EdgeX Planning Conference Minnesota Release

Virtual Meeting

Nov 14th – 17th 2022

Conference Agenda

All times in PT

- Nov 14, Monday, 7am
  - Intros
  - Conclude any Levski Issues
  - Business Topics
  - Architectural Tee-up
- Nov 15, Tuesday, 9am
  - Cadence & Process
  - Architectural Topics and Discussion
- Nov 16, Wednesday, 7am
  - Minnesota Scoping
- Nov 17, Thursday, 9am
  - Finish Minnesota scoping and architectural issues
  - Levski lessons learned
  - Future planning meetings
  - Long term road mapping

Intros

Name, rank, serial number please

Day 1

Nov 14th

Conference Agenda – Day 1

All times in PT

- 7am Introductions & Logistics
- 7:15am Levski release (any unfinished business, issues, etc.)
  - TBD
- 8:00am Minnesota release – high level agreement
- 8:30am Business Topics
- 9:00am Architecture topic tee-up
- 10am day 1 adjourn

If we have extra time today, we may bring more topics forward

Levski Wrap-Up

Remaining Levski Release Items

- Still be done
  - Release of all Device Services? Which are still TODO?
    - Camera USB still to come
    - Grove… TBC (currently only on version 1.x).. Hardware dependencies, we've already unofficially deprecated this.
      - Decision – let's officially archive this. Iain has some code to merge. Device Services WG will manage this. Good ticket for a newbie to bring up to date
  - C SDK items? Nothing outstanding, just for the next release
  - Tag all repos including TAF? Cloud: Have verified the testing today. Already contacted Bill
  - Examples
  - Docs
    - Levski docs board: https://github.com/orgs/edgexfoundry/projects/51
    - TODO: Update the docker descriptions (on dockerhub) – Lenny this week, + Bill, Ernesto
  - Performance metrics report (+ blog/marketing efforts)
    - Cloud's team will run this week. X86 = HP Mp9, ARM = RP4. Now done, James will create the report
  - Others?
- Issues/Concerns
  - None yet?
- Reminder we are still supporting EdgeX 2.1 (Jakarta) LTS until Nov 2023

Manual Release Testing (what we said at end of Kamakura testing)

Test	Owner/Actions
Test any configuration not already handled by new TAF tests - Test without Core Data - With various Message Bus implementations (MQTT, Redis) - Without scheduler, notifications, rules engine	Jim
Test reboot of the system. Make sure EdgeX can come back up	Lenny
Test device services (with real hardware where possible)	Jim – GPIO, Modbus, SNMP Intel – LLRP (DS and AS)
Working with service authors to identify minimal tests	Not done yet
~~Test different combinations/permutations of app functions in app services~~	Already covered by TAF tests
Test running with multiple services of same type (device service and app services)	Cloud and team – device service Lenny – app service

Q: Which of these are still manual tasks?

Starting Minnesota

Minnesota

Minnesota Release – HIGH LEVEL (As discussed in PreWire)

- Major Version - move to 3.0? Rather than keeping to a minor version - 2.4?
    - The Architect's Meetings have been leading towards a new major versions. TSC vote now passed: 3.0 !
    - Opportunity to remove a lot of deprecated items and technical debt
    - Not an LTS? Our previous LTS was a stabilized V2 (at version 2.1)
    - LTS Policy: https://wiki.edgexfoundry.org/pages/viewpage.action?pageId=69173332
        - Per LTS Policy, Jakarta will be supported for a period of 2 years **(November 2023)** unless otherwise stipulated by the EdgeX TSC
        - A November 2023 release (likely a 3.1) would be a candidate for LTS and continue that trend…
- Current major additions under consideration
    - Global Configuration
    - Use of URIs for configuration files etc
    - Configuration Formats
    - Size concerns – chance to replace Kong, Consul, Vault etc with lighter options
    - EdgeX data modelling and entity relationships
    - Message bus enhancements (MQTT5, NATS)
    - Notification Service Improvements
    - Authenticated service to service communication
- Do we have some initial priority on the above?

EdgeX Project Boards

- Time for all to transition to Minnesota – By December ??
  - Remove the Dones
  - Clean out In progress, QA Review, etc.
  - Move appropriate Icebox items back to the Backlog (leave issues not going to be addressed in the Icebox)
- Start a new Minnesota Documentation update (or rename Levski)
- Do we want to explore the new style of boards
  - Multi-view, More customizations, Better issue repository access
  - Any WG chair comment on what works, what doesn't?
  - Decision: Continue to be a WG Chair decision on which board type to use
- EdgeX Slack vs GitHub discussions…
  - James will gather details ahead of a formal decision later this week
  - Perhaps need a little more info wrt notifications and best practices

