EDGEXFOUNDRY™

EdgeX Planning Conference Odessa Release

Virtual Meeting

Dec 4th – 6th 2023

Conference Agenda

- Dec 4, Monday, 7am - Kick Off and Main Architecture Discussions
  - Introductions and Meeting Logistics
  - Conclude any Napa Issues
  - Odessa Aims and Objectives
  - Odessa Architecture Discussions
- Dec 5, Tuesday, 9am - Architecture Discussions and Odessa Scoping
  - Odessa Architecture Discussions Cont.
  - Odessa Scoping by Working Group
- Dec 6, Wednesday, 7am - Complete Odessa Scoping, Business and Marketing Topics, Lessons Learnt
  - Odessa Scoping by Working Group Cont.
  - Business and Marketing Topics
  - Project Process Improvements
  - Lessons Learnt, Continuous Evaluation, etc,

All times in PST

Intros

Name, rank, serial number please

Day 1

Conference Agenda – Day 1

All times in PST

- 7:00am - Introductions
- 7:15am - Napa release (any unfinished business, issues, etc)
- 7:30am - Odessa Aims and Objectives release – high level agreement
- 8:00am - Odessa Architecture Discussions
- 10:00am - Day 1 adjourn

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

Napa Wrap-Up

Remaining Napa Release Items

- Still be done
  - Release of all Device Services? Which are still TODO?
  - Any remaining doc updates?
    - Odessa docs now marked as WIP
  - Performance metrics report
  - Others?
- Collateral / Marketing:
  - https://www.edgexfoundry.org/software/releases/
  - [LF Edge - Napa Release Blog]()
- Issues/Concerns
  - None yet?
- Reminder we are now supporting EdgeX 3.1 (Napa) LTS until Nov 2025

Starting Odessa

Odessa Release

Odessa Release Objectives/Size – As discussed in Prewire

- Minor Version - move to 3.2?
  - Expectation so far to not be an LTS (remains at 3.1)
  - LTS Policy: https://wiki.edgexfoundry.org/pages/viewpage.action?pageId=69173332
    - Per LTS Policy, Napa will be supported period of 2 years **(to November 2025)** unless otherwise stipulated by the TSC
- Current major additions under consideration – all changes must consider V3 compatibility
  - URI for Files (completion work)
  - Common Configuration (completion work)
  - Hashicorp replacement services
  - Data Modelling and Relationships
  - Microservice authentication based on end-to-end encryption
  - Continue EdgeX docs redesign

EdgeX Cadence Check

- April/May & Oct/Nov to remain target release months?
  - Napa release (Ernesto) - Nov 2023 ✓
  - Odessa release (Mengyi & Farshid) - April 2024
  - Palau release (Rodney) - Nov 2024
  - Queensland release (Bill) - April 2025
- Next Planning Meeting (likely May 2024)
  - With travel restrictions lifted – is a F2F possible?
  - Volunteers or suggestions on where to hold this meeting?
  - Consider a target conference or event?

