EDGE X FOUNDRY™

Edinburgh TSC F2F
Working Agenda and Deck

Edinburgh, UK

Oct 23-25, 2018

WORK IN PROGRESS

Please connect with Jim W, Brett P, or Keith S for additions, re-action, commentary to this deck

EDGE X FOUNDRY™

EdgeX Chairperson's Day

Meetings by WG Chair to discuss particular issues/designs

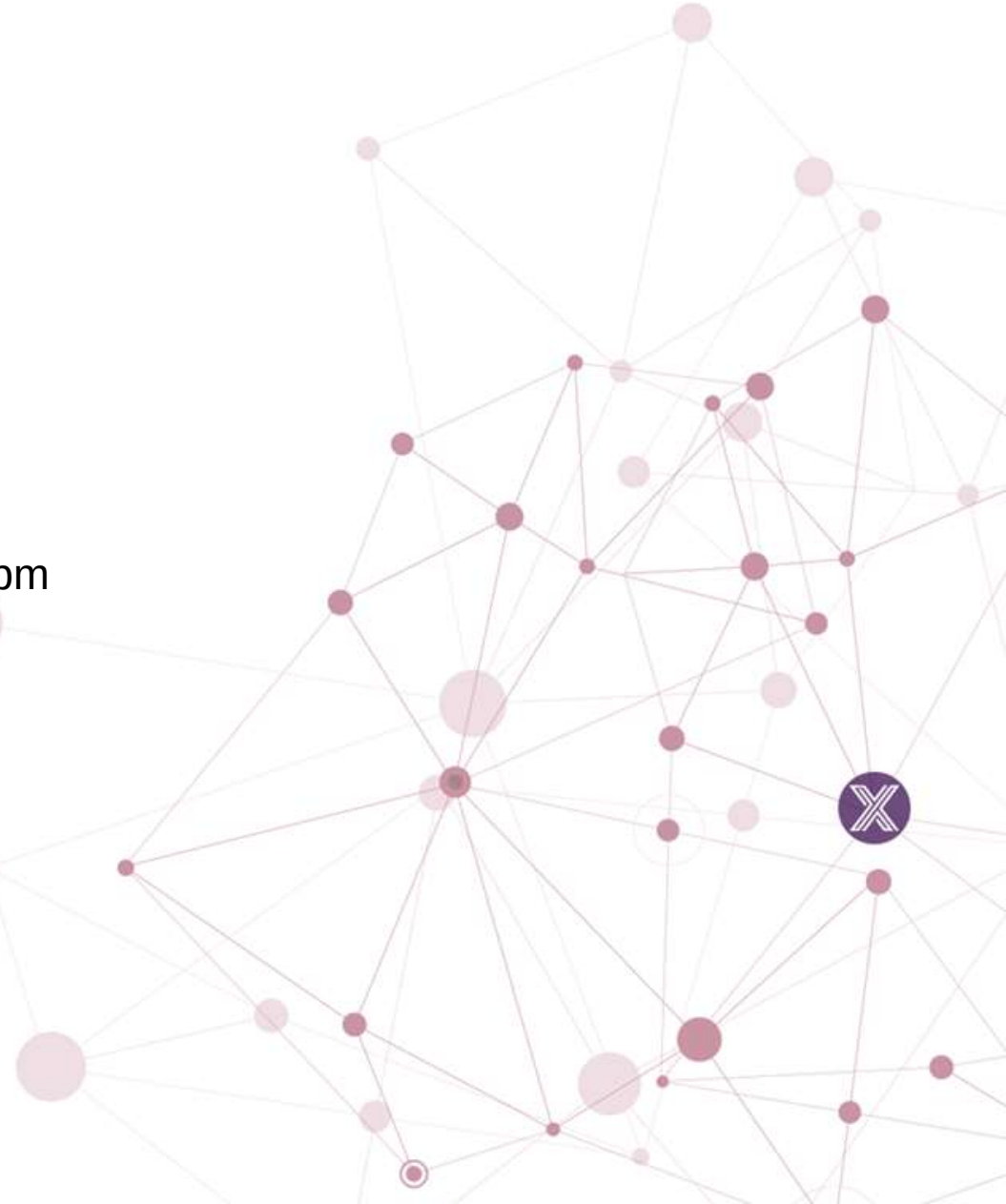
Edinburgh Work Group Meetings (pre-F2F) – Oct 23

- Device Service/SDK
  - Device onboarding; DS callbacks; adding device through metadata and next iteration of SDK
- System Management WG
  - What is EdgeX system management versus gateway management
  - Scoping future system management capabilities
- Security WG
  - HW based security discussion
- Test/QA
  - Performance testing strategy
- Application Services Design
  - The new export services; more scalable; more configurable & loosely coupled

EdgeX TSC 2-day F2F

Edinburg F2F Meeting

- Agenda
  - Day 1 – Edinburg planning day
    - Welcome and intro by Keith Steele (IoTech): 9-9:30am
    - Architecture day tee-up: 9:30-11am
      - Review and explanation of upcoming items
    - Edinburgh Planning – what's in/out: 11-2:30pm
      - Scope definition
    - Future Release Roadmaps - Fuji and beyond: 2:30-3:00pm
      - Long range scoping and roadmap review
    - Developer Advocate Perspective: 3:00-3:30pm
      - Better onboarding
    - DevOps Changes – 4-5pm
      - Release management, developer involvement, etc.
  - Day 2 – Architecture issues day
    - TBD architecture discussion and decisions: 9am-2:30pm
    - Action items for TSC Face-to-Face: 2:30-3pm
    - Business Issues/Discussion: 3-5pm



Architecture Issues Tee-up – Technical Debt

Edinburgh

- Database replacement/options
- Device services and SDK in mono repo
- Export Services (soon Application Services) – state of rework

Fuji or later

- Windows developer support (0MQ issue)
- ARM 32 support (are we directly supporting or not??)
- Support for device hierarchy in metadata (and elsewhere)
- API Documentation
  - Automate generation of API documentation (RAML)
  - Replacement of RAML or alternate to RAML (Swagger)

