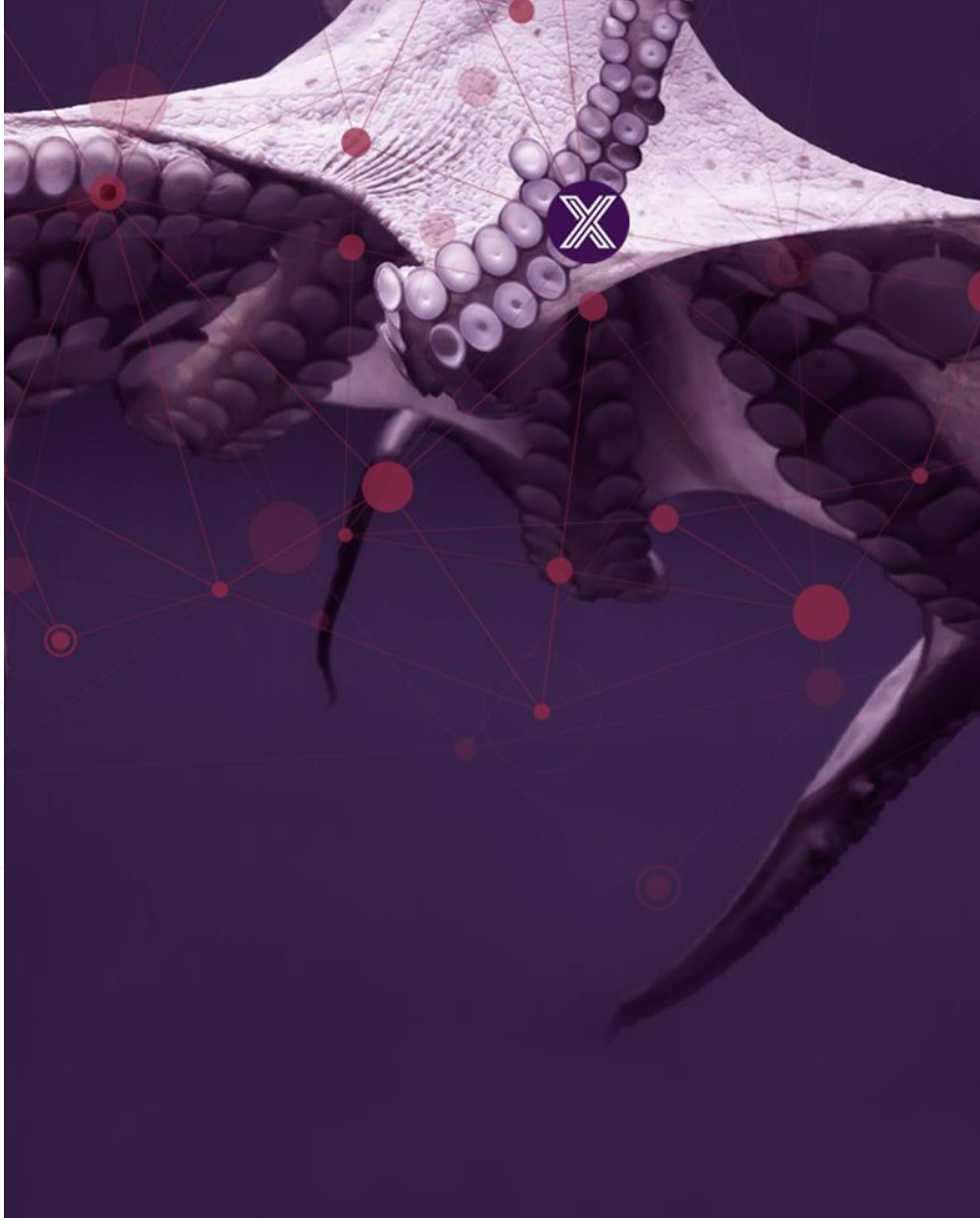
EDGEXFOUNDRY™



Jakarta Scope

As of June 24, 2021

Release Theme and Major Objectives

- Jakarta will be a stabilization release
- Jakarta will be an LTS release
- Jakarta will explore implementation of the service list and registry ADRs
  - May only offer a design or beta implementation for the release.

