Secure Device Onboard (FIDO Device Onboard)

Status

- **Current Project Stage:** Stage 1 - At Large
- **Website:** https://www.lfedge.org/projects/securedeviceonboard/
- **Wiki:** FIDO Device Onboard
- **TAC Sponsors:** Joe Pearson (IBM), Jim St. Leger (Intel)
- **Project Overview:** PowerPoint; TAC Presentation (May 6)

- Presented during the Wednesday, May 6, 2020 TAC call: Meeting Recording (https://zoom.us/rec/share/69MlNur70HhOaZWK6k3BAbEoA73oT6a8hCAc_vJYxEsi4LRQjA7QRWt_fjnd5Q0E)
- TAC two-thirds vote approval reached on June 3, 2020.
- Governing Board Strategic Planning Committee approval reached on June 4, 2020.

Project Proposal - Project Introduction:

Required Information	Responses (Please list N/A if not applicable)
Name of Project	Secure Device Onboard
Project Description (what it does, why it is valuable, origin and history)	**What does Secure Device Onboard do?** Secure Device Onboard (SDO) is an automated “Zero-Touch” onboarding service. To more securely and automatically onboard and provision a device on edge hardware, it only needs to be drop shipped to the point of installation, connected to the network and powered up. SDO does the rest. This zero-touch model simplifies the installer’s role, reduces costs and eliminates poor security practices, such as shipping default passwords. **Why it is valuable?** Easier, faster, less expensive, and more secure onboarding of devices. Expands TAM for IOT devices, accelerates resulting ecosystem of data processing infrastructure. Most “Zero touch” automated onboarding solutions require the target platform to be decided at manufacturer. · Forces Custom Build-to-Order Model- ODMs must manufacture unique device SKUs for each customer/cloud combination. SDO “Late Binding”- allows the device’s target platform to be selected “late” in the supply chain, at first power-on. · Enables Build-to-Plan Model - ODMs can build identical IOT devices in high volume using a standardized manufacturing process. Reduces inventories, supply cycle times, and costs. · Open – service & cloud independent. Devices are bound to target ecosystem at install. Works with existing cloud services, it does not replace them. **Origin and History** Secure Device Onboard was released as open source software by Intel Corporation in February 2020, based on Intel® SDO Version 1.7. The original Intel® SDO launched in September 2017 as a stand-alone Intel product reflecting the original SDO protocol and architecture specifications. With the complex ecosystem needed for success of this product, we decided to open source and donate the core functions of Intel® SDO to the community in order to drive an industry standard, resolve key industry friction points, and allow the IOT market to grow faster. We believe that open sourcing with a vibrant ecosystem will allow SDO to evolve into a true industry standard.
Statement on alignment with Foundation Mission Statement	One of the primary objectives of Secure Device Onboard is to expand TAM for IOT devices. To achieve this goal, a cross-industry collaboration of device manufacturers; distributors; systems integrators; cloud service providers and device management software vendors is required to accelerate adoption. The Linux Foundation is the ideal organization to facilitate this collaboration and accelerate adoption of this important technology.

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.	We believe that Secure Device Onboard will accelerate adoption of devices into Home and Industrial ecosystems, helping drive the need for all of the current projects in the LFEdge community. Integration with FLEDGE enabled devices could simplify the production process and installation of newly manufactured devices.
Link to *current* Code of Conduct	https://lfprojects.org/policies/code-of-conduct/
Sponsors from TAC, if identified (a sponsor helps mentor projects)	Intel, Jim St. Leger IBM, Joe Pearson
Project license	Apache License 2.0
Source control (GitHub by default)	https://github.com/secure-device-onboard
Issue tracker (GitHub by default)	https://github.com/secure-device-onboard
External dependencies (including licenses)	