Architecture Issues Tee-up - Enhancements

Edinburgh

- Move to Go 1.11
- Modules / vgo
- Tracing of request through all services to allow better debugging and support
- Performance testing (who, what, how)
- Automated security testing
- HW root of trust (HW storage abstraction) in Security
- Improved resiliency
  - What do we need to do next? Services now are more resilient to timing issues
- Support for distribution
  - What do we need to do to better support truly distributed EdgeX

Fuji or later

- Config changes and callback/watcher
- Alternate deployment / orchestration (ex: adding Kubernetes support)
- How to supply command information to the north side systems (ex: how to give Azure IoT the ability to command devices)?
- Code signing
  - Exe/JAR/etc. artifacts
  - Docker containers
- Downsampling @ device service level
- Min/Max values (or other checks) on command parameters
- Message bus intercommunications between more micro services.
- Configuration versioning

Cadence Check

- April & Oct remain target release months
  - Edinburgh – April 2019
  - Fuji – Oct 2019
  - Geneva – April 2020
- F2F planning around time of completion of each release
  - Korea – April 2019
  - ??? – Oct 2019
- Conferences
  - At least 2 x large marketing/promotional events (Hannover Messe, IoT SWC)
  - At least 1 x developer focused event
    - Internet of Things World – May 13-16 (Dell Tech sponsoring)

Edinburgh Scope

- Major themes
  - Certification program
  - Export Services -> Application Services
  - MongoDB alternative (Redis) with database abstraction
  - Device Services galore
- Other efforts
  - Performance testing
  - HW Root of Trust

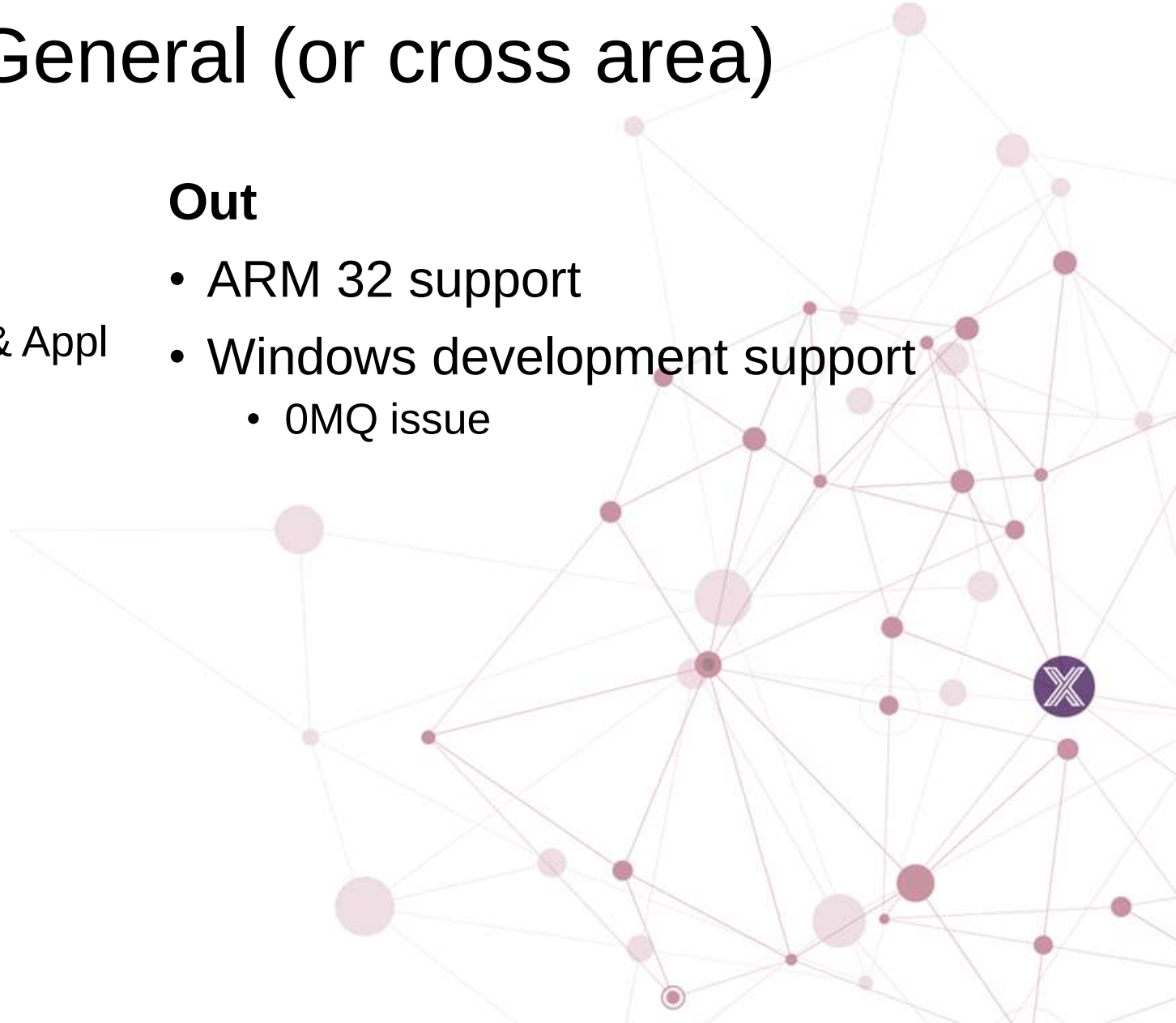
Edinburgh Planning – General (or cross area)

In

- Support binary data with CBOR
  - DS -> Core Data -> Export Distro (& Appl Services)
- Use of GoKit for all Go Services
- Versioning with modules/vgo
- Tracing

Out

- ARM 32 support
- Windows development support
  - 0MQ issue



Edinburgh Planning – Export (Application WG)

