

Open Horizon

Status

- Current **Project Stage**: Stage 1 - At Large
 - **Website**: <https://www.lfedge.org/projects/openhorizon/>
 - **Wiki**: TBA
 - **TAC Sponsors**: [Joe Pearson](#) (IBM), [Frank Zdarsky](#) (Red Hat), [Jim St. Leger](#) (Intel), [Sukhdev Kapur](#) (Juniper), [Jim White](#) (EdgeX Foundry)
 - **Project Overview**: [Open Horizon Project Proposal for LF Edge TAC.pptx](#)
-
- Presented during the Wednesday, March 11, 2020 TAC call: Meeting Recording (https://zoom.us/rec/share/xetqDO7vqjJJT9bOtHHiCoQvH4n7X6a8hiRI8qdey0e7Y87S8KqUkg3JL_uk9pJq)
 - TAC two-thirds vote approval reached on Monday, March 30, 2020.
 - Governing Board Strategic Planning Committee approval reached on April 9, 2020.

Project Proposal - Project Introduction:

Required Information	Responses (Please list N/A if not applicable)
Name of Project	Open Horizon
Project Description (what it does, why it is valuable, origin and history)	Description Open Horizon is software to autonomously and securely manage the service software lifecycle of global or local fleets of edge compute nodes in various connected states. It interoperates with existing orchestration tools and conventions. It can be considered as a single component of a complete Cloud-to-Edge workload management solution; covering both services and related metadata, including machine learning assets. Value and Ethos • Autonomy – When fleets of devices number in the tens of thousands to millions and more, conventional software management techniques do not scale. Open Horizon provides a policy-driven mechanism for specifying device properties, business requirements, and service objectives and monitors those conditions continually for matches. • Security – The underlying philosophy is to minimize attack surfaces. Each Agent has authority only for its own device and each acts autonomously. All other components have tightly restricted scopes of authority, and none have authority over the Agents. Components know the Agents only by their public keys (not even their IP addresses) and all communications are encrypted between sender and receiver, so even the central switchboard component cannot read/alter/inject messages. Only signed and verified workloads are run. • Connectivity – Device nodes may not always be connected to the Internet. Software management must allow for partially- and completely-disconnected operations. History • IBM began Horizon development mid 2015, demo at IoT World Forum 12/2015 • Previously named Project Mountain, then Blue Horizon, now Open Horizon • IBM has a commercial offering, IBM Edge Application Manager, built with Open Horizon Focus & Technology • Targeting the Edge Gateway/Controller and Device tiers of Edge Computing • Works with Linux variants and OSX, on x86_64, and arm32v6 through arm64v8. Running the Agent only requires a single core. • Both the Agent and Docker together consume around 100 MB RAM, leaving excess memory for workloads on an edge node with 512MB RAM. • The Agent and Docker combined use about 400 MB storage on disk. • Written in Golang and Scala

Statement on alignment with Foundation Mission Statement	"LF Edge provides a neutral structure for building a diverse open source community capable of driving better, more secure development at the edge. It also unifies an aligned vision for the diverse and complex edge projects being built today." - from the LF Edge FAQ. In order to further the mission of LF Edge, Open Horizon's goals include: • To enlist partners as stakeholders • To build an active, thriving community of developer contributors • To establish our approaches as de facto standards and blueprints and speed industry adoption • To create an ecosystem of components and services from LF-contributed code to enable enterprise-ready solutions • To provide a migration path from individual projects to complete enterprise-ready solutions
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.	Whether thinking about component solutions that could be assembled for a complete Cloud-to-Edge workload management implementation, or projects that would work together to solve a specific use case, Open Horizon is designed to work together with other solutions in an open, standardized approach. Open Horizon is already incubating with EdgeX Foundry and building an integration project with them that will live in their holding repository. We are hoping to learn valuable techniques and processes from this effort that can be used to inform other integration and collaboration efforts within LF Edge going forward. Open Horizon works with EdgeX Foundry, and potentially Fledge, to deliver and manage their solution and to synchronize any needed ML assets with their origin. Open Horizon is planning to collaborate with Home Edge to synchronize ML assets from the cloud to the home gateway. Open Horizon could use Project EVE as a device management solution and to bootstrap the Open Horizon Agent. Open Horizon is in discussions with several Akraio Blueprint Family owners on contributing to their blueprints as a portion of their solutions in the areas of workload management on edge devices and delivery of assets in an ML deployment pipeline.
Link to <i>current</i> Code of Conduct	none yet, in progress
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	Joe Pearson, IBM Frank Zdarsky, Red Hat Jim St. Leger, Intel Sukhdev Kapur, Juniper Jim White, EdgeX Foundry
Project license	Apache 2.0 https://github.com/open-horizon/anax/blob/master/LICENSE.txt
Source control (GitHub by default)	https://github.com/open-horizon
Issue tracker (GitHub by default)	https://github.com/open-horizon
External dependencies (including licenses)	Auto-generated dependency lists: • Anax • Sync Service
Release methodology and mechanics	Agile with Kanban Scrums twice a week Two-week sprints ending with a scrum of scrums Releases when features complete, historically 3-6 months
Names of initial committers, if different from those submitting proposal	Joe Pearson, IBM David Booz, IBM Bruce Potter, IBM Carl Girouard, IBM Jon McGuire, IBM
Current number of code contributors to proposed project	14