'R' release naming…

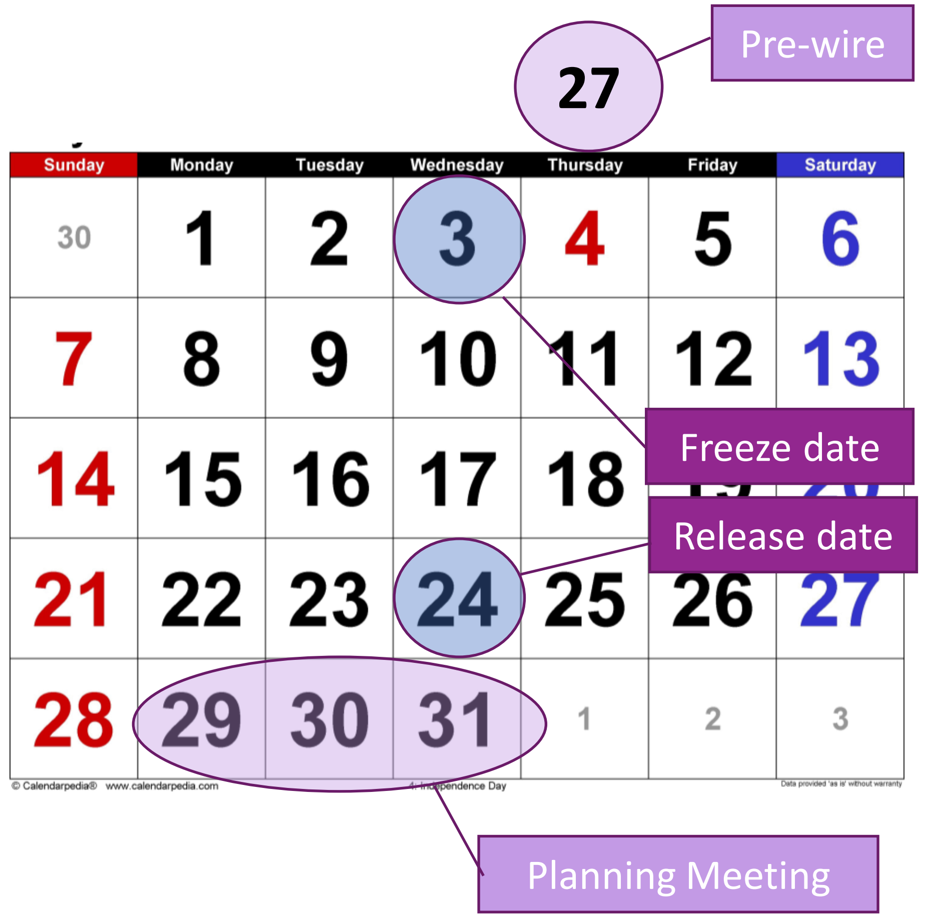
Release Timing Details

- EdgeX has 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 – Wednesday
  - In person: Tuesday-Thursday

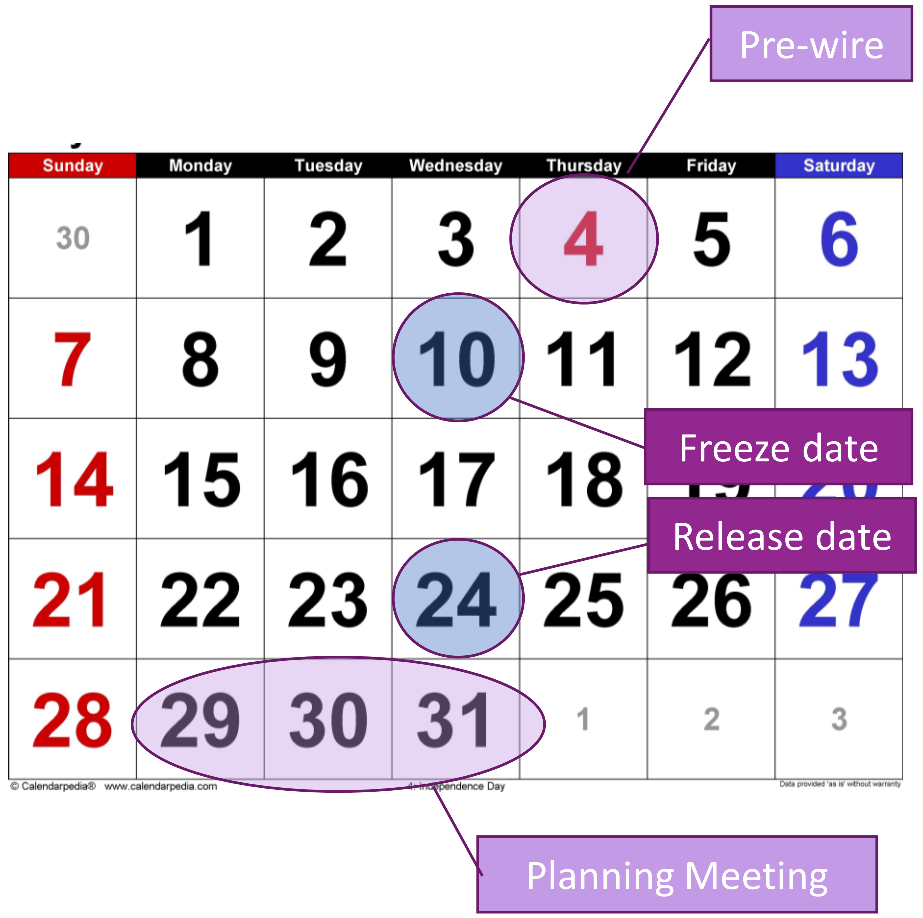
- If travel is necessary – planning meeting may need to be a week out
- Include manual testing in the release schedule

Formalized Release Timing/Planning

Example Calendar for Major/LTS Release



Example Calendar for Minor Release



Scoping and Planning

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 / Prewire List

- In Scope
  - URI for Files (completion work)
  - Common Configuration (completion work)
  - Hashicorp replacement services
  - Data Modelling and Relationships
  - Microservice authentication based on end-to-end encryption
  - Continue EdgeX docs redesign
- Under consideration
- On the Fence
  - Notification Service Improvements
  - NanoMQ
  - EdgeX Lite