In

- Export Services replaced with Application Services

Out

- Additional Northbound endpoints - Fuji
  - AWS
  - IBM Watson
  - IoTivity
  - DDS
  - AMQP
- Support additional northbound formats
  - Haystack
- Integrate to edge software/agents
  - AWS Greengrass
  - Microsoft IoT Edge
- Rules Engine replacement - Fuji

Edinburgh Planning – Core (and Supporting)

In

- Better database abstraction architecture
  - Core data, Metadata, Exports, Notifications, Logging
  - Removing the BSON
  - Hiding domain IDs
  - Replace driver
  - Implementation of Core Services Using Redis
  - Certification/marketplace for alternatives
- Support for alternate logging format
  - XML & CSV in addition to JSON

Out

- Watchers/callbacks for config or data changes - Fuji
  - Config watcher already in place
  - Service registration/action needs to be implemented as needed
  - A more universal approach to metadata changes is needed
- Scheduling service rework
- Logging service rework

Edinburgh Planning – DS & SDK

In

- Devices added via 3rd party
  - Calls to Metadata vs calls to DS
  - How to handle
  - How to involve sys management
- Tutorials and How-to-guides
  - Device Service SDK Tutorials
- Demo/Virtual Device Services
  - Simple and Complex device-virtual
- DS for:
  - Modbus
  - BACnet
  - BLE
  - MQTT
  - SNMP
- Improvements to the SDK??
  - What did we not implement for Edinburgh that we thing we really need?

Out

- Additional Device Services - Fuji
  - Profinet/Profibus
  - CANBus
  - LORA
  - IoTivity
  - Zigbee
  - Zwave
- Support for mesh network protocols
- Tooling for SDK (CLI, JetBrains, or Eclipse plugins, etc.)
- Downsampling – DS throttle back on readings if nothing is changing

Edinburgh Planning – System Management

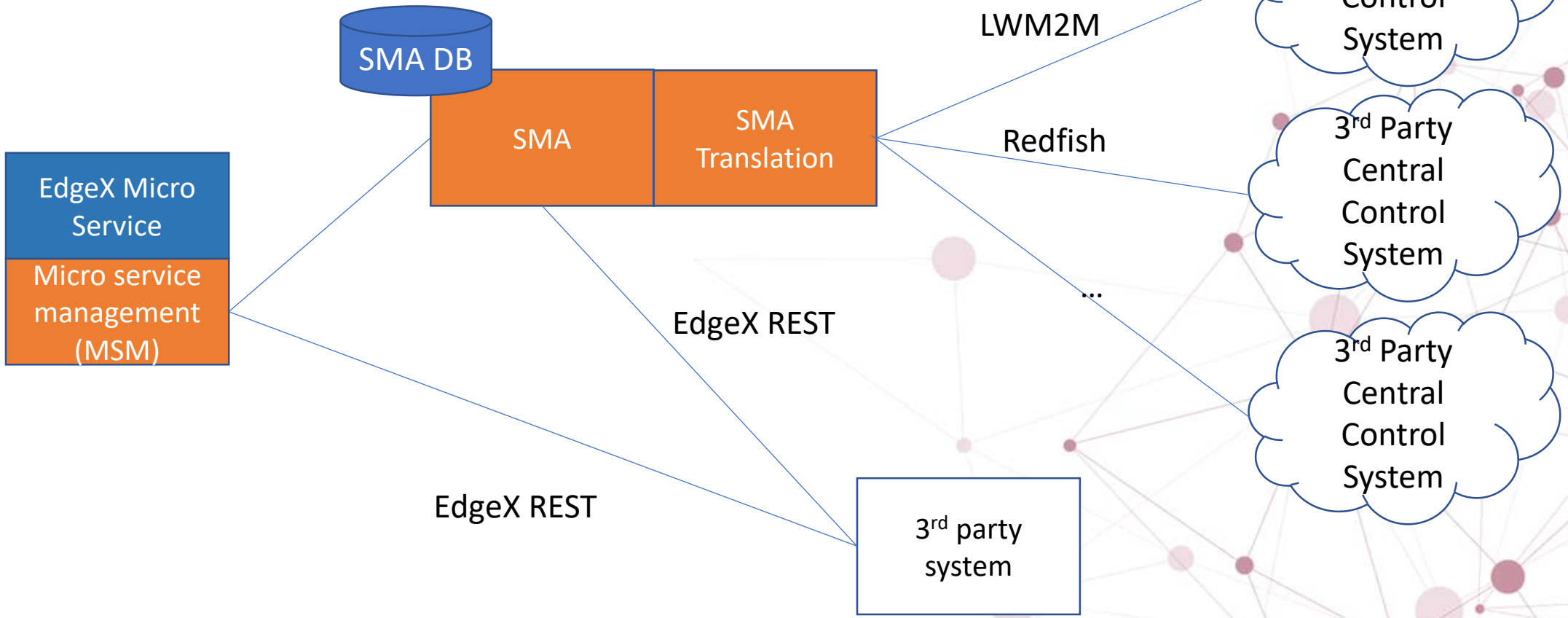
In

- Add Metrics
  - # of objects detected
  - Inferences per second
  - I/O
- Storing metrics collected locally
- SMA Translation layer (stretch)
  - Pick one protocol to start (LWM2M)

Out

- Setting configuration - Fuji
- Callbacks (alert on changes to config/metric) - Fuji
- SMA translations to other protocols
  - Redfish
  - OMADM
- Actuation based on metric change - Fuji
  - “rules engine” for control plane data
- Software updates/deployments