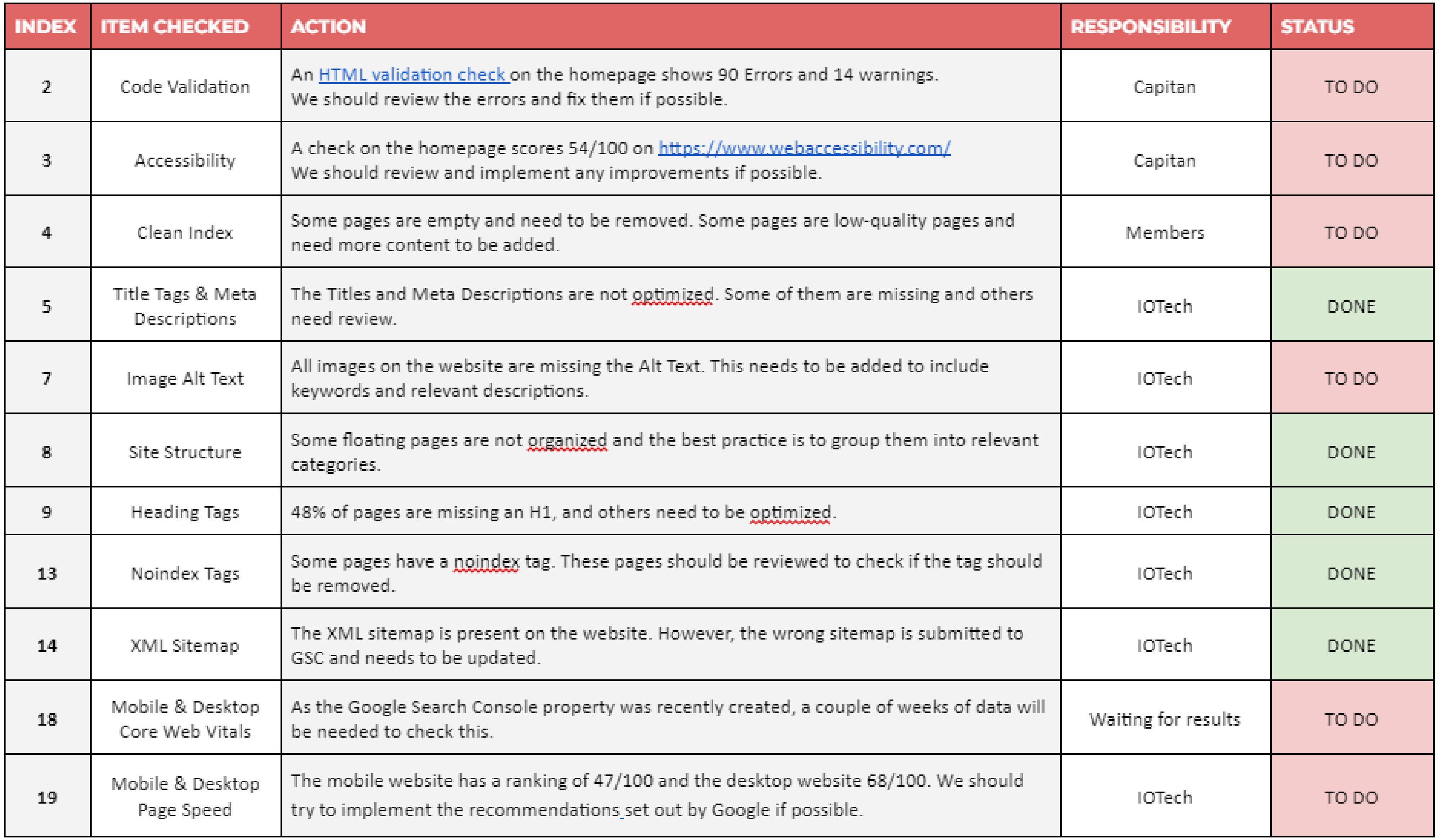
Business Topics

Business Topic Agenda

- Web Site (Andy/Gavin & Melvin on China site)
  - Current stats and status (Next Slide)
- Developer Evangelism (Jim/James)
  - EdgeX Ready (Rodney) Yes, keep this going. Maybe more interesting to Sis? EdgeX becoming mature
  - More Adopter Series? Yes. Eaton will (towards end of Minnesota). Chinese companies too
  - More Tech Talks? How-To videos? More commercially-savvy talks (real devices, real clouds)
  - English-speaking Hackathon? Attendance has been hard, geographic spread makes it difficult. Put on hold
  - Return to in-person events in 2023? Intel still have restrictions but Canonical is business as usual
- Next Level of OpenSSF Badge? https://openssf.org/
  - Currently "Passing". Can we get "Silver", "Gold"? Yes, silver next. Maintain this under the Security WG
- Do we still need an Outreach WG? No longer a TSC Voting Position
  - From Levski planning – perhaps move work into TSC?
  - Does EdgeX outreach still need independence from LF Edge?
  - Wider topic – how best to accomplish Outreach work?
  - Better coordination with LF Edge marketing teams
  - Decision: Make this a once monthly meeting. Jim to action with LF Edge Marketing team

EdgeX English Language Website Update

- Live since 2020
- IOTech responsible for
  - Release updates
  - Tech Talks updates
  - Reporting of site stats
    - In Google data studio
- Technical SEO Audit
  - Detailed site review
  - 6 Action Items
  - 5 Items Addressed
  - 8 Items Passed
- Keyword and Content Audit to follow



INDEX	ITEM CHECKED	ACTION	RESPONSIBILITY	STATUS
2	Code Validation	An HTML validation check on the homepage shows 90 Errors and 14 warnings. We should review the errors and fix them if possible.	Capitan	TO DO
3	Accessibility	A check on the homepage scores 54/100 on https://www.webaccessibility.com/ We should review and implement any improvements if possible.	Capitan	TO DO
4	Clean Index	Some pages are empty and need to be removed. Some pages are low-quality pages and need more content to be added.	Members	TO DO
5	Title Tags & Meta Descriptions	The Titles and Meta Descriptions are not optimized. Some of them are missing and others need review.	IOTech	DONE
7	Image Alt Text	All images on the website are missing the Alt Text. This needs to be added to include keywords and relevant descriptions.	IOTech	TO DO
8	Site Structure	Some floating pages are not organized and the best practice is to group them into relevant categories.	IOTech	DONE
9	Heading Tags	48% of pages are missing an H1, and others need to be optimized.	IOTech	DONE
13	Noindex Tags	Some pages have a noindex tag. These pages should be reviewed to check if the tag should be removed.	IOTech	DONE
14	XML Sitemap	The XML sitemap is present on the website. However, the wrong sitemap is submitted to GSC and needs to be updated.	IOTech	DONE
18	Mobile & Desktop Core Web Vitals	As the Google Search Console property was recently created, a couple of weeks of data will be needed to check this.	Waiting for results	TO DO
19	Mobile & Desktop Page Speed	The mobile website has a ranking of 47/100 and the desktop website 68/100. We should try to implement the recommendations set out by Google if possible.	IOTech	TO DO

Architecture Tee Up

Minnesota Major Themes and Objectives

- A Major Release – 3.0
- Major additions under consideration
  - Global Configuration
  - Use of URIs for configuration files etc
  - Size concerns – chance to replace Kong, Consul, Vault etc with lighter options
  - EdgeX data modelling and entity relationships
  - Message bus enhancements (MQTT5, NATS)
  - Notification Service Improvements
  - Authenticated service to service communication

Scoping Exercise

- Categories used in our Pre-Wire
  - In scope
    - It is “In scope” – must be done; no debate; near unanimous consent
    - Should not take (much) time in the planning meeting to discuss
  - Under consideration
    - It is definitely under consideration for the planning meeting and release
    - It is not in scope yet, but it worthy of some time to discuss; with a strong tendency to put it in scope
    - Has a majority of support to at least consider it; must be put in or out of scope at the end of the meeting
  - Not sure/On the fence
    - There are some that believe it should be under consideration or in scope but others are unsure or even against it.
    - To be reviewed and debated during the planning meeting as time permits; placed out of scope by default if not covered in the planning meeting
  - Out of scope (Not discussing in this meeting)
    - A majority believe this work will not be covered in the next release
    - A potentially valid need, but just not going to be accomplished for the next release (example: non-backward compatible change)
    - These items will not be discussed during the planning meeting but will be added to the backlog/roadmap
  - Never in scope
    - A majority believe this work will not be (ever) accomplished in EdgeX
    - Remove from the backlog or future scope (with rationale)
    - These items will not be discussed in planning meetings going forward