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

Out of Scope - potentially valid, kept in backlog

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

Planning 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

URIs for Files (completing V3 work)

- 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 - i.e., this is not just about the "common" configuration files
- Already completed except for C Services
  - Q: this is the only remaining work here?
- Previous Progress
  - UCR Approved - https://docs.edgexfoundry.org/3.0/design/ucr/URIs-for-Files/
  - ADR Approved - https://docs.edgexfoundry.org/3.0/design/adr/0027-URIs%20for%20Files/
  - EdgeX 3.1 Docs - https://docs.edgexfoundry.org/3.1/microservices/general/#uri-for-files

Common Configuration (completing V3 work)

- Biggest item completed in 3.0, but any more aspects to complete/improve?
  - UCR: https://docs.edgexfoundry.org/3.0/design/ucr/Common%20Configuration/
  - ADR: https://docs.edgexfoundry.org/3.0/design/adr/0026-Common%20Configuration/
  - EdgeX 3.1 Docs: https://docs.edgexfoundry.org/3.1/microservices/configuration/CommonConfiguration/

1. One known bug to fix https://github.com/edgexfoundry/go-mod-bootstrap/issues/534

   Previously said: Out of scope for 3.1. Stretch: Analyze/design a better code solution for writable. Review again in 3.2 planning.
   - Internal code refactoring (as needed to address the above issue)
   - Bug and refactoring of interfacing to Consul
   - Refactoring of interfacing to support slices
   - Longer term plan to switching to full document configuration (perhaps for a V4)
2. Implement Common Config for C Devices (this was completed in 3.1)

Hashicorp Product Replacements – Situation Reminder

- In August 2023, HashiCorp announced they are moving to a Business Source License (BSL)
  - Vs Mozilla Public License (MPL)
  - MPL is compatible with Apache 2
  - https://www.hashicorp.com/blog/hashicorp-adopts-business-source-license
- Applies to future releases
  - “The license change is not retroactive. This means all source code and releases prior to the change remain under the MPL 2.0 license.”
  - Vault 1.14.5, Consul 1.16.3 are the last versions under MPL
  - EdgeX uses Vault 1.14 and Consul 1.16
- Regarding Patches to current releases
  - “HashiCorp will continue to backport critical security patches, as available, to existing versions under the MPL 2.0 license until December 31, 2023. Any patches after that date will be provided under the new license.”

Hashicorp Product Replacements – Meaning of the change to BSL

- The intent of BSL is to protect HashiCorp commercial interests
  - “Organizations providing competitive offerings to HashiCorp will no longer be permitted to use the community edition products free of charge under our BSL license. “
  - “All non-production uses are permitted. All production uses are allowed other than hosting or embedding the software in an offering competitive with HashiCorp commercial products, hosted or self-managed.”
  - Change only impacts production uses
- EdgeX, as an open-source project, is not directly affected – but adopters could be
  - The BSL license does not prohibit or restrict use of HashiCorp products.
  - Our adopters may be impacted – depending on whether they only use EdgeX internally and/or offer a “competitive offering”
    - “A “competitive offering” is a product that is sold to third parties, including through paid support arrangements, that significantly overlaps the capabilities of a HashiCorp commercial product. “
    - Remember, an adopter may offer non-EdgeX products that compete (ex: IOTech has a product that could be considered competitive with Nomad from HashiCorp)

Hashicorp Product Replacements – Plan for EdgeX

- Recommended for TSC approval
    - Vault -> OpenBao; a forked, open source   (See next slide)
    - Consul -> core-edgex-keeper; Eaton/IOTech developed drop-in replacement   (See next slide)
- Lift
    - EdgeX participation/liaison with OpenBao
    - Work to update/review/approve core-edgex-keeper (code, docs, tests)
- Plan
    - Have both OpenBao and core-edgex-keeper available by Odesa release
    - Backward compatibility? Do we need to still allow for Vault/Consul with EdgeX 3.x releases

Hashicorp Product Replacements – Consul

- See https://github.com/edgexfoundry-holding/edgex-core-keeper
- Project started by Eaton/IOTech Systems to provide a lighter weight alternative to Consul
- API compliant with Consul
  - Not a 100% replacement of all Consul features
  - But a drop-in replacement for EdgeX use of Consul
- Currently in holding for community review
  - Needs to be brought up to EdgeX 3.0/1 standards
  - Documentation needed
  - Testing needed