mbedTLS	Apache License 2.0
Apache Commons Collections	Apache License 2.0
Apache Commons IO	Apache License 2.0
Apache Commons Pool	Apache License 2.0
Apache HttpComponents HttpClient	Apache License 2.0
Apache ServiceMix :: Bundles :: spring-aspects	Apache License 2.0
Apache Tomcat	Apache License 2.0
Bouncy Castle PKIX, CMS, EAC, TSP, PKCS, OCSP, CMP, and CRMF APIs	MIT License
BouncyCastle	MIT License
CryptoAuthLib	Microchip Proprietary License
epid-sdk	Apache 2.0
FasterXML jackson-core	Apache License 2.0
guava-libraries	Apache License 2.0
H2 Database Engine	Apache License 2.0
Hamcrest	BSD 3-clause "New" or "Revised" License
Jackson-annotations	Apache License 2.0
jackson-core	Apache License 2.0
jackson-databind	Apache License 2.0
Jackson-modules-java8	Apache License 2.0
javax.annotation API	Sun GPL With Classpath Exception v2.0
jaxb-v2	GNU General Public License v2.0 w /Classpath exception
jedis	MIT License
JUnit	Eclipse Public License 1.0
junit 5	Common Public License 1.0
jwtk/jjwt	Apache License 2.0
Legion of the Bouncy Castle	MIT License
Log4J API	Apache License 2.0
Log4J Core	Apache License 2.0
Log4j Implemented Over SLF4J	Apache License 2.0

Logstash - logstash-logback-encoder	Apache License 2.0
MariaDB Client Library for Java Applications	GNU Lesser General Public License v2.1 or later
Mockito	MIT License
MS SQL Server JDBC Driver	MIT
Objenesis	Apache License 2.0
OpenJDK	GNU General Public License v2.0 w /Classpath exception
OpenJRE 8	GPL-2.0-with-classpath-exception
openssl	OpenSSL Combined License
OpenSSL	OpenSSL Combined License
PowerMock	Apache License 2.0
Project Lombok	MIT License
reactor reactor-core	Apache 2.0
Safestring	MIT License
SLF4J	MIT License
Spring-Boot	Apache License 2.0
spring-framework	Apache License 2.0
System Rules	Common Public License 1.0
TestNG	Apache License 2.0
tpm2-abrmd	BSD-2
tpm2-tss	BSD-2

Release methodology and mechanics	Secure Device Onboard currently follows a release cadence of approximately 12 weeks, typically with 9 weeks allocated for development, two weeks for integration test, and one week for final validation. Defects identified in the two-week integration test phase are resolved and the code base updated to create a release candidate for the final week of validation. Release artifacts are generated by a fully automated CI system. Integration test and validation includes both automated and manual testing and provides end-to-end testing of the SDO component running in concert to execute all phases of the SDO protocol and service lifecycle across multiple platforms.
Names of initial committers, if different from those submitting proposal	N/A
Current number of code contributors to proposed project	20
Current number of organizations contributing to proposed project	Intel Corporation