Architect’s Topic List

- In Scope
  - Global Configuration
  - URIs for Configuration
  - Configuration Format
  - (ADR) EdgeX Data Modelling and Relationships
  - MQTT5 Support
  - (Discuss) NATS
  - Notification Service Improvements
  - (UCR) Core Data Cache + ADR (if needed), implementation (TBD)
  - (Discuss) Lowering the JSON overhead
  - Authenticated service to service communication
- Under consideration
  - EdgeX Size Concerns – Replace Kong, Consul, Vault, etc
  - EdgeX Data Modelling and Relationships (Implementation)
  - EdgeX Lite
- On the Fence
  - None yet

Day 2

Nov 15th

Conference Agenda – Day 2

All times in PT

- 9am Release cadence check & release naming
- ~~9:30am Process improvements~~ - already covered
- ~~10:30am~~ 9:30 am Architecture topics discussion and decisions
  - Topics relevant to upcoming release having priority, discussion of architectural design - with goal of making decisions that are particularly relevant and impactful to upcoming release
- With any remaining time, we will start to explore Minnesota scope
- 12pm day 2 adjourn

Cadence and Release Naming

Nov 15th

Cadence Check

- April/May & Oct/Nov remain target release months (namer)
  - ✓ Levski release – Nov 2022 (Diana & Malini)
  - Minnesota release – May 2023 (Brett)
  - Napa release – Nov 2023 (Ernesto)
  - Odessa release - May 2024 (Mengyi & Farshid)
  - Palau release – Nov 2024 (Rodney)
- Venue for next F2F Meeting (May 2023)
  - With travel restrictions lifted – is a F2F possible?
    - Perhaps… Intel largely focused in Phoenix or Portland
    - Rodney – US sounds doable, depends on Covid etc. What about timezones?
      - Hybrid support would be important, i.e., link by zoom
    - Decision: Sensible deadline for a decision around Feb..
- Conferences
  - Target events to return to and when
  - Potential piggy-back with a F2F meeting?

“P” release named by Rodney Hess

Release Timing Details

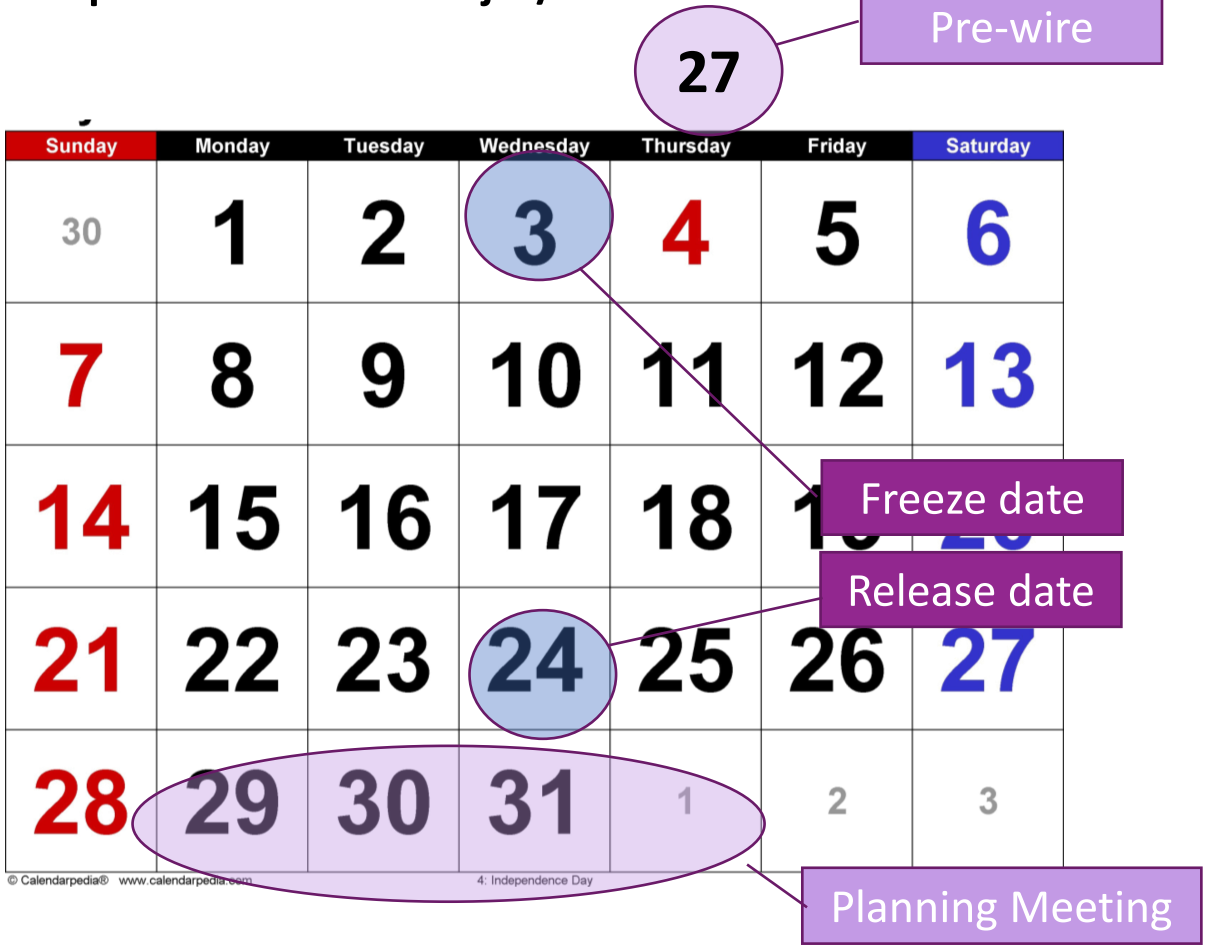
- Last cycle we established a semi-formal release/planning schedule as follows:
- Generally attempt to select release date about 2 months in advance of the release
  - Typically April and October for spring/fall releases
  - Release date preferred to be a Wednesday
  - Adjust per circumstances and TSC review
- Release schedule for minor release
  - Freeze date 2 weeks in advance of release date
  - Prewire, Thursday prior to freeze date
- Release schedule for major or LTS release
  - Freeze date 3 weeks in advance of release date
  - Prewire, Thursday prior to freeze date
- Planning meeting – generally the week following the release
  - Virtual meeting: Monday – Thursday with Friday for potential training/non-conference
  - In person: Tuesday-Thursday with Thursday afternoon for potential training/non-conference