Management - Edinburgh



Edinburgh Planning - Security

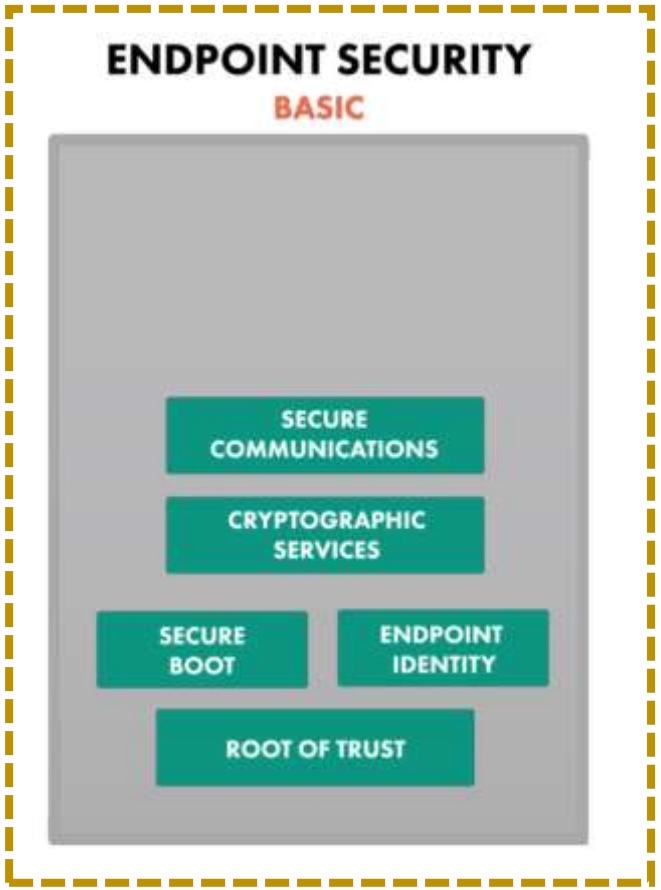
In

- HW Root of Trust
  - Storage Abstraction Layer
- Service-Service comms via Kong
- Security testing
  - Automated

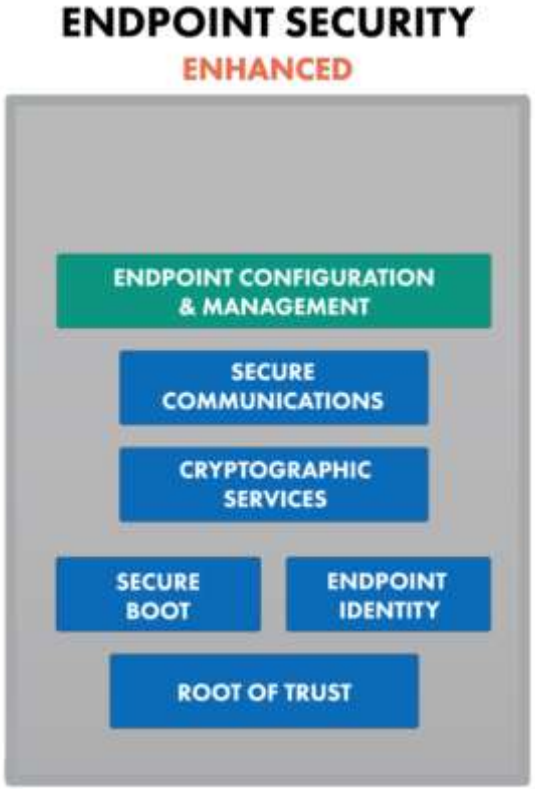
Out

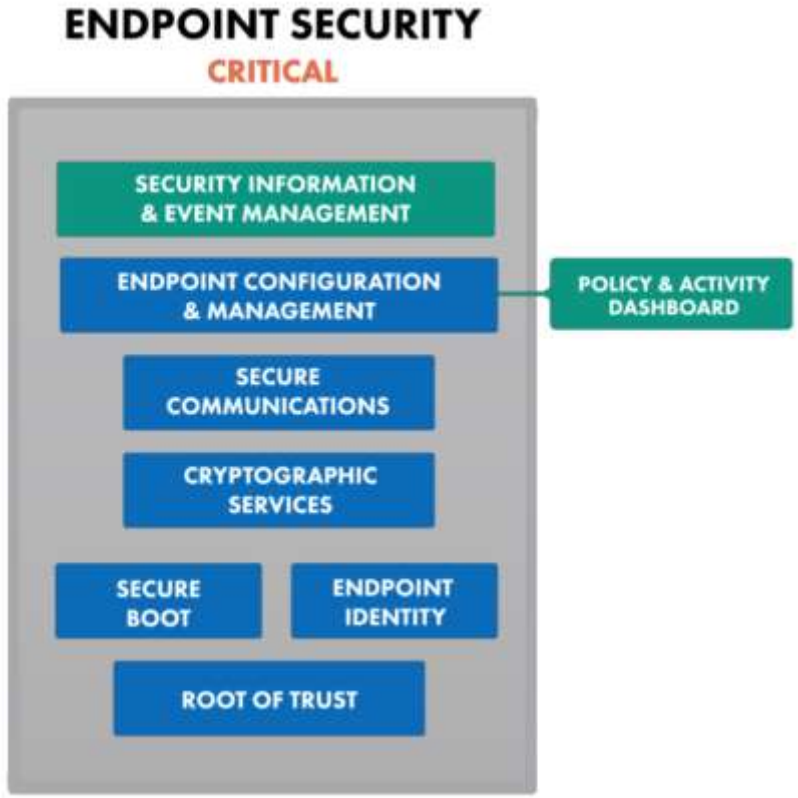
- Code signing
- How to securely provision new devices/sensor
- Hyperledger/blockchain/digital ledger integration
- Protect data at rest
  - In DB like Mongo
  - In log files
- Privacy concerns (HIP-A, GDRP, …)
- Renew/refresh threat model

IIC Endpoint Security Best Practices and EdgeX



EdgeX will begin here





Protect Perimeter Ingress: Details and Roadmap

Feature	California	Delhi	Edinburgh	Beyond
API Gateway	Single Ingress Point for ALL HTTPS traffic (no HTTP) using Kong	X	TBD	TBD
Authentication	Simple JWT based authentication (via kong plugin)	Oauth based AuthN (Client Credentials, Bearer Token Flow)	X	Identity Management Features (User Lifecycle Management, password change, revoke)
Authorization	None	Via Kong ACL plugin that enables group based AuthZ	TBD	TBD
TLS	Server Side Only Primary Cert stored in Vault	X	Mutual Certificates	TBD
Service to Service	None	None	Enabled via one of (mutual certs or Token based AuthN)	Secure service registration (Considering Consul Connect)

