



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

Kamakura Release
Oct 17, 2021, 8-11am PDT

Conference Agenda

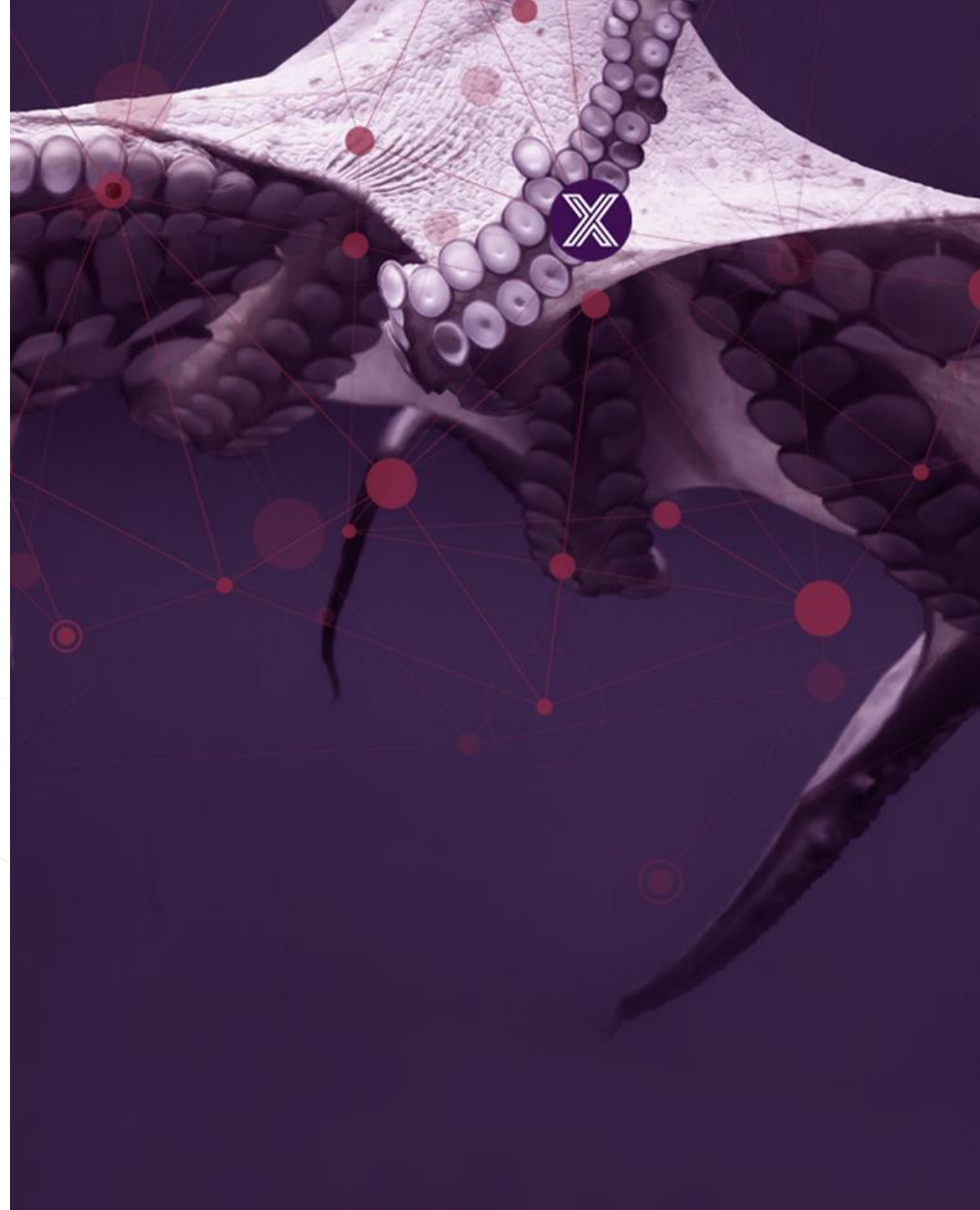
All times in
PDT

- 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



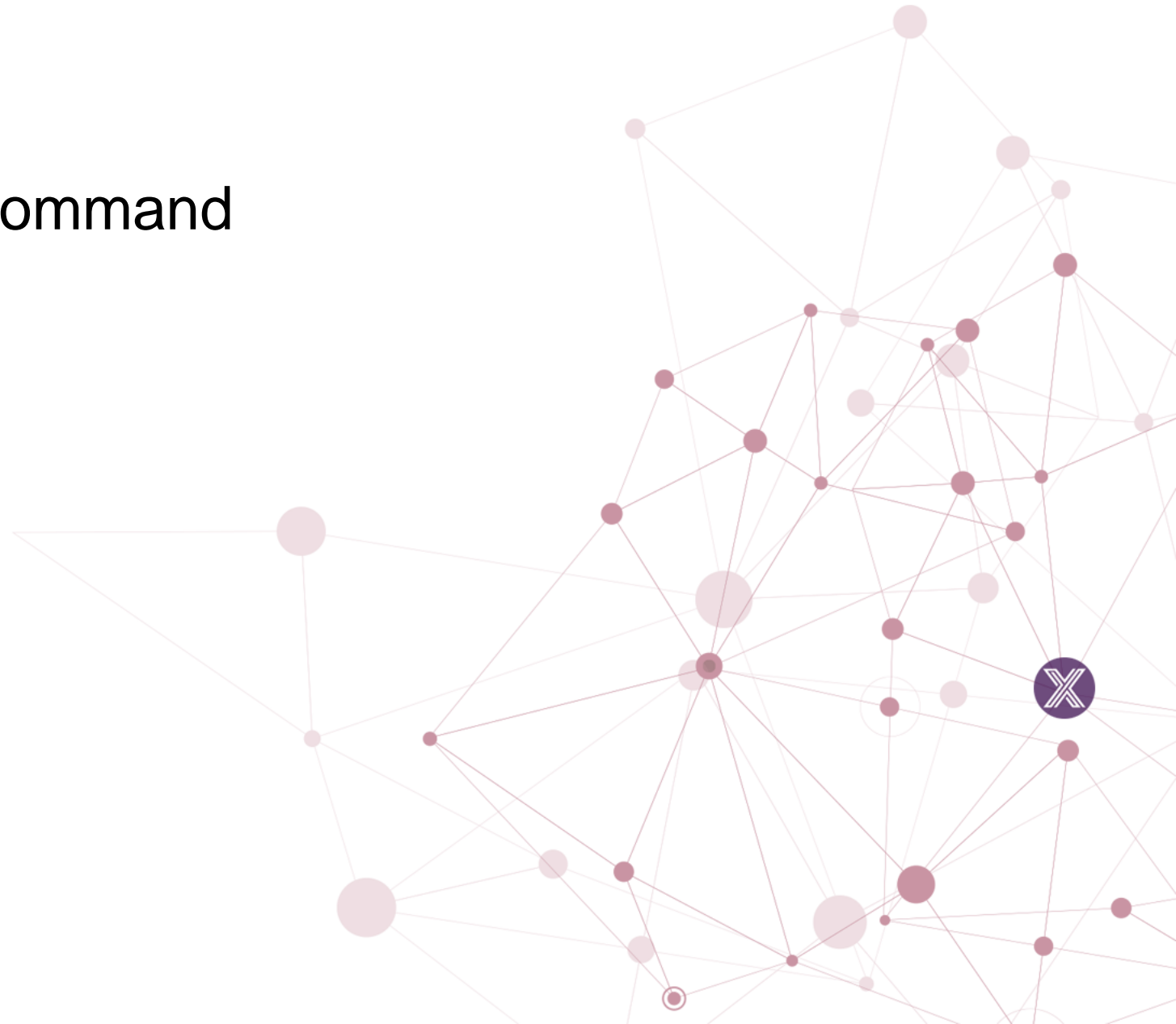
Intros

Name, rank, serial number please



Jakarta Items

- Object data type in get/set command



Kamakura Release Objectives/Size

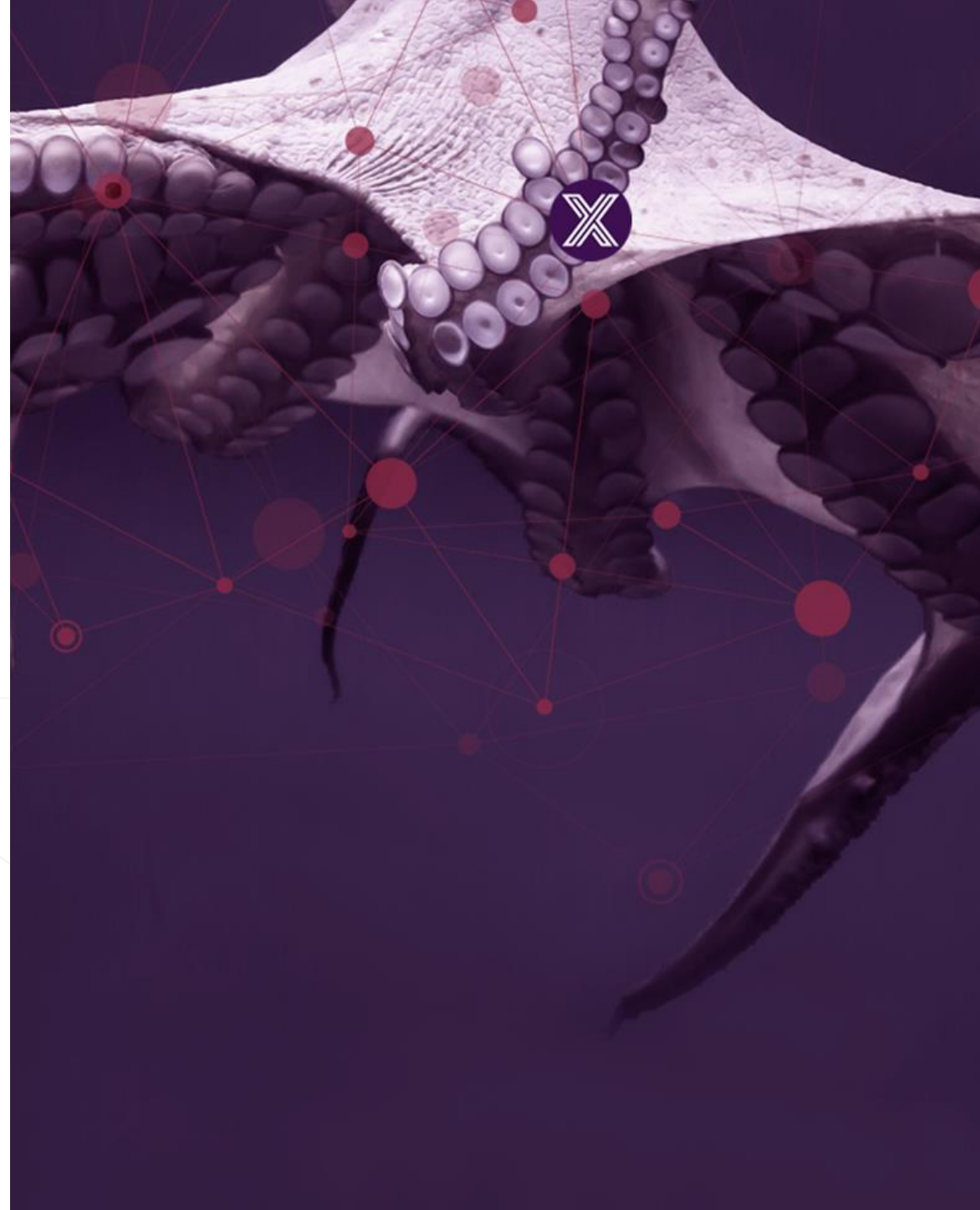
- Version 2.2?
 - Must still be backward compatible
 - Can add, but not remove deprecated items yet
 - Not LTS, but must consider we have LTS now on the books
- Major additions under consideration
 - North to south message bus
 - App service to command to start??
 - Metrics (per ADR)
 - Service List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

EDGE X FOUNDRY™

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

Metrics Collection

- 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

Dynamic Service List

- ADR - <https://github.com/edgexfoundry/edgex-docs/pull/377>
- Many modules need a (dynamic) list of services
 - As DS and AS get added/removed
 - As future 3rd party services (like analytics) get added/updated
- Implementation
 - Where does it live for secure and non-secure modes

UoM

- 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

- 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

- 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

- 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)

Declarative Kong

- 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

- 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)

