EDGEXFOUNDRY™

EdgeX Kamakura Pre-wire Meeting (virtual meeting – in advance of Planning meeting)

Kamakura Release

Oct 27, 2021, 8-11am PDT

Conference Agenda

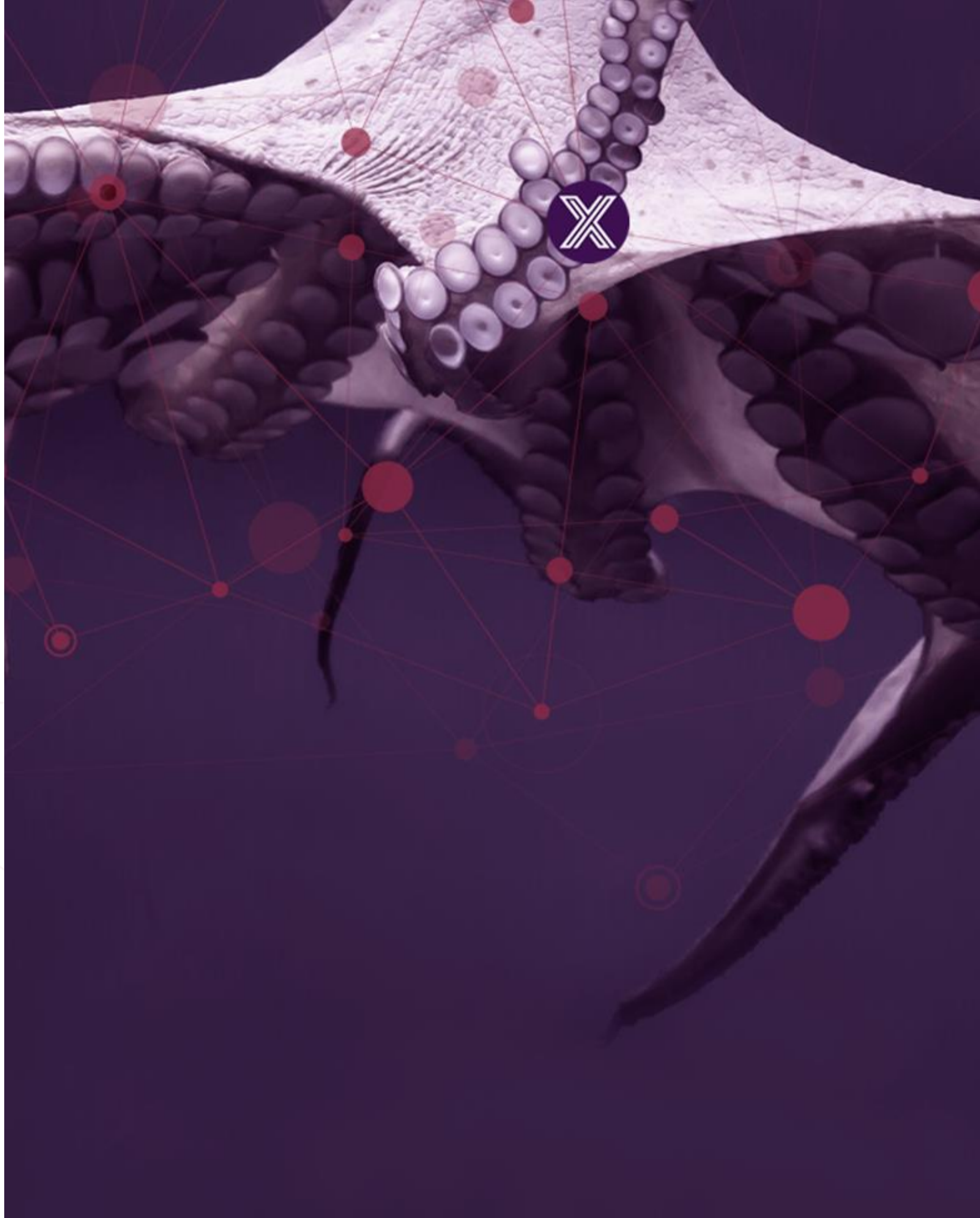
All times in PDT

- 8am – Introductions (if needed)
- 8:10am – Address any Jakarta items (if needed)
- 8:20am – Kamakura release objectives/size
- 8:30am – Kamakura Architecture Topic Scoping
- TBD – Kamakura Release Scoping
- 11am PDT - Adjourn