Current number of organizations contributing to proposed project	One, IBM
Briefly describe the project's leadership team and decision-making process	The project leadership team is composed of the Project Chair, Chief Architect, Dev Lead, QA/Test Lead, DevOps Lead, and Offering Lead. At the beginning of a release cycle the backlog, roadmap, and any outstanding community requests are weighed to determine what can be reasonable delivered in the next release.
Preferred maturity level (see stages here)	Stage 1, aiming for Stage 2 in about six months or so
For Projects applying at the Growth (Phase 2) or Impact Stage (Phase 3), please outline how your project successfully meets/exceeds the requirements as defined under each category. Responses may be included both here and /or in accompanying documentation.	n/a
List of project's official communication channels (slack, irc, mailing lists)	using EXF's Slack and Groups.io for integration work, plan to use LF Edge Slack and Groups.io after acceptance at Stage 1
Link to project's website	under development
Links to social media accounts	none, will use LF Edge social media
Existing financial sponsorship	IBM
Infrastructure needs or requests (to include GitHub /Gerrit, CI/CD, Jenkins, Nexus, JIRA, other ...)	GitHub, Jenkins Pipelines, investigating additional solutions including Cloud-based offerings provided free to open source projects
Currently Supported Architecture	x86, ARM 32 & 64
Planned Architecture Support	none
Project logo in svg format (see https://github.com/lf-edge/lfedge-landscape#logos for guidelines)	https://github.com/open-horizon/artwork/blob/master/color/open-horizon-color.svg
Trademark status	n/a
Does the project have a Core Infrastructure Initiative security best practices badge? (See: https://bestpractices.coreinfrastructure.org)	no
Any additional information the TAC and Board should take into consideration when reviewing your proposal?	Open Horizon Info GitHub: https://github.com/open-horizon Open Horizon lives on the network edge, and typically on constrained devices. Open Horizon is designed to help place your software as close as possible to where the data is being created and right where actions need to be taken. A fully autonomous Open Horizon Agent runs on every edge device to orchestrate and manage the software lifecycles of your containers (e.g., download, verify, sandbox, run, restart, remove) on its device. Open Horizon is mostly decentralized, and designed for massive scale. Driven only by a software deployment pattern you have configured, or by policies (properties and constraints) that you have set, the Open Horizon Agent makes its own decisions about which software to run on its device. The Agent has a very small footprint, currently requiring less than 30MB of RAM at runtime, and supports small Linux amd64, arm64v8, and arv32v6 /v7 devices (with as little as 512MB RAM). Video resources: Introduction to Open Horizon (12:07): https://www.youtube.com/watch?v=NUFRKtn-ED0 Inter-component Communications (9:22): https://www.youtube.com/watch?v=vPTIWD-OY_w Patterns and Policies (9:11): https://www.youtube.com/watch?v=f7KBD17LbxY

Project Proposal - Mapping Criteria and Data:

Stage 1: At Large Project

Criteria	Data
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	Joe Pearson, IBM Jim St. Leger, Intel Jim White, EdgeX Foundry Frank Zdarsky, Red Hat Sukhdev Kapur, Juniper
A presentation at an upcoming meeting of the TAC, in accordance with the project proposal requirements	March 11, 2020 (link at top of page)
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.	Apache 2.0 for code CC-BY 4.0 for documentation DCO for all contributions
Upon acceptance, At Large projects must list their status prominently on website/readme	will do, no website yet

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
Cloud Connectivity	Consume
Container Runtime & Orchestration	Provide, Consume, Facilitate
Data Governance	N/A
Data Models	Consume, Facilitate
Device Connectivity	Consume
Filters/Pre-processing	N/A
Logging	Consume

Management UI	N/A
Messaging & Events	N/A
Notifications & Alerts	N/A
Security	Provide, Consume, Facilitate
Storage	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)	Support (Agent)
Customer Premises (DC and Edge Gateways)	Support (Agent, Management Hub)
Telco Network Edge (MEC and Far-MEC)	Support (Management Hub)
Telco CO & Regional	N/A
Cloud Edge & CDNs	N/A
Public Cloud	N/A
Private Cloud	N/A

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	X
Manufacturing	
Metal & Mining	
Oil & Gas	
Pharma	
Health Care	X
Power & Utilities	
Pulp & Paper	X
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 TAC-subgroup@lists.lfedge.org 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	X
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	Require
Applications	Require
Firmware	N/A
Hardware	Require
Orchestration	Provide
OS	Require
VM/Containers	Require

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
Configuration (drive)	Provide
Content (management system)	Provide
IaaS	N/A
PaaS	N/A
Physical Infrastructure	N/A
SaaS	N/A