Project Background

Projects are governed individually by their own Technical Steering Committee (TSC). Each TSC lead or chair attends the Technical Advisory Council (TAC).

- *Resources:*
  - *Overview of All Projects Community Insights*
  - *LF Energy Community Calendar*

OperatorFabric

lfenergy.org/projects/operatorfabric/

A modular, extensible, industrial-strength and field-tested platform for use in electricity, water and other utility operations, OperatorFabric is a smart assistant for system operators.

OperatorFabric:

- was initiated by RTE, and contributed to LF Energy in 2019
- and all its sub-projects are licensed under Mozilla Public License V2.0
- is written in Java and based on the Spring framework. This simplifies writing and integrating better-coordinated software. It is developed using continuous integration and continuous delivery (CI/CD) through Gradle, Travis CI, and Sonarcloud.
- falls in the Shared section of LFE's High-Level Architecture, under Unified Operators UX Components and Framework
  - This section focuses on the digital functionalities supporting operators in their interaction with systems and stakeholders.

- **Capabilities**
  - System visualization and console integration
  - Precise alerting
  - Workflow scheduling
  - Remedial action manager
  - Historian
  - Scripting (ex: Python, JavaScript)
  - Reference implementation: Let's Coordinate

- **6-month Roadmap**
  - Technical Roadmap

      - Lighten the software architecture to simplify development and deployment
      - Upgrade the technical stack
      - Implement API testing using KARATE DSL
    - Functional Roadmap
      - Propose new features such as card routing, sound playing on card's arrival, triggering actions, etc.
      - Implement end user tests using tools such as Cypres and Selinium
      - Enhance the UX experience
- **Resources:**
  - OperatorFabric Community Insights
  - Project Logo
  - Webinar
  - Demo video
  - GitHub
  - Wiki
  - Releases

PowSyBl

lfenergy.org/projects/powsybl/

*PowSyBl (**P**ower **S**ystem **B**locks) is an open source library dedicated to electrical grid modeling and simulation. Using PowSyBl, developers can create applications able to perform dynamic power flow simulations and security analyses on the network, handle a variety of formats including CGMES for European data exchanges, etc. PowSyBl has a completely modular design, enabling developers to extend or customize its features by providing their own plugins. Modularity is a key value of PowSyBl's open source design.*

PowSyBl:

- was initiated by RTE and contributed to LF Energy in 2019
- is licensed under the Mozilla Public License 2.0
- falls in the Power System Calculation section of LFE's High-Level Architecture, under System Management
  - This section focuses on the digital functionalities used to calculate targets or variables related to power delivery and system stability.
- all the features are exposed as web services, so as to make it easy to build web-based applications on top of the framework.

- **Capabilities**
  - **Features:**
    - Grid data model, described with Java classes. Has the possibility to define extensions with plugins
    - Data management system
    - Importers and exporters for several formats (CIM, CGMES, UCTE, etc.)
    - APIs to various computation modules (load-flow, security analysis, short-circuit computation, sensitivity computation, optimizers, AMPL)
    - Distribution framework for HPC (tested on a 10,000 cores platform)

        - JavaFX user interface framework
        - Scripting
        - Advanced functions: grid model merging, remedial actions manager
        - Modular architecture based on plugins
    - **Provides several APIs for power system simulations and analyses, including:**
        - Power flow computations
        - Security analyses
        - Remedial action simulations
        - Short circuits computations
        - Sensitivity computations
        - Time domain simulations
        - These simulations can run on a supercomputer -- this has been tested using the CORDIS iTesla project with the Curie supercomputer.

- **6-month Roadmap**
  See further details about each section here.
    - Documentation
    - Demonstrator
    - Converters
    - Grid modeling
    - Simulators
    - Data management
    - Viewing
    - High level services
    - Functional tests

- **Resources:**
    - PowSyBi Community Insights
    - Project Logo
    - Landing page
    - Webinar
    - Demo
    - GitHub
    - Wiki
    - Use cases

RIAPS

lfenergy.org/projects/riaps/

The goal of RIAPS (The Resilient Information Architecture Platform for Smart Grid) is to provide a run-time and design-time software environment for building applications that execute on the Smart Grid. The main distinguishing characteristic of RIAPS is its completely distributed computing model: software applications are distributed across a multitude of computing nodes attached to a communication network, and each node has *access to local measurements and actuators. Applications include, but are not limited to: monitoring and control, data collection and processing, microgrids, energy management, transactive energy, and safety applications.*

RIAPS:

- was developed as an ARPA-E project under Award Number DE-AR0000666 at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University, in coordination with NCSU's FREEDM Center and Washington State University. RIAPS was contributed to LF Energy in 2019.
- RIAPS falls in the Edge Node Control section of LFE's High-Level Architecture, under Acquisition and Control
  - This section focuses on the digital functionalities that are shared amongst all nodes.