IIC Endpoint Security Best Practices Reference: Secure Communications

Secrets/Key Management

Feature	California	Delhi	Edinburgh	Beyond
Vault	Init and store primary Kong Cert	Non-root token and namespace	Initial Services use of Vault for secrets	System wide usage of vault for secrets
Certificate Management	Generate certs for Vault and API gateway	X	Generate certs for service to service communication	X
Initial Power Up Secrets	X	Design pluggable abstraction Layer for HW based secure storage	Deliver abstraction layer	Use abstraction layer to encrypt Initial Power up secrets
Service to Service Communication	X	X	Enabled via one of (mutual certs or Token based AuthN)	Secure service registration

IIC Endpoint Security Best Practices Reference: Secure Communications, Endpoint Identity, Cryptographic Services

Cryptographic Services

Feature	California	Delhi	Edinburgh	Beyond
X.509 v3 Certs	RSA: 1024 bits 2048 bits 4096 bits << recommended >>	Elliptic Curve secp224r1 NIST P-224 secp256v1 NIST P-256 secp384r1 NIST P-384 << recommended >> secp521r1 NIST P-521	X	X
Vault Encryption	AES256 W/ GCM mode using 96-bit nonces for IV	X	X	X
File System Encryption	X	X	TBD	TBD
TLS	Server Side	X	Mutual Certs	X

IIC Endpoint Security Best Practices Reference: Cryptographic Services

Hardware Based Security

Feature	California	Delhi	Edinburgh	Beyond
Secure Boot	X	Information Sessions with HW Vendors	Recommendations and Guidelines	X
Root of Trust	X	Information Sessions with HW Vendors	Recommendations and Guidelines	X
Secure Secrets Storage	X	Design pluggable abstraction Layer	Deliver pluggable Abstraction layer	Add 3rd party plugins

IIC Endpoint Security Best Practices Reference: Secure Boot, Root of Trust, Cryptographic Services

Future Security Features

Data Protection

- DAR Encrypted Storage
- Data Protection Policy

Guidelines

- Privacy

Identity and Access

- Administration Local and Remote

Operational Security

- Security Monitoring
- Audit
- SW Update Management
- Attestation
- Secure Auto-configuration
- Operational Security Policy

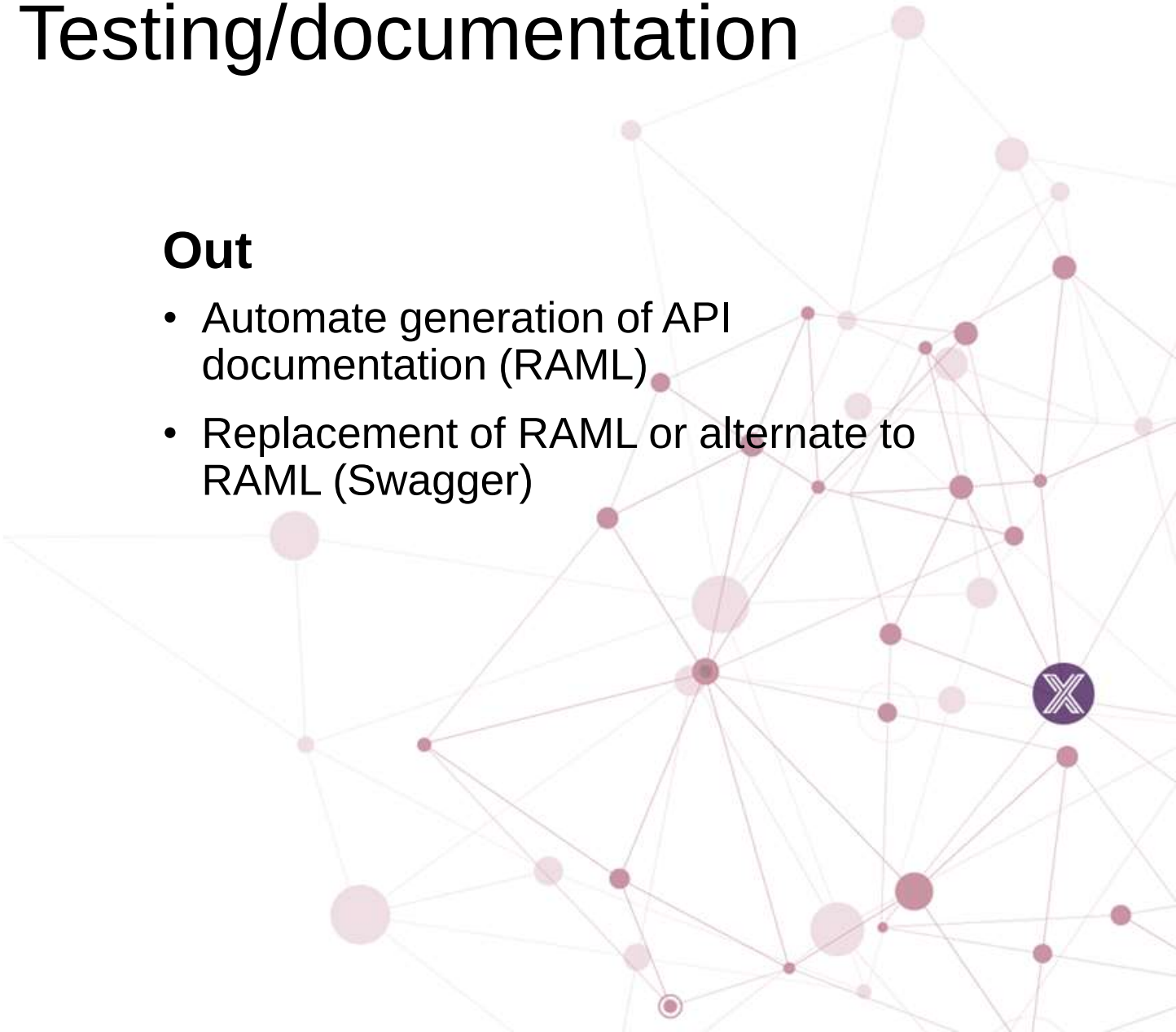
Edinburgh Planning – Testing/documentation

In

- Performance Testing
- Automate security testing
  - What does this include??

