



EDGE X FOUNDRY™

EdgeX Planning Conference (virtual meeting)

Levski Release
May 16-19, 2022

Conference Agenda

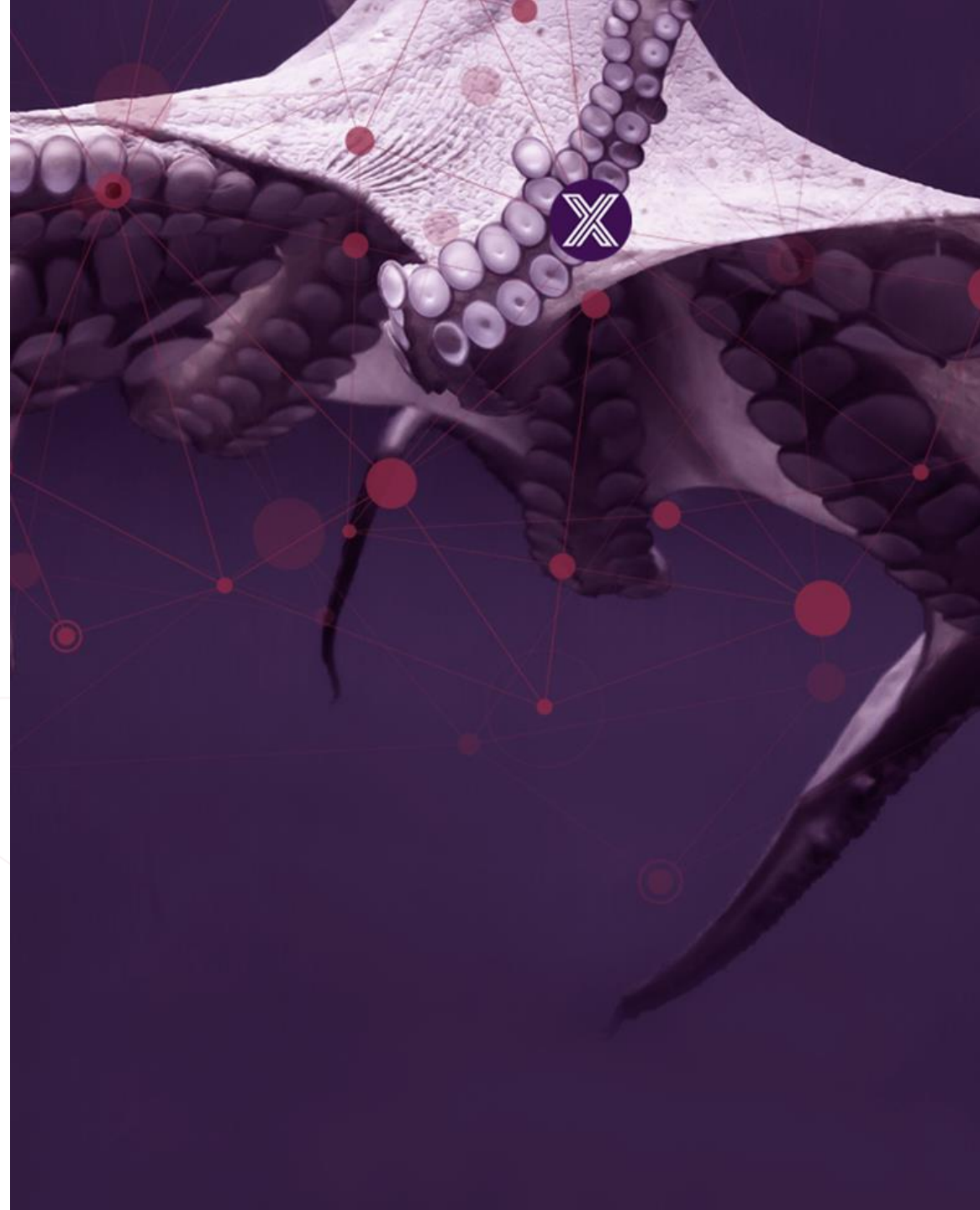
- May 16, Monday, 7am
 - Business Topics
 - Architectural Tee-up
- May 17, Tuesday, 9am
 - Cadence & Process
 - Architectural issues
- May 18, Wednesday, 7am
 - Levski Scoping
- May 19, Thursday, 9am
 - Finish scoping and architectural issues
 - Kamakura lessons learned
 - Future planning meetings
 - Long term road mapping

All times in
PDT

EDGE X FOUNDRY™

Intros

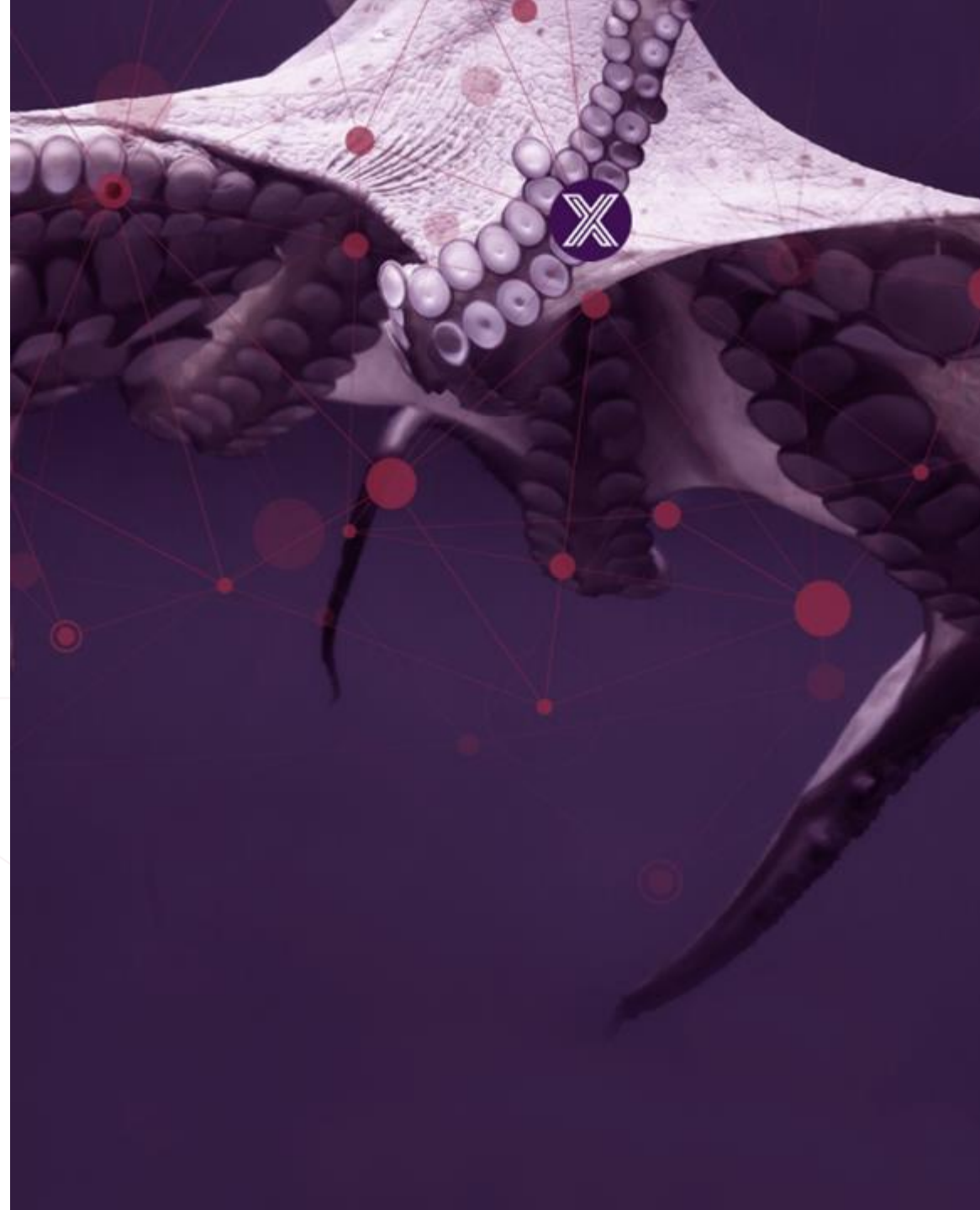
Name, rank, serial number please



EDGE X FOUNDRY™

Day 1

May 16



Conference Agenda – Day 1

- May 16, Monday (all times in PDT; all times tentative and fluid)
- 7am PDT Introductions & Logistics
- 7:15am Kamakura release (unfinished business, issues, etc.)
 - TBD
- 8:00am Levski release
- 8:30am Business Topics
- 9:00am Architecture topic tee-up
- 10am day 1 adjourn

Kamakura Release

- Still be done
 - TBD
- Issues/Concerns
 - TBD

Manual Testing To Do

Test	Owner/Actions
Test any configuration not already handled by new TAF tests	
Test reboot of the system. Make sure EdgeX can come back up	
Test device services (with real hardware where possible)	
Working with service authors to identify minimal tests	
Test different combinations/permutations of app functions in app services	
Test on Windows	
Test running with multiple services of same type (device service and app services)	

Levksi Release

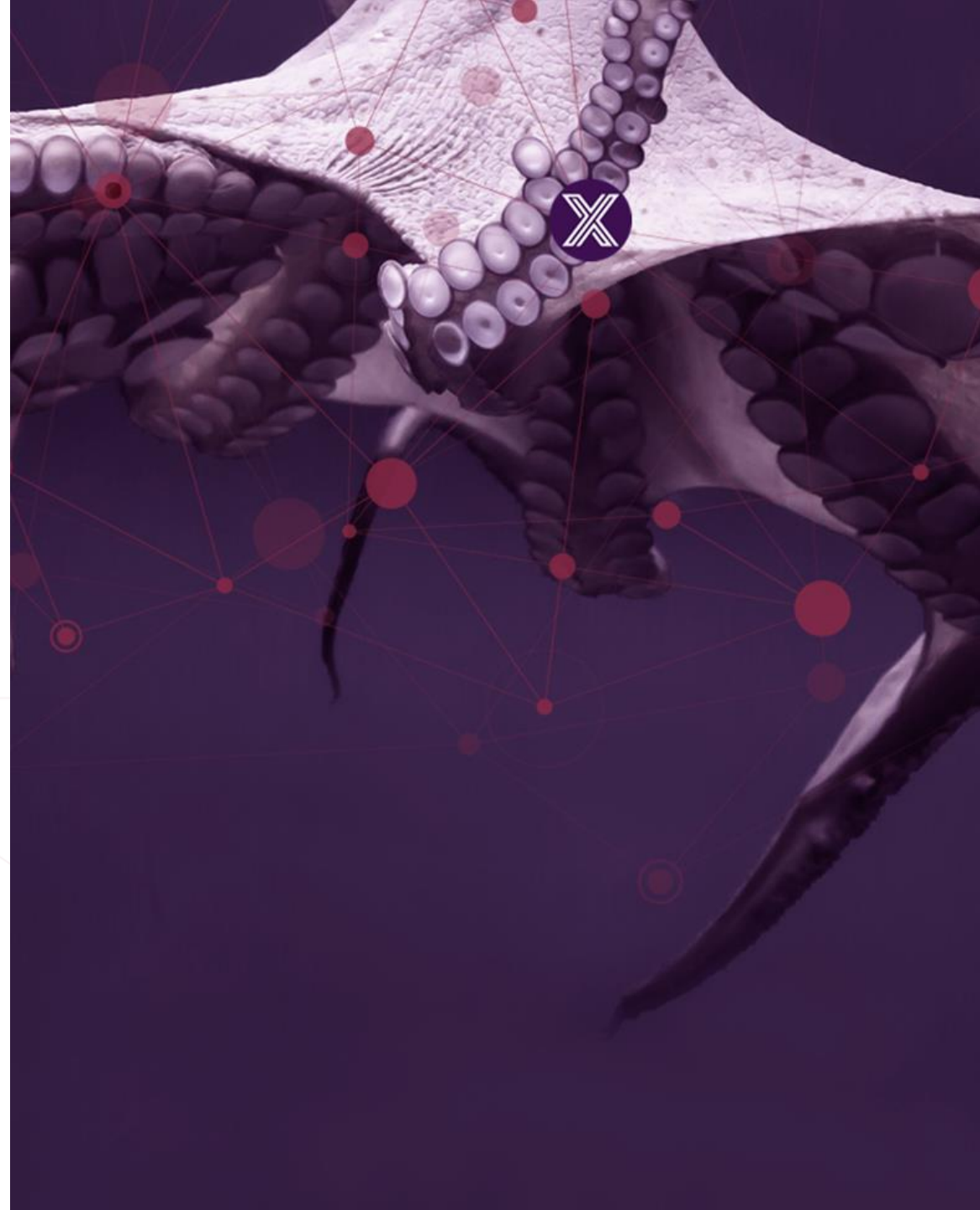
- Version 2.3?
 - Must still be backward compatible
 - Can add, but not remove deprecated items yet
 - Not LTS, but must consider we have LTS now on the books
- Major additions under consideration
 - Unit of measure implementation
 - More metrics
 - North to south message bus implementation
 - CPE implementation
 - Core Data Cache implementation (under consideration)
 - DS Filtering (under consideration)
 - Record and replay DS (under consideration)

