ARRAS
Status and Challenges

David P. Chassin, PhD
SLAC National Accelerator Laboratory, Menlo Park, California*

7 September 2024

LF ENERGY

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

Overview

1. Brief history of Arras Energy
2. GridLAB-D capabilities adopted in Arras Energy
3. User interfaces available to Arras Energy users
4. OSS challenges and opportunities

Contributing organizations

SLAC NATIONAL ACCELERATOR LABORATORY

Stanford University

U.S. DEPARTMENT OF ENERGY

POST ROAD FOUNDATION
CONNECTED, SUSTAINABLE INFRASTRUCTURE FOR THE 21ST CENTURY

NRECA America's Electric Cooperatives

Pacific Northwest NATIONAL LABORATORY

HITACHI Inspire the Next

SOUTHERN CALIFORNIA EDISON
An *EDISON INTERNATIONAL* Company

NREL NATIONAL RENEWABLE ENERGY LABORATORY

LF ENERGY

FUNDING PROVIDED BY THE CALIFORNIA ENERGY COMMISSION

aws

Energy Delivery Systems Simulation

Arras Energy is a commercial release of HiPAS GridLAB-D developed by the California Energy Commission

→ Make the US Department of Energy's agent-based electricity delivery system simulator available to California utilities

→ Support emerging use-cases in California

- Hosting capacity analysis
- Extreme event resilience
- Deep electrification
- Modern tariff design.

LF ENERGY ARRAS

Arras Energy - 21st Century Electricity System Analysis Tools for 21st Century Electricity Utilities

https://arras.energy/

Timeline of Arras Energy development

2002 - PNNL creates PDSS, an agent-based solver technology demonstration

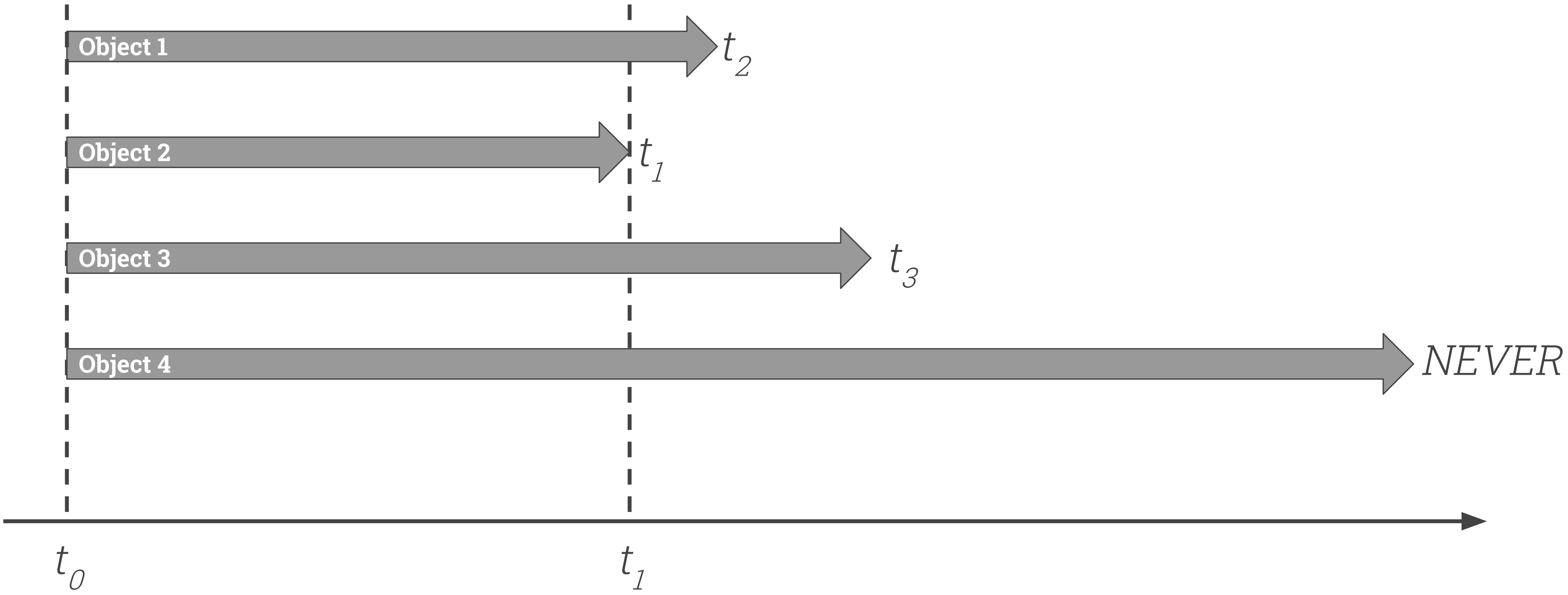
2007 - DOE funds development of GridLAB-D based on PDSS

2008 - First version of GridLAB-D released as free open-source software

2018 - CEC funds SLAC to develop HiPAS GridLAB-D

2022 - LF Energy adopts HiPAS GridLAB-D and renames it Arras Energy

Agent-based Solver Concept of Operations



Multiple Concurrent Solvers Mechanism

- **Initialization** (construct problems for solvers from object data and set clock to starttime)
- **Variable Time-step iteration** (advance clock to next object state change)
  - **Precommit** (one-time initial data exchange, once per timestep)
    - **Presync** (top-down data exchange and state updates, once per iteration)
    - **Sync** (bottom-up data exchange and solver updates, once per iteration)
    - **Postsync** (bottom-up data exchange and state updates, once per iteration)
  - **Commit** (final data exchange when all objects return future times, once per timestep)
- **Finalize** (final data save when clock reaches NEVER or stoptime)

LF ENERGY

Arras Energy Status

- **Sandbox project -- needs updated roadmap, thriving community & growth**
- **Current simulator capabilities include**
  - Transmission and distribution system powerflow solvers (transient dynamics is experimental)
  - Models for residential, commercial, and industrial loads (data-driven, physics-based)
  - Full tariff database (OpenEI)
  - Retail market simulation (includes transactive energy)
  - Optimization (e.g., CVX w/Clarabel, MPC)
  - Weather data (North America online, elsewhere offline)
  - IEEE, DOE, and PG&E test/taxonomy feeders
  - CSV players and recorders, mysql and influxdb support
  - Time-series geodata scatter/gather
  - Python integration (numpy, pandas, scipy, sklearn, etc.)

