

Project Nexoedge

Presented to the TAC on June 28, 2023 - [Recording](#), [Slides](#), [Minutes](#)

Subgroup Readout to the TAC on Wednesday, July 26, 2023 - [Recording](#), [Slides](#), [Minutes](#)

TAC Vote concluded on August 9, 2023 and have made the recommendation to the Governing Board Strategic Planning Committee to approve this project's proposal.

Required Information	Responses (Please list N/A if not applicable)
Name of Project	Nexoedge (previously named nEdge-nCloud)
Project Description (what it does, why it is valuable, origin and history)	<i>What does nEdge-nCloud do?</i> nEdge-nCloud is a reliable and efficient multi-cloud distributed storage which enables applications and devices at the edge to store data into one or more clouds. nEdge serves as a storage proxy at the edge to pre-process data for reliability and provide unified storage (over multiple cloud storage destinations) to edge applications. nEdge persists and access data in one or more clouds via nCloud, which is deployed in the clouds. <i>Why is nEdge-nCloud valuable?</i> Leveraging Cloud Storage with Ease at the Edge In edge deployment, IoT devices and applications generate a vast amount of data. nEdge-nCloud enables them to store data in a reliable and efficient storage pool of data across private and public clouds, e.g., for archival and backup. nEdge-nCloud supports utilizing multiple public cloud storage as storage destinations, e.g., AWS and Alibaba Cloud. It transparently manages data storage on behalf of the applications, freeing the applications from complex storage setup and management. nEdge-nCloud currently provides a standard network file interface, SMB, for file storage, and a socket-based client interface for custom application integration. Efficient and Configurable Data Reliability nEdge-nCloud adopts erasure coding to reduce the storage overhead and hence the cost compared to replication. nEdge-nCloud also provides configurable reliability which can be customized according to application storage requirements. It automatically repairs data to ensure data reliability. <i>Origin and History of nEdge-nCloud</i> nEdge-nCloud is a distributed storage solution co-developed by CU Coding Ltd. and CUHK ADSLab. Ideas of applying nEdge-nCloud to edge deployments, .e.g, smart factories, have won awards in several competitions, including the EdgeX China challenge 2022 and 5th "Bloom Cup" 5G Application Competition.
Statement on alignment with Foundation Mission Statement	nEdge-nCloud aligns with the LF Edge Mission Statement: • nEdge-nCloud is neutral to edge and cloud platform vendors, providing a unified cloud storage through standard storage interfaces to edge applications over a diversity of cloud storage. • nEdge-nCloud enables edge applications to reliably store data into the cloud in a cost-efficient manner. • nEdge-nCloud accelerates edge application development and innovation by offloading cloud-tier data management from applications logic.