Business Topic Agenda

- Web Site (Andy/Gavin & Melvin on China site)
 - Current stats and status
 - Any planned improvements or changes
 - Security issues
- Developer Evangelism (Jim)
 - EdgeX Ready (Rodney)
 - Tech Talks
 - Hackathon
 - Return to events in 2022 (Fall?)
- Do we still need an Outreach WG (and TSC voting position)?

EDGE X FOUNDRY™

Architecture Tee Up



Levski Major Themes and Objectives

- Dot release
- Major additions under consideration
 - Unit of measure implementation
 - More metrics
 - North to south message bus implementation
 - CPE implementation
 - Core Data Cache implementation (under consideration)
 - DS Filtering (under consideration)
 - Record and replay DS (under consideration)



Scoping

- Categories used in our pre-wire
 - In scope
 - It is “In scope” – must be done; no debate; near unanimous consent
 - Only discussion is what will be completed in this release
 - Under consideration
 - It is not in scope yet, but it worthy of some time to discuss
 - A strong tendency to put it in scope; in scope by default if not reviewed or decided
 - 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 being discussed in this meeting

Architect's Topics list

- In scope
 - ADR process improvements (will be covered separately)
 - Additional Metrics collection
 - What's telemetry is collected outside of core data
 - Last Connected/Last Reported
 - Possibly some other technical debt in Profile/DS/Device area
 - GUI and calls to SMA
- Under Consideration
 - How can we reduce the build times (especially for ARM)
 - Securing the MQTT message bus communications
 - Mostly for internal
 - ADR for global configuration
 - An EdgeX 3.0 topic but an ADR is needed sooner versus later because this could be hard to solve
 - Bring your own Vault
 - ADR for URI reference to point to config/profile/etc.

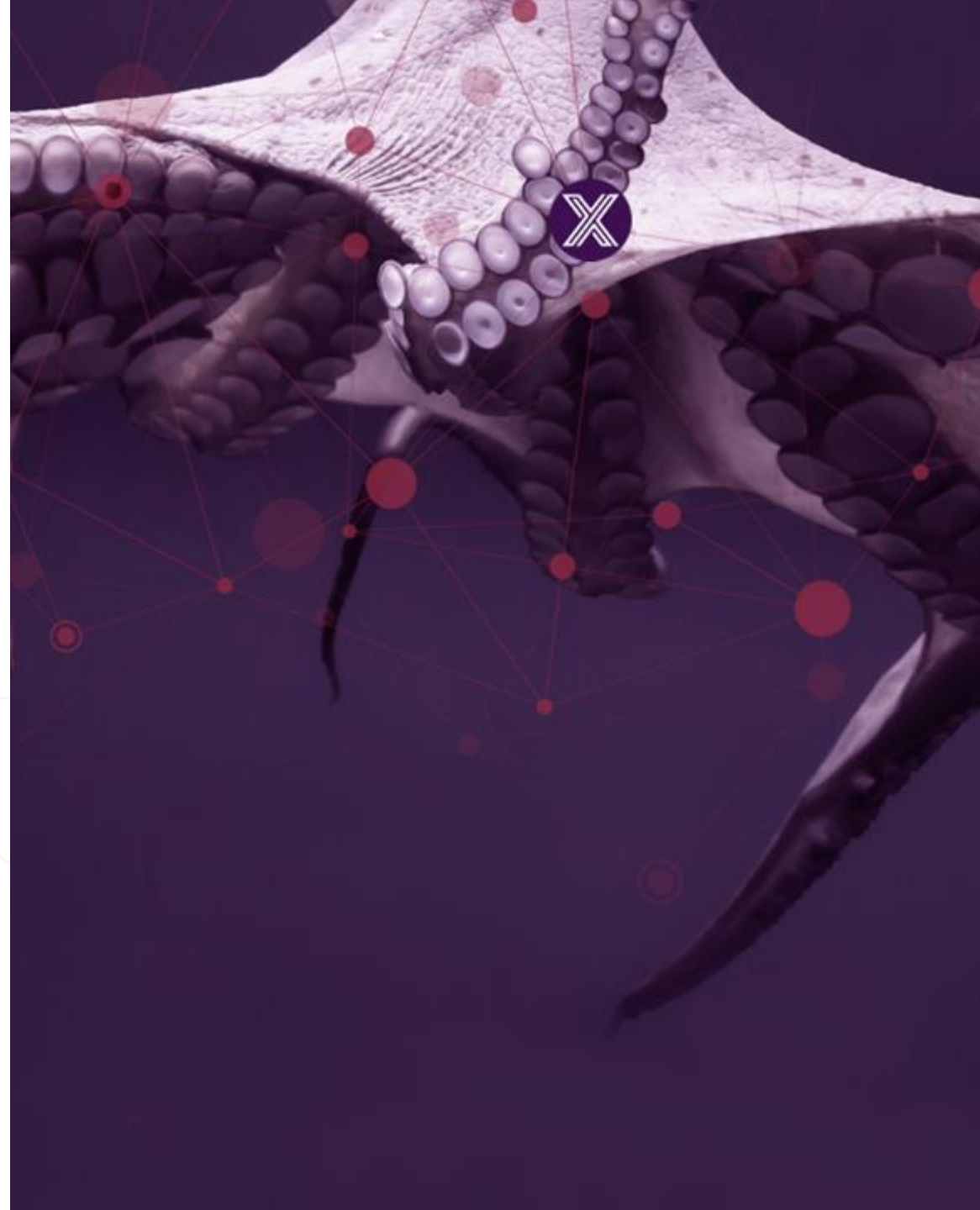
- On the fence
 - NATs implementation of go-mod-messaging
 - I18N/L10N support
 - Allow profile, service, device resource names, etc to use other character sets

[Details in slides for Day 2](#)

EDGE X FOUNDRY™

Day 2

May 17

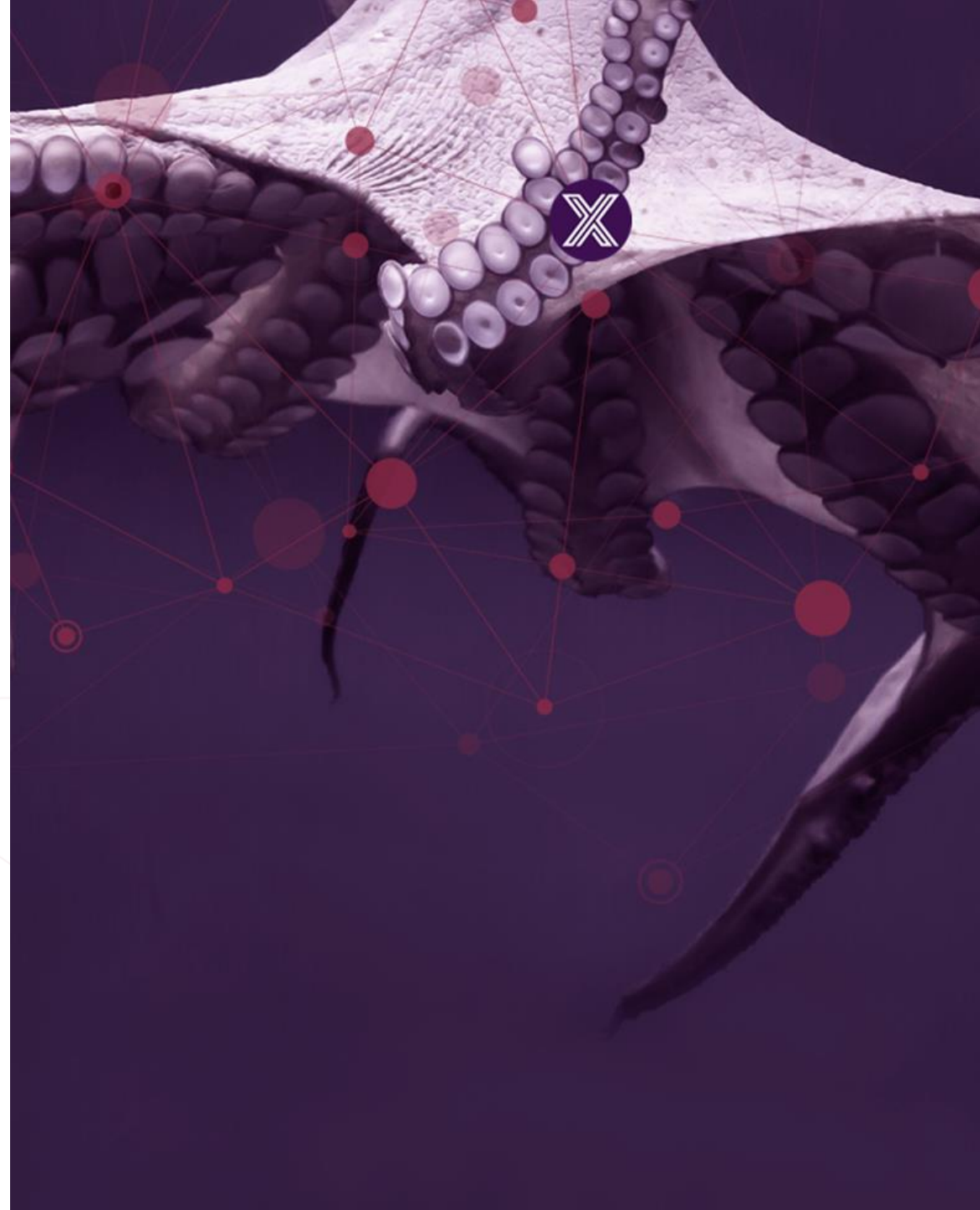


Conference Agenda – Day 2

- May 17, Tuesday (all times in PDT; all times tentative and fluid)
- 9am PDT Meeting Start
- 9:15am Release cadence check & release naming
- 9: 45am Process improvements
- 10:30am 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 Levski scope
- 12pm day 2 adjourn



Cadence & Release Naming



Cadence Check

- April & Oct/Nov remain target release months
 - Kamakura release – May 2022
 - Levski release – Nov 2022
 - Minnesota release – May 2023
 - Napa release – Nov 2023
- Venue for next F2F Meeting (Nov 2022)
 - With travel restrictions lifted – everyone in favor or a F2F
 - Volunteers, suggestions on where to hold this meeting?
- Conferences – marketing committee update
 - Target events to return to and when