LF ENERGY

Available Solvers

Distribution Systems

3-phase unbalanced solvers

- Newton-Raphson current injection
- Kersting's method
- Transient dynamics (experimental)
- CYME converter available

Detailed asset models

- Cable geometry/configuration
- Transformers, relays, switches, etc.
- Pole failure due wind, ice, vegetations
- Distribution generation and storage

Load models by sector (R/C/I/A)

- Physics based (residential w/appliances)
- Data-driven (all)

Transmission Systems

Positive sequence solver (pypower)

- PSS/E converter available

Aggregate asset models

- Powerline composition
- Transformer, relays
- Generators, including costs for OPF
- Loads

Geodata

- Weather
- Loadshapes
- Other parameters that vary regionally

Other Useful Modules

Modeling
- Behavior
- Connection
- Powernet
- Reliability
- Resilience
- Revenue
- Tariff
- Transactive

Support
- Assert
- Influxdb
- Mysql
- Tape
- Optimize

PyPower

CVXPY

MySQL

influxdb

CSV

GridLAB-D Subcommands

Building - generate load models from data
Convert - file format converters (to/from)
Geodata - manipulate geographic data
Library - manipulate asset libraries
Loaddata - manipulate load datasets
Model - download example/test models
Plot - generate plots from models
Python - access python environment
Server - control gridlabd servers
Template - access analysis templates
Timezone - access timezone database
Version - control gridlabd version
Weather - access weather libraries

david — -bash — 75×32

```
MacBook-Pro-2:~ david$ gridlabd weather index ^WA
WA-Bellingham_Intl_Ap.tmy3
WA-Bremerton_National.tmy3
WA-Ephrata_Ap_Fcwos.tmy3
WA-Fairchild_Afb.tmy3
WA-Felts_Fld.tmy3
WA-Gray_Aaf.tmy3
WA-Hanford.tmy3
WA-Hoquiam_Ap.tmy3
WA-Kelso_Wb_Ap.tmy3
WA-Moses_Lake_Grant_County_Ap.tmy3
WA-Olympia_Airport.tmy3
WA-Pasco.tmy3
WA-Pullman_Moscow_Rgnl.tmy3
WA-Quillayute_State_Airport.tmy3
WA-Renton_Muni.tmy3
WA-Seattle_Boeing_Field_Isis.tmy3
WA-Seattle_Seattletacoma_Intl_A.tmy3
WA-Snohomish_Co.tmy3
WA-Spokane_International_Ap_Cheney_Uo.tmy3
WA-Stampede_Pass.tmy3
WA-Tacoma_Mcchord_Afb.tmy3
WA-Tacoma_Narrows.tmy3
WA-The_Dalles_Municipal_Arpt.tmy3
WA-Toledowinlock_Mem.tmy3
WA-Walla_Walla_City_County_Ap.tmy3
WA-Wenatchee_Pangborn.tmy3
WA-Whidbey_Island_Nas.tmy3
WA-William_R_Fairchild.tmy3
WA-Yakima_Air_Terminal.tmy3
MacBook-Pro-2:~ david$
```

GridLAB-D Tools

create_* - create submodels (ductbanks, filters, meters, poles, schedule, etc.)
fire_* - get fire reports and fire danger for location and date
fit_filter - fit a *z*-transform to data and generate a gridlabd filter
group - identify potential islands in a gridlabd model
market_* - access market data and generate market models
noaa_forecast - access NOAA weather data (realtime)
nsrdb_weather - access historical weather data from NREL

Python Integration

Module-level

- Full python access
- All global variables
- Model construction (before initialization)
- Model access (after initialization)
- Event handlers
  - Initialization
  - Precommit
  - Presync/sync/postsync
  - Commit
  - Finalize
  - Terminate