If travel is necessary – planning meeting may need to be a week out

Include manual testing in the release schedule

Formalized Release Timing/Planning

Example Calendar for Major/LTS Release

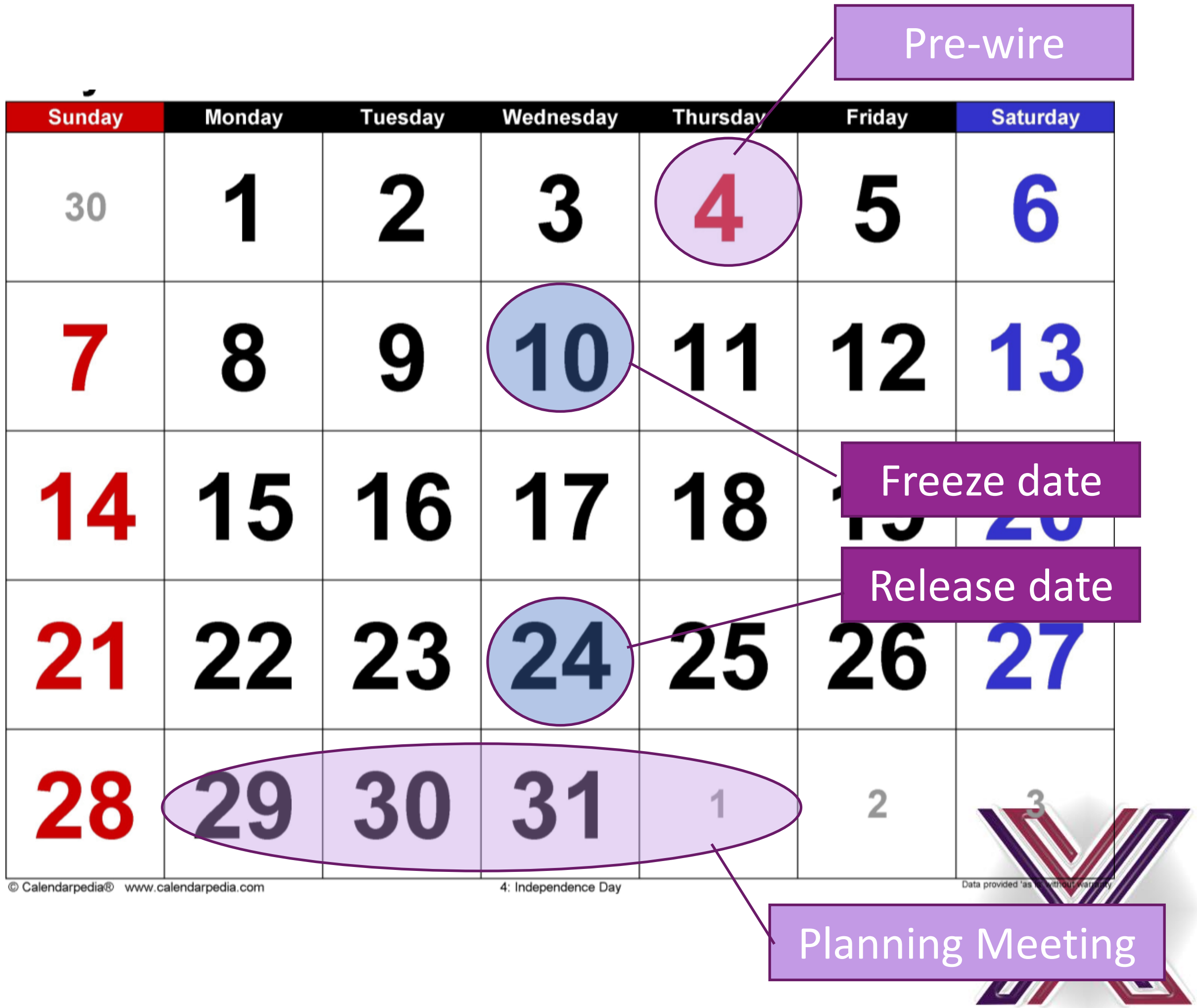


Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

© Calendarpedia® www.calendarpedia.com

4: Independence Day

Example Calendar for Minor Release



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

© Calendarpedia® www.calendarpedia.com

4: Independence Day

Data provided 'as is' without warranty

Current EdgeX Meeting Times

- Mondays
  - Device Services @ 8am PT
  - Certification & Outreach/Marketing @ 9am PT (3rd Monday each month?)
  - Architect's Meeting @ 10am PT (3rd Monday each month)
  - UI Project Group @ 7pm PT
- Tuesdays
  - DevOps @ 9am PT
  - Application Services WG @ 3:30pm (alternate weeks – Arizona time doesn't change)
  - Core @ 5pm (Arizona time – doesn't change)
- Wednesdays
  - TSC @ 8am PT
  - Security @ 9am PT
- Thursdays
  - Vertical Solutions - China Project @11pm PT
- Friday
  - No meetings

Any changes?

- TSC meeting
- WG meetings
- Monthly architect meetings
  - More/less frequent?

Process Improvements

ADR / UCR Process

- In the last cycle we added Use Case Record (UCRs) to the process
  - https://docs.edgexfoundry.org/2.3/design/Process/
  - Aim was to better agree needs and requirements before diving into architectural details (ADR)
  - Is this working for us?
  - Continue this into future? Yes
    - Refine the description of the Market Segments. Remove the Target Users section - Farshid
    - Double check that links for creating the UCRs, properly documented - Farshid

3rd party library support

- We have a process in place to review and vet 3rd party libraries being used by the project
  - We perform a "paper study" to explore the use, ongoing development status, upkeep, and contributors to a library
- In many cases, 3rd party libraries have less than ideal maintenance records, use, etc.
  - But often, there are few or no alternatives to using a library we might otherwise reject
- What to do about this going forward?
  - Fork and maintain libraries (at least those with potential support issues) in EdgeX
    - Only do this when the library team is not responding to an issue
  - Try to provide assistance to these library projects
    - This is our main recourse for issues
  - Do nothing – hope for the best; react to issues as they arise
  - We have and maintain our vendor copy
    - Default – as long as there are no issues
- Decision: apply more scrutiny to non-version libs but not a hardline to refuse them
  - Engage DevOps here

Architectural Discussions / Decisions

Architect’s Topic List

- In Scope
  - Global Configuration
  - URIs for Configuration
  - Configuration Format
  - (ADR) EdgeX Data Modelling and Relationships
  - MQTT5 Support
  - (Discuss) NATS
  - Notification Service Improvements
  - (UCR) Core Data Cache + ADR (if needed), implementation (TBD)
  - (Discuss) Lowering the JSON overhead
- Under consideration
  - EdgeX Size Concerns – Replace Kong, Consul, Vault, etc
  - EdgeX Data Modelling and Relationships (Implementation)
  - EdgeX Lite
- On the Fence
  - None yet