Release Detail Timing

- Given circumstances of the last few releases, we have been flexible
 - Should we try to standardize the release timeline a bit for everyone's expectations and planning?
- Formalizing this for future releases (**Proposed**)
 - Release date selected (attempt to select 2 months in advance of release)
 - Wednesday / mid week target
 - September for fall
 - March for spring
 - Freeze date selected off the release date
 - Wednesday target to match release date
 - 3 weeks prior for substantial major release or LTS
 - 2 weeks prior for minor release
 - Planning meeting
 - Monday – Thursday virtual or Tuesday/Wednesday when in person
 - 2.5 weeks prior to a major or LTS release
 - 1.5 weeks prior to a minor release
 - Pre-wire meeting
 - Wednesday or Thursday prior to the Freeze date

Release Timing

- Formal release/planning schedule is now:
- Generally attempt to select release date about 2 months in advance of the release
 - Typically early March and early September for spring/fall releases
 - Release date preferred to have 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 training/non-conference
 - In person: Tuesday-Thursday with Thursday afternoon for training/non-conference

Formalized Planning

Example Calendar for Major/LTS Release

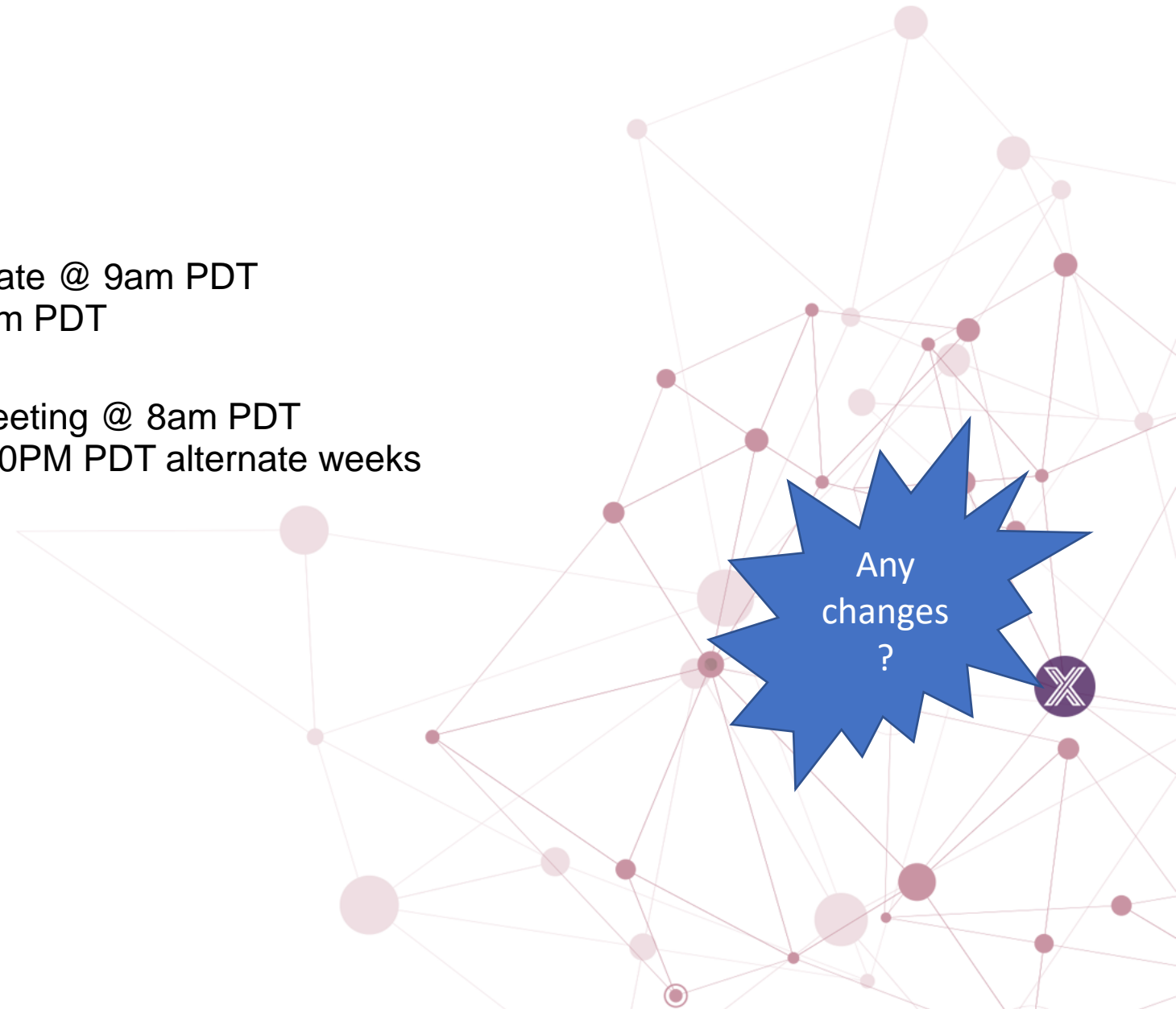


Example Calendar for Minor Release



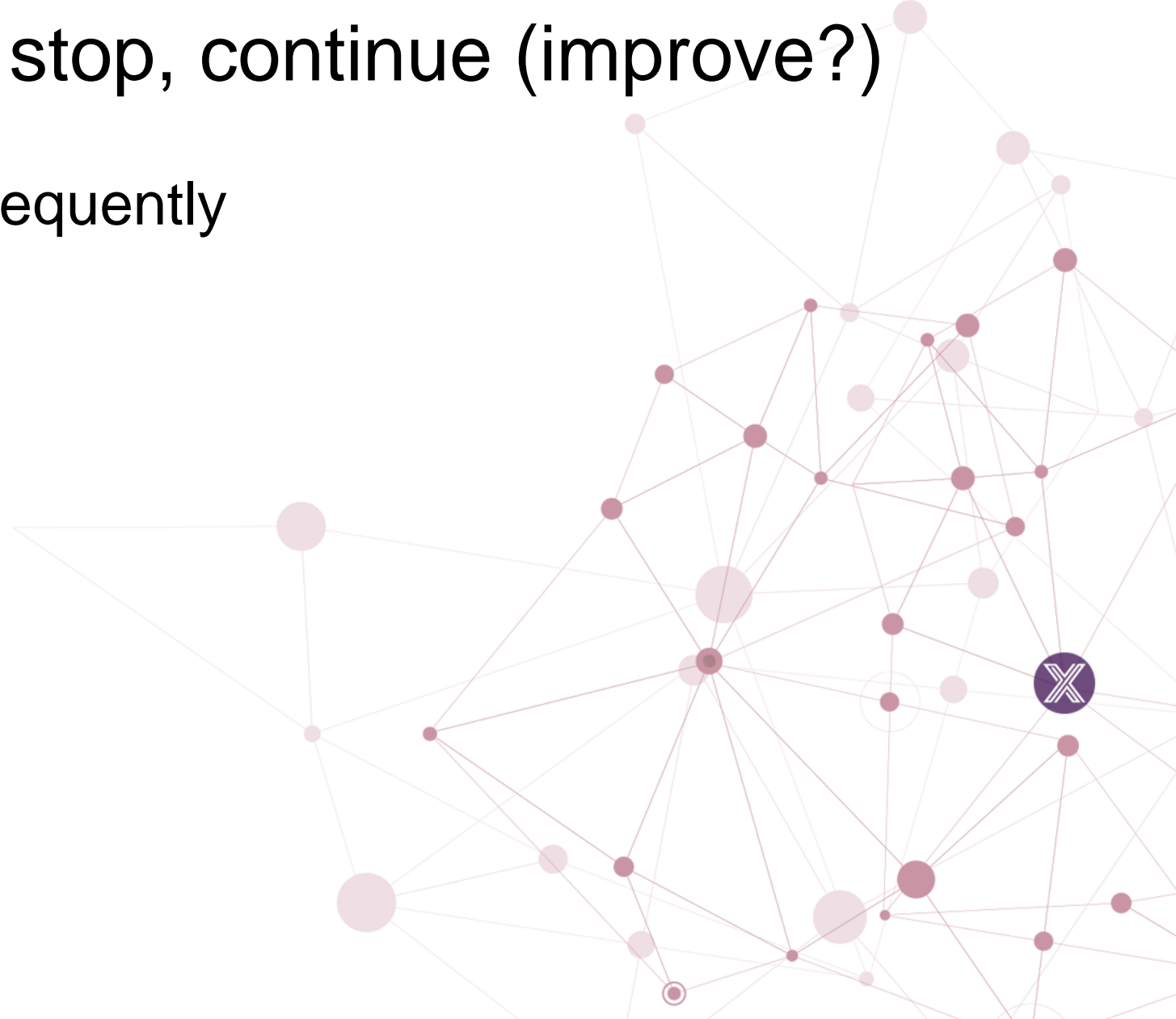
EdgeX Meeting Times

- Mondays
 - Device Services @ 8am PDT
 - Certification & Outreach/Marketing alternate @ 9am PDT
 - Architect's meeting once a month @ 10am PDT
- Tuesdays
 - Occasional Adopter series or tech talk meeting @ 8am PDT
 - Application Services WG @ 9am and 3:30PM PDT alternate weeks
- Wednesdays
 - TSC @ 8am PDT
 - Security @ 9am PDT
- Thursdays
 - Core @ 8am PDT
 - DevOps @ 9am PDT
- Friday
 - No meetings



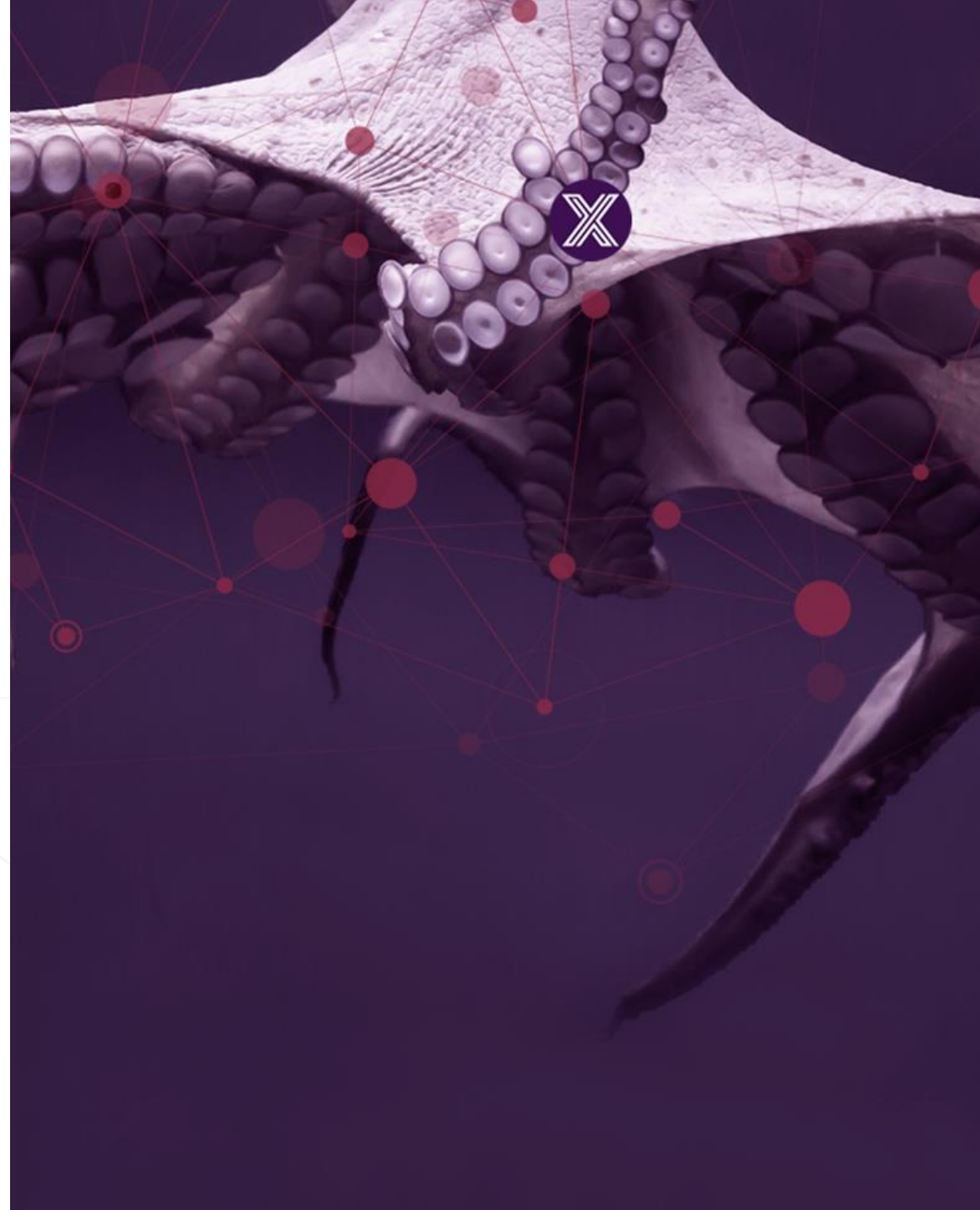
EdgeX Meetings Start, stop, continue (improve?)