Hashicorp Product Replacements – Vault

- OpenBao – open-source Vault replacement
- Name provisional pending LF legal review
- Intention is to be a sub-project under LF Edge (like EdgeX)
- Wiki: https://wiki.lfedge.org/display/OH/OpenBao+%28Hashicorp+Vault+Fork+effort%29+FAQ
- They are working on a web site
- Led by IBM
  - Looking for other companies and organizations to support it officially (EdgeX is listed as a supporting org)
- Project currently meeting weekly (Thursday mornings at 9am EST)
  - Zoom: https://zoom-lfx.platform.linuxfoundation.org/meeting/94640473133?password=34a5aead-aef2-4a39-9e91-c16082afcfc5
  - Calendar URL (to subscribe to the meetings): https://lists.lfedge.org/g/openbao/ics/12823919/1232060308/feed.ics
- Hashicorp Vault code has been forked
  - Github: https://github.com/lf-edge/openbao/tree/development
  - IBM engineers and others are working to get it building
  - First task is to rid the code of all Hashi references
- Taking feature requests for the future of the project
  - Immediate need is to provide a drop-in replacement for Vault

EdgeX Data Modelling and Relationships – rolled over from 3.1

- Parent/Child Devices – Last time said scope: Small
  - 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
- Virtual Device Resources – Last time said scope: Small
  - 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)
    - Updated UCR to describe virtual device resource concept
- Provision Watch via Device Metadata Last time said 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

EdgeX Data Modelling and Relationships – New UC1

- **Protocol-specific Attribute Values in Device**
- There are many different manufacturers of the same type of device (e.g. HVAC). The Device Profile of a specific type of device could used be for many or all of the Devices of that type, reducing the need to duplicate a Device Profile just to account for differences in the protocol-specific attribute values.
- Example:
    - Two different manufacturers may build an HVAC using Modbus, and another may build an HVAC using SNMP. A single Device Profile cannot be created for 3 HVACs with three since HVAC1 and HVAC2 use the same protocol entry for the attribute - 'HoldingRegister'.
    - Since these three devices are the same type of device, all HVACs with temperature attributes, just having different protocols, the protocol information should reside in the Device, which describes a specific (instance of a) device. In this scenario all three devices can use the same Device Profile. This becomes more important as you have more devices, potentially increasing the number and management of Device Profiles when one will do
- Requirement Summary
    - The requirement is to support the protocol-specific attribute values (e.g. HoldingRegister) in the Device as well as the Device Profile (for backward compatibility).
    - If only the Device contains a protocol-specific attribute, it is used for that device.
    - If only the Device Profile contains a protocol-specific attribute, it is used for all Devices that have that Device Profile.
    - If both the Device Profile and the Device contain a protocol-specific attribute, the entry in the Device overrides the one in the Device Profile.
- For Odessa: Discuss, UCR, ADR, Implementation?
    - In scope (UCR at least)

EdgeX Data Modelling and Relationships – New UC1

```
Device Profiles
  ...
  name: HVAC1DP
  deviceResources
    -
      name: temperature
      attributes:
        { HoldingRegister: "4028" }
        { oid: "1.3.6.1.4.1.20440.4.1.5.1.2.1.3.2" , community: "private" }
  ...
  name: HVAC2DP
  deviceResources
    -
      name: temperature
      attributes:
        { HoldingRegister: "7055" }

Devices
  name: HVAC1
  deviceResources
    -
      name: temperature
      attributes:
  ...

  ...
  name: HVAC2
  deviceResources
    -
      name: temperature
      attributes:
```