Class/Object-level

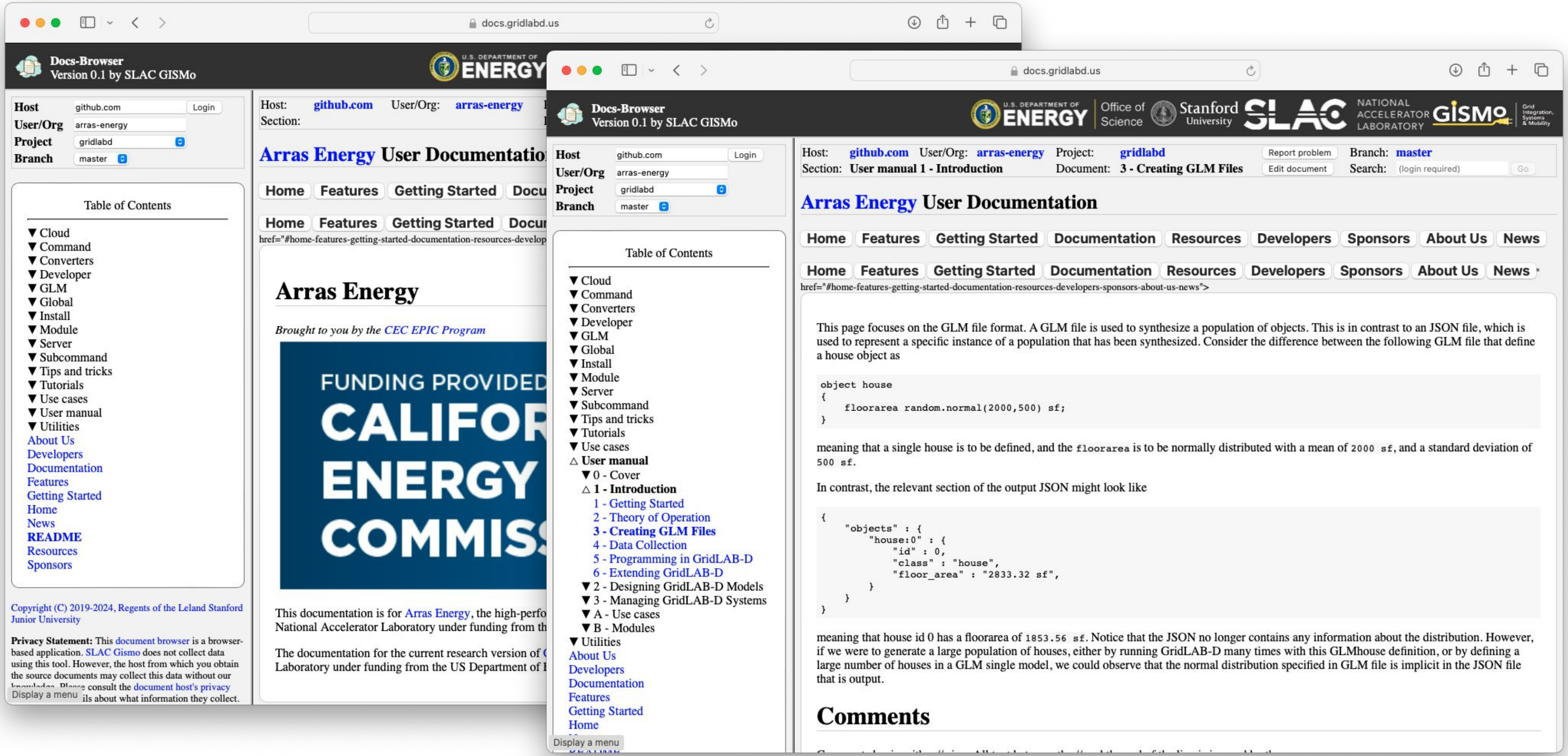
- Class event handlers
- Object event handlers
- Full object property access
- Automatic unit conversion
- Fast property accessors

Other Python Integrations

- Require tool adds python modules
- Python/shell subcommands access venv
- OpenFIDO access to pipelines/workflows

LFENERGY

Online Documentation



Validation

Challenge

- ~500,000 lines of C, C++, and Python
- Dozens of external datasets
- Many use cases include bad outcomes
- Failures are expected/handled
- Emergent behavior by design