- WG meetings – more/less frequently
- Monthly architect meetings
 - More/less frequent?



EDGE X FOUNDRY™

Process Improvements



Process Improvement Topics

- ADR process changes/improvements
- Real Hardware Testing
- 3rd party library support



ADR Process

- Is it working for us?
- Does it need improvements?
- [Farshid's recommendations](#):
 - Need to cover requirements better
 - Separate requirements and design
- Need to avoid getting into implementation details (how?)

Real Hardware Testing

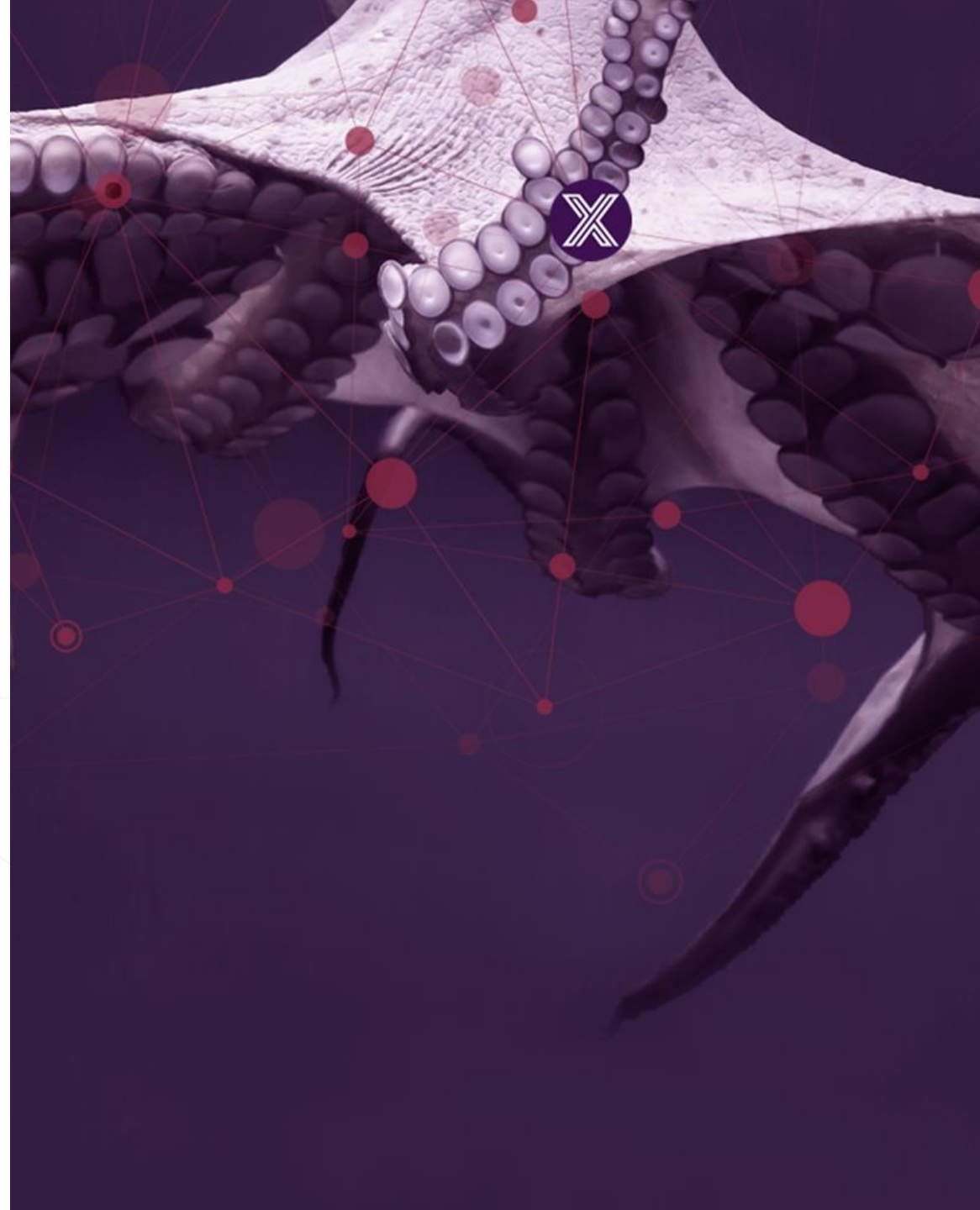
- How can EdgeX validate the platform with real hardware?
- How can the project get real hardware experiences/feedback?
- Can we establish a “framework” for outside testing with real hardware?

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
 - Try to provide assistance to these library projects
 - Do nothing – hope for the best; react to issues as they arise

EDGE X FOUNDRY™

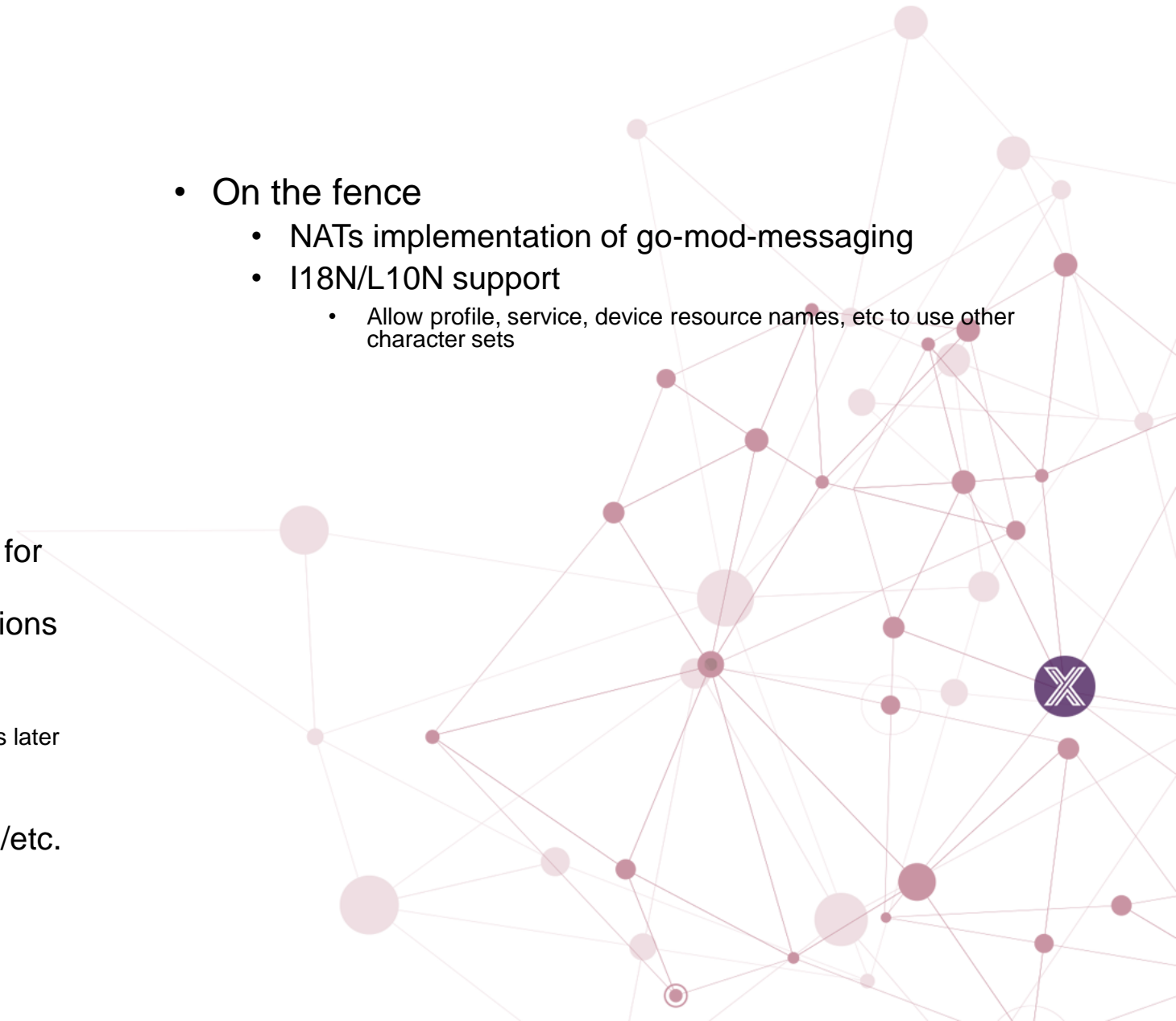
Architectural Discussions/Decisions



Architect's Topics list

- In scope
 - ADR process improvements (will be covered separately)
 - Additional Metrics collection
 - What's telemetry is collected outside of core data
 - Last Connected/Last Reported
 - Possibly some other technical debt in Profile/DS/Device area
 - GUI and calls to SMA
- Under Consideration
 - How can we reduce the build times (especially for ARM)
 - Securing the MQTT message bus communications
 - Mostly for internal
 - ADR for global configuration
 - An EdgeX 3.0 topic but an ADR is needed sooner versus later because this could be hard to solve
 - Bring your own Vault
 - ADR for URI reference to point to config/profile/etc.

