

# EdgeLake project proposal

Presented to the TAC: March 20, 2024

Subgroup reviewed on: March 27, 2024

Subgroup readout to the TAC: April 3, 2024

## Project Proposal - Project Introduction:

Required Information	Responses (Please list N/A if not applicable)
Name of Project	EdgeLake
Project Description (what it does, why it is valuable, origin and history)	<p>EdgeLake provides a software solution to manage and store data at edge nodes such that: a) distributed edge data appears as a unified collection of data (without centralizing the data), b) data processes at the edge are fully automated (for example: automated schema creation) and c) the distributed edge nodes appear and managed from a single point as a single machine.</p> <p>The EdgeLake community intends to provide this software under an OSI-approved open-source license, led by a community run under open governance principles. This project is a fork of AnyLog's Network.</p>
Statement on alignment with <a href="#">Foundation Mission Statement</a>	We agree with the foundation's mission statement.
High level assessment of project synergy with existing projects under LF Edge, including how the project compliments/overlaps with existing projects, and potential ways to harmonize over time. Responses may be included both here and/or in accompanying documentation.	<p>Open Horizon (OH) is incorporating EdgeLake. As OH is deploying and orchestrating software at the edge (from a single point), and AnyLog manages and queries data from the distributed edge - for many projects - OH + EdgeLake replace the need in the cloud and for all other use cases lowers cloud dependency, and automates data management at the edge.</p> <p>Projects EdgeX Foundry, Akraino, Hyperledger, are interested in integration with EdgeLake. AnyLog brings all these projects together: EdgeX Foundry serves the data to AnyLog Nodes (rather than cloud nodes) and a project like Smart-Transactions in Akraino queries Edge Data (as if the data is centralized), whereas data remains in-place. The blockchain is a key component in EdgeLake and Hyperledger is interested in providing the blockchain functionality.</p> <p>Additional Info: <a href="#">Value Prop</a> and <a href="#">a Speed Layer at the Edge</a>. See also diagrams below.</p>
Link to <i>current</i> Code of Conduct	We will adopt LF Edge's Code of Conduct.
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on <a href="#">Project Stages: Definitions and Expectations</a>	Joe Pearson (Open Horizon) Request is pending
Project license	<a href="#">Mozilla Public License, Version 2.0</a>
Source control (GitHub by default)	<a href="https://github.com/EdgeLake">https://github.com/EdgeLake</a> - Being used in Open-Horizon, currently private, will make public as requested in the process.
Issue tracker (GitHub by default)	<a href="https://github.com/EdgeLake/EdgeLake/issues">https://github.com/EdgeLake/EdgeLake/issues</a>
External dependencies (including licenses)	<a href="https://github.com/EdgeLake/EdgeLake/network/dependencies">https://github.com/EdgeLake/EdgeLake/network/dependencies</a>
Release methodology and mechanics	GitHub Releases
Names of initial committers, if different from those submitting proposal	Moshe Shadmon, Mark Davidson, Faisal Nawab
Current number of code contributors to proposed project	4
Current number of organizations contributing to proposed project	1
Briefly describe the project's leadership team and decision-making process	EdgeLake is involved in Open-Horizon features request subgroup and has been following their Technical Charter.
List of project's official communication channels (slack, irc, mailing lists)	Slack & mailing lists
Link to project's website	In construction
Links to social media accounts	In construction
Existing financial sponsorship	AnyLog

Infrastructure needs or requests (to include GitHub /Gerrit, CI/CD, Jenkins, Nexus, JIRA, other ...)	EdgeLake needs a place to host its community supported plugins. Access to development/test hardware to support other architectures (RISC-V).
Currently Supported Architecture	x86/64, ARM
Planned Architecture Support	Generic
Project logo in svg format (see <a href="https://github.com/lf-edge/lfedge-landscape#logos">https://github.com/lf-edge/lfedge-landscape#logos</a> for guidelines)	TBD
Trademark status	Trademark will need to be pursued by the Linux Foundation upon project proposal acceptance
Does the project have a Core Infrastructure Initiative security best practices badge? (See: <a href="https://bestpractices.coreinfrastructure.org">https://bestpractices.coreinfrastructure.org</a> )	No
Any additional information the TAC and Board should take into consideration when reviewing your proposal?	EdgeLake is an incubation project under Open Horizon and is ready to become its own standalone member project under LF Edge.

#### Project Proposal - Mapping Criteria and Data:

### Stage 1: At Large Projects (formerly 'Sandbox')

Criteria	Data
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on <a href="#">Project Stages: Definitions and Expectations</a>	
A presentation at an upcoming meeting of the TAC, in accordance with the project proposal requirements	
The typical IP Policy for Projects under the LF Edge Foundation is Apache 2.0 for Code Contributions, Developer Certificate of Origin (DCO) for new inbound contributions, and Creative Commons Attribution 4.0 International License for Documentation. Projects under outside licenses may still submit for consideration, subject to review/approval of the TAC and Board.	
Upon acceptance, At Large projects must list their status prominently on website/readme	

\*\*\* For existing Projects requesting Stage 2 or Stage 3 consideration, please update the above with the Stage 2 or Stage 3 Mapping criteria, available at [Project Stages Mapping: Criteria and Data](#)