→ *Atypical procedure for a simulator*

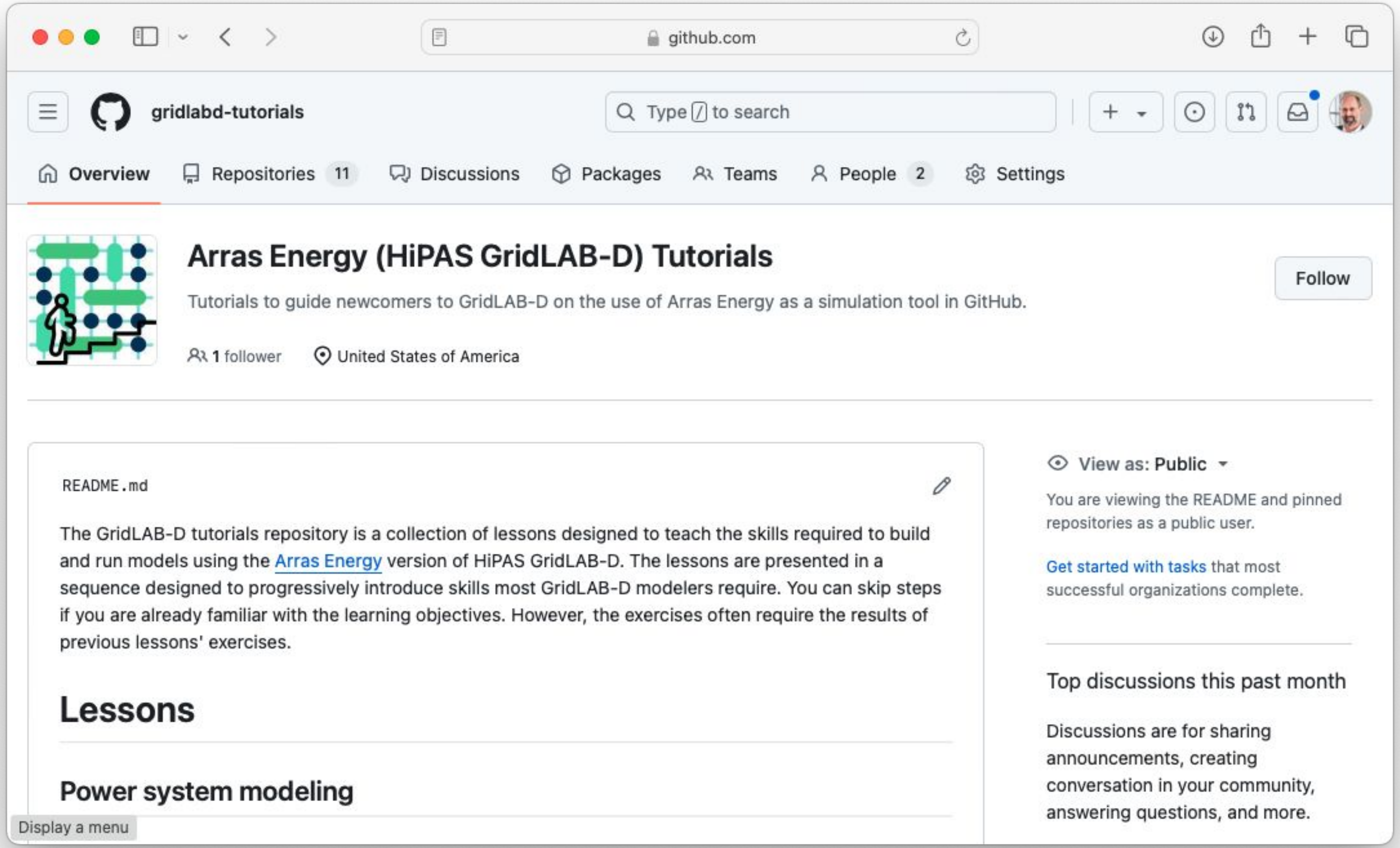
Approach

- ~750 test cases
- By module/class/feature
- By data set/provenance
- By use-case/scenario
- Includes expected error/exceptions

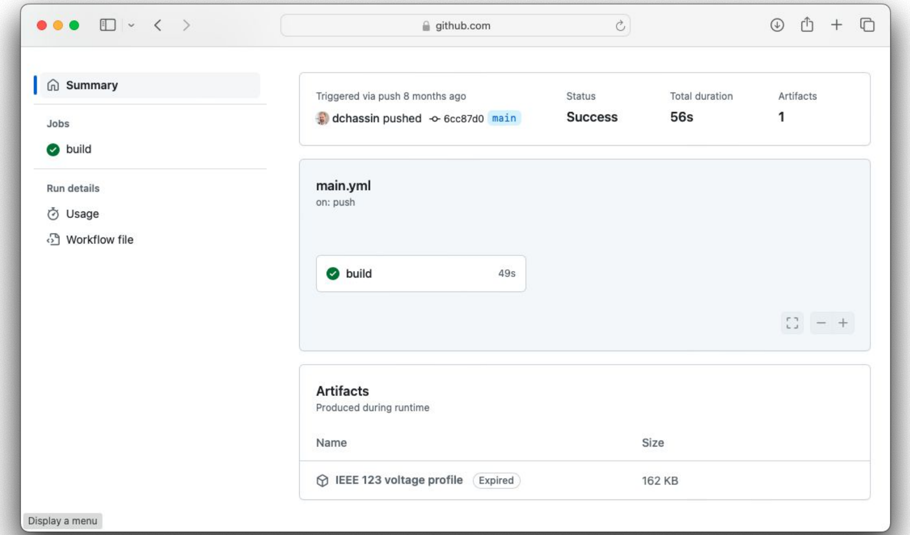
User Interfaces

- **GitHub Projects**
  - Easy setup and management of GridLAB-D models and simulation results
  - Ready-to-run template projects with github actions workflows
  - Online tutorials/howtos
- **OpenFIDO**
  - Developed by SLAC National Accelerator Laboratory
  - Funded by CEC as an open grid analysis environment for utilities and regulators
- **Marimo Notebook**
  - Developed by Akshay Agrawal, Stanford PhD student of Stephen Boyd
  - Funded by GISMo at SLAC and US Department
- **Hitachi GLOW**
  - Developed by Energy Solutions Lab, Hitachi America Ltd.
  - Funded by CEC as a commercial front-end to HiPAS GridLAB-D

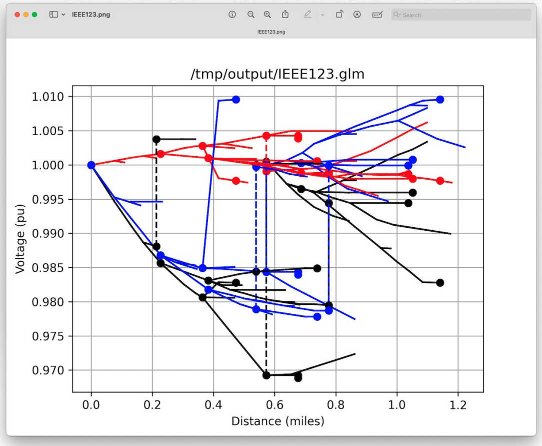
GitHub



github.com
gridlabd-tutorials
Type / to search
Overview
Repositories 11
Discussions
Packages
Teams
People 2
Settings
Arras Energy (HiPAS GridLAB-D) Tutorials
Tutorials to guide newcomers to GridLAB-D on the use of Arras Energy as a simulation tool in GitHub.
Follow
1 follower
United States of America
README.md
The GridLAB-D tutorials repository is a collection of lessons designed to teach the skills required to build and run models using the Arras Energy version of HiPAS GridLAB-D. The lessons are presented in a sequence designed to progressively introduce skills most GridLAB-D modelers require. You can skip steps if you are already familiar with the learning objectives. However, the exercises often require the results of previous lessons' exercises.
Lessons
Power system modeling
View as: Public
You are viewing the README and pinned repositories as a public user.
Get started with tasks that most successful organizations complete.
Top discussions this past month
Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.
Display a menu