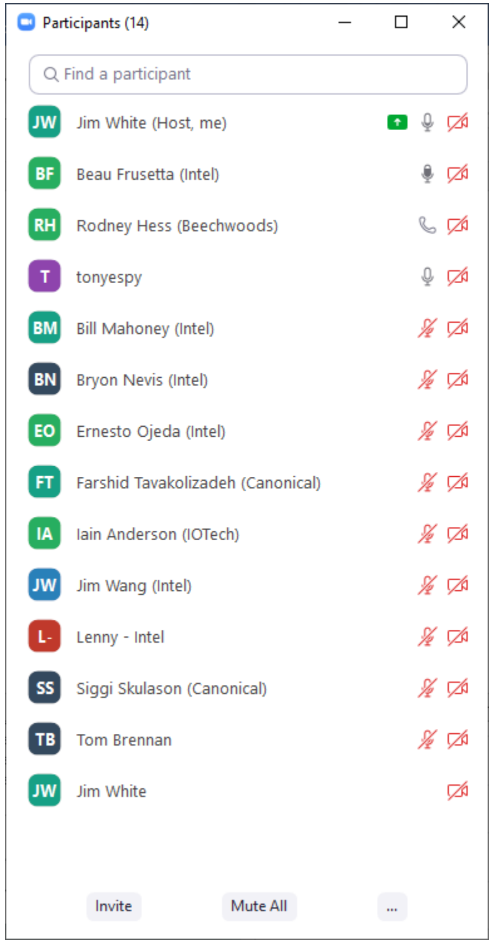
EDGEXFOUNDRY™

Intros

Name, rank, serial number please



Attendance (some attendees may have attended after this was captured)



Participants (14)

Find a participant

JW Jim White (Host, me)

BF Beau Frusetta (Intel)

RH Rodney Hess (Beechwoods)

T tonyespy

BM Bill Mahoney (Intel)

BN Bryon Nevis (Intel)

EO Ernesto Ojeda (Intel)

FT Farshid Tavakolizadeh (Canonical)

IA Iain Anderson (IOTech)

JW Jim Wang (Intel)

L- Lenny - Intel

SS Siggi Skulason (Canonical)

TB Tom Brennan

JW Jim White

Jakarta Items

- Object data type in get/set command
  - See Cloud's closed issue for current resolution
  - Jim to add to next week meeting topics
    - How should REST/SOAP web device be addressed in DS going forward?
  - May need ADR going forward
    - Intel team's approach for LLRP/RFID reference
- CORS support
  - Not just about setting headers
  - Need logic to validate domain (see flowchart provide by Bryon)
    - "preflight checks"
    - Happening in secure mode (using Kong gateway) but not non-secure mode
    - Core working group topic
    - Must be addressed before release
    - Replicating in C SDK is also an issue
- Bug on eKuiper connections file (prevents Snap from installing)
  - File not available in 1.3.0 or 1.3.1 – only 1.4.0
    - Issue to be resolved this week
  - Tight coupling between eKuiper and EdgeX that needs to be examined
    - Should be addressed as part of Kamakura (maybe just EdgeX issue)
- Pipeline LTS testing – done for go services and one C service
  - Looks like we are set for Jakarta