Briefly describe the project's leadership team and decision-making process	We recognize that in order to be a viable open source project, a neutral diverse technical governance is critical. We will be actively seeking TSC leaders from companies who are committed to SDO success. Our initial proposal is that Intel will contribute 3 of 7 TSC seats and by the end of Q2'20 we will hold elections for the other 4 seats. Currently the leadership of the project is as follows: Rich Rodgers (Intel) is the "product owner" and is responsible for identifying the feature roadmap for the secure device onboard project. He collaborates with members of the Secure Device Onboard ecosystem to identify emerging requirements and features. We anticipate that this process will expand to include others in a Secure Device Onboard technical steering committee comprised of community contributors and ecosystem stakeholders. Tom Barnes (Intel) is the project manager for Secure Device Onboard where he is responsible for planning and processes. He has previously contributed to the LF Hyperledger Sawtooth, Avalon, and Private Data Objects projects. We anticipate that Tom will be an initial maintainer for the Secure Device Onboard project. Saurabh Dadu (Intel) is the chief architect for Secure Device Onboard. He is responsible for translating the feature roadmap into technical requirements and architectural specifications, for maintenance of the Secure Device Onboard protocol specification, and for the overall security architecture of Secure Device Onboard. We anticipate that he will continue in this role as part of the Secure Device Onboard Technical Steering Committee. John Easterday (Intel) and Tushar Ranjan Behera (Intel) are the technical leads for the Secure Device Onboard project. They are responsible for software development as well as for oversite of devops and validation activities. We anticipate that they will be initial maintainers for the Secure Device Onboard project, with responsibility for ensuring contributions are properly and promptly reviewed and approved, and that they will eventually be joined by other contributors as the community of contributors grows. Richard Kerslake: Secure Device Onboard – Director. (Intel) Hussein Alayan: Secure Device Onboard – Program Manager/Deputy Product owner (Intel) SDO is a complex project comprising five sub-components spanning embedded devices to cloud services. As the community of contributors grows, we anticipate that the governance model will evolve into a core team/sub-team model similar to the one used by the Rust project as described here: https://github.com/rust-lang/rfcs/blob/master/text/1068-rust-governance.md.
Preferred maturity level (see stages here)	Secure Device Onboard is applying for Stage 1: At Large Projects
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)	As a recently opening open source project, we plan to work with LFEdge in setting up communication and net presence (slack, website, social media, etc...)
Link to project's website	As a recently opening open source project, we plan to work with LFEdge in setting up communication and net presence (slack, website, social media, etc...)
Links to social media accounts	As a recently opening open source project, we plan to work with LFEdge in setting up communication and net presence (slack, website, social media, etc...)
Existing financial sponsorship	Intel Corporation
Infrastructure needs or requests	· Secure Device Onboard is moving its continuous integration (CI) infrastructure to Jenkins running on Amazon Web Services. If LF Edge has an alternative solution, we would be interested in learning more about its capabilities and associated costs. · The Secure Device Onboard plans to maintain both its source and documentation repositories on GitHub, and to publish binaries using GitHub. · The Secure Device Onboard project plans to move its documentation to GitHub Pages. IF LF Edge has an alternative solution, we would be interested in learning more about its capabilities and associated costs. · The Secure Device Onboard project would benefit from access to a Jira instance (or equivalent) managed by LF Edge. · The Secure Device Onboard project would benefit from access to a Slack channel (or equivalent) managed by LF Edge.
Currently Supported Architecture	x86, x86-64, ARM

Planned Architecture Support	N/A
Project logo in svg format (see https://github.com/lf-edge/lfedge-landscape#logos for guidelines)	As a recently opening open source project, we plan to work with LFEdge in setting up communication and net presence (slack, website, social media, etc...)
Trademark status	N/A
Does the project have a Core Infrastructure Initiative security best practices badge? (See: https://bestpractices.coreinfrastructure.org)	No - however, the team is familiar with the Core Infrastructure security badge process and will consider pursuing that badge in the future.
Any additional information the TAC and Board should take into consideration when reviewing your proposal?	No

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

Criteria	**Data**
2 TAC sponsors to champion the project & provide mentorship as needed	TBD
A presentation at an upcoming meeting of the TAC, in accordance with the project proposal requirements	March 25, 2020
Adherence to the Foundation IP Policy	Yes
Upon acceptance, At Large projects must list their status prominently on website/readme	Yes

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	Provides
Cloud Connectivity	Consumes
Container Runtime & Orchestration	Consumes
Data Governance	Provide and Facilitate
Data Models	N/A
Device Connectivity	Consumes – HTTP, HTTPs, with architectural support for other connectivity protocols not yet implemented
Filters/Pre-processing	N/A
Logging	Provides
Management UI	Provides (some use cases)
Messaging & Events	Provides
Notifications & Alerts	Provides
Security	Provides
Storage	N/A

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
Customer Premises (DC and Edge Gateways)	Support
Telco Network Edge (MEC and Far-MEC)	Possible
Telco CO & Regional	Possible
Cloud Edge & CDNs	Support
Public Cloud	Support
Private Cloud	Support

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	X
Facilities / Building automation	
Consumer	X
Manufacturing	
Metal & Mining	
Oil & Gas	
Pharma	X
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	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)	

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.	X

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 and Cloud 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	Provide
Firmware	Require
Hardware	Require
Orchestration	N/A
OS	Require
VM/Containers	Require, vm is optional

Cloud Stack Layers	**Does Proposed Project Currently Include (Yes, No or Planned/Roadmap)**
Applications	Yes
Configuration (drive)	Yes
Content (management system)	Yes
IaaS	No
PaaS	No
Physical Infrastructure	No
SaaS	No