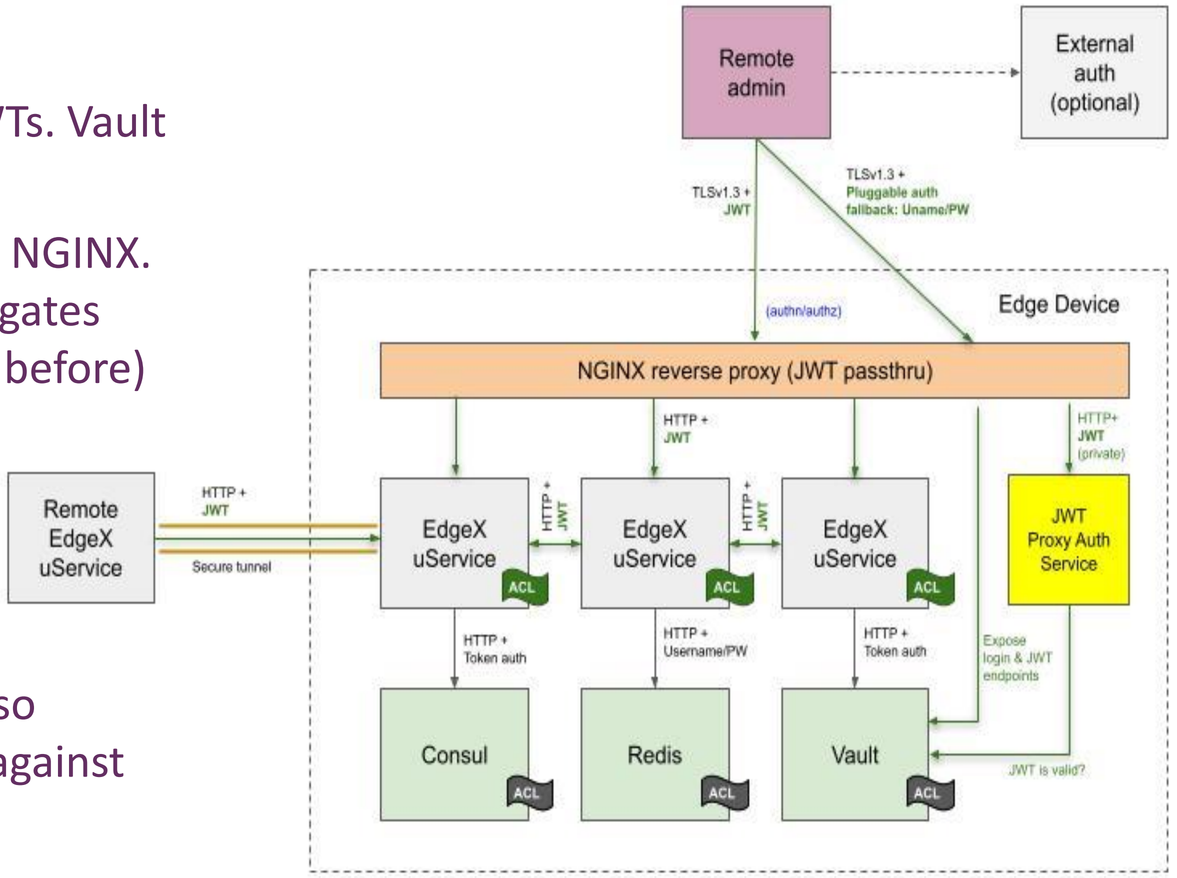
```
Device Profile
  ...
  name: HVAC
  deviceResources
    -
      name: temperature
  ...

Devices
  ...
  name: HVAC1
  deviceResources
    -
      name: temperature
      attributes:
        { HoldingRegister: "4028" }
  ...
  name: HVAC2
  deviceResources
    -
      name: temperature
      attributes:
        { HoldingRegister: "7055" }
  ...
  name: HVAC3
  deviceResources
    -
      name: temperature
      attributes:
        { oid: "1.3.6.1.4.1.20440.4.1.5.1.2.1.3.2" , community: "private"
  ...
```