Kamakura Release Objectives/Size

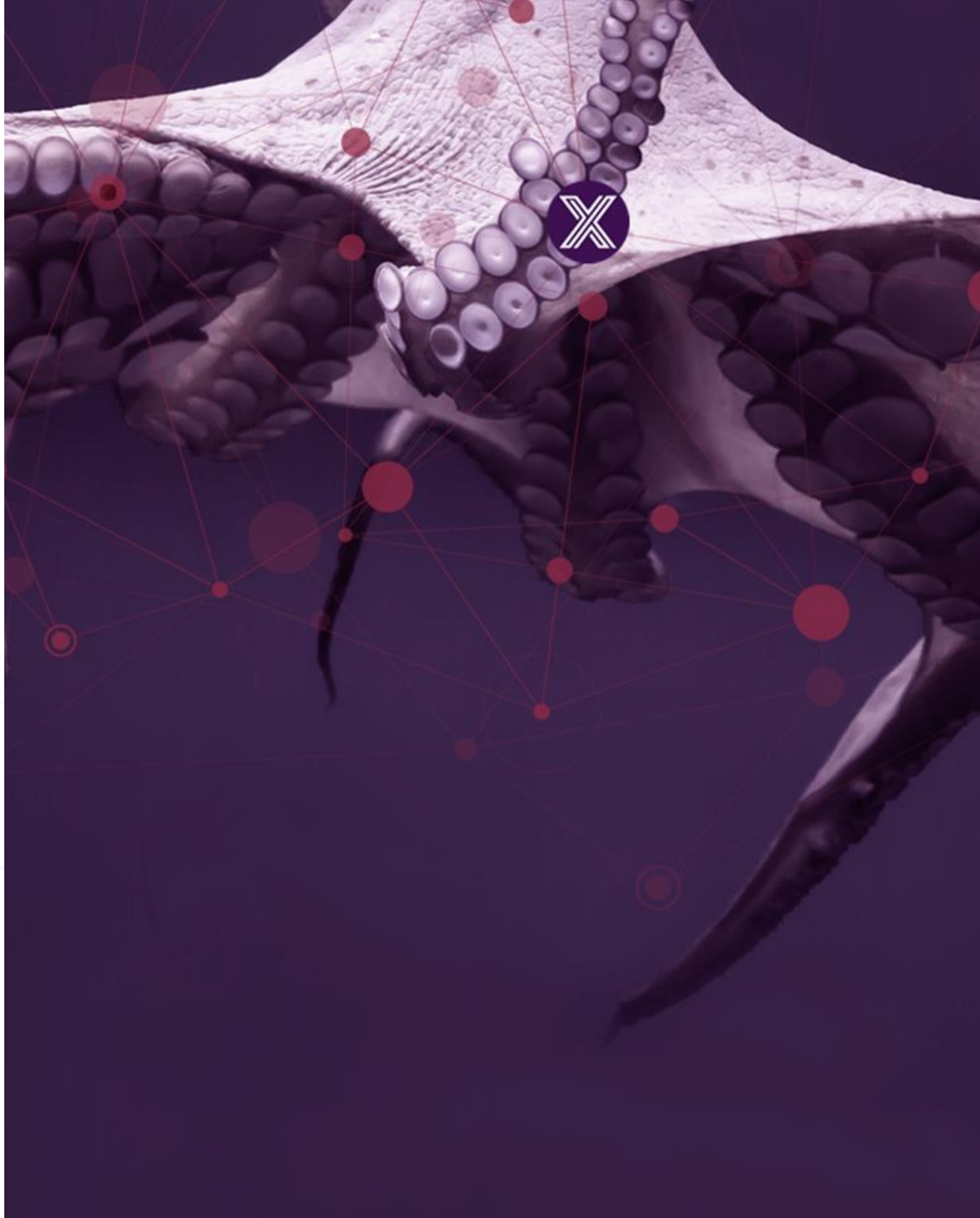
- Version 2.2?
  - Must still be backward compatible
  - Can add, but not remove deprecated items yet
  - Not LTS, but must consider we have LTS now on the books
- Major additions under consideration
  - North to south message bus
    - App service to command to start??
  - Metrics (per ADR)
  - ~~Service List ADR implementation~~
  - DS Filtering
  - Record and replay DS
  - ~~Consul available through the API Gateway~~

Scoping Exercise

- Categories we'll use
  - 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 Kamakura 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
    - A majority believe this work will not be covered in Kamakura
    - A potentially valid need, but just not going to be accomplished in Kamakura (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

EDGEXFOUNDRY™

Architecture Topics



Architect’s Topics list

- Metrics collection (ADR)
  - Control plane events too?
- ~~Dynamic service list (ADR)~~
- UoM implementation (ADR)
- Tracking project decisions (small to the large)
- Dynamic profiles (ADR)
- Global configuration
  - Remove duplication across configuration
- Declarative Kong
  - To remove Postgres
  - To reduce EdgeX image
- North-South message bus
  - Allow cloud or north side system to message command
  - Allow command to message device services
- Reducing EdgeX Size
  - Research only this release
- Getting rid of SMA (already deprecated)
  - What replaces start/stop/continue, config, service metrics
- Dynamic changes to any config
  - Callback on service for any change
- Expand notifications
  - Allow other protocols (SMS, messages, websockets)
  - Alarm based on metrics/resource usage
- Windows Support
  - ZMQ prohibits native support (how to provide option to remove ZMQ)
  - Package support (MSI)
- Executables containing multiple services as runtime option
  - Research only this release
- Things provisioning
  - More autonomous
  - Better device services
- Kubernetes
  - What’s next?