- On the fence
 - NATs implementation of go-mod-messaging
 - I18N/L10N support
 - Allow profile, service, device resource names, etc to use other character sets



FYI – considered out of scope (per prewire)

- “EdgeX-lite” w/o security
- Explosion in the number or build artifacts and how we reduce it
 - Do we let adopters do more of regard to binaries, images, artifacts
- Global configuration (implementation)
 - Remove duplication across configuration
- EdgeX size concerns (Kong Consul)
- Dynamic Config
- Use of URIs to point to configuration profiles, etc (implementation)
- Various message bus changes (messaging connects, topic names, topic aliases)
- Getting rid of SMA (already deprecated)
- What replaces start/stop/continue, config, device metrics
- Running on Windows (more? MSI or other artifacts)
- ARM support
- Create tool or script to create a new device or application service (discussed during the planning meeting - Nov 21).
- Artifact signing
- Requirement of real hardware test environment



Scoping

- Categories used in our pre-wire
 - In scope
 - It is “In scope” – must be done; no debate; near unanimous consent
 - Only discussion is what will be completed in this release
 - Under consideration
 - It is not in scope yet, but it worthy of some time to discuss
 - A strong tendency to put it in scope; in scope by default if not reviewed or decided
 - 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 being discussed in this meeting

In Scope



Metrics Collection

- ADR - <https://github.com/edgexfoundry/edgex-docs/pull/268>
 - DONE!
- What telemetry do we collect?
 - Impact to size
 - Laundry list in the ADR
 - Revisit scope and what additions we make in Levski (if any)
- **Lenny** to lead the research/discussion

Last Connected & Last Reported

- Bugs discovered : ([1151](#), [1150](#)) these fields on device service and device have not been populated since ? (perhaps Barcelona but at least Edinburgh)
 - Obviously the empty fields need to be addressed
 - Requirement is still there
 - Options:
 - Just deprecate those fields (not recommended)
 - Use Levski release to fix these
 - Fix them in DS
 - Use alternate means to provide this information (example: another metric to report)
 - May require short term and long term fix based on backward compatibility
- If we are going into this section of the code, might there be some other tech debt we want to address?
 - - Command chaining
 - device-sdk-go #26 and device-sdk-c #3
 - The fields specifying these operations were dropped in the profile
 - rework for v2 so there's an implication there that these will be WONTFIX
 - - Configurable size limits in device service
 - MaxCmdValueLen is a configuration setting in the Go SDK that does nothing. It is labelled as "MaxCmdValueLen is the maximum string length of a command parameter or result (including the value descriptor name) that can be returned by a Driver."
 - In the C SDK this is called MaxCmdResultLen and it doesn't do anything either

Customer Reference	IOtech Work Package	Scope / IOtech Effort
Row 92: System Alerts Disconnection Cases	Analyze how Edge Xpert can provide users with disconnection information	<ul style="list-style-type: none"> • IOtech to analyze if/how existing Edge Xpert features can provide users with information relating to disconnects from physical devices and Cloud/IT systems • Analyze and verify different disconnection scenarios • Provide recommendations and design for adding further functionality as necessary • 5 man-days effort
	Total IOtech Effort:	5 man-days

GUI and calls to SMA

- Today, the GUI calls on the SMA for service metrics, configuration, etc.
 - SMA is deprecated (although won't be removed until EdgeX 3)
 - Do we want GUI to use a different means to get metrics/config/etc.?
 - Per <https://github.com/edgexfoundry/edgex-ui-go/issues/486>
- Calling on each service directly presents some challenges
 - Service name from the registry isn't directly mapped to metrics APIs
 - GUI doesn't call on all services today (one of the reasons for the SMA)
 - Addition of new service, GUI would have to be notified or do a refresh of service list periodically
 - GUI wouldn't know the proxy path when Kong is deployed/used (GUI is outside of Kong)
- Suggested
 - Service registration includes URL to get metrics and other SMA related data for each service

Under consideration

In scope by default



Build Time Issues

- With SPIFFE/SPIRE and other related items, builds are taking a long time
 - 6 minutes for Intel builds
 - Approaching an hour on ARM builds
- Future discussions about how to do other builds for other needs (with/without security, etc.) could make this worse in the future
- Are there things we can do or try to reduce build times
- **Bryon and Ernesto** leading this discussion

Securing the internal message bus – specific to MQTT

- Communications between services
- How to generate credentials and tell the services about these
- How to initialize the MQTT secrets for the internal message bus comms
 - This is just about initializing the broker and putting the credentials can get at the in Vault
 - Question: why is an alternative secure implementation for MQTT – we already have secure Redis Pub/Sub? Should we consider deprecating Redis Pub/Sub if it is not meeting needs?
 - MQTT does provide more “knobs” (QoS, durability, keep alive, etc.) for using MQTT vs Redis today
- **Lenny, Tony** to lead the discussion

Global Configuration

- <https://github.com/edgexfoundry/edgex-docs/issues/580>
- 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)
- Hard to solve – one of the downsides of microservices
- EdgeX 3.0 consideration – would likely be non-backward compatible
- Architecturally significant – calls for an ADR
- Implementation likely out of scope for this release
- Jim to lead the discussion

Bring your own Vault

- Allowing users to potentially run Vault in a cloud
- One customer did run-your-own; it is physically possible
 - Had to rip open secret store setup to make it happen (lot of work)
 - Bootstrapping is the complex part
 - This is really technical debt refactoring (not interfaced/abstracted enough right now)
- “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.” – Bryon N
- Bryon to lead this discussion

URI reference to point to config/profile/etc.

- Came up as part of UoM ADR discussion
- <https://github.com/orgs/edgexfoundry/projects/48#card-77954532>
- We are looking to use a URI to specify an external configuration file in the future
 - 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 in the future.
 - 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).
- May need EdgeX 3.0 to implement (backward compatibility)
 - It could be done in a backward compatible way
 - Need requirements/use case better documented
- Requirements/ADR needed
- Implementation probably out of scope for this release
- Jim to lead this discussion

On the fence

Out of scope by default



NATs (back again?)

- Add NATS implementation for MessageClient interface
- <https://github.com/edgexfoundry/go-mod-messaging/issues/37>
- <https://github.com/edgexfoundry/go-mod-messaging/pull/137>
- Size concerns – add size to the binaries
 - Conditional compilation “trick” may be a way address this (this is EdgeX Lite concept)
- Rationale
 - 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 settings
- Requested by at least one adopter and discussed in the Kamakura planning meeting (others by Slack more recently)
- NATS JetStream?? How does this apply/differ from NATS at large?
- You are going to want to secure it – what does that look like?
- Alex to provide pros/cons and use case for NATS prior to the planning meeting
- Presenter of this discussion to be determined and based on Alex feedback

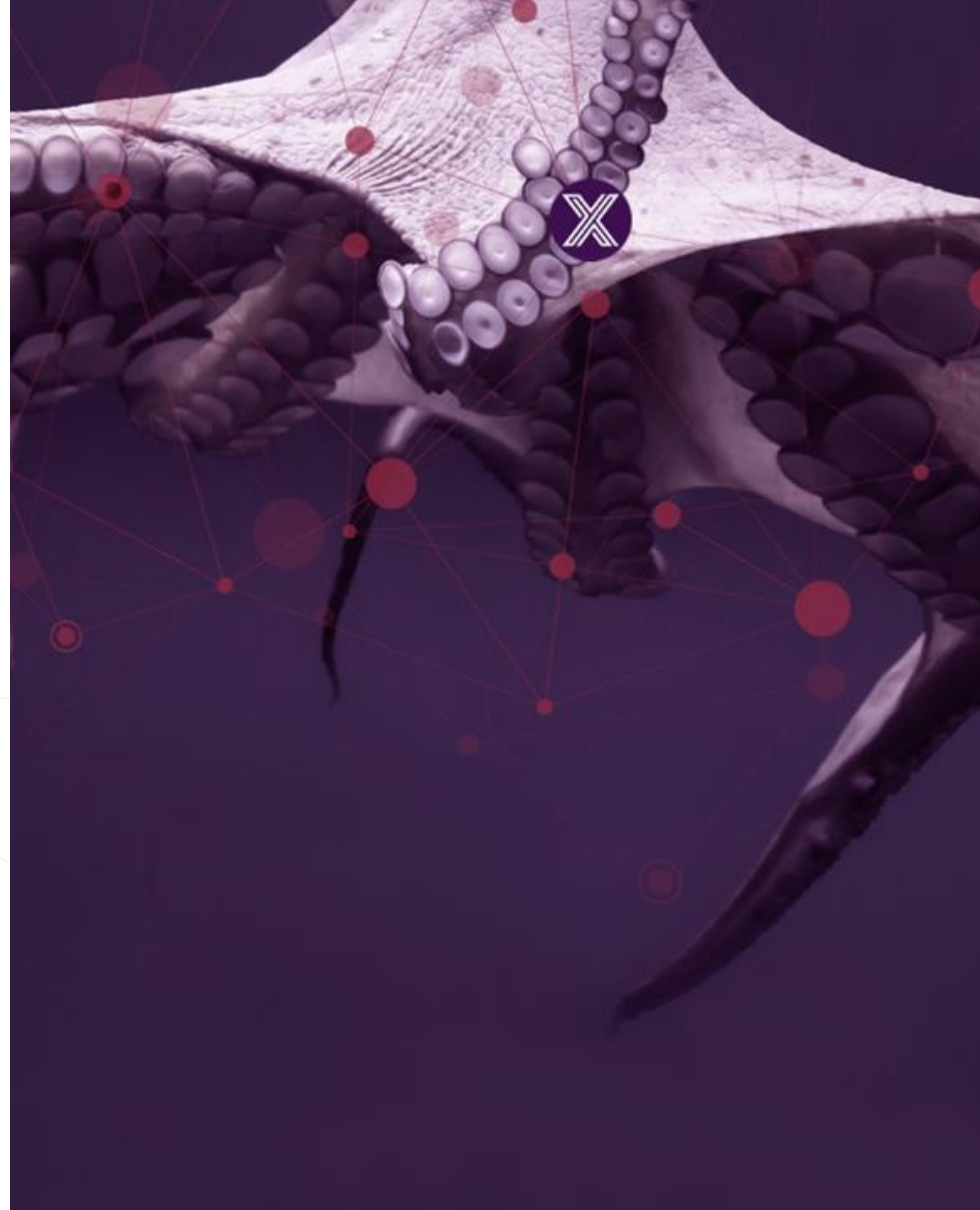
I18n/L10n metadata

- <https://github.com/edgexfoundry/edgex-go/issues/3935>
- An adopter has requested that things like device names be allowed to contain other language characters. Specifically, in this instance, Vietnamese characters. Presumably, much more of EdgeX could be internationalized or localized.
- May be applicable to all services working with user defined JSON or YAML (like device profiles).
- Validation of “names” and such would have to be updated to allow for I18n/L10n
 - Think this might be in a single place
- How would this work with URLs, topic names, etc.?
- Input from China – how about a name to character of choice translation kept somewhere that can be used for UX purposes?
- Jim to lead the discussion