Microservice authentication based on end-to-end encryption

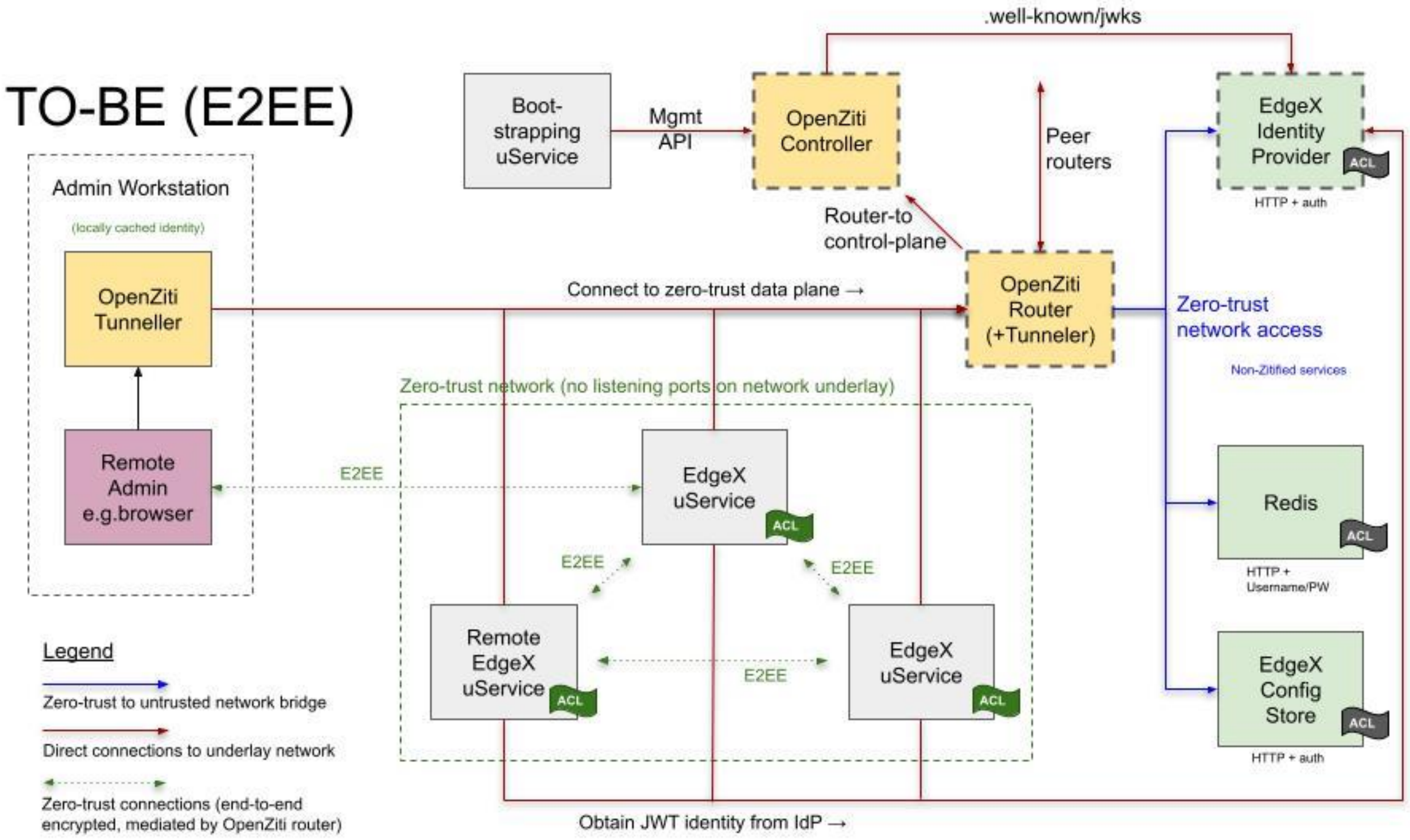
- Previously referred to as “Zero Trust / Secure Distributed EdgeX (OpenZiti Integration)”
  - See As-is and To-be diagrams on next slides
- UCR [Approved]: https://docs.edgexfoundry.org/3.1/design/ucr/Microservice-Authentication/
- ADR [In Review]: https://github.com/edgexfoundry/edgex-docs/pull/935
  - Oriented at the minimal implementation of including OpenZiti
- Implementation in this cycle? Clint @ NetFoundry is ready to push PRs
  - See Clint’s demo: https://www.youtube.com/watch?v=ARuGmzNuz-w
  - In this prototype, OpenZiti client libraries have been directly linked in to EdgeX's basic microservices and have replaced the standard TCP/IP listeners and dialers that most REST-based microservice architectures rely on. The demo also includes a “Zitified” a third-party component, the eKuiper rules engine, which was done with only a few lines of code. The Zitified services have no open HTTP ports that can be attacked, and all inbound REST calls are authenticated by an OpenZiti-linked identity. Only the OpenZiti control plane and edge router components bear the risk of exposed ports.
  - For Odessa - ADR and Implementation?

EdgeX Security - Minnesota (3.0) and Napa (3.1)

- Expanded Vault to include issuing and validating JWTs. Vault and not Kong own JWT issuance and validation
- Kong was actually replaced with the lighter/simpler NGINX. All inbound requests authorized by plugin that delegates JWT checking to Vault. Transparent to the caller (as before)
- Every service requires a JWT to be passed as part of the request that is validated
- Every service uses Vault-supplied JWT to authenticate outgoing calls
- Also every service will be assigned a Vault identity, so possible to configure Vault authentication engines against Vault identifies – e.g. Auth0, Kubernetes services



EdgeX Security - Outlined in ADR



Continue EdgeX docs redesign

- Big strides made for Napa release. Thanks everybody
- Latest here: https://docs.edgexfoundry.org/3.1/
- Noted remaining work:
  - Docs Refactoring of individual Device Services
  - Others? Tutorials, How-to-guides

Under Consideration

On The Fence

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: Last year: Small/Medium, self-contained

Ideas:

- Perhaps build on the back of an "alarm" service – Jim's requirements work
- Pipeline function to send notification via Support Notifications (see App Services WG later)
- eKuiper/rules integration