Deprecated Items List

- References to ZeroMQ (allowing for the refactor of go-mod messaging to support single host messaging)
- Remove SMA, executor and all references
- Username/password for SMTP access in notifications
- App Services old AES encryption
- Legacy Snap options
- Old tagging in edgex-UI
- lastConnected and lastReported in Device Services
- Kong Oauth2 plugin (keeping JWT)
- Deprecated aspects of App Functions SDK:
  - MessageProcessor interface
  - SetFunctionsPipeline and LoadConfigurablePipeline
  - NewTags factory method in app functions

Note: these items will be completely removed from this release

Items already set as “out of scope” or “never in scope”

Out of Scope - potentially valid, kept in backlog

- Time Series DB in place of Redis
- Bring Your Own Vault
  - allowing users to potentially run Vault in a cloud
- Support for RISC-V/64 bit architecture
- Integration with Prometheus
- Improve “things provisioning” to support zero touch
  - Concluded more research and requirements needed
- Move some configuration to static code variables
  - Example: internal message bus topic names
- Internationalization/localization of metadata

Never in Scope - will not be accomplished, removed from discussion

- Move Core Command API into Core Metadata
- Device resource units to be required field

Scoping Considerations

- Is there a champion / developer to drive the solution & get the work done
  - Who
  - Timeline
  - Dependencies
- What is high priority and what is a stretch goal?
  - T-shirt size it
  - Small – one release; one WG; one service; approximately one man to complete
  - Med – one release; many services;
  - Large – one release; all services; could take someone all release to finish
  - X-Large – multi-release and probably more than one service

In Scope

Global Configuration

- Is it time to consider some sort of global configuration?
  - We have a lot of duplication
    - Example: most of SecretStore section is the same for all services. These codes could be pulled from a common Global config or defaulted if left empty
  - Requires a lot of config to be touched when an adopter wants to change one item
    - Examples: common message bus configs (e.g. topic name), logging
    - Turning on/off metrics collection
- Considerations
  - Where would the configuration go?
  - With or without Consul, with or without security (and Vault)?
  - Potential effects on service discovery and registry
- Open UCR: https://github.com/edgexfoundry/edgex-docs/pull/892
- In Scope: Large
- Need to complete UCR asap, ADR to follow. Good to hold architectural discussions as early as we can (likely 1st week of Dec due to Lenny's holidays). Aim to start impl in early 2023
- Consider non-Consul options too. Jim volunteers to create an ADR-options file/wiki to kick off discussion
- This is key part of the next release; TSC needs to agree whether this issue could even delay Minnesota
- James to organize Architect's meeting on Nov 28th 9am PT

URIs for Configuration

- Better configuration capability
  - Potential to use a URI to specify an external configuration location (allowing for file, HTTP, HTTPS or other protocol access support)
  - URI mechanism may be used to point to device profiles, configuration files, UoM and other "configuration" information
  - This would even allow multiple EdgeX instances to use the same configuration or profile (multiple EdgeX instances using the same URI to use a shared profile for example)
  - Support better deployment scalability
- Considerations
  - Orthogonal to Global Configuration Discussion?
  - Fairly large impact? Potentially touching all services and configuration
  - Consider internationalization
- In Scope: Med/Large (TBC). Or XL if we talk of whole capability
- A universal utility that can be attached to any service to get files and configurations
- We need a UCR asap. ADR to follow. Crawl/walk/run on impl
- Local file changes can break with 3.0 (if needed)

Configuration Formats

- Consistent configuration formats: https://github.com/orgs/edgexfoundry/discussions/4
- A device service can have four types of configuration files:
  - Server configurations (TOML/JSON?)
  - Device(s) configurations (TOML/JSON)
  - Profile(s) configurations (YAML/JSON)
  - Provision Watcher (JSON)
- Confusing and difficult to work with? 3.0 gives an opportunity to simplify/unify?
- Scope: Large
- Suggestion to migrate all TOML usage to YAML, and ideally drop TOML for 3.0
- Documentation to describe the need here

(ADRs for) EdgeX Data Modelling and Relationships

- Parent/Child Devices – In scope: Small/Med
  - Add optional relationship metadata property that indicates a parent device to build hierarchical relationships between devices (often required at the northbound side for groupings management and presentation)
    - ADR not required, not cross cutting. Eaton to raise a usual issue/PR and document the need here
- Extending Device Data – In scope: Large
  - Means to extend a southbound device's profile with new resources (e.g., min, max, alarms, trends) that are added and managed by a higher-level service (e.g., analytics, utility or exporter)
    - ADR required since this is Core Metadata and Core Command related, potentially device SDKs – Eaton to action
- Provision Watch via Device Metadata In scope: Small/Med
  - Allow device provision watchers to utilize both device metadata (e.g., serial number, MAC address, etc) and the existing protocol properties as needed
    - General enhancement on provisioning; ADR not required. Eaton to raise an issue/PR as normal
- UCRs for all of the above approved in the Lesvki cycle

MQTT 5 Support

- Add support for MQTT 5 features
  - In particular, its bridging feature, topic aliases, and better (standardized) response or "return codes" for various errors
  - Add at the Message Bus, Device Service and/or App Service levels?
- Considerations
  - Requested during the Levsksi planning meeting May 22
  - Might Sparkplug integration be considered as part of this?
  - Does Mosquitto support MQTT5 yet (answer: yes)? So doesn't prompt a change in broker?
- Previous Groups Discussions:
  - Per Levski Planning meeting (5/17/22) - explore NATS vs MQTT5 vs other message bus implementations for future default implementation (and other supported implementations).
  - Performance and feature comparison. https://github.com/orgs/edgexfoundry/projects/48#card-82314596
  - Idea to experiment in the MQTT Device Service initially… maybe stretch goal for next release?
  - Potential opt-in options
- Backwards compatible with MQTT?
- Out of Scope

(Discuss) NATS

- NATS rationale (reminder):
  - Lightweight protocol with potential for on-device use
  - Ability to use common messaging infrastructure at all levels of stack
  - Protocol-level metadata allows 'native' conveyance of the EdgeX message envelope
  - Flexible deployment options for HA in some scenarios
- Redis Pub/Sub and MQTT currently provided as message bus implementations
- Levski added an optional/experimental NATS configuration. Changes here: #137
- What should the default message bus implementation be for this release?
  - What about securing it? MQTT was secured in Levski cycle
  - NATS and MQTT require brokers. Redis Pub/Sub likely an easy default
- Promote NATS to be a default built-in option. Need to consider C here
- Lenny’s proposal to build in by default so always available to users. Currently it has to be especially built to be used
- Out of Scope

NATS built-in by default (from Lenny)

- NATS is currently not built-in by default due to size increase concerns.
    - Must opt-in to NATS at build time
    - Most users deploy our pre-built binaries (snap) or docker images (compose) so don’t have NATS as an option
    - Therefore, NATS will not get much adoption if not built in by default
- Once we remove the ZeroMQ implementation there will be room for NATS

ASC	Baseline (no NATS)	Baseline +NATS	Baseline -ZMQ	Baseline -ZMQ +NATS
Binary	24.0 MB	25.4 MB	22.4 MB	23.7 MB
Delta from Baseline	0	+1.4 MB	-1.6 MB	-.3 MB
Docker image	34.2 MB	35.7 MB	29.7 MB	31 MB
Delta from Baseline	0	+1.5 MB	-4.5 MB	-3.2 MB

Note that docker has more size savings w/o ZMQ because we don’t have to include the ZMQ runtime library in each image.