EDGE X FOUNDRY™

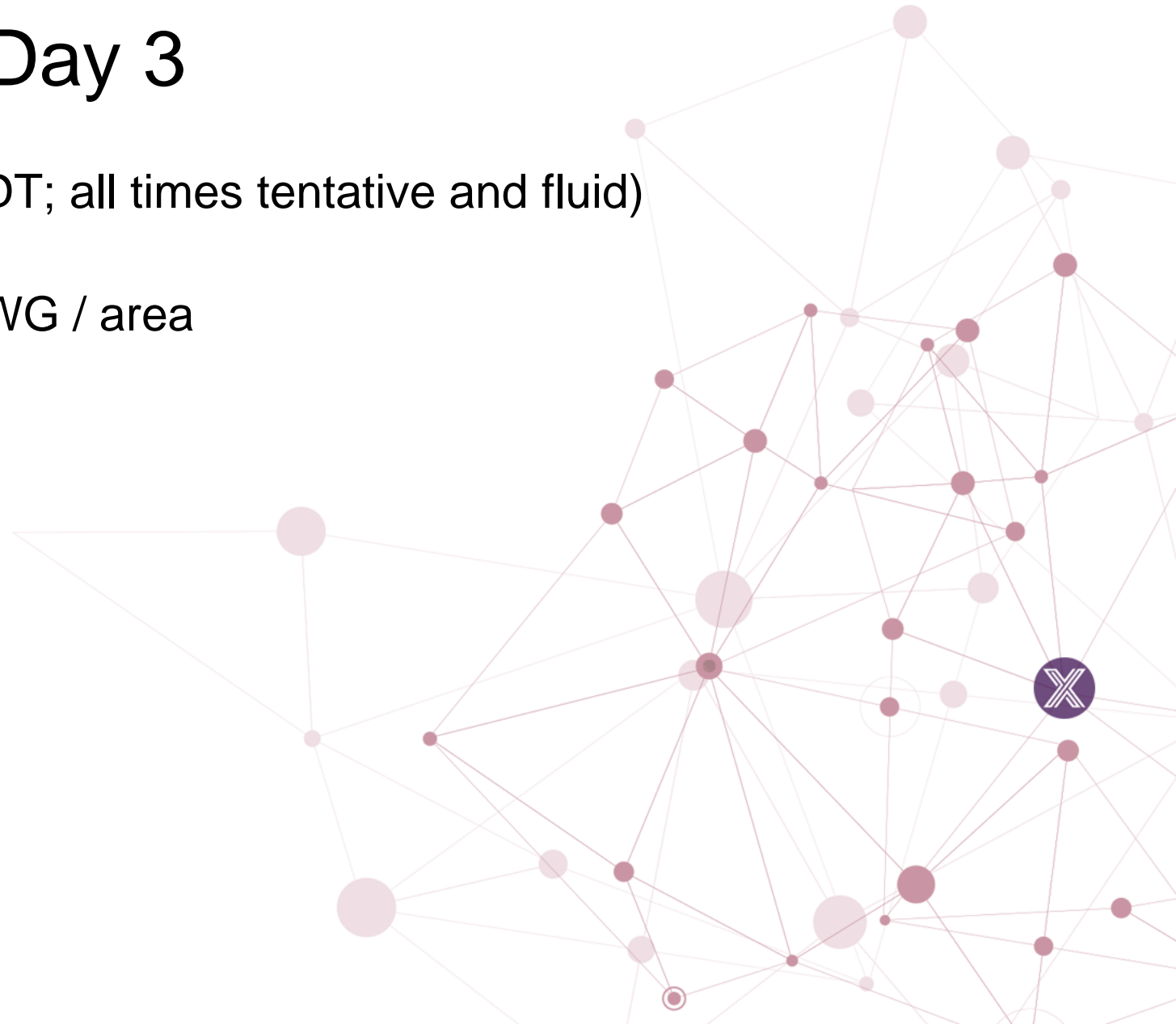
Day 3

May 18



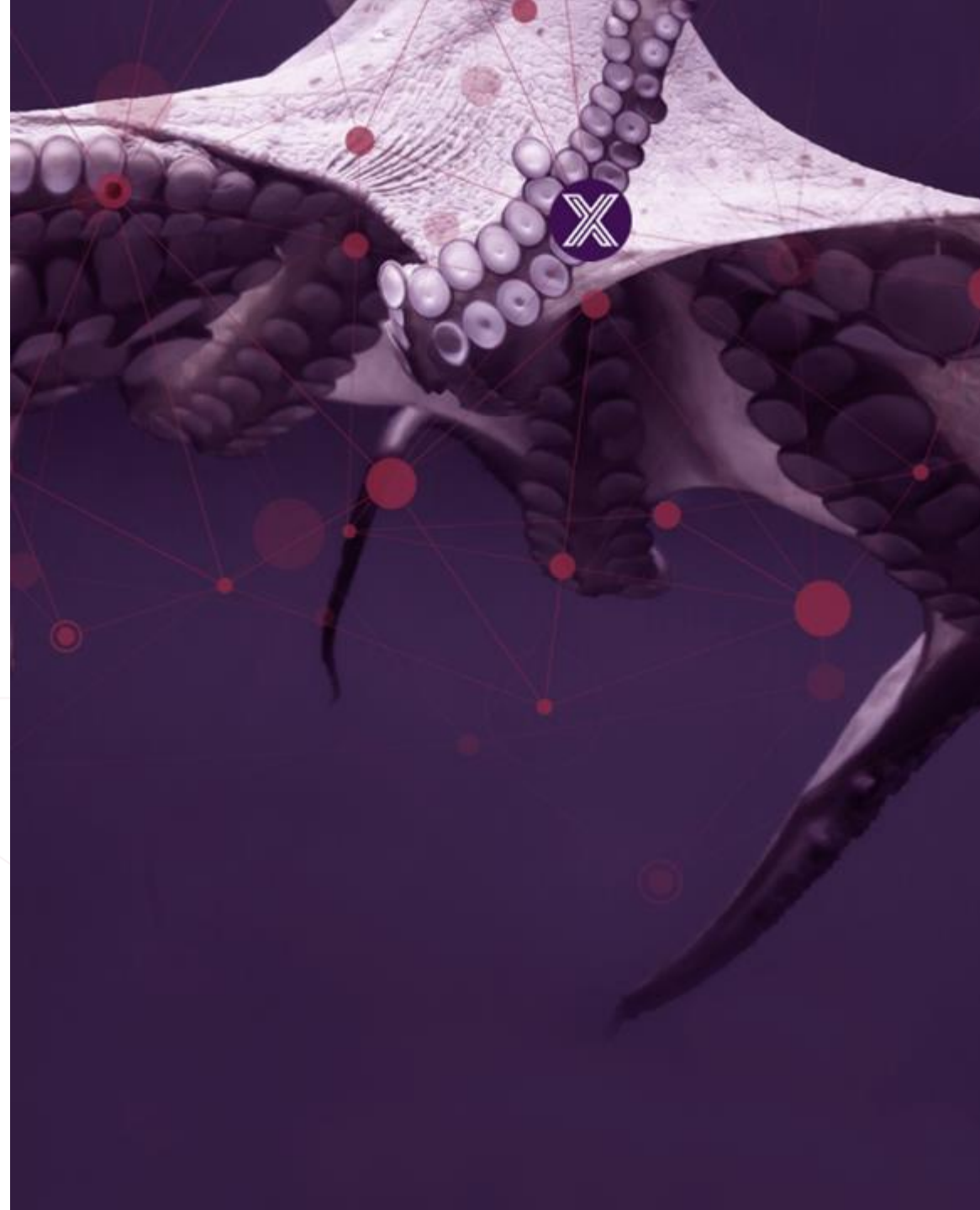
Conference Agenda – Day 3

- May 18, Wednesday (all times in PDT; all times tentative and fluid)
- 7:00am PDT
- 7:15am Levski scope planning by WG / area
 - General
 - Common
 - Cross cutting concerns
 - Core
 - GUI
 - CLI
 - System management
 - Test QA
 - Device Service
 - Application
 - eKuiper / rules engine
 - Security
 - DevOps
- 10am day 3 adjourn



EDGE X FOUNDRY™

Levski Scope



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

Caveats

- 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 architectural discussion earlier.**
 - These need to be added and scoped as well

Scoping

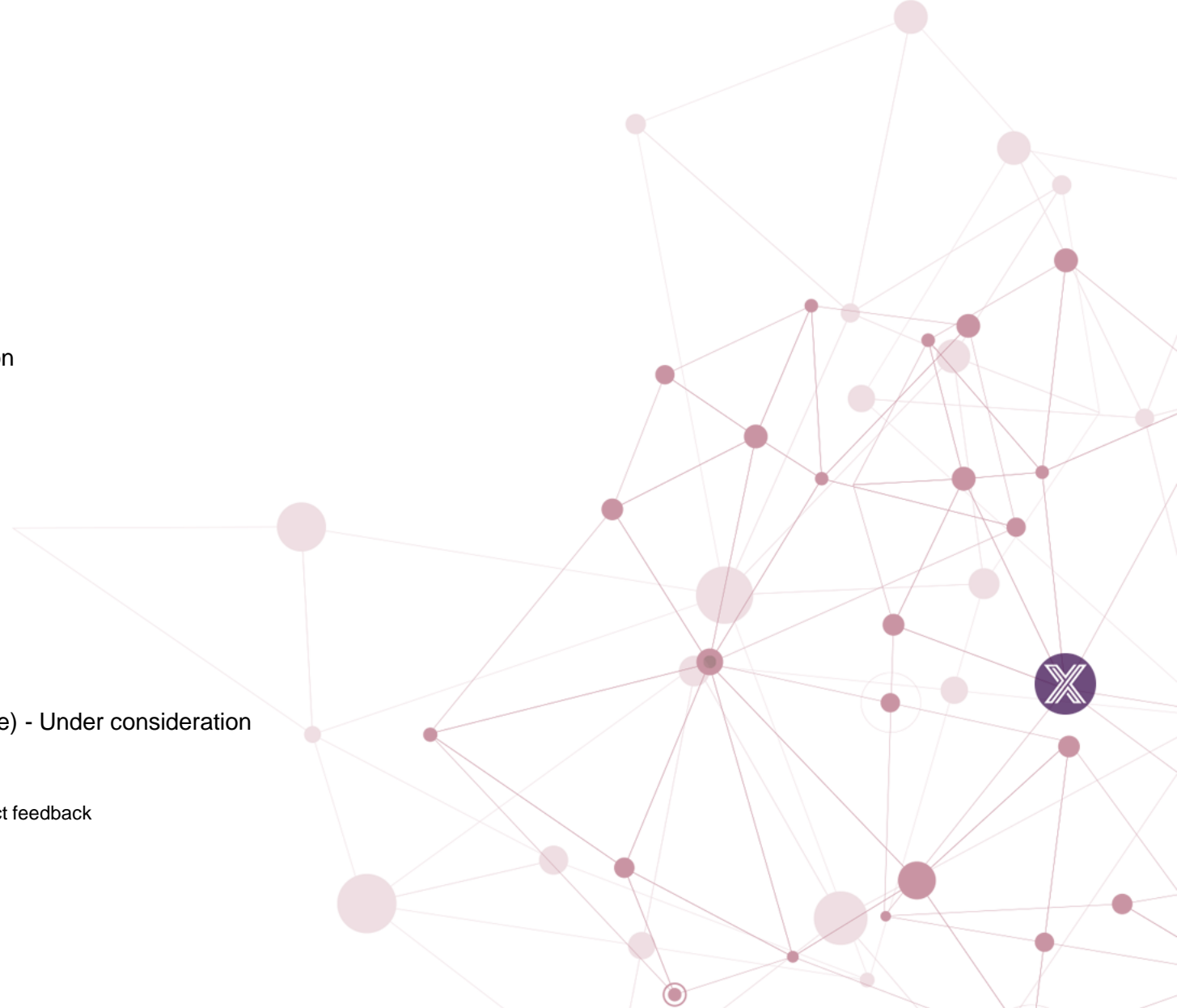
- Categories used in our pre-wire
 - **In scope**
 - It is “In scope” – must be done; no debate; near unanimous consent
 - Only discussion is what will be completed in this release
 - **Under consideration**
 - It is not in scope yet, but it worthy of some time to discuss
 - A strong tendency to put it in scope; in scope by default if not reviewed or decided
 - 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 being discussed in this meeting

General

- Upgrade Go (1.18, 1.19?) In scope
 - Generics important
- UoM implementation - In scope
- N-S message bus implementation - In scope
- Cloud templates need updating
 - Do we even keep these – they are way out date
 - EdgeX Ready impact?
 - Under consideration to deprecate