Out

- Automate generation of API documentation (RAML)
- Replacement of RAML or alternate to RAML (Swagger)



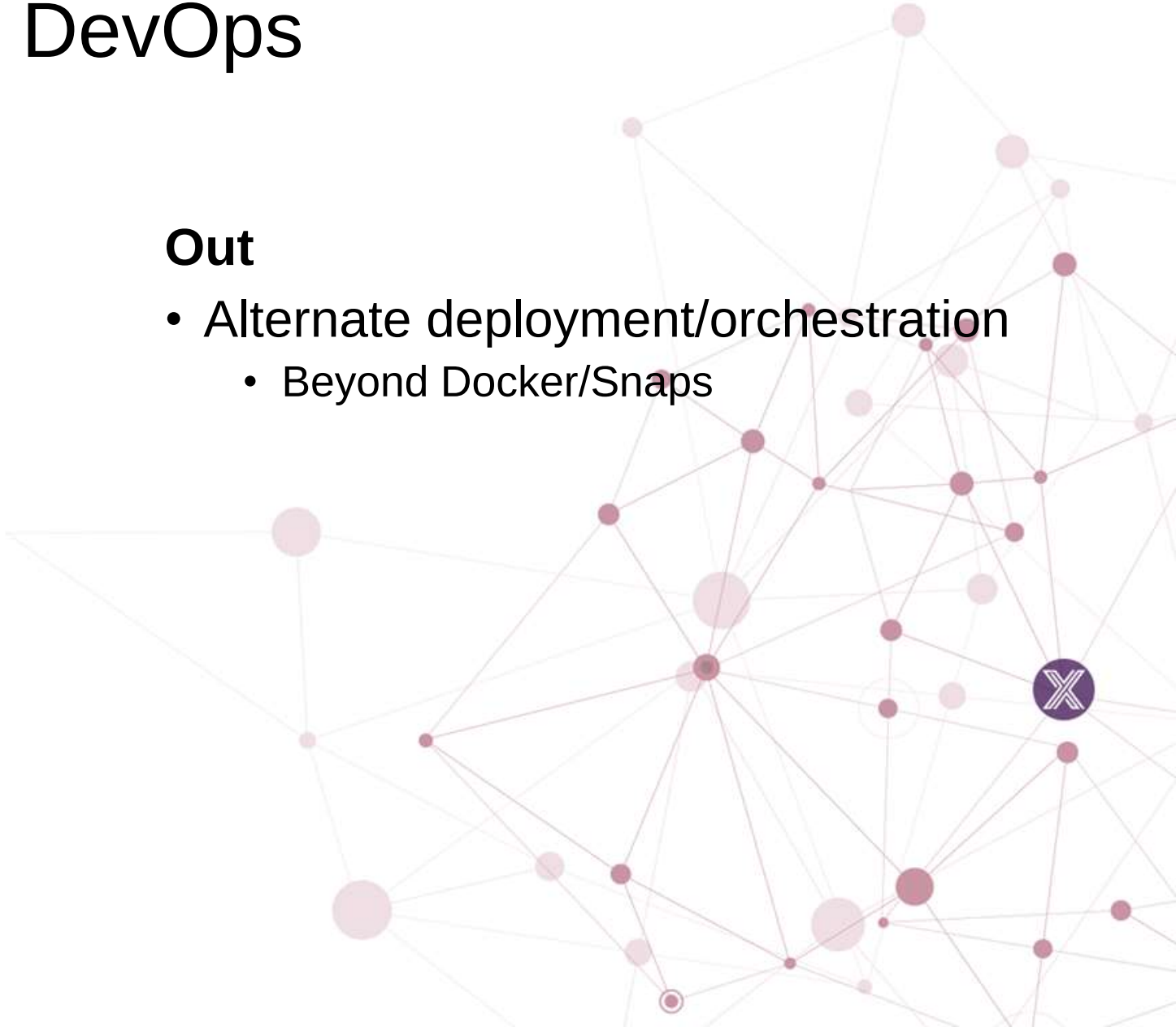
Edinburgh Planning – DevOps

In

- Go 1.11
- Vgo/modules

Out

- Alternate deployment/orchestration
  - Beyond Docker/Snaps



Edinburgh Planning - Vertical Solutions

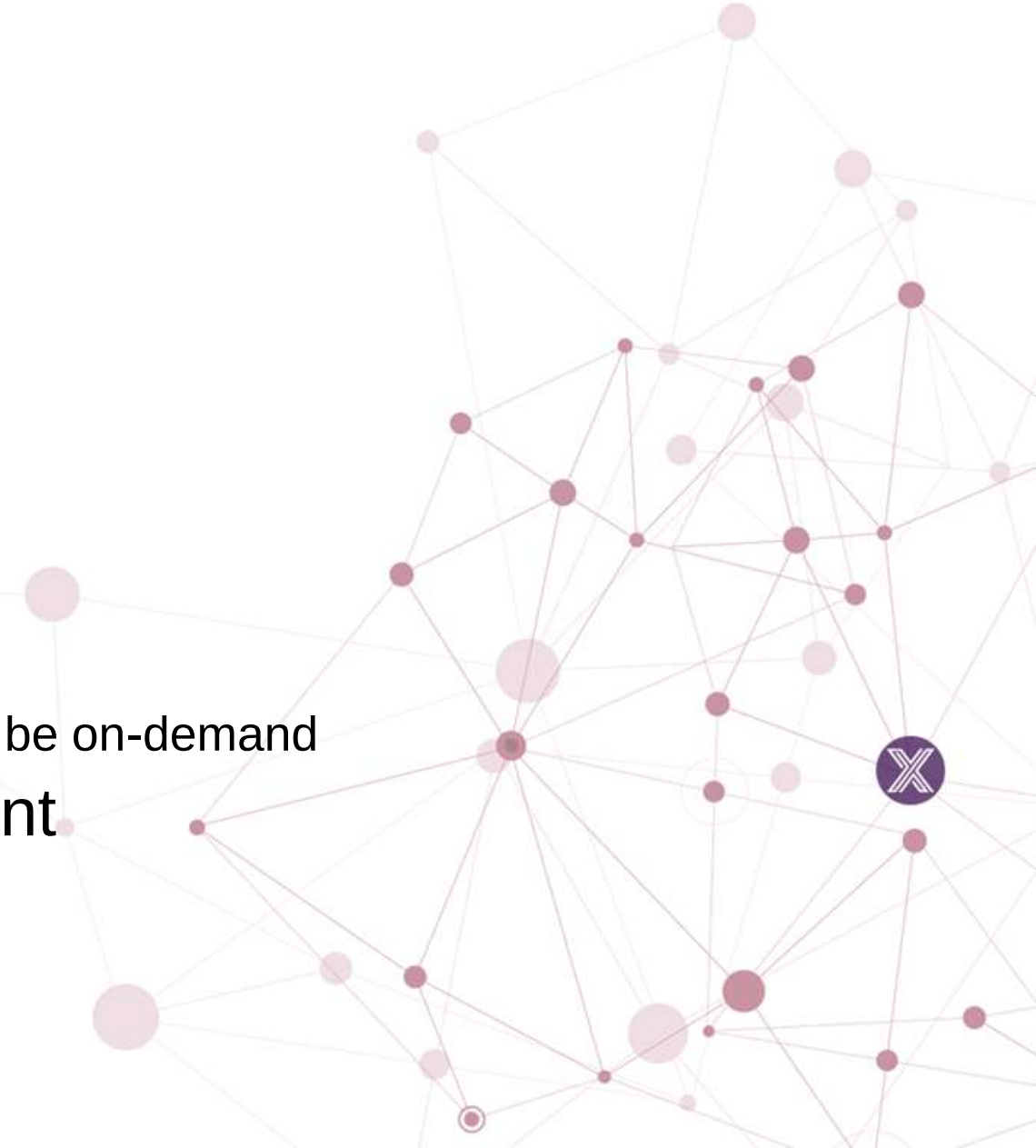
- TBD

Developer Advocate

- Improving onboarding
- Upcoming changes
- TBD

DevOps Issues

- Release Manager
  - Role & Duties
- Release management
  - Keeping track of release features
  - Procedures and planning.
  - Branch cutting
  - What do we want to change to improve?
  - Track release content via Wiki, Github, both??
  - Do we need to formalize dot releases or allow to be on-demand
- Procedures for tracking release content
- Developer participation



Fuji Planning

- What do we want as our major themes of this release?
  - Facilitate East/West capability
  - Micro service load balancing, failover, scale-over, ...
  - Device from EdgeX A triggers action on device on EdgeX B
  - Address data privacy concerns
    - EU laws and affirmation about data use/storage/etc.
    - HIP-A
- Roadmap refresh
- Backlog refresh
- TBD

Geneva Planning (Apr 2020)

- What do we want as our major themes of this release?