Notification Service Improvements

- Suggested Improvements
  - Currently only email or HTTP; Better to provide support for sending notifications via SMS, web sockets or message protocol (like MQTT or Redis Pub/Sub)
  - Request to make using/sending notifications easier. Today, a developer must write code directly into a service to be able to send a notification. Is there a way to do some notifications by configuration? At the very least, better examples need to be provided on how to use the notification service
  - Also needed is a way for the rules engine to trigger notifications.
- T-shirt size: Small/Medium, self-contained
- Scope: Stretch, help wanted, etc

Day 3

Nov 16th

Conference Agenda – Day 3

All times in PT

- 7:00am Minnesota scope planning – Big Ticket Items (Couple left)
- 7:30am Minnesota scope planning – Working Groups
  - General
    - Common
    - Cross cutting concerns
  - Core
    - GUI, CLI, Test QA
  - Device Services
  - Application Services
    - eKuiper / rules engine
  - Security
  - DevOps
- 10am day 3 adjourn

(UCR) Core Data Cache (ADR if needed, Impl TBC)

- Desire to have Core Data serve more as a cache, i.e., keeping only latest *n* readings
- Initial ADR: https://github.com/edgexfoundry/edgex-docs/pull/723
    - A lot of discussion already on that. UCR already done too (needs approving)
    - What are the benefits of such a change? Can keep DB smaller
    - Don't think breaking change. Only impacts Core Data, additional APIs
- UCR here: https://github.com/edgexfoundry/edgex-docs/pull/884
- Extending / Enhancement, ADR may not be necessary
- Farshid: Not necessarily done in Core Data; scheduling service and other APIs possible.
- In scope: UCR
- Implementation (+ ADR if necessary): stretch. Likely small/medium

(Discuss) Lowering the JSON Overhead?

- Do we have too much duplicated info in each JSON event?
- Considerations
  - **EdgeX 3.0** Potentially impacts backward compatibility if the JSON message structure changes
- Previous Group Discussions
  - Levski planning May 22: Unknown details about what specific changes can/should be made. Would probably require some testing to ensure savings warranted the changes
  - Needs more discussions and evidence to warrants changes
  - Farshid: gzip could be an option to greatly reduced the size. Perhaps both in message bus and when exporting
- Iain’s testing and analysis on next slide…

(Discuss) Lowering the JSON Overhead?

Iain's Analysis:

- I took an Event generated by Levski - this contains three integer readings. Then tried some different things to reduce the size:
  - Removing the duplicated apiVersion, deviceName and profileName fields
  - Using shorter names for things e.g., "api" rather than "apiVersion"
  - Combination of the above
  - Using GZIP to compress the JSON
- Conclusion: using compression looks like a big potential gain.
- Question: are there other well-supported compression algorithms that might be more suited to json data?
- Question: are we considering this for event data only (it's probably the most performance-sensitive)?
- Need to consider footprint and other performance issues here. Good to be configurable since benefits are use case dependent
- In scope: Need more stats and observations, customer cases. More project testing needed, more community involvement
- If we compress the payload (not the whole envelope) then the clients can cleanly decode whatever is received
- Compatibility: could be added without breaking changes in 3.1

JSON content	Size	GZ size
EdgeX 2.3	100%	37%
De-duplicated	74%	37%
Short field names	84%	34%
Short and de-duped	65%	34%

Under Consideration

EdgeX Size Concerns – Replace Kong, Consul, Vault etc

- EdgeX size is getting bigger
- EdgeX services are quite tiny, but 3rd party services are huge
  - They were created for the enterprise, by enterprise experts
  - They were not meant for the edge
  - We only use a fraction (10%) of the functionality provided by these systems
- Substitute Services Options
  - Eaton and IOTech have been researching options and technology and have produced lighter-weight alternatives to replace Consul and Kong
  - Willing to contribute these to EdgeX Foundry either as replacements or secondary alternatives
- Eaton engineers still testing these new replacements services but looking to contribute back. Keeper looking good so far, Nginx still to test.
- Kong changes: Potential breaking changes to consider… could be compatible
- Substitute services would be additive/optional so not breaking in 3.1 etc. Compose-Builder easy to make/add alternatives
- Bryon doubts value of Kong/DB options and go straight to Nginx
- More analysis and testing needed. Stretch for Minnesota at this stage

Component	Container Size
Consul	116MB
Kong (and its DB)	347MB
Vault	186MB
Average EdgeX service	~20MB

Component	Container Size
Keeper	18MB
Nginx + Proxy-Auth	46.5MB

EdgeX Data Modelling and Relationships (Implementations)

- Parent/Child Devices
  - Add optional relationship metadata property that indicat arent device to build hierarchical relationships between devices (often required at the n bound sid upings m gement and presentation)
- Extending Device Data
  - Means to extend a southbound device's profil min, max, alarms, trends) that are added and managed by a higher-level s r)
- Provision Watch via Device Metadata
  - Allow device provision watchers to utilize both de etada serial number, MAC address, etc) and the existing protocol properties as needed
- UCRs for all of the above approved in the Lesvki cycle

Already Covered
in Day 2
Discussions

EdgeX Lite

- Provide capability to build EdgeX services w/o certain capabilities to have a smaller footprint
  - Same as we did with Delay Start capability and ZMQ
  - Mainly focus on Security capabilities that are currently built into all the services that cause their footprint to increase
  - Gives adopters ability to build a lite version when they are not using secure mode
  - Can be done for other capabilities also that we identify as contributing to a larger footprint that some low resource deployments don't need
  - Could do this with the NATS implementation. Out by default, but could be added via build tag. Requires all services (which use MessageBus) that an adopter wants to use to be rebuilt with NATS included
- Previous Groups Discussions:
  - **EdgeX 3.0** – more easily done without backward compatibility concerns
  - Levski Planning Decision: Defer until we have stats and input from adopters. Currently in Icebox
  - Explore goweight - The Lesvki cycle determined “nothing actionable”??
- Decision: In Scope/Low: Target go-mod messaging to enable opt-out build flags for this release and describe as an “incubation” feature in the docs.
- Stretch to do this for security in this cycle