Metrics Collection – In Scope

- ADR - https://github.com/edgexfoundry/edgex-docs/pull/268
- Each service to send metrics to topic
  - Ex: number of events sent, average time per request, etc….
- How to implement
  - Go metrics, other package
  - Impact to C services
- Considerations
  - Impact to size
- Also need control plane events??
  - Ex: a device was provisioned, a service was stopped, config has changed…

UoM – In Scope

- ADR - https://github.com/edgexfoundry/edgex-docs/pull/386
- Can’t find a spec that is universally used
- Allow for adopter to specify but add to UoM to event/readings
- Validation - later

Tracking project decisions – In scope

- ADR has been used (successfully?) for large stuff
- What about all the smaller and project decisions by WGs?
- Project is losing track
  - PR process is too heavy to submit for all decisions
  - Needs to be accessible by everyone
  - Needs to be quick to update

- For reference: Kubernetes
  - https://github.com/kubernetes/community/blob/master/sig-architecture/README.md

Dynamic Profiles – In Scope (certainly for design)

- ADR - https://github.com/edgexfoundry/edgex-docs/pull/605
- What do we allow to be added, removed, updated on profiles (and their associations) once system is running
  - Adding easier than allowing updates/removals
- Needs to be enforced across the system
  - CLI, GUI, APIs, etc.

Global Configuration – Under consideration (at least for ADR/design)

- Is it time to consider some sort of global configuration?
  - We have a lot of duplication
  - Requires a lot of config to be touched when an adopter wants to change one item
    - Examples: topic name, logging
    - Turn on/off metrics collection in the future
- Difficulty – where would the configuration go?
- Considerations: with or without Consul, with or without security (and Vault)
- Additional consideration: containerized vs non-containerized (Snaps)

Declarative Kong – Never scope

- Kong is big and Postgres makes it bigger
- Questions
  - Does it help with size?
  - Can you configure groups/users ACL
    - Only supports JWT users
- IOTech research
  - Image size is the same as regular Kong, but saves 157MB image size for Postgres
  - Memory consumption of Declarative Kong increases a little (+ 20MB) because data is stored in memory (but saves 45MB memory used by Postgres)
  - It would make the API Gateway read-only.
  - Can’t create new users and tokens dynamically
  - Based on IOTech customer needs, dynamic needs outweigh size issue (but general EdgeX size is becoming a problem)

North-South message bus – under consideration (design/ADR)

- We’ve successfully implemented the ability to send sensor data from south to north via message bus
- No way to send requests from north to south in message bus
- Message bus provides means for:
  - Guaranteed delivery and better QoS where needed
  - Messaging preferred in many use cases
- Would require (and could be done in phases)
  - Command service to receive and relay messages
  - DS to receive/process messages
- Other questions
  - Would AS need message bus acknowledgement
  - Rules engine to call message bus or REST option?
  - What about other services using message bus (support)

EdgeX Size Concern – on the fence (research at least)

- EdgeX size is getting bigger
  - Of course there is more functionality too
  - Better security – but at a price
  - Message bus support – but requires infrastructure (potentially a broker)
- 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
- It is time to research alternatives or even consider our own implementations in some cases
  - Could we encourage 3rd party partners to develop edge specific alternate?
  - We need to be careful in creating custom security service that are not trusted
    - First rule in security – don't create your own security solutions
- Spend this release researching options

Component	Size
Consul	113MB
Kong (and its DB)	308MB
Vault	196MB
Average EdgeX service	~20MB

Deprecating SMA – on the fence

- It is already deprecated
- What replaces it? Does anything replace it?
  - Start/Stop/Restart capability
  - Metrics of services (memory, CPU)
  - Get config
- Docker or K8s provides this
  - What about non-containerized or Snaps
- What is the timing of removing