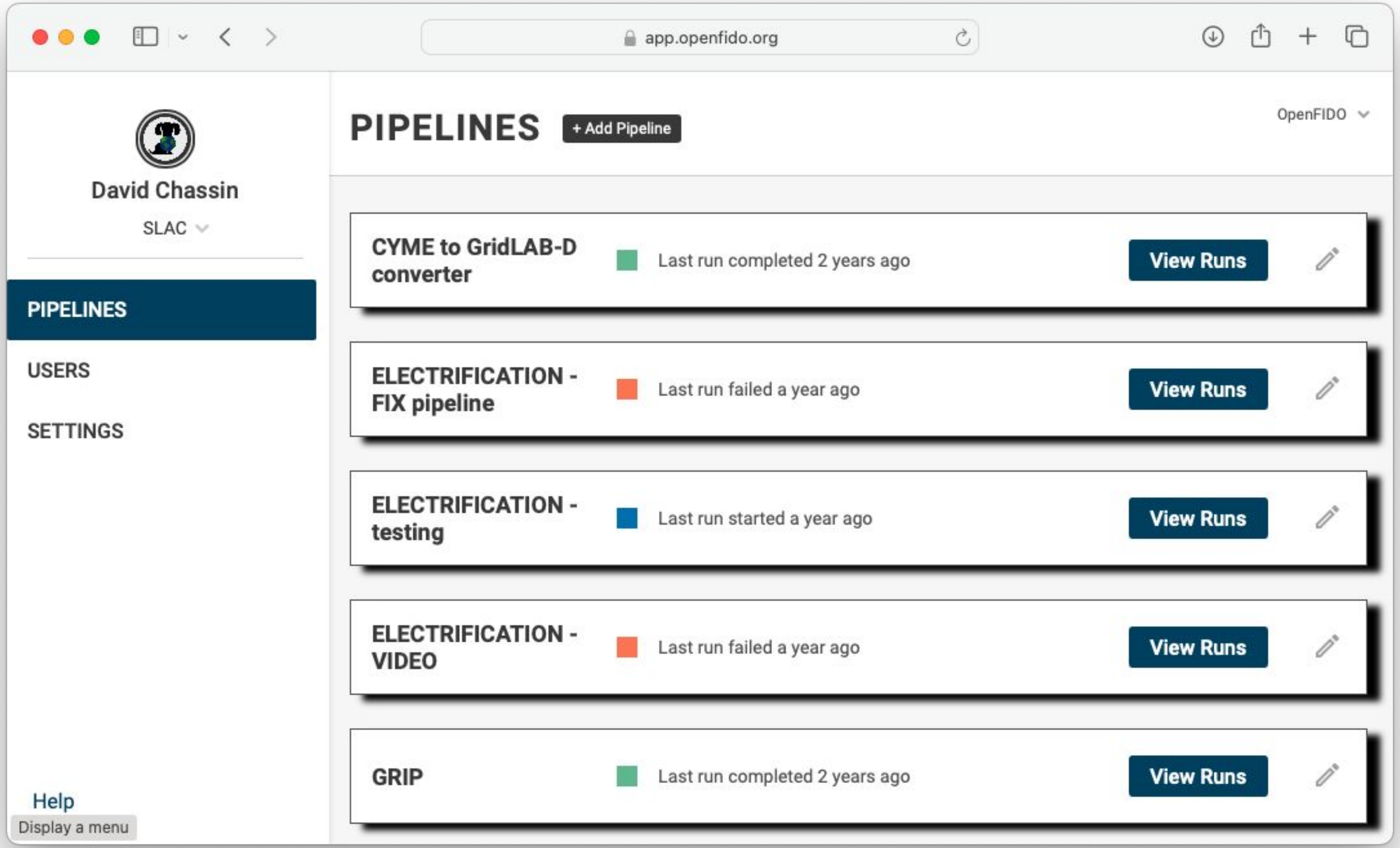


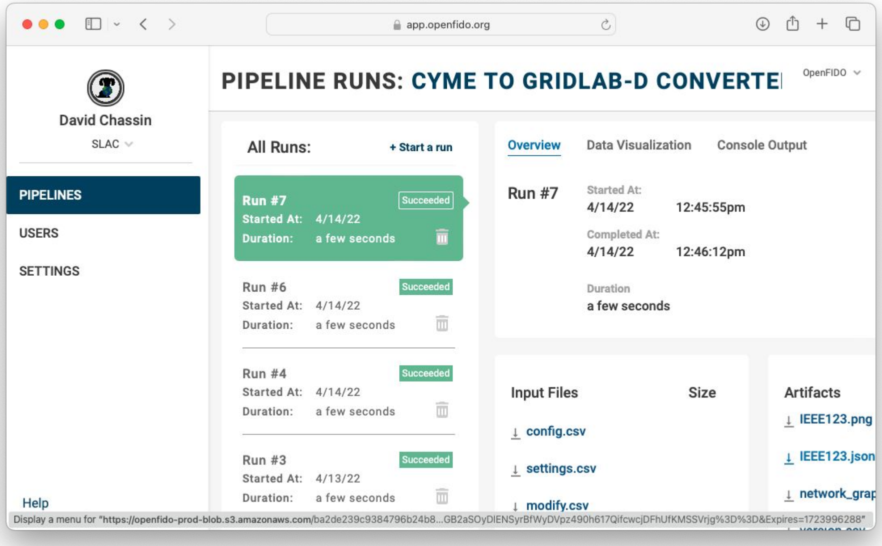
github.com
Summary
Jobs
build
Run details
Usage
Workflow file
Triggered via push 8 months ago
dchassin pushed 6cc87d0 main
Status
Success
Total duration
56s
Artifacts
1
main.yml
on: push
build
49s
Artifacts
Produced during runtime
Name
Size
IEEE 123 voltage profile
Expired
162 KB
Display a menu

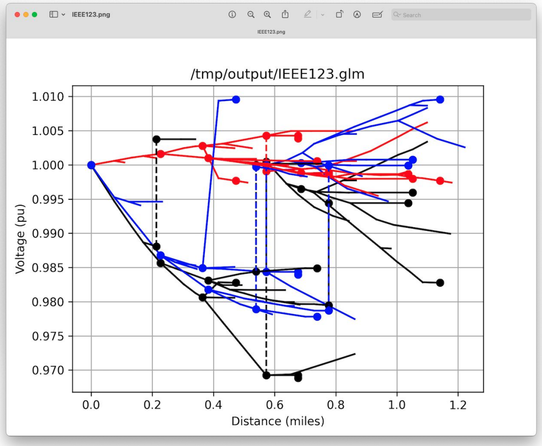


/tmp/output/IEEE123.glm
Voltage (pu)
Distance (miles)
1.010
1.005
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.0
0.2
0.4
0.6
0.8
1.0
1.2

