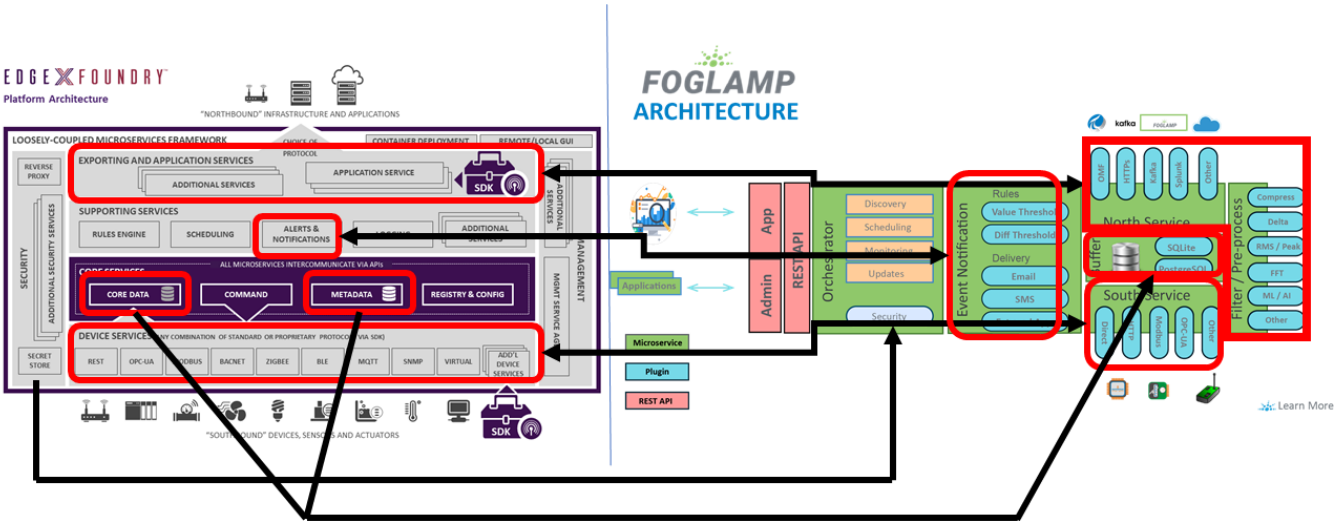
TAC Project Proposal Review Subgroup - FogLAMP

Subgroup Review Meetings

- June 4, 2019: Meeting Notes / Meeting Recording
- June 10, 2019: Meeting Notes / Meeting Recording



Functions (Provide, Consume, Facilitate, or N/A; Add context as needed)*

	Foglamp	EdgeX
APIs	Provide, Consume, Facilitate - Microcontroller Example - South Microservice (esp8266)	Provide
Cloud Connectivity	Provide, Consume and Facilitate - North repos, GCP, OSI-OMF	Provide
Container Runtime & Orchestration	Consume - (debian, rpm, docker, Project Eve)	Consume
Data Governance	Provide and Facilitate - Auditing logs and asset tracking.	
Data Models	Provide, Consume and Facilitate	
Device Connectivity	Provide, Consume and Facilitate	Provide
Filters/Pre-processing	Provide and Facilitate	Provide
Logging	Consume	Provide
Management UI	Provide, Consume and Facilitate	Facilitate
Messaging & Events	Provide, Consume and Facilitate	Provide
Notifications & Alerts	Provide, Consume and Facilitate	Provide
Security	Provide, Consume and Facilitate	Provide
Storage	Provide, Consume and Facilitate - Buffering	Provide

* as filled out by each project independently

Foglamp and EdgeX Foundry have significant functional overlap. Both can trace origins to a common reference architecture and both are largely attempting to serve many of the same IoT/edge use cases. There are technical differences in the two platforms (use in primary programming language and reference implementation persistent store for examples), but these were not explored in detail during the sub-group's review. It should be a comfort to both projects (and the LF Edge community) that two independent efforts solving some of the same edge needs arrived at implementations that have so much in common. In fact, of the functional differences between the platforms, many could be described as design choices that could be equally implemented in the alternate platform if given focus and a common use case (examples: deployment or local analytics and actuation).

Functional Commonalities

- Both contain a set of services that provide "south" (sensor, devices, things) and "north" (cloud, applications, enterprise systems, on-prem solutions, etc.) side connectors. South connectors connect the physical sensor to the platform via the protocol of the sensor and convey the sensor data to a nominal platform data schema for persistence. North connectors connect the platform to TCP/IP IT environments, applications or tools in a manner requested by the receiver.
- Both contain a central set of services that persist sensor collected data locally and offer metadata about the connected things and environment.
- Generally speaking, both Foglamp and EdgeX are dual transformation engines – converting IoT protocol data (Modbus, OPC-UA, etc.) to an internal platform representation and then transforming the data a second time (filtering, formatting, compressing, etc.) to deliver it to the doorstep of consuming systems, and applications.
- Both provide a scheduling service to initiate activities within the platform.
- Both platforms allow one instance of the platform to input to another instance of the platform (stacking instances).
- Both offer a notification service for alerting external parties/systems of an event via alternate delivery mechanisms (email, SMS, etc.).
- Both projects are microservice based – this is a comment on architecture and design versus functionality, but it does mean both projects support the idea of replacement components to provide 3rd party value adds and extensions (potentially even proprietary ones).
- Both offer REST API around each service – again, while this is a technical implementation detail, it provides users of either platform an easy to use means to query, monitor, configure, etc. the services that are part of the platform.

Functional Differences

- Foglamp targets Linux distributions. EdgeX is platform (hardware and OS) independent.
- Foglamp comes with orchestration and deployment facilities to get Foglamp to its host platform (as well as monitor and update it). EdgeX had intentionally steered clear of providing these services and instead provide tooling and packaging to help facilitate a user's own choice in orchestration and deployment.
- Foglamp provides monitoring tools and user interfaces, where EdgeX offers system management monitoring APIs that can be used by any monitoring system.
- Foglamp offers a built-in graphical user interface for exploring data collected, monitoring/managing assets, etc. EdgeX provides user interfaces but has been careful to describe these as demonstration and POC UIs – serving more as example functionality.
- Foglamp is built for multi-tenancy. EdgeX does not yet support multi-tenancy.
- EdgeX provides SDKs for creating new south side and north side connector services. The SDKs help provide all the scaffolding code to build a new connector and simplify connector creation. Foglamp south and north side connectors are plugin-based (versus service based) architectures and Foglamp provides documentation that helps template the construction of the new plugins.
- EdgeX offers services to actuate back onto devices and a replaceable rules engine to provide local analytics at the edge (turning sensor readings into local intelligence to provide for local command of devices).
- A technical detail that could be of interest to administrators and those looking to access the data repositories directly, Foglamp's persistence is SQL-based while EdgeX's persistence stores are NoSQL by default.

Difference (and commonality) in out-of-the-box south side connectors.

EdgeX	***Foglamp***
HTTP/REST	HTTP/REST
	CoAP
Modbus	Modbus
Modbus TCP	Modbus TCP
MQTT	MQTT
	MQTT Sparkplug B
OPC-UA	OPC-UA
	CSV
	Open Weather Map
	Director connectors for a variety of devices/sensors
SNMP	
ONVIF (Camera IP)	
BLE	
Random & Virtual (for test and demonstration)	Benchmark, Expression, Random and Sinusoid (for test and demonstration)