Dynamic Config – out of scope (keep in the backlog)

- We have some “writable” config elements
- Why isn’t all of the config “writable”
  - But may require service restart
- Would need callback capability in all services (assume this is there to some extent already)
- Consideration: Consul vs config file

Expand Support Notifications – on the fence

- Adopter use case needs
  - Send notifications via SMS, web socket or message protocol (MQTT)
  - Make it easier to use/send notifications
    - Better examples
- Allow rules engine to trigger notifications
- Allow configuration (vs coding) of alarms
  - Notifications based on certain conditions
    - Examples
      - Sensor value falls outside of a threshold
      - Service metric falls outside of a threshold (time to respond is > than a number of milliseconds)
      - Service memory use exceeds a threshold of MB

Windows Support – under considersation scope

- We claim platform independence
- Running natively on Windows is now nearly impossible (outside of Docker or something like WSL2)
  - ZeroMQ does not compile on Windows 10 (Mike J script broken)
- Can we keep backward compatibility but allow for no-ZMQ windows option?
- Can we provide Windows MSI for base EdgeX install (like Ubuntu Snap with everything included?)
- And testing in CI/CD for Windows? Ugh

Container with multiple executables (collapse multiple containers) – out of scope

- For some use cases or some environments, can we simplify EdgeX deployment
- Can all of core and supporting services be deployed in one container?
- Can all of app services and rules engine be deployed in one container?
- Are there other likely combinations?
- A research project for this release?
- People could do this on your own
  - Consideration of env vars (like port) and such make this complex
- Moving command API in to metadata (another scope option) – under consideration
  - Command API is tiny
  - Doesn’t have its own DB or anything; current implementation make this trivial
- Another scope option: one image, multiple containers (out of scope)
  - Need to reduce number of containers in general
    - Pull times
    - Security / sign containers

Things Provisioning – under consideration

- Adopters are asking for more automatic discovery/provisioning
  - “Zero touch” provisioning
- How can we improve and simplify the discovery of devices?
- How can we improve and simplify the automatic provisioning of devices?
- Device configuration file is helping
- How about some tooling too?