OpenFIDO







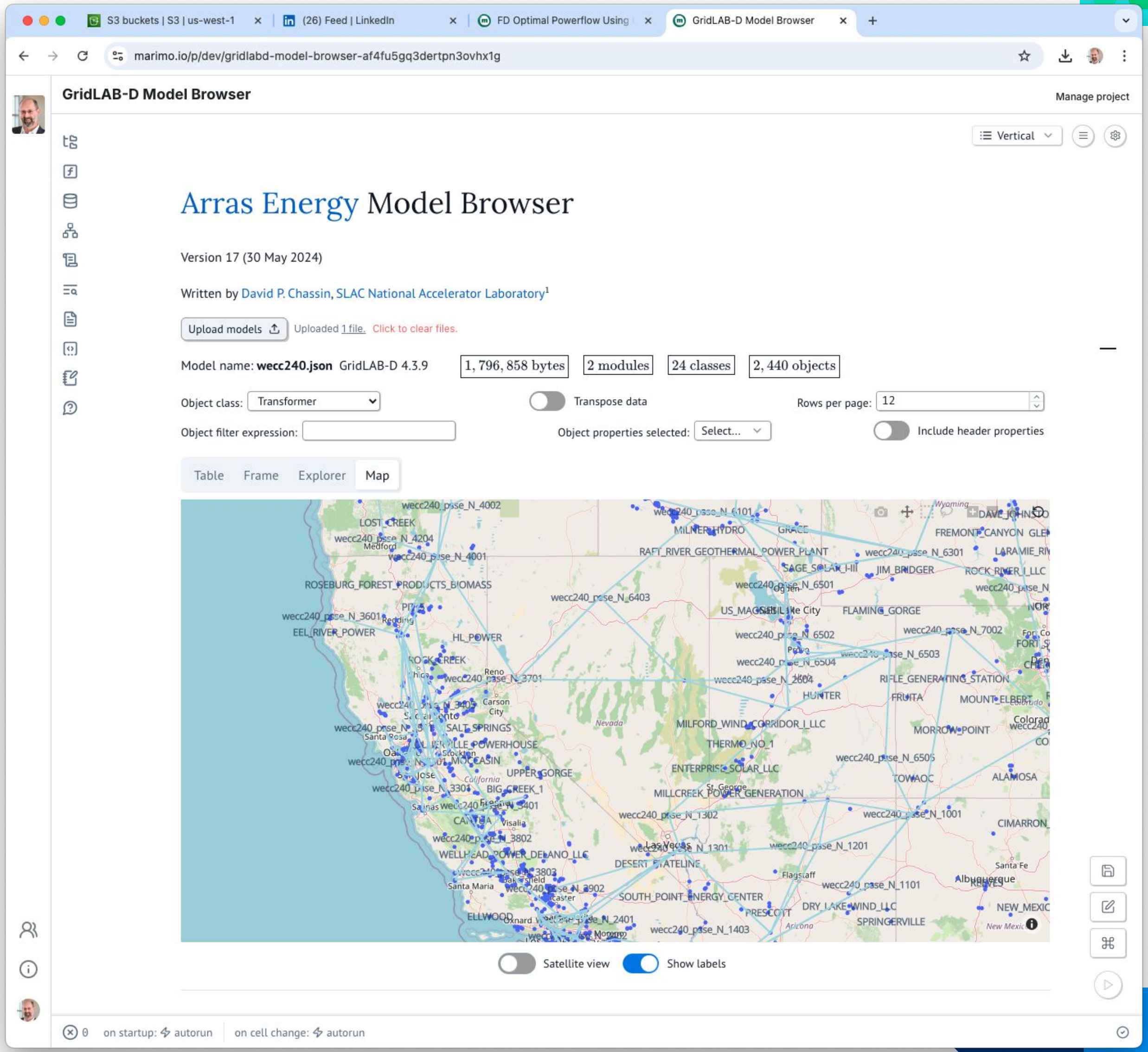
LF ENERGY

Marimo Notebook

A new kind of Python notebook

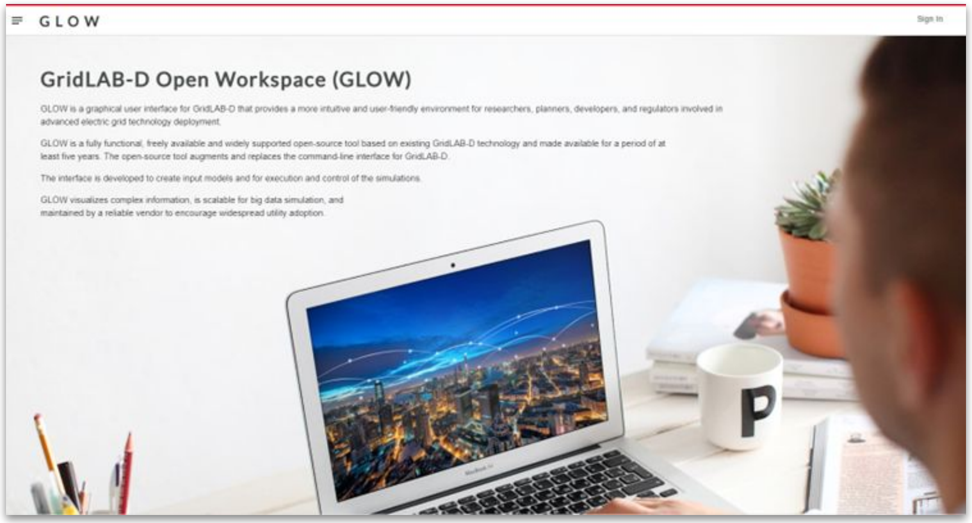
- Reactive notebook
- Tracks and manages data flows
- Maintains notebook consistency
- Addresses notebook problems
  - Reproducibility
  - Maintainability
  - Interactivity
  - Reusability
  - Shareability