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.	nEdge-nCloud can provide storage to projects that gather and manage the data from heterogeneous IoT devices, such as eKuiper, EdgeX, Fledge. One example would be storing video recordings and images of surveillance cameras from the edge to nEdge-nCloud.																									
Link to <i>current</i> Code of Conduct	https://lfprojects.org/policies/code-of-conduct/																									
2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	Sven van der Meer Tina Tsou																									
Project license	Apache 2.0																									
Source control (GitHub by default)	https://github.com/nexoedge/nexoedge																									
Issue tracker (GitHub by default)	https://github.com/nexoedge/nexoedge																									
External dependencies (including licenses)	<table><tr><td>Aliyun OSS C SDK</td><td>MIT</td></tr><tr><td>AWS CPP SDK</td><td>Apache 2.0</td></tr><tr><td>Azure Storage CPP SDK</td><td>Apache 2.0</td></tr><tr><td>Boost</td><td>Boost 1.0</td></tr><tr><td>Cpp-netlib</td><td>Boost 1.0</td></tr><tr><td>Google glog</td><td>Free BSD</td></tr><tr><td>Hiredis</td><td>FreeBSD</td></tr><tr><td>Intel Storage Acceleration Library</td><td>FreeBSD</td></tr><tr><td>Nlohmann JSON</td><td>MIT</td></tr><tr><td>OpenSSL 3.0</td><td>Apache 2.0</td></tr><tr><td>ZeroMQ</td><td>MPL 2.0</td></tr><tr><td>ZeroMQ CPP</td><td>Apache 2.0</td></tr></table>		Aliyun OSS C SDK	MIT	AWS CPP SDK	Apache 2.0	Azure Storage CPP SDK	Apache 2.0	Boost	Boost 1.0	Cpp-netlib	Boost 1.0	Google glog	Free BSD	Hiredis	FreeBSD	Intel Storage Acceleration Library	FreeBSD	Nlohmann JSON	MIT	OpenSSL 3.0	Apache 2.0	ZeroMQ	MPL 2.0	ZeroMQ CPP	Apache 2.0
Aliyun OSS C SDK	MIT																									
AWS CPP SDK	Apache 2.0																									
Azure Storage CPP SDK	Apache 2.0																									
Boost	Boost 1.0																									
Cpp-netlib	Boost 1.0																									
Google glog	Free BSD																									
Hiredis	FreeBSD																									
Intel Storage Acceleration Library	FreeBSD																									
Nlohmann JSON	MIT																									
OpenSSL 3.0	Apache 2.0																									
ZeroMQ	MPL 2.0																									
ZeroMQ CPP	Apache 2.0																									
Release methodology and mechanics																										
Names of initial committers, if different from those submitting proposal	Helen H. W. Chan																									
Current number of code contributors to the proposed project	8																									
Current number of organizations contributing to the proposed project	CUHK Applied Distributed System Lab (ADS Lab)																									
Briefly describe the project's leadership team and decision-making process	Aldous Ng / CEO, CU Coding Ltd. Shakeel Salamat Ullah / CTO, CU Coding Ltd. Patrick P. C. Lee / Prof. at The Chinese University of Hong Kong Helen H. W. Chan / Postdoc. at The Chinese University of Hong Kong Andy Jiao / Advisor to CU Coding Ltd.																									
List of project's official communication channels (slack, irc, mailing lists)	N/A																									
Link to project's website	https://nexoedge.io/																									
Links to social media accounts	N/A																									
Existing financial sponsorship	N/A																									

Infrastructure needs or requests (to include GitHub/Gerrit, CI/CD, Jenkins, Nexus, JIRA, other ...)	GitHub
Currently Supported Architecture	x86-64
Planned Architecture Support	N/A
Project logo in svg format (see https://github.com/lf-edge/lfedge-landscape#logos for guidelines)	https://nexoedge.io/wp-content/uploads/2023/05/nexoedge-logo-blue.png
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?	N/A

Project Proposal - Mapping Criteria and Data:

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

2 TAC Sponsors, if identified (Sponsors help mentor projects) - See full definition on Project Stages: Definitions and Expectations	N/A
A presentation at an upcoming meeting of the TAC, in accordance with the project proposal requirements	Yes (slides (pdf))
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.	Yes
Upon acceptance, At Large projects must list their status prominently on website/readme	TBD

Project Proposal - Taxonomy Data:

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

APIs	Provide
Cloud Connectivity	Provide
Container Runtime & Orchestration	Consume
Data Governance	Provide (users have their choice on the cloud storage destinations), Consume
Data Models	Provide
Device Connectivity	Consume
Filters/Pre-processing	N/A
Logging	Consume
Management UI	Consume
Messaging & Events	N/A
Notifications & Alerts	N/A
Security	N/A
Storage	Provide, Consume, Facilitate - Reliability

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

Customer Devices (Edge Nodes)	N/A
Customer Premises (DC and Edge Gateways)	Support
Telco Network Edge (MEC and Far-MEC)	Support
Telco CO & Regional	Possible
Cloud Edge & CDNs	Cloud Edge – Support; CDNs: Possible
Public Cloud	Support
Private Cloud	Support

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

Automotive / Connected Car	X
Chemicals	X
Facilities / Building automation	X
Consumer	
Manufacturing	X
Metal & Mining	X
Oil & Gas	X
Pharma	X
Health Care	X
Power & Utilities	X
Pulp & Paper	X
Telco Operators	
Telco/Communications Service Provider (Network Equipment Provider)	X
Transportation (asset tracking)	X
Supply Chain	X
Preventative Maintenance	X
Water Utilities	X
Security / Surveillance	
Retail / Commerce (physical point of sale with customers)	X
Other - Please add if not listed above (please notify TAC-subgroup@lists.lfedge.org when you add one)	No

Deployments (static v dynamic, connectivity, physical placement) - (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.	X

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

APIs	Provide
Applications	Provide
Firmware	Required
Hardware	Required
Orchestration	Required
OS	Required
VM/Containers	Required

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

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