Kubernetes – under consideration

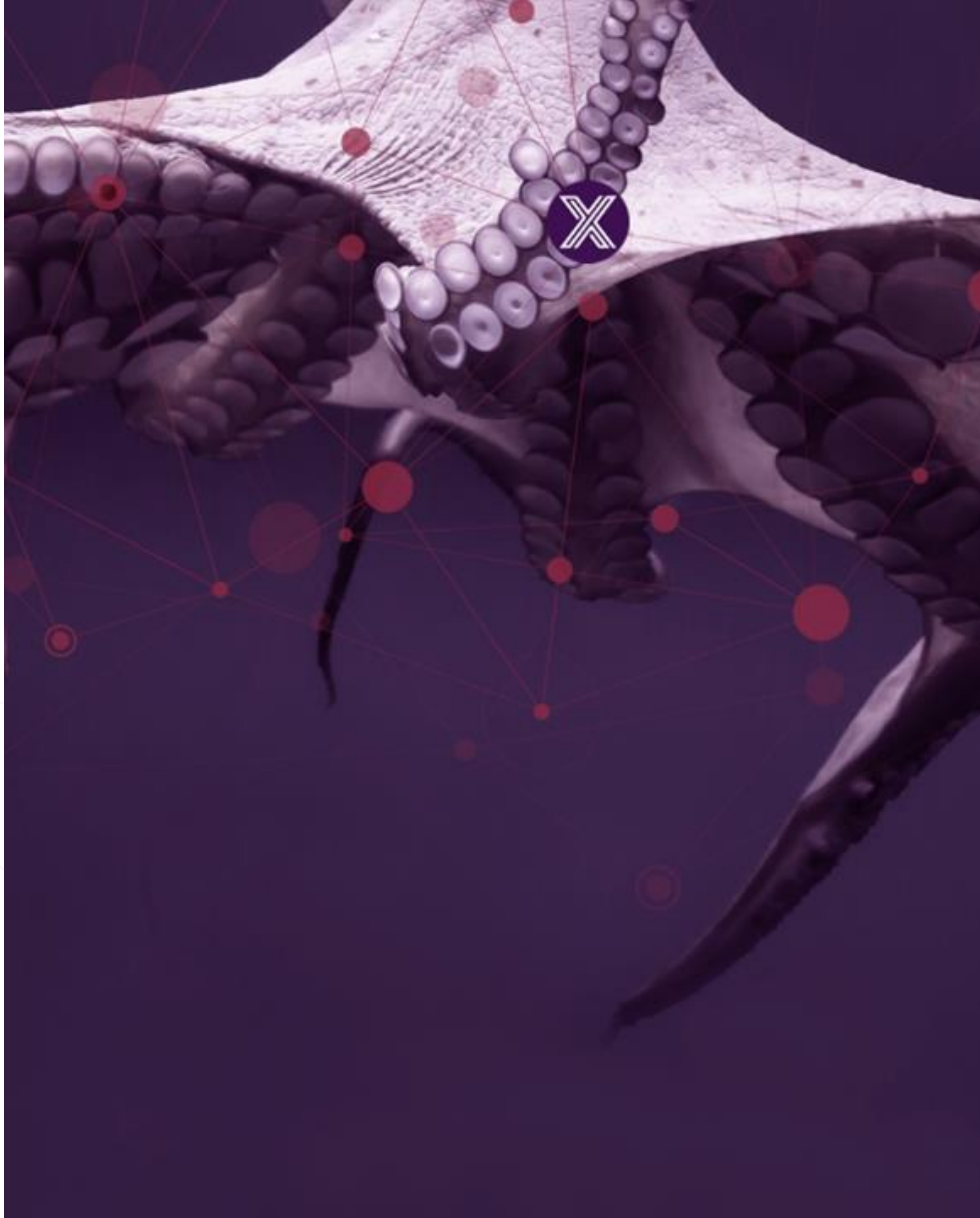
- What’s next?
- We have an example
- In the past we had a helm chart example
- Intel members say “can’t run EdgeX in K8s”
    - How can we get more (services) of EdgeX running in K8s
        - What’s blocking from running all of EdgeX in K8s?
        - Why and why is not in a form that most in K8s field accept?
        - Expectation of scalability – how do we address?

Architecture Scope

- In
  - Dynamic Profiles
  - Tracking project decisions
  - Metrics
  - UoM
- Under Consideration
  - K8s
  - Thing provisioning
  - Window Support
  - North/South message bus
  - Global configuration
  - Merge Command/metadata
- On the fence Consideration
  - Expand support notifications
  - Deprecate SMA
  - EdgeX size
- Out of scope
  - Multiple execs per container
  - Single image/multiple container
  - Dynamic config
- Never
  - Declarative Kong

EDGEXFOUNDRY™

General and Work Group Scoping



General

In scope
Under consideration
On the fence

- Upgrade Go (1.17?)
- Remove ZMQ to allow for Windows native
  - Backward compatibility issue – all go to remove issues
  - In scope to try
- Always send CBOR option for performance
  - Test first and apply if it makes sense
  - What would “significant” savings look like?
  - >100 milliseconds?
- Replace poorly supported 3rd party libs
  - bitbucket.org/bertimus9/systemstat
  - github.com/armon/circbuf
  - github.com/go-logfmt/logfmt
  - github.com/mitchellh/consulstructure
  - github.com/pascaldekloe/goe
  - github.com/ryanuber/columnize
  - github.com/sean-/seed
  - 3 V1 CLI’s as well

- Global Configuration
- Metrics ADR implementation
- Service List ADR implementation
- UoM support
  - ADR for review
- Dynamic change to any configuration
  - Watcher for all config

Core

In scope
Under consideration
On the fence
Out of scope

- Core
  - Expand the notification service
    - SMS, message bus, web sockets
    - Raise alarms (when data or metric out of threshold)
- GUI
  - Fix unallowed activity (ex: mod device profile)
  - Testing
- CLI
  - V2
  - Looking for input (support app and device service)
  - Support for registry, proxy
- System management
  - What replaces current SMA capability (config get, service metrics, start/stop/restart)
- Test QA
  - Real hardware testing
    - In the LF lab?
    - At least a pilot (small)
    - CI/CD integration?
  - Improve testing (from our Jakarta planning)
    - Test without optional services, message bus v REST, etc.
    - Improved integration test (Kuiper)
    - Improved stress tests
    - Improved performance testing (other device types, CBOR, etc.)
    - Break-tests/robustness tests (reboot testing)/ chaos testing