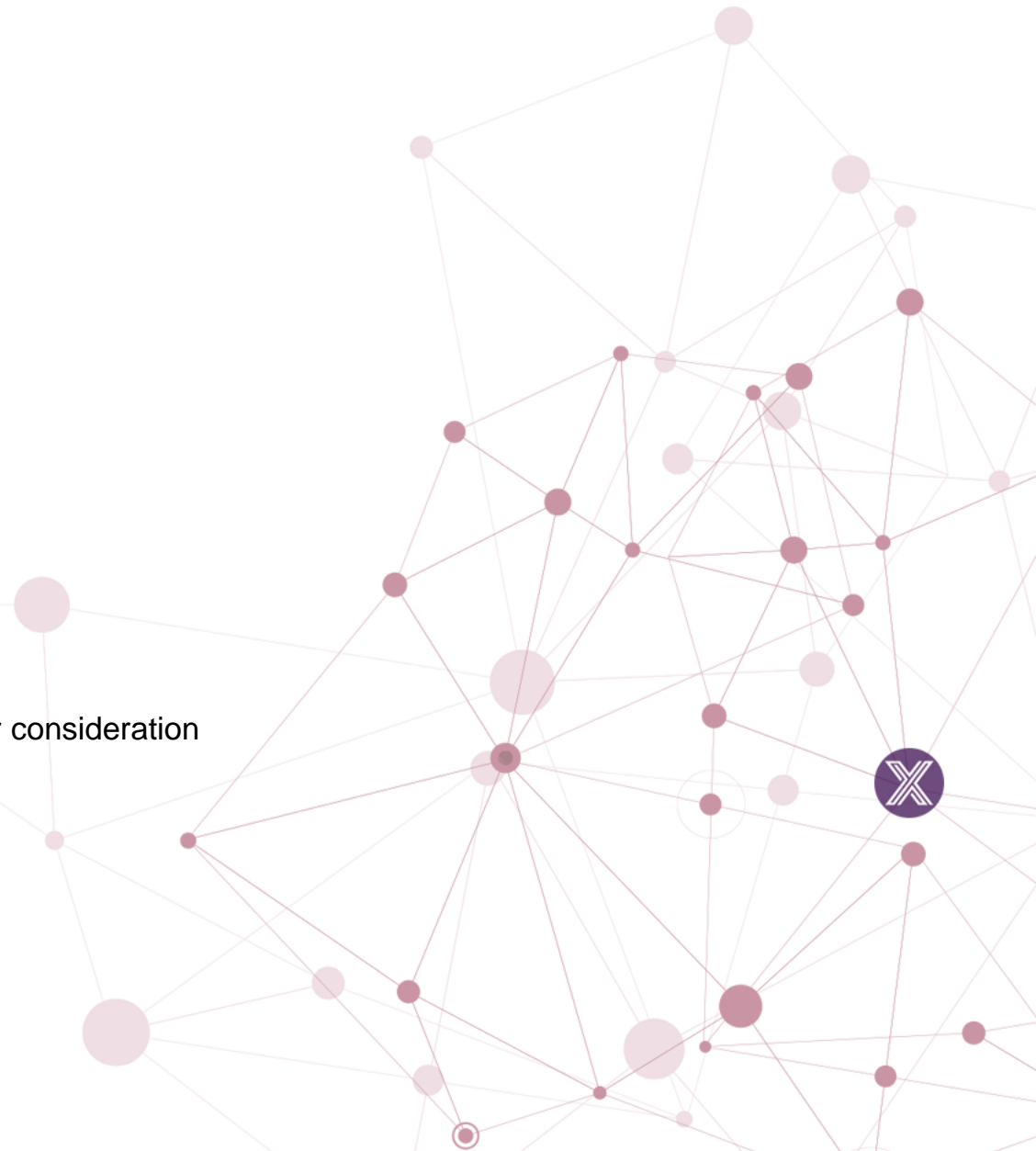
Core

- Core
 - Prometheus integration
 - Could do with app service configurable
 - Have an example for Influx – could do an example for Prometheus
 - Not sure/On the fence
 - CPE ADR and implementation - In scope
 - Core Data Cache ADR and implementation - Under consideration
- K8s support
 - Bryon has submitted issues for how to refine
 - Review Bryon's issues and figure out what's next – In scope
- CLI
 - No resource availability at this time
- GUI
 - Add unit tests to CI/CD
 - Some bug fixes/cleanup
- Test QA
 - Always CBOR option for performance (never got to test this cycle) - Under consideration
 - Nice to have tests for delayed start - In scope
 - Framework for outside testing with real hardware
 - How organizations can validate with real hardware and provide project feedback
 - Under consideration–process improvement



Device Services

- DS upgrage/approvals (CoAP, GPIO, LLRP, UART, Grove, ONVIF, USB) In scope
 - With exception of UART which is stalled – should we deprecate this work?
- BACNet DS - In scope
- DS Filtering implementation (ADR already in place)
 - Not sure/On the fence – resource availability concerns
- C SDK library – Eaton input otherwise Out of scope
 - Produce a release artifact vs clone the repo and start a service
- cmake options to optimize dependencies – Eaton input otherwise Out of scope
 - Eaton proposed ADR
- Publish multiple readings in 1 msg in MQTT DS #[299](#) – Under consideration
 - 2 end user requests for this feature
 - Can this be done in a backward compatible way??
- Handle both JSON request bodies as well as CBOR request bodies #[488](#) – Under consideration
 - Need binary PUT for consistency
 - Hasn't high priority in the past
 - In C but not Go
- Metrics / Telemetry collection in C services (per ADR) – ???
- Record and Replay – under consideration; see next slide



Record and Replay

- <https://github.com/edgexfoundry/edgex-docs/issues/471>
- <https://wiki.edgexfoundry.org/display/FA/Record+and+Replay>
- Application Service takes Events and logs them to a file. (Maybe a single service to both this and DS work)
- Device Service reads a file generated by that Application Service, and generates new Events based it.
 - Event generation will occur either at fixed intervals defined in the service configuration, or based on the timestamps of the Events which are read in.
 - New Events should have their origin field re-created; the created field should not be set by the device service.
 - Enhancement: Replay the contents of multiple files simultaneously
- Impacts EdgeX Ready
 - We would need a How-to-Guide and possibly update the submission materials to include the “recorded” sample data.
- Is anyone asking for this – not within recent memory but there have been requests
- Could be useful for debugging/testing; could be beneficial beyond EdgeX Ready
- Need of ADR and/or additional requirements?
- A real nice to have
- Not really architectural decision “significant”; could it be handled by a single team (AS or DS)?

Application Services

- NO items in scope or under consideration at this time



DevOps

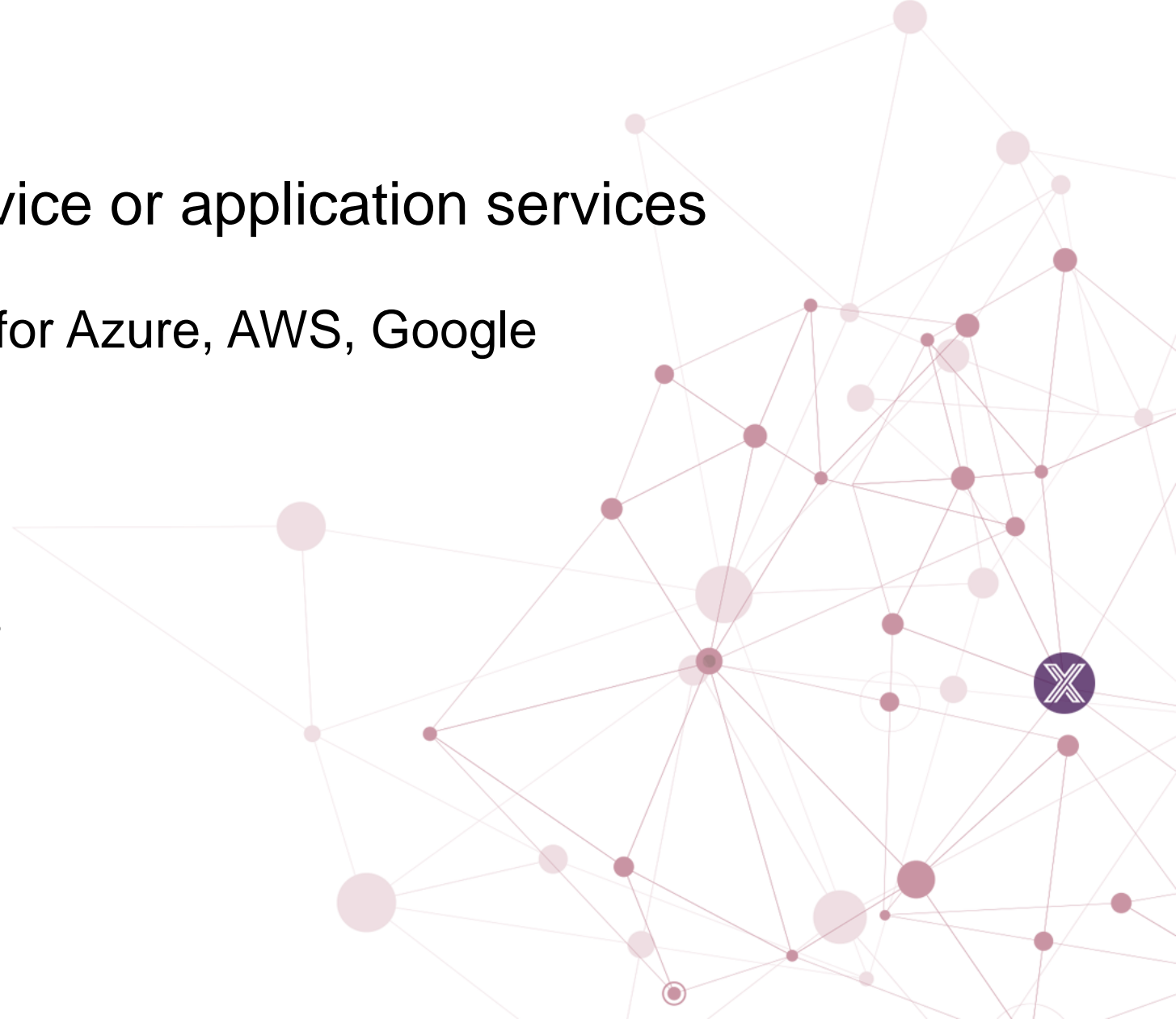
- Credly API integration if needed (carry over from Kamakura)
- htmlproofer checks on each edgex-docs PR
- Automate version bump in swagger api files on release, i.e. https://github.com/edgexfoundry/edgex-go/blob/main/openapi/v2/*.yaml (Ask from Lenny)
- Assist GUI team automate testing

Snaps

- Upgrade snaps base to core22 (built from Ubuntu 22.04 LTS)
- Guide on headless deployment of EdgeX on Ubuntu Core
- Add support for SPIFFE/SPIRE
- Standalone device-virtual snap