NanoMQ https://nanomq.io/

- EMQ (the company behind EdgeX rules engine eKuiper) have built a lightweight message broker called NanoMQ
  - Claimed to be extremely lightweight and very fast (compared to Mosquitto and HiveMQ)
  - Implements Sparkplug, includes an embedded rules engine, brokerless options
  - EMQ pushing to start another LF Edge open-source project
  - Could/should EdgeX consider NanoMQ as an internal messaging option (alongside Redis Pub/Sub and Mosquitto)?
    - LF Edge community and support/maintenance benefits
- EMQ presented at architects meeting in May. Recording here: https://wiki.edgexfoundry.org/display/FA/EdgeX+Architects%27+Meetings
- Completed crawl phase in EdgeX 3.1
  - I.e. Added changes in compose-builder to swap in NanoMQ (documented as experimental option)
- Jim and Lenny discussed some options with Jaylin and the EMQ team:
  - Inject secrets into NanoMQ so that when the container comes up it is at least minimally secured
    - E.g. equivalent to how Mosquitto knows to use security bootstrapping process and use the /mosquito/config/pwdfile to secure the broker
  - Provide some comparison (benchmarks) performance data between Mosquito MQTT and NanoMQ (and potentially Redis Pub/Sub) when running EdgeX
    - Look to provide guidance for users where/how NanoMQ might be better than Mosquitto under certain environments (like resource constrained devices)

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
- Previously marked as Stretch goal for Minnesota:
  - On The Fence - Needs some prioritization, stats based
  - In Scope: Stripping debug from Go binaries
  - In Scope: Look at Tiny Go https://github.com/tinygo-org/tinygo
  - Consider results from survey of EdgeX adopters on different usage (message bus, security, etc,)

Day 2

Conference Agenda – Day 2

All times in PST

- 9:00am - Napa Architecture Discussions cont…
- 10:00am - Napa Scoping by Working Group
- 12:00pm - Day 2 adjourn

Day 1 Summary - TODO

Day 1 Scoping Summary - TODO

Theme / Title	In Scope	Priority	T-Shirt	Notes

Odessa Scope

Caveats / Notes

- Items in the pages following are options at this point
- May be research or design tasks for this release
    - May be partial implementation
- No level of effort has been determined
- No priority has been assigned
- NOTE: scope does not include tasks and work items that come from the main architectural discussions earlier
    - These need to be added and scoped as well

General

- Upgrade Go version? Napa currently using 1.21 (released Aug)
  - Expected February, so in scope
- More Metrics added to microservices?
  - Out of Scope unless feedback emerges (see survey)
- Helm Charts as release artifacts
  - Move example Helm Chart to its own edgex-helm repo- done? Likely outside of LTS
  - Maintain and release as we do with edgex-compose
    - In Scope
- General EdgeX Project Management enhancements… adjustments to the dev process?
  - Review of the meetings (especially Core and Devices WG)

Core Working Group

- Core
  - URI for Files (completion work)
  - Common Configuration (completion work)
  - Consul replacement
- User Interface
  - General discussion about future of UI needed
    - Support and maintenance difficulties?
  - Bug: Fields for Provision Watcher Tab don't match the Provision Watcher DTO #642
  - Other new requested features? Still uses MUX
- Test/QA
  - Fuzzing and formalizing fuzzing – On the Fence
- Security
  - Other than microservice authentication and Hashicorp changes?
  - eKuiper API security is not enabled when EdgeX is running in secure mode #4538 – see OpenZiti work
  - Enable eKuiper to make authenticated EdgeX API calls #4539 – see OpenZiti work
  - edgex-vault logs contain errors about revoking consul token leases #3544 – Tech debt to be looked at

- Icebox – mostly don’t warrant architectural discussion
  - Websocket support for support-notifications #2304
  - [Tech-Debt] Refactor the knowSecrets in compose-builder/makefile to use script or function #284
  - Support for EdgeX holding services via a fork in the edgexfoundry-holding organization #233
  - Add support for services to notify systemd socket when ready #1301
  - Support automatic migration of Devices between Device Services #1451
  - Module exploration - remove or replace github.com/kr/logfmt #2616
  - Various noted fuzzing errors

Device Services

- Previously mentioned Device Relationships etc
- New CAN Device Service
  - Likely 3.1.x
- New S7 Device Service
  - Repo in holding – review in progress
  - Likely 3.1.x
- Allow AutoDiscovery to generate Device Profile https://github.com/edgexfoundry/edgex-go/issues/4422

- Icebox – all seem to need user pull and/or resources
  - Handle both JSON request bodies as well as CBOR request bodies #488
  - DS operatingState doesn't go down post DS stop (C) #356
  - Implement size constraints for devices and profiles #18
  - Support regular expressions in assertions #839

    Real-world UCR needed
  - DS Filtering implementation (ADR already in place)
  - Downsampling - Throttling device data