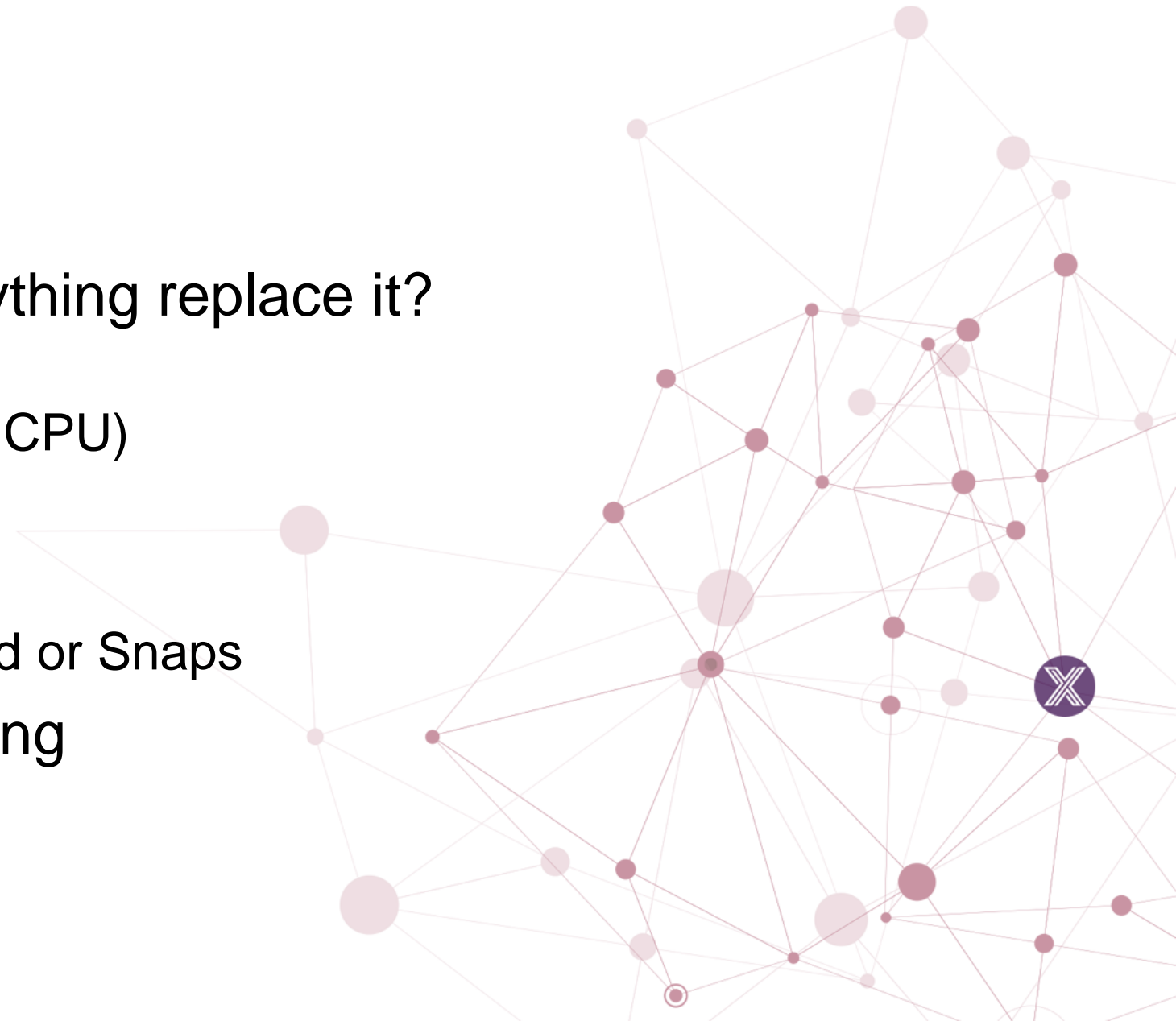
EdgeX Size Concern

- 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

- 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

- 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

- 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

- 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?)