#### Project Proposal - Taxonomy Data:

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

Functions	(Provide, Consume, Facilitate, or N/A; Add context as needed)
APIs	Provide, Consume EdgeLake is a protocol that Consumes and Service data using industry interfaces like REST, MQTT and SQL.
Cloud Connectivity	Provide - Cloud process can Query and Pull Edge Data.
Container Runtime & Orchestration	Consume
Data Governance	Consume
Data Models	Provide
Device Connectivity	Consume
Filters/Pre-processing	Consume, Provide, Facilitate
Logging	Provide
Management UI	Provide, Consume

Messaging & Events	Provide, Consume
Notifications & Alerts	Provide, Consume
Security	Provide, Consume
Storage	Provide, Consume

## Deployment & Industry Verticals (Support, Possible, N/A; Add context as needed)

Deployment Type	(Support, Possible, N/A; Add context as needed)
Customer Devices (Edge Nodes)	
Customer Premises (DC and Edge Gateways)	
Telco Network Edge (MEC and Far-MEC)	
Telco CO & Regional	
Cloud Edge & CDNs	
Public Cloud	
Private Cloud	

## Deployment & Industry Verticals ( or X; Add context as needed)

Directly applicable Industry/Verticals use cases	( or X; Add context as needed)
Automotive / Connected Car	
Chemicals	
Facilities / Building automation	
Consumer	
Manufacturing	
Metal & Mining	
Oil & Gas	
Pharma	
Health Care	
Power & Utilities	
Pulp & Paper	
Telco Operators	
Telco/Communications Service Provider (Network Equipment Provider)	
Transportation (asset tracking)	
Supply Chain	
Preventative Maintenance	
Water Utilities	
Security / Surveillance	
Retail / Commerce (physical point of sale with customers)	
Other - Please add if not listed above (please notify <a href="mailto:TAC-subgroup@lists.lfedge.org">TAC-subgroup@lists.lfedge.org</a> when you add one)	

## Deployments (static v dynamic, connectivity, physical placement) - ( or X; Add context as needed)

Use Cases	( or X; Add context as needed)
Gateways (to Cloud, to other placements)	
NFV Infrastructure	
Stationary during their entire usable life / Fixed placement edge constellations / Assume you always have connectivity and you don't need to store & forward.	
Stationary during active periods, but nomadic between activations (e.g., fixed access) / Not always assumed to have connectivity. Don't expect to store & forward.	
Mobile within a constrained and well-defined space (e.g., in a factory) / Expect to have intermittent connectivity and store & forward.	
Fully mobile (To include: Wearables and Connected Vehicles) / Bursts of connectivity and always store & forward.	

## Compute Stack Layers (architecture classification) - (Provide, Require, or N/A; Add context as needed)

Compute Stack Layers	(Provide, Require, or N/A; Add context as needed)
APIs	Provide
Applications	Require
Firmware	Require
Hardware	Require
Orchestration	Require
OS	Require
VM/Containers	Optional

## Cloud Stack Layers (architecture classification) - (Provide, Require, or N/A; Add context as needed)

Cloud Stack Layers	(Provide, Require, or N/A; Add context as needed)
Applications	Require - The infrastructure aims to service apps and AI
Configuration (drive)	Optional - different options are provided for configuration
Content (management system)	N/A
IaaS	Optional - we can be deployed on IaaS
PaaS	Optional - we can be deployed as PaaS
Physical Infrastructure	Require
SaaS	N/A

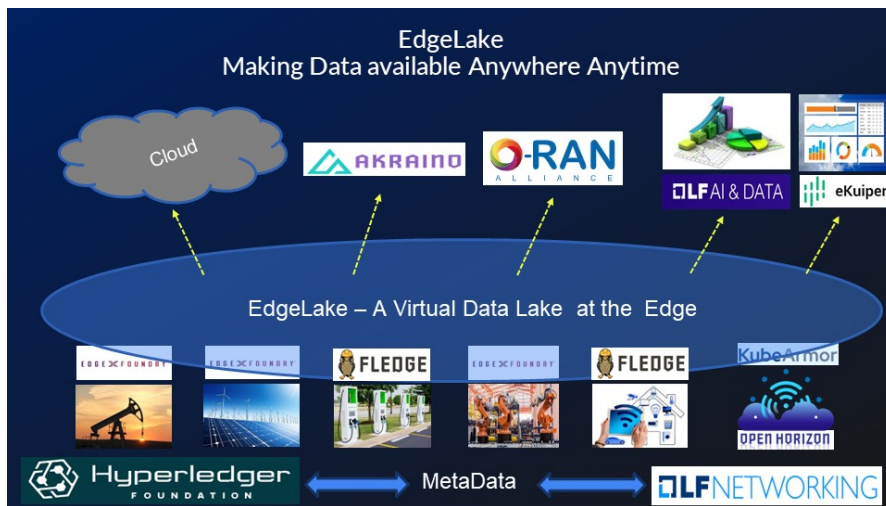
## Diagrams

Integration to LFEDGE Projects

EdgeLake

Making Edge Data Available Anywhere Anytime

Without Centralizing the Data



### EdgeLake & Open-Horizon Integration