Application Services

- Enhance documentation
  - E.g. how to guides
- Different northbound endpoints?
  - Sparkplug - IOTech may be able to help

- Icebox
  - Add pipeline function to send notification via Support Notifications
  - Use nano message IPC for ASC pluggable pipeline function
  - Cloud export examples
  - [LLRP] Add LLRP Inventory specific service metrics
  - Additional App Service metrics

DevOps

- #438 Implement git-cliff, a git-chglog replacement
- #376 Automate release branch creation
- Support for EdgeX holding services via a fork in the edgexfoundry-holding organization #233 – will be closing
- Separate go version for build and go version for compatibility #4430

Day 3

Conference Agenda – Day 3

All times in PST

- 7:00am - Odessa Scoping by Working Group
- 7:15am - Business and Marketing Topics
- 8:00am - Project Process Improvements
- 9:00am - Lessons Learnt, Evaluation, etc
- 10:00am - Day 3 adjourn

Business Topics

Business Topic Agenda - TODO

EdgeX Website Update - TODO

EdgeX Website Update – Keyword Research - TODO

EdgeX Website Update – Traffic Analysis - TODO

Social Media - TODO

Process Improvements

Process Improvements - Agenda

- EdgeX Meetings – Days and Times check
- UCR and ADR process
- Issue and PR process
- Community Chat: GitHub Discussions
- Third-Party Libraries
- EdgeX Project Boards
- Example Repo Tagging – Napa decision: change tagging process to have examples work against specific releases – like the rest of the repos and dev cycle

EdgeX Meeting Times

- Mondays
  - Device Services @ 4pm PT (alternate weeks)
  - Outreach/Marketing @ 9am PT (3rd Monday each month)
- Tuesdays
  - DevOps @ 9am PT (2nd Tuesday each month)
  - Application Services WG @ 3:30pm AZ Time (alternate weeks)
  - Core @ 4pm PT
- Wednesdays
  - TSC / Architect's Meetings @ 8am PDT (alternate weeks)
- Thursdays
  - No meetings
- Friday
  - China Project Meeting 12:00am (1st Friday each month)

- Any Changes
  - More/less frequent?

ADR / UCR Process

- During Levski cycle we added Use Case Record (UCRs) to the process
  - https://docs.edgexfoundry.org/3.0/design/Process/
  - Aim was to better agree needs and requirements before diving into Architectural Design Records (ADRs)
  - Believe this is working well for us? Any suggested tweaks?

Issue and PR Process

- Any issues with Issues to report?
- Are we under-defining the Issue/Case creation template?
- Last time we added some minor changes:
  - Added “WIP” in commit message when fixing a bug introduced in current cycle
  - To ensure review standards are followed, added committers checklist in the pull request template
  - Are these working for us? Any other changes needed?

Community Chat: GitHub Discussions

- https://github.com/orgs/edgexfoundry/discussions
- Daily questions posted with much discussion – thanks to all contributors!!
- Suggestions for refining the process / management of discussions:
  - Mark discussions as closed when resolved
  - Mark specific comments as "answers" – Stack Overflow style
  - Keep conversations specific to original question. Open new discussions when needed
  - Promote "Show and Tell" section for EdgeX adopters / use cases
- How to ensure questions raised result in awareness of requirements/needs?
  - Just ask the person raising the question to create an issue (if needed) and reference the discussion.
- Other thoughts/comments?

EdgeX Project Boards - https://github.com/orgs/edgexfoundry/projects

- Time for all to transition to Odessa boards – By End of 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 Odessa Documentation update (or rename Napa)
- Feedback on the different project boards that are being used?
  - Most working groups use “new” GitHub boards
  - Time to migrate all remaining boards from “classic” to “new”
  - Some old boards to delete/close – e.g. Certification WG, Kamakura Docs, etc

Manual Release Testing

- https://wiki.edgexfoundry.org/display/FA/Manual+Testing
- Have any important/new manual tests emerged?
- Plan to add a checklist for future manual release testing

Lessons Learnt, Continuous Evaluation, etc

Lessons Learned in Napa (What we said last time)

To review and add new thoughts:

- Positives
  - V3 Breaking Changes were handled smoothly. TAF and management of breaking changes etc
  - Doc changes and migration guide, all done within the time
  - Managed workload well to hit the release targets
  - Release process gets better

- Areas for improvement
  - Continue to improve on commit messages
  - Continue to improve documentation
  - More community coding members
  - Awareness of different support options for a wide range of requirements people have

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?

EDGEX FOUNDRY™

Thank You

EDGEX FOUNDRY™

A Linux Foundation project under LF EDGE