Container with multiple containers

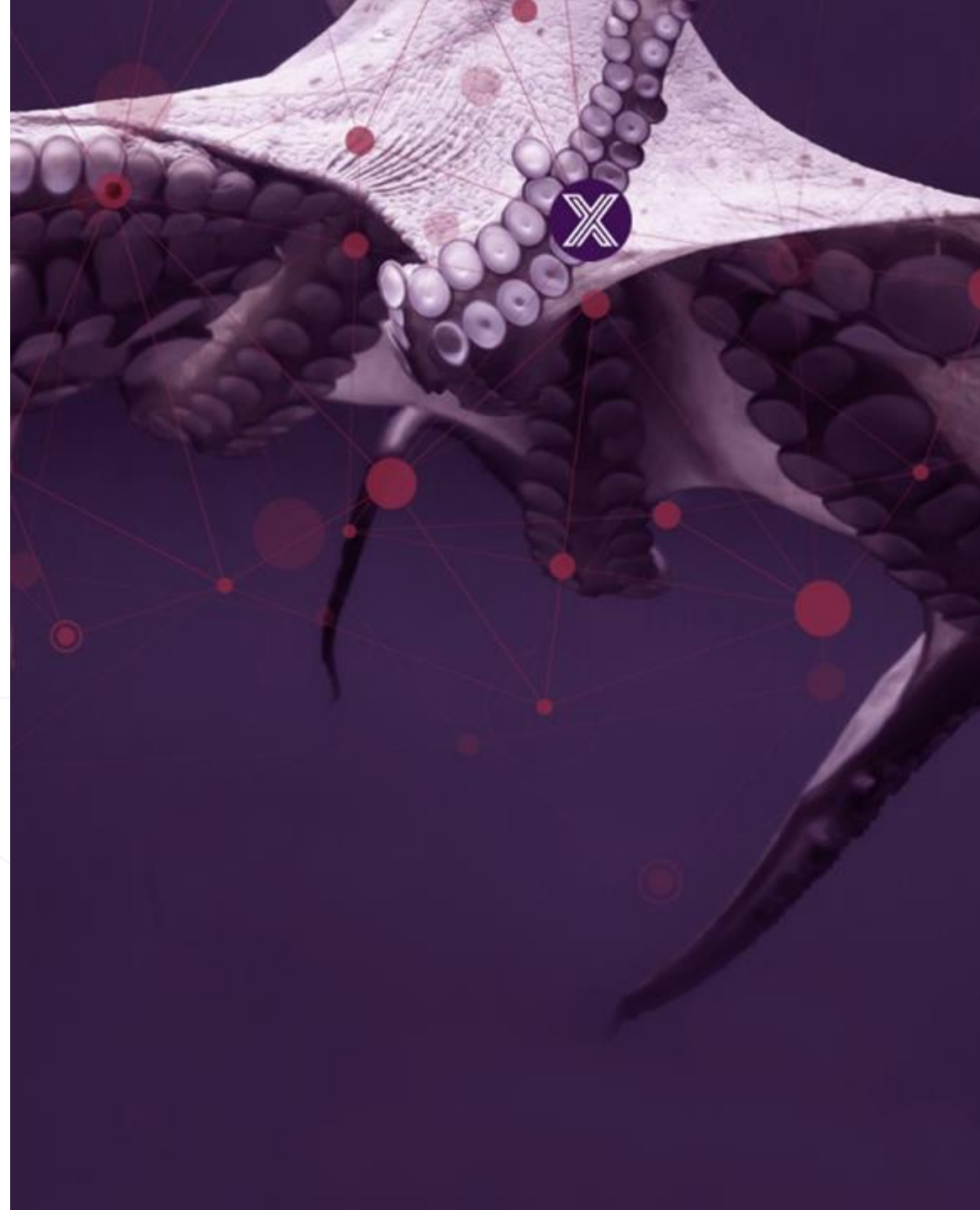
- For some use cases or some environments, can we simplify EdgeX deployment
- Can all of core and supporting services be deployed in one container?
- Can all of app services and rules engine be deployed in one container?
- Are there other likely combinations?
- A research project for this release?

Things Provisioning

- 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?



General and Work Group Scoping



General

- Upgrade Go (1.17?)
- Remove ZMQ to allow for Windows native
 - Backward compatibility issue
- 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

- 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 and then what
- System management
 - What replaces current SMA capability (config get, service metrics, start/stop/restart)
- Test QA
 - Real hardware testing
 - In the LF lab?
 - 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.)

- From the Icebox
 - Core
 - Prometheus integration
 - Move to typing in JSON
 - Support automatic migration of devices between services
 - Services to notify system socket when ready
 - Add Nats implementation for message bus
 - Add Websocket support for support notifications
 - Support custom attribute in Device object
 - Update lastConnected should not trigger device service callback
 - Support object value type in Set Command
 - Allow Get Command to pass parameters in the payload
 - Test
 - Trigger performance tests from edgex-taf
 - Jakarta Strech
 - MQTT implementation doesn't handle different hosts for pub/sub
 - Refactor the pkg.Encode func name

Device Services

- 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
 - Support regular expressions in assertions
- Icebox
 - DS operatingState doesn't go down post DS stop (C)
 - Support object value type in Set Command
 - Allow Get Command to pass parameters in the payload
 - Support certain dynamic updates to device profiles (arch issue too)
 - Support custom attributes in Device object
 - Implement size constraints for devices and profiles
 - 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)
 - 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

- Make lint
 - Done on edgex-go
- Expose Consul UI through the API Gateway
- From the backlog
 - Kong alternative
 - Remove delay in consul_wait_install.sh
 - edgex-vault log contains errors
 - Remove dead code from API gateway flows
 - Define public constant for EDGEX_SECURITY_SECRET_STORE
 - Refactor anonymous function inside secretstore-setup
 - Microservice communication security
 - Update HOW-TO enable DS service via service mesh
 - Image signing for Docker
 - Integrate GoKart security scanning
 - Secret store unsealing daemon
 - Research for Docker base image recommendation

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
 - Any changes, needs or recommendations for improvement?

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