- **Capabilities**
  - The RIAPS technology stack features:
    - A component-oriented programming model for distributed real-time software running on embedded nodes dispersed throughout the grid or microgrid
    - Support for low-latency, hard real-time applications via a low-overhead messaging layer
    - Services for application management, fault tolerance, security, high-precision time synchronization, distributed coordination
    - Uniform device access with support for various industrial protocols
    - A development toolkit for developing and deploying apps
    - Implementation languages include Python, C++, Simulink/Stateflow, and others
  - Detailed feature descriptions can be found here.

- **6-month Roadmap**
  See further details about each section here.
  - Apps
  - Interfaces
  - Platform Ports
  - Features

- **Resources:**
  - RIAPs Community Insights
  - Project Logo
  - Landing page
  - Webinar
  - Demo video
  - GitHub
  - Wiki
  - Further project background
  - Publications

OpenEEMeter
lfenergy.org/projects/openeemeter/

OpenEEmeter is an open source toolkit for implementing and developing standard methods for calculating normalized metered energy consumption (NMEC) and avoiding energy use. The OpenEEmeter library contains routines for estimating energy efficiency savings at the meter.

Using the OpenEEmeter, private companies, utilities, and regulators can consistently calculate changes in energy consumption for building efficiency projects and portfolios with confidence in the methods and replicability of results.

The OpenEEmeter generates consistent and replicable results by always using the same methods to determine changes in energy consumption-there are no discretionary independent variables that change from calculation to calculation. Site level changes in consumption will reflect the same underlying methods across programs and implementations.

OpenEEMeter

- was contributed by Recurve, formerly Open Energy Efficiency, in 2019.
- falls in the Customer Relationship & Communication section of LFE's High-Level Architecture, under Customer & Market
    - This section focuses on the digital functionalities supporting customer relationship management and communication.

- **Capabilities**
    - Contains reference implementations of standard CalTRACK methods
    - Enforces standards compliance by incorporating data sufficiency checking and first-class warnings reporting
    - Facilitates integration with external systems and testing of methodological variations with modular design
    - Uses public weather sources by default, but allows for flexibility
    - Is built on top of the popular python scientific stack (scipy/pandas)
    - Includes visualization and debugging tools

- **6-month Roadmap**
  See further details about each section here.
    - Community goals: we want help our community thrive and continue to grow
        - Develop project documentation and tutorials
        - Make it easier to contribute
    - Technical goals: we want to keep building the library in new ways that make it as easy as possible to use
        - Implement new CalTRACK recommendations
        - Hourly model visualizations
        - Weather normal and unusual scenarios
        - Greater weather coverage

- **Resources:**
  - OpenEEmeter Community Insights
  - Project Logo
  - Landing page
  - Tutorial
  - Webinar
  - GitHub
  - Wiki
  - Other
    - Energy Market Methods Consortium
    - CalTRACK
    - Project History

EM2

lfenergy.org/projects/em2/

The Energy Market Methods Consortium (EM2) is an energy industry collaboration to create and curate market methods and algorithms, with a focus on distributed energy resource grid impacts, regulatory planning, and data privacy. EM2 has developed and is maintaining and improving standardized methods, linked to open source code, to enable demand flexibility as a resource, supporting energy programs and distributed energy resource (DER) markets.

EM2:

- was contributed by Recurve, formerly Open Energy Efficiency, in 2019.
- is a project affiliated with LF Energy with governance mirroring the Joint Development Foundation (JDF), which is also part of the Linux Foundation. Governance is described in the charter.
- falls in the Metering and Compensation section of LFE's High-Level Architecture, under Customer & Market
  - This section focuses on the digital functionalities supporting determination and financially handling realization of market contracts and consequences of system operation.