Minnesota Scope

Scoping Considerations

- Is there a champion / developer to drive the solution & get the work done
  - Who
  - Timeline
  - Dependencies
- What is high priority and what is a stretch goal?
  - T-shirt size it
  - Small – one release; one WG; one service; approximately one man to complete
  - Med – one release; many services;
  - Large – one release; all services; could take someone all release to finish
  - X-Large – multi-release and probably more than one service

General

- Upgrade Go Version (1.19). 1.20 should be available in Feb 2023.
  - In scope. And consider some of the available enhancements
  - Decision: Wait until Feb rather than upgrade to 1.19 right away and then need to upgrade again
  - Make sure to also bump the Linter
- ~~TUI for docker compose file generation~~ already included
- Cloud templates need updating
  - Decision: Out of date, should be removed from main branch

Core

- Core
  - **EdgeX 3.0** Move protocol properties to JSON typing (Device Service discussion too) #592 in scope
    - Iain: should also correctly use numeric value types instead of strings as value props in device resources
  - Note that we will need to change the API to V3
    - Lenny suggesting: Get Command API could change and improve conciseness. Speak to Cloud.
    - Task for each WG early in cycle.
- CLI – could do with some usage info, or better promotion in docs
  - Need to discuss long term support for CLI
  - Support app and device services
  - Support registry
  - Support proxy setup going through API Gateway
    - Looks like this should be deprecated… unless there are community users. Canonical could continue to support, but to what end… customer and community feedback needed
      - James to ask on GitHub discussions as poll, with messages on Slack to point people at it
- Test QA
  - Improve testing (from last planning session)
    - Test without core data in place
    - Test without optional services, message bus v REST, etc.
    - Improved integration test (Kuiper – data comes back from eKuiper)
    - Improved stress tests
    - Redis use of memory tests
    - Improved performance testing (other device types, CBOR, etc.)
      - James: check this list with Cloud
  - Testing for new 3.0 features
- GUI
  - Add Provision Watcher Tab to Metadata section – nice to have

- Key Items from Icebox:
  - [EdgeX 3.0] Collapse pub/sub hosts into single host #138 In Scope
  - [EdgeX 3.0] Refactor Redis Pub/sub topic wild cards to match MQTT #166 In Scope
  - [EdgeX 3.0] Add JSON attributes to MessageEnvelope #169 In Scope
  - [EdgeX 3.0] Refactor database config to single DB #4131 In Scope

Device Services

- Implement ProtocolProperties validation mechanism
  - C-Catch up. IN SCOPE
- Areas where C Device SDK is missing features:
  - NATS Support. On the fence – resource dependent
  - N/S messages with anything other than Redis, i.e. MQTT Support as a subscriber. In Scope
  - Secure operation with delayed service start. Stretch (at best)
- DS Filtering implementation (ADR already in place)
  - Move to icebox
- Downsampling
  - Throttling device data send based on ability to consume/use Icebox
- Create a tool or script to create a new device or application service. Icebox and help wanted. Maybe more about example usage and better templating
- C SDK library creation Not doing – close this
  - i.e., produce a release artifact vs clone repo and edit code
- cmake options to optimize dependencies In Scope
- Use Device System Events instead of callbacks
  - #1259 and #4235

- Icebox
  - MQTT DS needs to be redesigned - Publish multiple readings in 1 msg in MQTT DS #299
    - >> Improve MQTT Device Service. In scope to discuss
    - In Scope, pending resources. A few parts to it.
  - Handle both JSON request bodies as well as CBOR request bodies #488
  - DS operatingState doesn’t go down post DS stop (C) #356
  - Support custom attributes in Device object #3685
  - Implement size constraints for devices and profiles #18
  - Support automatic migration of Devices between Device Services #1415
  - Support regular expressions in assertions #839

Application Services

- Tight coupling between EdgeX and eKuiper releases
  - Discussion with eKuiper team is in scope
- Record and Replay Implementation
  - In Scope, help wanted. Medium size
- SDK refactoring given we are going to V3 – In Scope, resource dependent
  - E.g. separating message bus config from triggering

- Key Items from the Icebox
  - Add support for regular expressions in filters #1037 Stay in icebox, help wanted
  - Use nano message IPC for ASC pluggable pipeline function Stay in icebox
  - SDK hangs when calling docker-compose stop #500 Try to reproduce, or else close
  - [EdgeX 3.0] Factory and receiver functions for all transforms should use pointers #1131 In scope, discussion not needed
  - Enhance MessageError to implement the Error interface #1125 In scope , discussion not needed

Day 4

Nov 17th

Conference Agenda – Day 4

All times in PT

- 9am Meeting Start
  - Scope and architectural questions not addressed in prior 3 days will be finished as priority work on day 4 as needed.
- 10:00am Levski lessons learned
- ~~10:30am Spring release - in person, face to face?~~
  - (already agreed to consider in February)
- 10:30am Long term roadmap - beyond Minnesota
  - EdgeX 3.1 and next LTS forecast
  - Planning meeting lessons learned and changes for Napa
- 12pm day 4 adjourn

Quick Summary: EdgeX 3.0 High Level Updates

EdgeX TSC has just approved the 3.0 version as the next release

Theme / Title	In Scope	Priority	T-Shirt	Breaking	Notes
Global Configuration	✓	High	L	✓	UCR #892 needs approving
URIs for Configuration	✓	High	M/L	-	UCR needed asap
Configuration Formats	✓	Med	L	✓	Documentation to describe the need
Parent/Child Devices	✓	Med	S/M	-	ADR not needed; Eaton to raise PR
Extending Device Data	✓	Med	L	-	ADR required (Eaton)
Provision Watch via Device Metadata	✓	Med	S/M	-	ADR not needed; Eaton to raise PR
Notification Service Improvements	Stretch	Low	S/M	-	
Core Data Cache	✓ (UCR)	Med	S/M	-	UCR in scope; options in Core Data, Scheduling
Lowering the JSON Overhead	Stretch	Low	S/M	-	Need more customer stats (IOTech)
Replace Kong, Consul, Vault with Lighter Options	Stretch	Med	L/XL	-	Eaton solutions available, testing ongoing
EdgeX Lite	Stretch	Low	S	-	Incubation work
MQTT5 Support	-				
NATS built-in by default	-				