Try it: https://tinyurl.com/yer2uzde



GLOW (GridLAB-D Open Workspace)

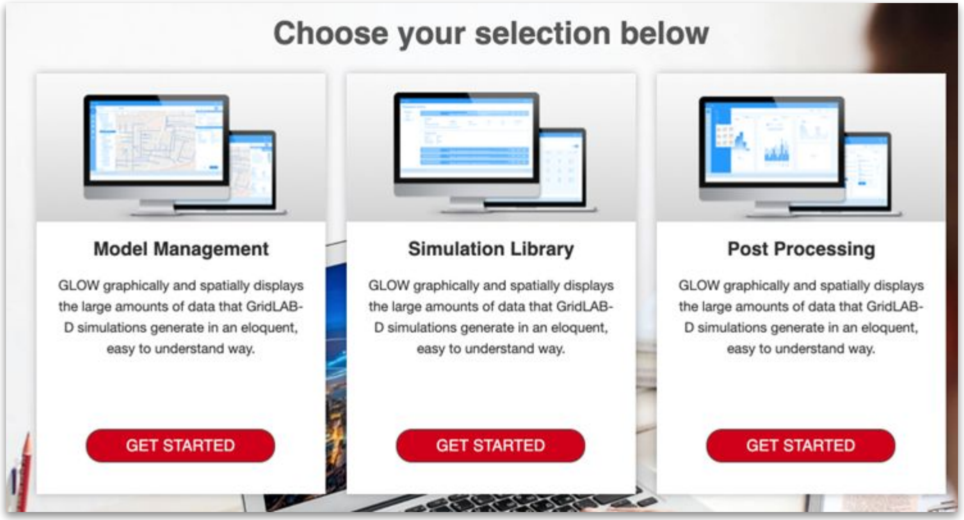
- **Key Features**
  - Web-based GUI for Arras Energy (HiPAS GridLAB-D)
  - An open-source platform
  - Designed for distribution resources planning
  - Support deployment on workstation/cloud
  - A user can register through GLOW website to access both versions of GLOW.
  - Use cases:
    - Power flow
    - Electrification
    - Integration Capacity Analysis (ICA)



- GLOW is developed by Hitachi America Ltd. with support from the California Energy Commission
- GLOW is using HiPAS GridLAB-D 4.3.1-220805 as a simulation engine.
- Arras Energy is a commercial-grade release of HiPAS GridLAB-D developed by the California Energy Commission

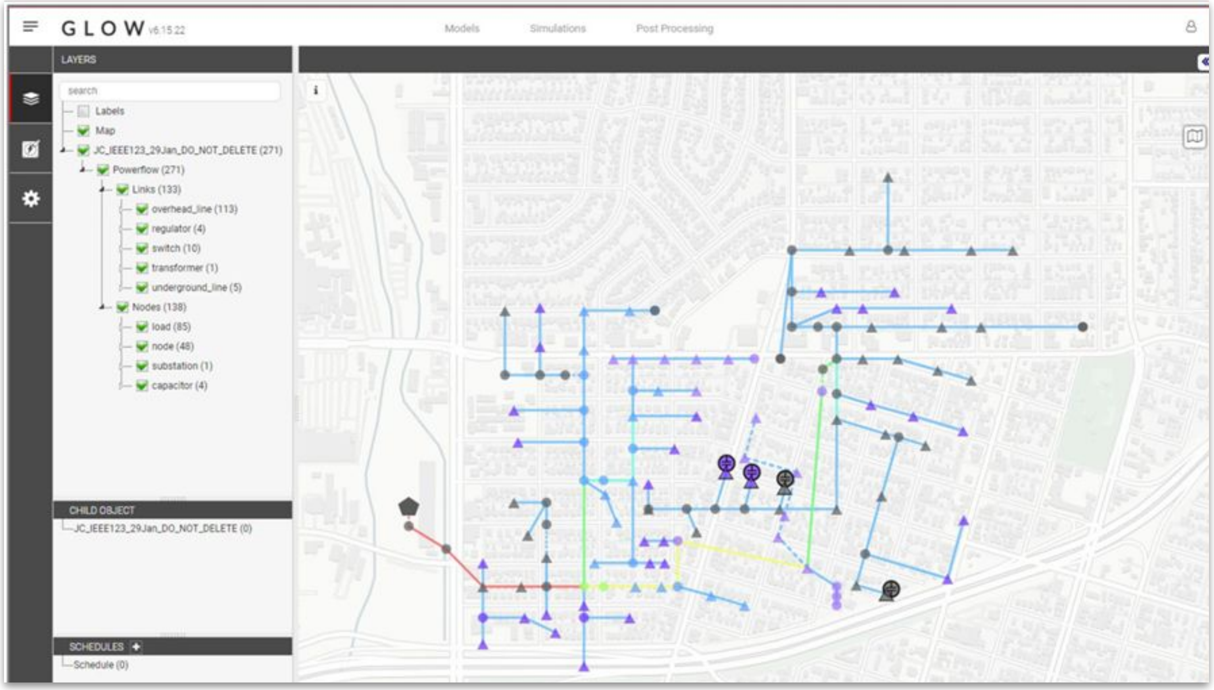
GLOW (GridLAB-D Open Workspace)

- **Other Features**
  - Post-processing for result realization
  - Generate feeder topology for visualization
  - Designed for cross-organizational collaborations
- **Benefit**
  - Attract more user community
  - Facilitate the adoption of DER integration
  - Facilitate decision-making process