Post Ireland work to do

- Documentation
- Examples
- Document migration
  - Doc that outlines v1 to v2 configuration migration (what changed, was added, removed, etc.)
  - Needed or SDKs and core/supporting services
  - Needed for app or device services if they have specific config not found in SDK
  - Need one doc that outlines changes from v1 to v2 device profile

General

- Upgrade Go - TBD
  - Assume 1.17 will be out
  - Evaluate when it comes out (Core WG)
- Patch release to the LTS post Jakarta release
  - Triggered by high/critical bug or CVE issue
  - Will also trigger evaluation of latest patch releases for any 3rd party library or tool (Consul, Kong, etc.)
- Replace any poorly supported/used 3rd party library (ZMQ, etc.)
  - Only if there is no redesign/architect work required
  - It creates no backward compatibility issue
  - This is assigned to Core WG task to relook each library
    - Make determination on scope of change, impact.
  - The libraries should be archived into EdgeX repositories to support LTS
    - This is assigned to DevOps
  - Add a relook of policy/evaluation criteria to bring in 3rd party libs to architect's meeting agenda (especially for DS)

Core / Test QA

- *Includes core and supporting services, testing, CLI, GUI, system management*
- New
  - Add some additional endpoints/APIs
    - Query on readings by start/end timestamp and reading name (see issue) (small)
    - Query device resources by name and device name and device profile name (small)
  - CLI to V2 (large)
  - GUI
    - Data visualization (large)
    - TAF testing (large)
  - Mark SMA deprecated (small)
  - Enable CodeQL for Go repository services (small)
    - This was initially done for App Services SDK
    - Nothing found yet (checking code coming in and maybe not existing code); doesn’t appear disruptive
    - Next test – edgex-go
  - Install some linter (small)
    - Part of make test; looking for potential bugs
    - Requires config file to know what to look at
    - Core WG to explore initial config file; DevOps to implement against Guinee pig repository

Core / Test QA

- Tech Debt
  - Fix MaxRequest Size a config setting – there now but not being used (small)
  - Check that LastConnected is not being set in core data (with call to metadata) (x-small)
  - Test config alternates (x-large)
    - Configuration testing – testing non-default and dynamic config changes
    - Additional TAF tests, need to identify critical different configs we need to test (need a test plan – prioritize high.med.low the tests)
    - Implement identified high priority tests needed in this area
    - High priority tests for this release – things that people are really going to use/change (ex: message bus vs REST, persistence on/off vs security config which won't change much)
  - Test without optional services (small)
    - TAF test without core data running is a separate test
    - TAF tests without support services (scheduler/notifications)
  - Check whether running with all CBOR (vs JSON messages) improves performance (small?)
    - TSC to take decision and other actions given results
  - Improved performance testing (small)
    - Test with security on
  - Bring back backward compatibility tests with TAF (medium)
  - Test other device services (large)
    - Ideally, test all device services and test with real hardware
    - Device service testing is lower priority than above
    - TAF tests for MQTT (not requiring hardware)
    - TAF tests for camera (requiring hardware)
    - TAF tests for BACNet (with simulator and a C DS)

Core / Test QA

- Stretch Goals
  - Improved integration test (Kuiper) – needs to be scoped (out of scope for now until we have Kuiper release details)
  - Improved stress tests with MQTT in addition to Modbus
  - TAF tests for REST device service

App Services & Analytics

- New
  - AI / ML example – Stretch goal; try to use something already created or call to the community for help (medium)
  - LLRP to v2/2.0 (medium)
    - Who does this work (LLRP team, EdgeX community)?
    - Post LLRP acceptance
- Tech Debt
  - Fix MaxRequest Size a config setting – there now but not being used (small)
  - TAF tests for app services (medium)
    - Some examples of testing needs
      - Store and Forward
      - Batch send
      - Multiple endpoints
      - Mqtt publish placeholders
    - Need to explore unit test vs TAF tests and figure out how to cover
  - SDK hangs when calling docker-compose stop (small)
    - Can’t recreate – need to disposition whether this is an issue or not
    - In scope to disposition (to be done with LLRP – if not recreatable – close it)
    - Dependent on v2 upgrade of LLRP AS

App Services & Analytics

- Stretch Goals
  - Pipeline per topics
  - Fork the pipeline (lower priority)
  - In scope: create design for these (small); stretch implementation (medium for pipeline per topic)

DevOps

- Documentation for more edgex-global-pipeline functions (medium)
  - Some work started in Ireland
  - Developer facing part is to be done
- Figure out if it is possible to remove Dockerfile.build (small)
- Figure out how if it is possible to commit go.sum back with Go 1.16 (small)
  - First determine feasibility
- Upgrade all DevOps Python scripts to Python 3.9 (small)
  - Get rid of older unsupported Python
- Address any build images that need to be updated over time (more process in nature)
- Developer Badging (medium)
  - Almost done – implement automation/CI
  - Lower priority but really nice to have