Remaining Minnesota Scoping

Security

Theme / Title	OSSF?	ID	T-Shirt	Breaking?	In/Out
Big Ticket Items / Features					
Microservice authentication (Vault JWT-based) UCR, ADR, Implementation	✓	#613	L	✓	In. OpenZiti can possibly do this and below. UCR needs approving.
Secure microservice distribution (OpenZiti) UCR, ADR, implementation		TBD	XL	-	Stretch. UCR needed
Capture metrics for additional security events		#374	M	-	In
Delayed start services support in snaps (upstream systemd attestor)		TBD	L	-	Stretch, dependent on "Device System Events instead of callbacks" (#1259 and #4235) and resources
BYO Vault				?	UCR in scope. ADR/Impl Stretch
Security Assurance					
Check in STRIDE threat model		#858	S	-	In. Jim to help lead
Jakarta LTS maintenance (version bumps)		TBD	M	-	In
Fuzz testing for REST/MQTT interfaces	✓	#714	XL	-	Check TAF now. Make plan to expand. Stretch. Iterative approach
Publish SPDX SBOM for Minnesota release		#4173	M	-	Stretch
Investigate (and integrate?) GoKart static analysis tool		#3715	M	-	Stretch
Put security.txt in our release artifacts		#4151	S	-	Stretch
Help LFX improve BluBracket scanning of EdgeX		#3881	M	-	Stretch
Tweaks / Tech-Debt					
API gateway dead code removal		#3583	S	✓	IN
Secret APIs rename "path" to "secretname"		#370	M	✓	IN
CamelCase configname keys		#352	M	✓	IN
Pull postgres out of startup services list		#4032	S	-	Stretch
Remove superfluous delay in consul_wait_install.sh		#3584	S	-	Stretch
Cap # of redis connections		#3594	S	-	Stretch

DevOps

- Harden release process to avoid release failures with parallelized builds. In Scope. Medium
- POC Simple GitHub Actions and Simple Reusable workflow for Go builds. In Scope: PoC/Experiment. Small
- POC Build function for multi-arch docker images (using docker manifest) In Scope: PoC/Experiment. Small
- POC embedded swagger docs into the edgex-docs markdown (mkdocs-swagger-ui or mkdocs-swagger-ui-tag)
  - Put on hold due to ongoing discussions
- Misc pipeline support - TBC
- Automate dockerhub description release updates. In Scope. Small
- Add openssf best practices badge to EdgeX repos [ossf silver] #231 - Stretch

- From the Icebox
  - Automate the staging of github release assets for EdgeX Releases #411
  - Fix didChange in edgex-global-pipelines #191
  - Investigate Snyk API

Miscellaneous

- Docs
  - Migration guide needed for V3. Do this as we go

Lessons Learned and Future Plans

Kamakura – Lessons Learnt (we said we would try to fix)

- Positives
  - Docs: Incremental adds to automated release
    - More done in automated vs manual fashion for docs
  - DevOps – more automated; easier; automated release of compose
  - Scheduling – much better without overlap with planning
  - Security – hit our targets for the most part
  - Testing identified issues early/ ahead of the release
    - Identified the size issue quickly (smoke tests)
  - Kanban Board maintenance
    - More consistent use
  - Lenny in crisis switch mode without losing anything

- Areas of improvement
  - ADR process
    - “Farshid” use case approach promises to fix
    - Still an issue (carry over from Jakarta)
    - Decisions still taking too long
    - >> Think this is helping/working
  - Lack of critical testing usually bites us
    - No tests to check delayed start, etc.
  - May not have gone deep enough in our implementation considerations
    - Ex: delayed start impl
  - All the steps in release and post release processes need to be better documented
    - Ex: when/how to do change logs, manual testing, etc.
    - Make sure we don’t forget something

Lessons Learned in Levski – TO FILL IN

- Positives
  - Release Checklist works well
  - UCR helps focus our aims before diving into tech details
  - We hit all our main 2.3 implementation targets
  - Docs released on schedule (done as went along)
  - Security process worked well, e.g., handling CVEs
  - Automated testing continues to rise
  - Seamless transitioning of people

- Areas of improvement
  - Better Commit messages for change logs (next slide)
  - We should make architectural decisions earlier in the cycle
    - 1 month frequency too low
    - Can we do these more ad-hoc and on-demand
    - More expectation of moving through the UCR/ADR/Impl process in a timely manner
  - Alternate TSC and Architect's Meetings on the Wednesday.

Improve our Commit Messages – From Lenny

- Our Conventional Commit Messages are used to generate our change logs
- CI Pipeline auto generates draft change logs
  - https://jenkins.edgexfoundry.org/view/EdgeX%20Foundry%20Project/job/edgexfoundry/job/cd-management/job/changelog-generator/89/artifact/edgex-go/edgex-go-CHANGELOG.md/*view*/
  - We copy latest draft into service's ChangeLog.md file and edit for clarity
    - Remove changes not important to end user
      - Dependa-a-bot
      - Bugs created and fixed during current dev cycle
      - Build commits insignificant to users
      - etc
    - Edit messages to be clear and concise
    - Move messages to appropriate section. I.e. not a bug fix, etc.
    - We do our best edits without full knowledge of the changes.
  - Should we have additional Conventional Commit actions for in cycle fixes, build commits, etc that can easily be removed?
    - i.e. wip-fix: Fixed backward compatibility with XYZ
- **Call to Action**
  - Developers
    - Write all commit messages with end user in mind
    - Keep commits to a single scope
    - Use of appropriate Conventional Commit Action and optional Scope
  - Committers
    - Scrutinize commit messages, not just PR title.

High Level Napa Planning

- 3.1 – i.e., stabilization of the 3.0?
- LTS?
- Refining of some of the just-added features

Planning Meeting Lessons Learned

- Release Planning Meeting
  - Any lessons learned?
  - Any thing that could be done better?
  - Start doing, continue doing, stop doing?
  - What worked well and what did not?
- For the next Planning Meeting, what would we do differently?
  - Business and Marketing Topics can be moved to a later day
  - Review of Technical Matters upfront not needed so much now with our smaller, more focused group
  - Meeting pace was considered about right, i.e., we got everything done
  - How to get more engagement? Live polls throughout? Might be a zoom upgrade? James to look at tools we could use
  - Should try to have more homework done so our decision isn't just "need a UCR", "need an ADR", etc
  - UCRs should already be defined before the planning meeting. Needs a valid use case to warrant a real discussion
  - Back in original F2F meetings, people used to be assigned to a particular use case requirement
  - Need more customer input either directly from customers on the calls, or passed indirectly. More industry focused and avoid speculation of what people need

Thank You

EDGE X FOUNDRY™

A Linux Foundation project under LF EDGE