and review each potential style or formatting issue listed against the linter's recommended fix.

Workflow Missing Coverage Testing Results

File location	**Missed Statements**	**Proposed Remedy**
app/workflows/queries.py	2/52 uncovered	Add additional tests for uncovered statements.

Workflow Lint Issues Testing Results

Test Result	Issue Description	Proposed Remedy
92.2% lint score	Code Style and Formatting	Run the *lint* command, and review each potential style or formatting issue listed against the linter's recommended fix.

Summary of Recommendations for Follow-Up and Corrections

The following recommendations have been made for future development.

Pipeline development.

1. **Load decomposition**: analysis to permit extraction of end-use load shapes from AMI data and SCADA data.
2. **Census data**: access consumer demographic data to enhance load models using economic and population data.
3. **Advanced building loads**: utilize satellite data to identify the location, size, and type of buildings in communities, and automatically link the building models to power system network models.
4. **Grid resilience**: public safety power shutoff (PSPS) alternative/standardized outage optimization methodology, and long-term climate change impacts on electric load, distribution infrastructure, and data-driven asset planning and hardening.

Shared/public data/models security and integrity

The current data sharing model uses widely accepted digital security methods. However, this may not be fully satisfactory for utilities to satisfy critical infrastructure cyber-security standards. The project team recommends follow-up research and development work to identify and deploy digital artifact sharing mechanisms that allow utilities to share selected data from software, databases, and repositories with authorized users in a reliable and secure manner .

Legacy data handling and data aging strategies

Historical data is a critical component in developing long-term system performance models such a building loads, asset degradation, and human behavior. Consistent long-term access to weather, demographic, building stock, energy consumption, and power demand data is an ongoing challenge and a barrier to adopted of new analytics tools and methods, particularly as modern and more advanced data management and data sharing capabilities become widely accepted. The project team recommends further research and development into data handing strategies to future proof critical existing data sets and ensure that new data analytics have access to much of the valuable existing historical data that makes these analysis worth deploying at scale.

User Documentation

User documentation is available online at https://help.openfido.org. In addition a PDF version of the user documentation is available at https://github.com/slacgismo/openfido under Task 5.1, entitled "Developer and User Training Documentation".

The documentation for each of the published pipelines include recommended settings to access the pipeline, input and output specifications, and documentation references related to the specific pipeline implementation if appropriate.

The user documentation is provided for the following pipelines

- Tariff design
- Loadshape analysis
- Weather history and forecasting
- GridLAB-D simulation
- Census geographic data
- Resilience analysis
- Hosting capacity analysis
- Electrification analysis
- Address resolution
- Cyme model extraction

Developer Documentation

Developer documentation is available online at https://github.com/openfido. In addition a PDF version of the developer documentation is available at https://github.com/slacgismo/openfido under Task 5.1, entitled "Developer and User Training Documentation". The main contents of the developer documentation are as follows:

OpenFIDO Overview

This chapter provides developers with documentation on getting started, signing to access the shared application, hosting OpenFIDO locally, deploying OpenFIDO to AWS, assessing the status of the OpenFIDO CI/CD system.

OpenFIDO Application Service

This chapter provides developers with documentation on the OpenFIDO's application service which provides organizational control over workflows.

OpenFIDO Authentication Service

This chapter provides developers with documentation on OpenFIDO's authentication service which implements the platform user access control system, including email integration, building, and deployment management.

OpenFIDO Workflow Service

This chapter provides developers with documentation on the API that executes pipeline jobs, including setting up and managing the local development environment.

OpenFIDO Client

This chapter provides developer with the information needed to create, test, and deploy clients, including input data processing pipelines, configuration files, and pipeline run-control files.

Documentation of Full Functionality and Free Availability

A fully functional online version of OpenFIDO is available at https://app.openfido.org/. For access credentials to the online version, please contact the project administrators through GitHub at the OpenFIDO organization at https://github.com/openfido.

Users may freely install OpenFIDO on workstations, Docker servers, and AWS systems by following the instructions at https://github.com/openfido.

Developers may contribute to and submit new functionality to OpenFIDO and to pipelines be registering as a GitHub user and requesting to join the OpenFIDO organization at https://github.com/openfido.

Documentation of Full Support

On 17 January 2023 the Linux Foundation Energy Technical Advisory Committee voted unanimously to adopt OpenFIDO as a LF Energy project. The project Technical Steering Committee will be formed and have the project's organizational meeting in the winter or spring of 2023, at which point ownership and responsibility for OpenFIDO and all the pipelines will be transferred from SLAC National Accelerator Laboratory to LF Energy. This will ensure ongoing support for the GitHub repository, the AWS infrastructure, and the oversight of technical contributions to the open-source code.

The welcome letter from the Linux Foundation Energy is attached in Appendix A.

Appendix A - Linux Foundation Welcome Letter

From: Yarille Kilborn <ykilborn@linuxfoundation.org>
Date: Wed, Jan 18, 2023 at 12:02 PM
Subject: Welcome OpenFIDO to LF Energy!
To: Chassin, David P <dchassin@slac.stanford.edu>
Cc: LF Energy <operations@lfenergy.org>

Welcome to LF Energy!

We are pleased to welcome OpenFIDO LF Energy as a Technical Project. The information below is intended to onboard your project, please do read carefully. And of course, you'll have the LF Energy Staff available to help you with questions along the way, you can reach us at operations@lfenergy.com.

We suggest you review the New Project/Working Group Onboarding page as a guide to the overall onboarding process. For now, here are the immediate actions for you.

Scheduling First Technical Steering Committee Meeting

We are looking forward to kicking off the Technical Steering Committee (TSC) for the project. Actions to get this going are:

- Let us know who are the initial TSC members to include in this meeting
- Let us know a general timeframe to hold the call and we can help coordinate a day/time with the TSC members.

This first meeting will be facilitated by the LF Energy staff, and we will cover the following:

- Welcome/Introductions
- Role of the Technical Steering Committee
    - Nominate & Elect a Chairperson/Secretary
- Approve/Review Legal Docs
    - Project Charter
- DCO - Provide an overview
- Onboarding checklist review
- Launch plan
- Meeting Cadence going forward

TAC Engagement

As an LF Energy Technical Project benefit you will have an LF Energy Technical Advisory Council (TAC) sponsor assigned to help guide your project. We encourage you to connect with your TAC sponsor and

leverage the wealth of knowledge that they can provide. They will also be helpful to you in your preparation for the TAC Technical Project review which your project will be required to complete annually.

We also encourage you to regularly attend TAC meetings. Subscribe to the TAC email list to have the calendar invites sent to you.

Key Resources

The following are key resources we recommend you review and bookmark for future reference:

- Web Site: https://www.lfenergy.org/
- Interactive Landscape: https://landscape.lfenergy.org/
- GitHub: https://github.com/lf-energy
- Wiki: https://wiki.lfenergy.org/
- Mail Lists: https://lists.lfenergy.org/g/main
- Twitter: @LFE_Foundation
- LinkedIn: https://www.linkedin.com/company/lf-energy-foundation
- Email: operations@lfenergy.org

Again, welcome and we look forward to seeing the growth and success of your project as part of the LF Energy. If you have any questions please don't hesitate to reach out.

--
Yarille Ortiz Kilborn
Linux Foundation
Project Coordinator