DevOps

- Prep for switch to semantic-release (large)
  - Spike: research changelog plugin
  - Spike: research DockerHub plugin (Rebuild in nexus and push to dockerhub)
  - Spike: research GitHub Release plugin (edgex-cli)
  - Spike: research Custom plugins (for edgex-compose)
  - Spike: research best way to automate release (centralized vs decentralized)
  - Spike: misc plugin research
  - Spike: Small POC cut-over sample-service to use semantic-release
  - Helps to eliminate tech debt
- Archive (store in an EdgeX repo) anything so we can build in the future (availability of artifacts like Go, Consul, etc.) (x-large)
  - First get a list of that which needs to be archived (go.sum, npm,)
  - Figure out how to archive Go, C, JS, etc. libs
  - Figure out how to archive the 3rd party products (Consul, Kong, …)
  - Archiving may not be automated for Jakarta - may be manual at first
- Install some linter (small)
  - Part of make test; looking for potential bugs
  - Requires config file to know what to look at
  - Core WG to explore initial config file; DevOps to implement against Guinee pig repository

Security

- New
  - Micro service authentication (medium)
    - ADR; implementation is out
    - Build micro service communication security (service to service - #1950)
  - In priority order (concern that this might be too much for Jakarta)
    - Service restarting (security services at large) (large)
      - Build a secret store unsealing daemon (#1944)
      - Address Dynamic services (including services that are initially disabled)
      - Address stability concerns
    - Enable CORS (medium)
      - Medium because it is across all services
      - “Just a header” but there is some design work
      - Just in API gateway for Jakarta (should be straightforward)
      - Stretch to add in services natively (require other WGs)
    - Phase 3 of Consul implementation (medium)
      - Implement the ACLs
      - Impacts other services but no changes to other services – presuming services are well behaved (i.e. only register themselves only touch their configuration)
  - Re-engage LFX for code scanning (small)
    - Looking for EdgeX to be beta tester
  - Add service gated by ACL to provide capability for onboarding (small)
    - Security WG research
    - Tony homework to define; related to service list ADR
    - Any implementation as stretch goal
- Tech Debt
  - OAuth code clean up / removal

Snaps

- Token handling (with app services and profiles)
  - Canonical research and suggestions

Device Services

- New
  - Add record and replay function (new device service) (medium)
    - Low impact with new device service
  - Allow for dynamic updates to device profiles (small)
    - ADR first; implementation stretch and based on potential stabilization impact
  - Implement Validation of (device) ProtocolProperties (small)
    - We should do this. It is a convenience.
    - You can add a device without device service running or device actually available. Adding validation could be a breaking change in this instance.
    - Make validation off by default (by configuration) to keep backward compatibility
    - ADR first; implementation stretch and based on potential stabilization impact
  - Review/Accept remaining DS contributions: UART (small)
    - Review and upgrade to v2

Device Services

- Tech Debt
  - DS C side – secret provider for all and secrets in general (medium)
  - Move v1 Device services to V2 (GPIO, LLRP,. ..)
    - Ty to get help from authors
  - Device stanza in configuration should be drop (init commands and remove commands as example)
    - Structure does not have it in the SDK; configuration files just needs clean up
    - Should do that before Ireland release (Jim to check device services)
  - Archive holding DS that will not be done (Bluetooth, OPC-UA,) (small)
    - Take them off the list of pending services
    - Decision on which ones to archive should be architect meeting topic (Jim Action item)
  - SDK test suites (medium)
    - Which features should be tested – with unit/mocks? Do as much as we can of the high priority areas.
    - Some features may need integration tests (like SDK calls into metadata) – with TAF
    - Need to explore unit test vs TAF tests and figure out how to cove
- Distributed device service example for Jakarta (small or medium)
  - Might not be possible??
  - In scope to explore; either update the example if it is or mark example for v1 only
- Stretch
  - Sending many commands to a device service
    - Seems reasonable (have a way to send one command request to multiple devices)
    - Use of labels to implement?

Outreach

- Certification
  - Next steps for EdgeX Ready would be to increase protocol support
    - Allow people to submit EdgeX Ready applications using other protocols (Modbus, etc.).
  - Outreach/Cert WG to explore the option for potentially offering certification through this release cycle. Specific questions to be addressed by the Certification team for the TSC and community to move forward:
    - Are there other open-source projects offering a certification? If so, what do they offer and how does their program work?
    - Should the program focus on self-certification or have 3rd party administration (for fee) program? What is the target of a certification program and how does it benefit the project and the community?
  - Develop/present a potential strawman program (if required) to the community before the next planning meeting in the fall.
  - Dependency – record and replay may help with EdgeX Ready (worth exploration when that service ready)
- Seek funding for Web site post 2021
- Explore virtual hackathon for US/Europe
- (re)Explore industrial vertical solutions WG
- Better support for new users – QA session, meeting, online sessions?
- Marketing
  - Blog a month/Vlog a month
- PRC evangelism – how to support, how to leverage, etc.