- From the Icebox
  - Core
    - Prometheus integration
    - Move to typing in JSON
      - Tech debt to use interfaces vs string:string in properties
      - Complexity - using the string match on provision watcher
    - Support automatic migration of devices between services
      - Would need use case and research first
      - Need better discovery 'stuff' first
    - Services to notify system socket when ready
      - systemd – acknowledgement by service
    - Add Nats implementation for message bus
    - Add Websocket support for support notifications
    - Support custom attribute in Device object
      - Labels very simplistic and not what they are intended for
      - Eaton requirement help
      - Narrow discussion to support for parent-child
    - Update lastConnected should not trigger device service callback
  - Test
    - Trigger performance tests from edgex-taf
  - Jakarta Stretch
    - MQTT implementation doesn't handle different hosts for pub/sub
    - Refactor the pkg.Encode func name
      - Core metadata tech debt

Device Services

In scope
Under consideration
On the fence
Out of scope
Never

- C SDK work outstanding
  - Complete securing consul with access tokens
  - Complete securing the message bus
  - Secret provider for all
- BACNet DS
- Review/approve contributions (UART, GPIO, CoAP)
- DS Filtering
  - ADR already in place
- Record and replay DS
  - Wiki doc design already in place
- Downsampling
  - Throttling device data send based on ability to consume/use
- C SDK library
- Use cmake options to optimize dependencies
- Support regular expressions in assertions
  - Poorly designed assertions; maybe we don't need it at all
  - Use cases for it??

- Icebox
  - DS operatingState doesn't go down post DS stop (C)
  - ~~Support certain dynamic updates to device profiles (arch issue too)~~
  - ~~Support custom attributes in Device object~~
  - Implement size constraints for devices and profiles
    - Thinking this may never be done
  - Implement ProtocolProperties validation mechanism
  - Option to create multiple events with readings
  - ~~Replace modbus import lib (goburrow/modbus)~~
  - Implement resource command chaining
  - Use Yaml v3 library (vs V2) perspective
    - From code hygiene
  - Handle both JSON request bodies as well as CBOR request bodies

Application Services

- Fork the pipeline (carry over)
- Alternate language functions SDK
  - Python?
  - More script based language option
- AI/ML integration
  - TensorFlow Lite

- From the Icebox
  - Use nano message IPC for ASC pluggable pipeline function
  - CloudEvent as TargetType

Security

rank	url	title
10	https://github.com/edgexfoundry/device-camera-go/issues/104	Internal Digest implementation doesn't work with ONVIF Foscam/amcrest IPCAM
20	https://github.com/edgexfoundry/edgex-go/issues/3744	Expose Consul UI on the host when EdgeX is operating in secure mode (snap)
30	https://github.com/edgexfoundry/device-sdk-c/issues/335	[Secure Consul Ph2] Device SDK C: Enable Registry/Config access tokens
40	https://github.com/edgexfoundry/device-sdk-c/issues/314	Secure message bus connection for C SDK
50	https://github.com/edgexfoundry/edgex-go/issues/3773	Implement ADR 0019 - Delayed start secret store tokens (SPIFFE ADR)
60	https://github.com/edgexfoundry/go-mod-secrets/issues/129	Add "make lint" target and add to "make test" target (only for GO)
70	https://github.com/edgexfoundry/edgex-docs/issues/515	Update SSH Tunneling for remote Device Service how to for V2
80	https://github.com/edgexfoundry/edgex-docs/issues/516	Update Swarm for remote Device Service how to for V2
90	https://github.com/edgexfoundry/edgex-go/issues/3258	[Secure Consul Ph. 3] Fine tune Vault configuration as to Consul's secret engine token's TTL
100	https://github.com/edgexfoundry/edgex-go/issues/3257	[Secure Consul Ph. 3] Create ACL Policies and Roles for all services and per EdgeX service
110	https://github.com/edgexfoundry/edgex-go/issues/3158	[Secure Consul Ph. 3] Create a global Consul token (stored in a file) for use in remote configuration
120	https://github.com/edgexfoundry/edgex-go/issues/1950	ADR for microservice communication security (ADR first – not necessarily implement)