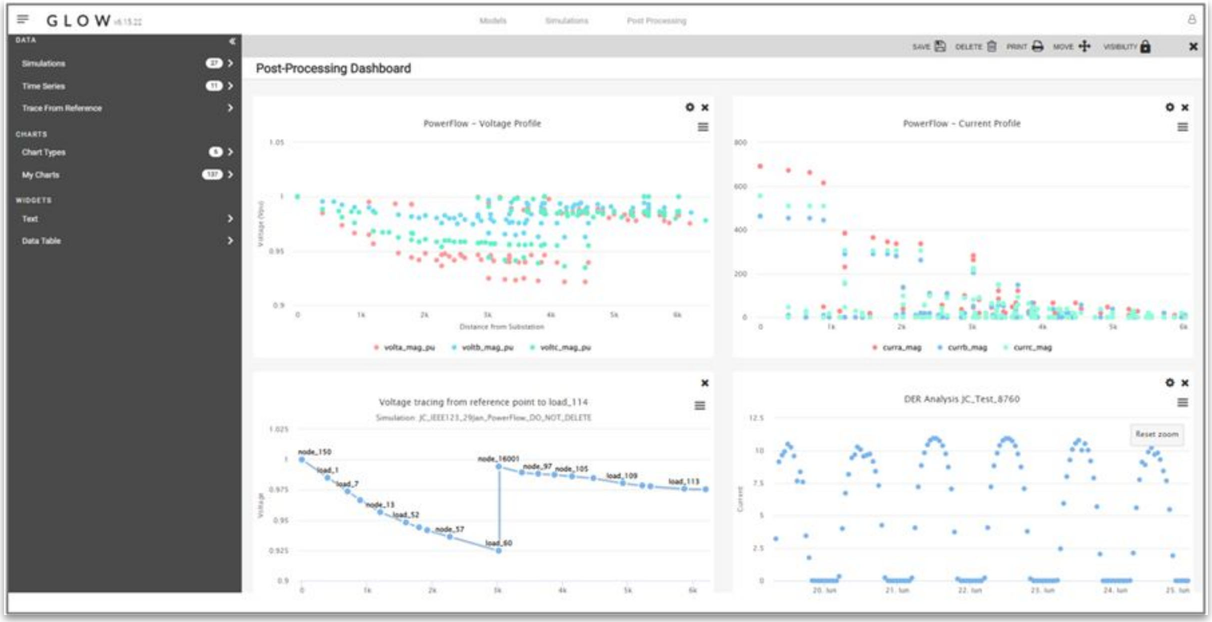


GLOW (GridLAB-D Open Workspace)

Power Flow Result in Viewer



Power Flow Result in Post-Processing



Open Source Challenges

Open source software challenges

- Utilities don't know what "open source" means/implies
- Community difficult to build and sustain
- Utility use-case validation is *ad hoc*
- Code/data provenance may be uncertain
- Utilities prefer white lists

LF ENERGY ARRAS

Arras Energy - 21st Century Electricity System Analysis Tools for 21st Century Electricity Utilities

https://arras.energy/

"Open source" is not like commercial products

- **Funding agencies and utilities still think open-source mean free**
  - Users can be surprised by license agreements, support fees, and runtime costs
- **Many types of OSS licenses**
  - Some are not user-friendly
  - Some are not developer-friendly
  - Government rights are often overlooked
- **Documentation may not be well-suited for utility needs**
  - Ad hoc / inline documentation
  - Lack of utility user-friendly training/tutorial/troubleshooting docs
  - Limited support may be available for some areas of tools/modules/data

LF ENERGY

It takes a community / Utility Support Model

- **Upgrades**
  - New features often introduced without enough review/input
  - Feature creep is a constant threat to well-documented code, testing, and validation
  - Controlled IT process can be incompatible with uncontrolled OSS distribution models
- **Maintenance**
  - Module updates are a constant source of testing and validation failures
  - Many updates are not backward compatible and break result reproducibility
  - Subtle "fixes" can break utility confidence if they get past testing/validation
- **Security Fixes**
  - Too many security alerts to respond to in a timely manner
  - Some security alerts don't have any fix available
  - Some security fixes can cripple existing capabilities

What do you want GridLAB-D for?

Utility use-cases and validation methods vary widely

- Emerging use-cases are ad hoc and often ill-defined
  - Often driven by local business/regulatory needs/mandates
  - Lack methodologies, data, and acceptance criteria
- Lack of standards for use-case definitions
  - IEEE committees work on these, but they're slow and lack industry participation
- Use-case validation datasets
  - Data sharing agreements are very difficult to obtain
  - Sensitive network/customer data is often necessary
  - Testing and validation on utility sandbox systems with limited access

Whose code is that?

The code provenance problem

- Modularity is an essential feature of OSS → complex dependencies
- Updates are critical but sometimes poorly supported/executed
- Code review/validation process increasingly complex and unwieldy
- Data more like code (think AI) → not handled in OSS the same way
- The xz "near miss" is much worse than we think

Utility white listing

"Everything is forbidden unless expressly permitted" is not OSS-friendly

- Utilities require a white-list of IP addresses and/or URLs
- GridLAB-D white list contains many URLs
- Some required modules have constantly changing URLs
- Some GridLAB-D resources do not have a fixed IPs

LFENERGY

Summaries: Pros and Cons of OSS in Energy

Benefits

- Cost-effectiveness
- Flexibility and customization
- Collaborative community
- Transparency
- Security *

Challenges

- Lack of dedicated support
- Hidden costs
- Fragmentation
- Compatibility
- IP and licensing

* Fast updates are a mixed blessing due to long utility security review cycles

LF ENERGY

For more information

Website	https://www.arras.energy/
Docker image	lfenergy/arras:latest
Source code	https://arras.energy/
GLOW	https://glow.hero-energy.com
GLOW Video	https://youtu.be/ep70nKCPct4
Documentation	https://docs.gridlabd.us/
Tutorials	https://github.com/gridlabd-tutorials
Contact	dpchassin@gmail.com

Download this presentation

LFENERGY