- **Capabilities/Process**
  - The methods development process is split into three working groups, each representing a core challenge in the development of scalable energy markets on the demand side. The core tenets uniting the methods development processes are transparency, empirical testing, and consensus. The three working groups are focused on standard methods for:
    - CalTRACK – calculating avoided energy use
    - GRID – adjustments to avoided energy use for grid integration
    - SEAT – enabling secure data sharing

- **Resources:**
  - EM2 Membership Structure (distinct from LF Energy)
  - Project Logo
  - Landing page
  - Webinar

  - Github
  - Wiki
  - Other:
    - Energy Market Methods Consortium
    - CalTRACK
    - GRID
    - SEAT

CoMPAS

lfenergy.org/projects/compas/

The CoMPAS (Configuration Modules for Power industry Automation Systems) project develops open source software components related to IEC 61850 model implementation (profile management) and configuration of a power industry Protection Automation and Control System (PACS). CoMPAS is designed to provide common software blocks for IEC 61850 profile configuration, using an open source shared development model.

The project seeks to:

1. *Leverage multi-vendor and multi-end-user development resources and 61850 competences to accelerate the development of common software blocks*
2. *Promote top-down configuration processes and common model implementation choices (thus also accelerate the conformity to IEC 61850 through software implementation);*
3. *Deliver a production grade and reference implementation of the standard.*

CoMPAS

- is the first-ever LF Energy project built from the ground up. It was launched in June 2020.
- falls in the Edge Node section of LFE's High-Level Architecture, under Acquisition and Control
  - This section focuses on the digital functionalities that are shared amongst all nodes.

- **Capabilities**
  - To ensure standardization and encourage broad adoption, CoMPAS's software components will be developed according to IEC 61850, an open, international standard that provides the framework to integrate a substation's PAC functions, regardless of the vendor or end-user. IEC 61850 can be complex to comply with, which may discourage independent companies from developing technology that abides by it. CoMPAS seeks to provide the standardized open-source software building blocks for PAC components vendors to use to create interoperable digital substation solutions.
  - The project does not duplicate standardization efforts. Instead, CoMPAS's goal is to deliver a production-grade reference implementation of the IEC standard.

- **6-month Roadmap**
  - Link to Roadmap
    - The initial roadmap of the project was drafted by a design team involving several grid operators and T&D automation vendors. See the initial roadmap here.

- **Resources:**
    - CoMPAS Community Insights
    - Project Logo
    - GitHub
    - Wiki
    - Other
        - Press Release

GXF

lfenergy.org/projects/gxf/

Formerly known as the Open Smart Grid Platform, Grid eXchange Fabric (GXF) allows you to monitor and control hardware in the public space. With several (generic) functions ready to use, the main benefits of GXF are: scalability & high availability, high security, its generic design, and no vendor lock-in. GXF is currently deployed in several public use cases, including microgrids, smart metering, public lighting, and distribution automation.

GXF

- was contributed by Alliander in February 2020.
- falls in the Central Hub and Infrastructure Management sections of LFE's High-Level Architecture, under Acquisition and Control
    - Central Hub: A central platform for data collection, monitoring and control equipment and nodes in the smart grid (e.g. SCADA or IoT platform).
    - Infrastructure Management: A central platform management equipment and nodes in the smart grid remotely.

- **Capabilities & Features**
    - a scalable and technology-agnostic industrial Internet of Things (IoT) platform that allows grid operators to securely collect data and monitor, control and manage smart devices on the grid
        - The architecture and implementation of GXF ensure scalability and availability.
    - Secure By Design:
        - GXF uses two-sided TLS for web applications to ensure that both parties trust each other's identity. All communication is encrypted.
    - Generic Design & Functions:
        - The core layer holds the generic functionality for device management, firmware management, time synchronization, device installation services, message routing to the correct device protocol, and other functions.
    - Domain Specific Functions:
        - GXF has several functional domains with domain-specific functions, including status updates, remote controls, automated events, and device filtering. For detailed information on these domains, see the documentation on Domains.

- **6-month Roadmap**
    - Transfer assets and governance to LF Energy

- Documentation
- Reimplement CLA using EasyCLA

- **Resources:**
  - GXF Community Insights
  - Project Logo
  - Landing page
  - GitHub
  - Wiki
  - Other
    - Press Release
    - FAQ
    - Use cases (here and here)