Wish list

200	https://github.com/edgexfoundry/edgex-go/issues/3715	Integrate GoKart security scanning
200	https://github.com/edgexfoundry/edgex-go/issues/3747	Investigate alternatives to Kong that have better platform support and use less memory
200	https://github.com/edgexfoundry/edgex-go/issues/3690	Research for Docker base image recommendation (busybox or etc. option vs alpine)

Security

500	https://github.com/edgexfoundry/edgex-docs/issues/136	HOW-TO enable remote device-service service via service mesh
500	https://github.com/edgexfoundry/edgex-go/issues/2470	Image signing for EdgeX images published to Docker Hub
500	https://github.com/edgexfoundry/edgex-go/issues/3227	Refactor security-bootstrapper Vault's Consul Secret Engine APIs using go-mod-secret (yet to be created)
500	https://github.com/edgexfoundry/edgex-go/issues/1944	Secret store unsealing daemon
900	https://github.com/edgexfoundry/edgex-go/issues/3584	Remove superfluous delay in consul_wait_install.sh and speed up Consul initialization
900	https://github.com/edgexfoundry/edgex-go/issues/3544	edgex-vault logs contain errors about revoking consul token leases
900	https://github.com/edgexfoundry/edgex-go/issues/3583	Remove dead code for API gateway flows
900	https://github.com/edgexfoundry/go-mod-secrets/issues/32	Define public constant for EDGEX_SECURITY_SECRET_STORE
900	https://github.com/edgexfoundry/edgex-go/issues/3182	[Tech Debt] Refactor the anonymous function inside secretstore-setup for init Vault

All tech debt – out of scope for Kamakura

Security Misc

- Declarative Kong is not the problem. The problem is that Kong itself is big. Fundamentally the question we need to answer is whether or not we need an API gateway. If we can push authentication into individual microservices, then we can replace Kong with just a URL-rewriting reverse proxy, possibly with TLS termination. If we put the API gateway in charge of authentication, then necessarily the microservices, to get to the next level of security, need to somehow validate that the incoming request has passed through Kong in order to authenticate it. This usually means mutual-auth TLS, which is often an enterprise feature, thus sucking us into using a feature-rich API gateway component.
- We should perhaps be thinking about enabling the "bring-your-own Vault" approach, similar to what I proposed for SPIFFE/SPIRE support. This would allow users to potentially run the Vault instance in a cloud where it is more secure.

- On EdgeX size. I'd like to throw out some unusual ideas, including:
  - Drive EdgeX to a monolith implementation like a Java server, k8s, or microk8s -- package up the core EdgeX microservices as libraries, and link them into a single executable that spawns a listening port for each microservice. The all-in-one service should have a reduced runtime footprint and also be smaller than the combined sizes of individually compiled microservices.
  - Turn CGO back on - compile EdgeX services with a C runtime library dependency. That means that each microservice wouldn't have it link in Go's implementation of the C runtime. There will be security implications to this wrt adding stack canaries and other things to mitigate the risk of the C library linkage.
  - Make an EdgeX uber-container that contains all of the EdgeX binaries in an internally-consisted file system layout. Use command line to determine which executable to run. Many solutions use this to reduce the number of containers that need to be downloaded.

DevOps

- Potentially automate badge awarding

- From the Icebox
  - Implementation of Docker CIS Security bench pipeline
  - Investigate Snky API

Miscellaneous

- Tool/script to create new device or application services
- ~~Snaps~~
  - ~~???~~
- Examples
  - Cloud export – more explicit for Azure, AWS, Google
  - Update cloud templates
  - Timeseries DB export (Influx)
  - RP4
  - More Kubernetes support/examples
- Docs
  - Device service restructure and adding undoc'ed features
  - Documenting the API (like discovery)

Outreach

- EdgeX badging – trial and eval in 2022
- EdgeX Ready – eval by early 2022
- EdgeX Certified – on hold
- Regular Web site and download stats production
- Hackathon (English)
- Regular blog/vlog posts
- Adopter series – time to make it ad hoc