Levski

Release Date: Nov 9, 2022

Code Freeze: Oct 26, 2022

Namers: Diana Atanasova and Malini Bhandaru (VMWare)

The Levski release is version 2.3 (a third minor release on the second major version - Ireland - made during the summer of 2021. It is therefore backward compatible with all EdgeX 2.x releases. The release is not an LTS release. Jakarta (2.2) remain an LTS release.

Release Major Themes

- Commands via Message Bus (also known as North to South message bus communication)
  - This is the delivery of commands to the device services via the EdgeX message bus infrastructure (previously this was REST only)
  - Allow asynchronous communications from 3rd party systems (applications, Cloud systems, etc) all the way to devices
  - Allow for better quality of control (QoS) given the broker implementations provide for various levels. QoS allows for guaranteed delivery and retransmission of messages when necessary. Broker levels allow adopters to pick what level of QoS they want in their implementation without having to change any EdgeX code
- System Events (also known as Control Plane Events)
- Unit of measure implementation. See https://docs.edgexfoundry.org/2.3/microservices/core/metadata/Ch-Metadata/#units-of-measure
- More metrics telemetry (following the beta capability provided in Kamakura) including common service metrics. See https://docs.edgexfoundry.org/2.3/microservices/general/#service-metrics

Additional Features/updates/project improvements

- Optional use of NATS as the EdgeX message bus (as an alternative to Redis Pub/Sub and MQTT). NATS is a lightweight protocol with potential for on-device use and has the ability to use common messaging infrastructure at all levels of the stack. Its protocol-level metadata allows 'native' conveyance of the EdgeX message envelope and provides flexible deployment options for HA in some settings
- Access to EdgeX MQTT Mosquitto broker is now fully authenticated, including eKuiper access
- Securing of Consul with access tokens
- Refine the architectural design process to include agreements of Use Case Requirements (UCRs) before the high level design, i.e. the Architecture Decision Records (ADRs) are discussed
- EdgeX has earned the "Passing" status at the Open Source Security Foundation (OpenSSF): https://bestpractices.coreinfrastructure.org/en/projects/1226?criteria_level=1. The project is aiming for the Silver level in the next release.
- Documentation: addition of building and running EdgeX on native platforms (Linux, Windows, ARM32, etc.) https://docs.edgexfoundry.org/2.3/getting-started/native/Ch-BuildRunNative/
- Provide build time options to allow an adopter to remove items that are not needed for their use of the platform (like security). This will help shorten builds as well as make the artefacts smaller.
- Upgrade to Go 1.18
- Addition of the UART device service (will be released shortly after the initial Levski release date)
- Initial construction of the EdgeX Threat Model (will be extended and completed in future releases)
- Addition of unit and Angular testing for the EdgeX UI
- Better link checking and versioning in the EdgeX docs build process
- Further automation in the DevOps release process, in turn eliminating most of the manual steps
- First test of GitHub Discussions in order to someday replace Slack (10K limit is not helpful to retain records of project communications

Known Bugs

None at this time

Changelogs

Core, Supporting, System Management, and Security Services Changelog

Go Device Service SDK Changelog

C Device Service SDK Changelog - coming soon

App Functions SDK Changelog

App Service Configurable Changelog

CoAP Device Service Changelog - coming soon

Modbus Device Service Changelog

MQTT Device Service Changelog

REST Device Service Changelog

SNMP Device Service Changelog

Virtual Device Service Changelog

EdgeX User Interface Changelog

EdgeX CLI