Miscellaneous

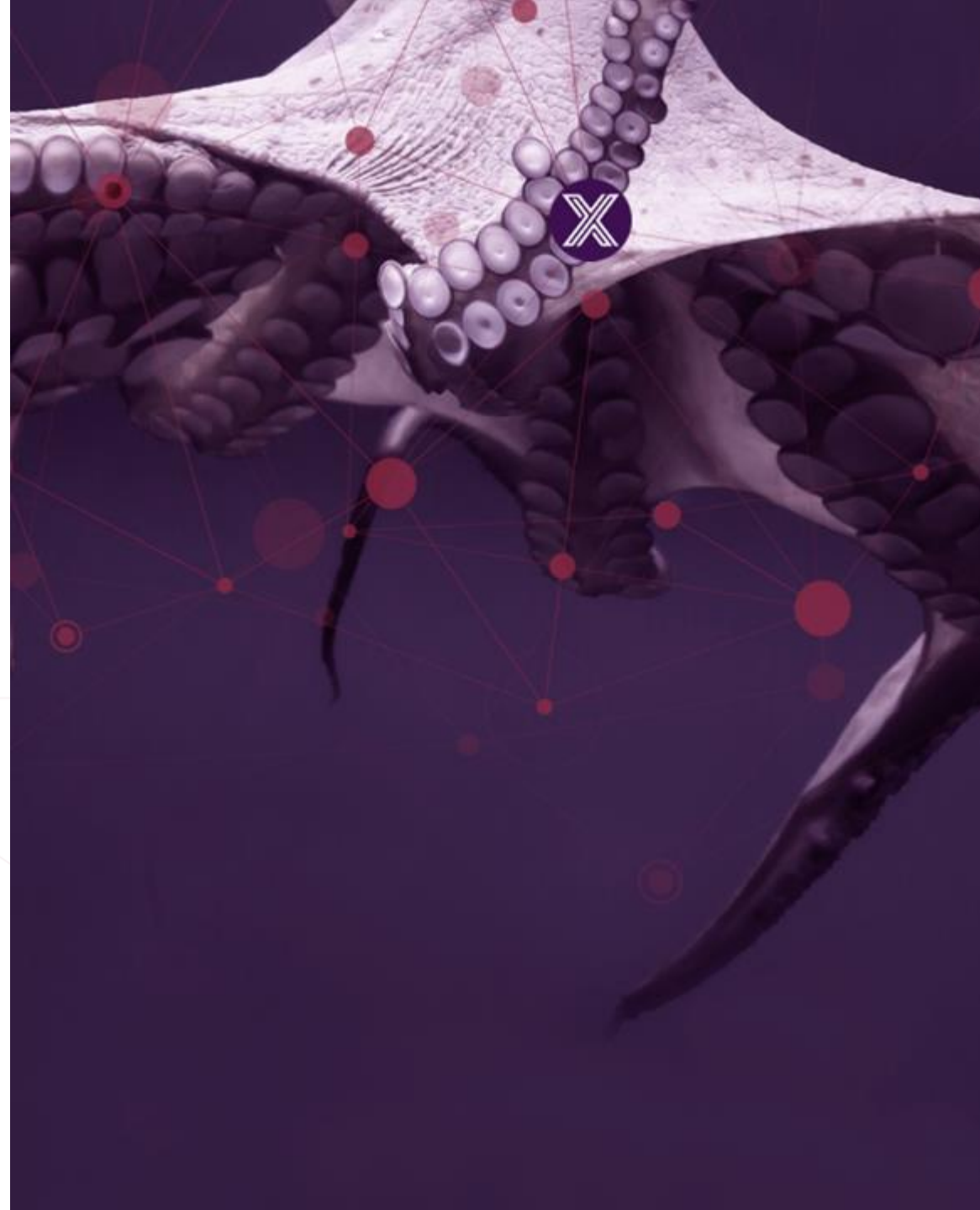
- Tool/script to create new device or application services
- Examples
 - Cloud export – more explicit for Azure, AWS, Google
 - Update cloud templates
 - RP4
 - Any new example requests
- Docs
 - Any new sections or features



EDGE X FOUNDRY™

Day 4

May 19

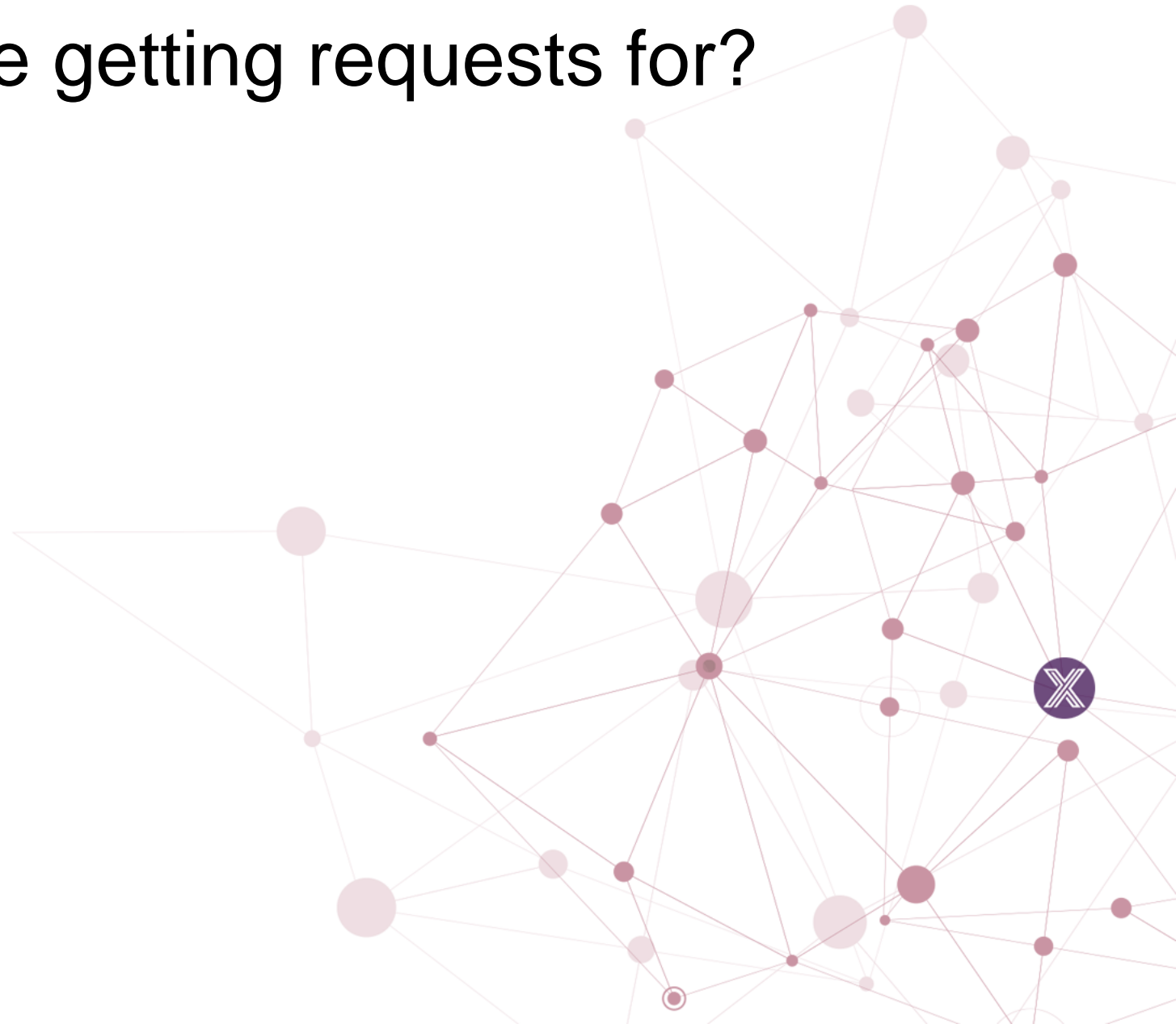


Conference Agenda – Day 4

- May 19, Thursday (all times in PDT; all times tentative and fluid)
- 9am PDT Meeting Start
- Scope and architectural questions not addressed in prior 3 days will be finished as priority work on day 4 as needed.
- 10am Requests of adopters
- 10:15am Kamakura lessons learned
- 10:30am Fall release - in person face to face??
 - Venue/time line
- 10:45am Long term roadmap - beyond Levski
 - Backlog revisit
 - EdgeX 3.0 and next LTS forecast
- 11:30pm Planning meeting lessons learned and changes for Minnesota
- 12pm day 4 adjourn

On Slack – what are we getting requests for?

- To be filled in



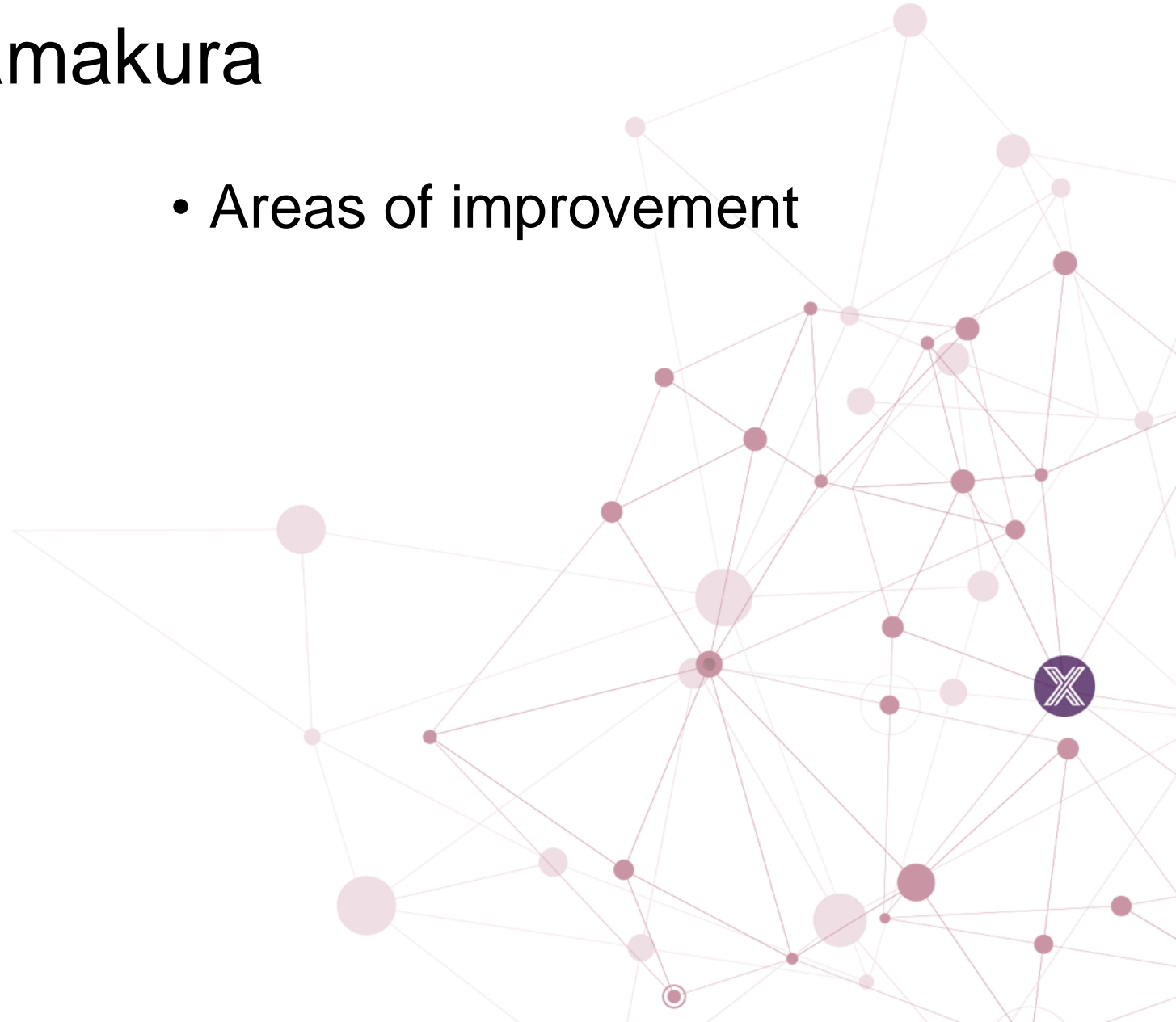
Jakarta – lessons learned we said we would try to fix

- Positives
 - Documentation done in parallel (mostly)
 - Docs done by release
 - DevOps/Release
 - Testing really good now and stable
 - Added a lot (more coverage, performance tests, etc.)
 - Smoke tests!
 - No code without tests
 - Fingers crossed – we have an LTS!
 - More adopters and more input from adopters
 - Streamlined PR templates/policy around when issues are required
- What could be improved
 - Takes a long time to make a decision
 - Architectural decisions, review process, collecting information about the real issue or what people need take a long time
 - More planning / describing a change we wanted
 - Object type as example
 - Need for hands for the process
 - ADRs may need to be brought back in to WG discussions

Lessons Learned in Kamakura

- Positives

- Areas of improvement



Fall (Minnesota) Release Planning

- Time for a face-to-face?
 - Organizational support
 - Venue suggestions/volunteers to host
 - Timing
 - November after the release (Thanksgiving time is close by)?
 - With potential early November release dates

Long-term Roadmap

- Backlog
 - <https://wiki.edgexfoundry.org/display/FA/Backlog>
- What would we see as likely big features in Minnesota?
- Open discussion about next EdgeX 3.0 and LTS
- Long term considerations and planning

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?
- If we shift to F2F for the next meeting, what would we do different?