EDGE X FOUNDRY™

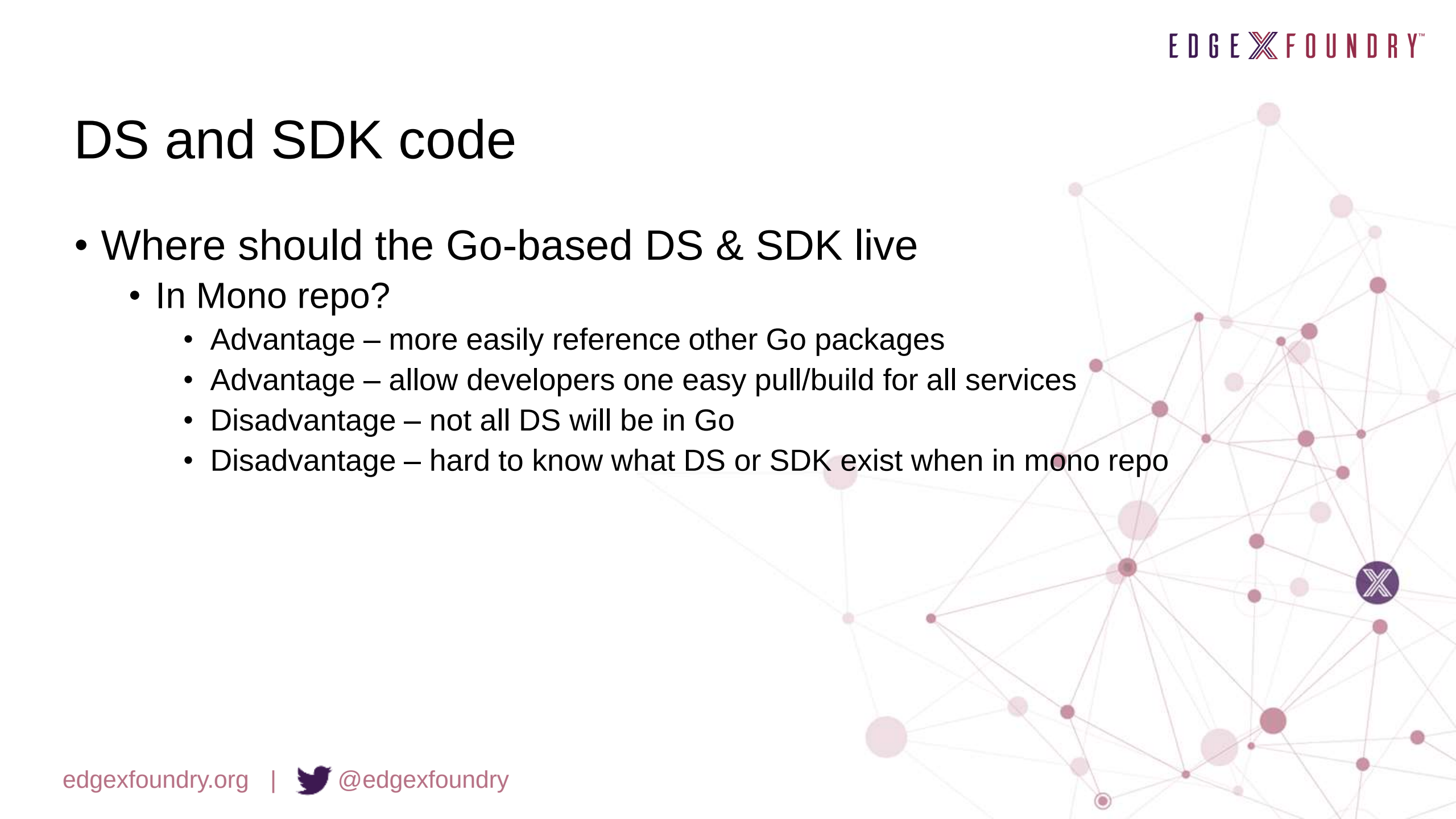
Architectural Issues/Discussion

Database Abstraction/Alternatives

- Per Data Persistence project group, the current plan:
  - Allow DB using services to more easily replace DB – proper abstraction around DB
    - Core data, Metadata, Exports, Notifications, Logging
    - Removing the BSON; hiding domain IDs
  - Replacing MongoDB driver to supported version
  - Implementation of Core Services Using Redis
  - Create a performance harness to test alternate services (using alternate DBs)
  - Create a Certification process to check alternate DB using services with alternate DB
  - Offer alternate DB implemented services via marketplace
  - Defer judgement on reference implementation(s) until Fuji

DS and SDK code

- Where should the Go-based DS & SDK live
  - In Mono repo?
    - Advantage – more easily reference other Go packages
    - Advantage – allow developers one easy pull/build for all services
    - Disadvantage – not all DS will be in Go
    - Disadvantage – hard to know what DS or SDK exist when in mono repo

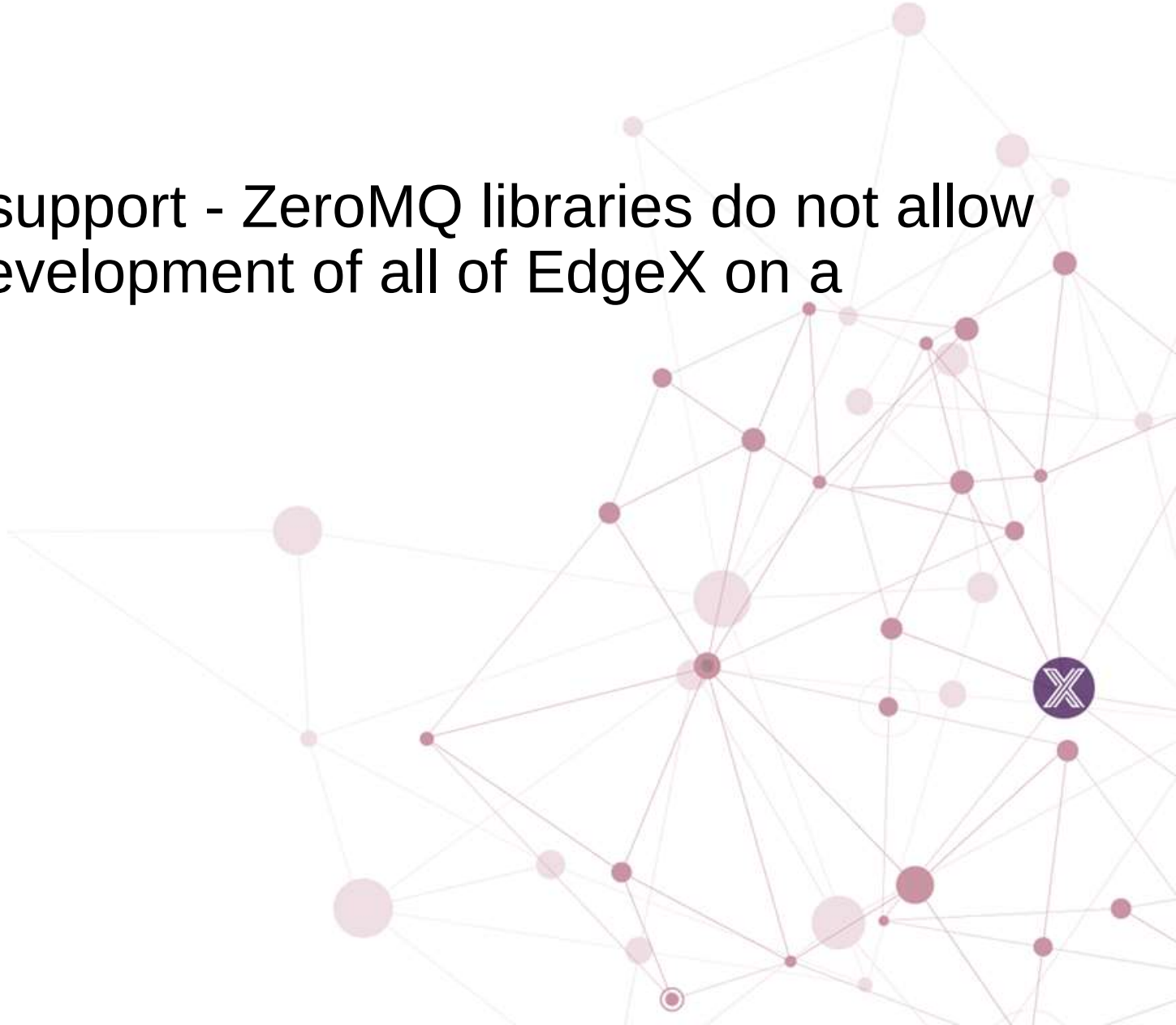


Application Services (aka – new Export Services)

- More to be provided

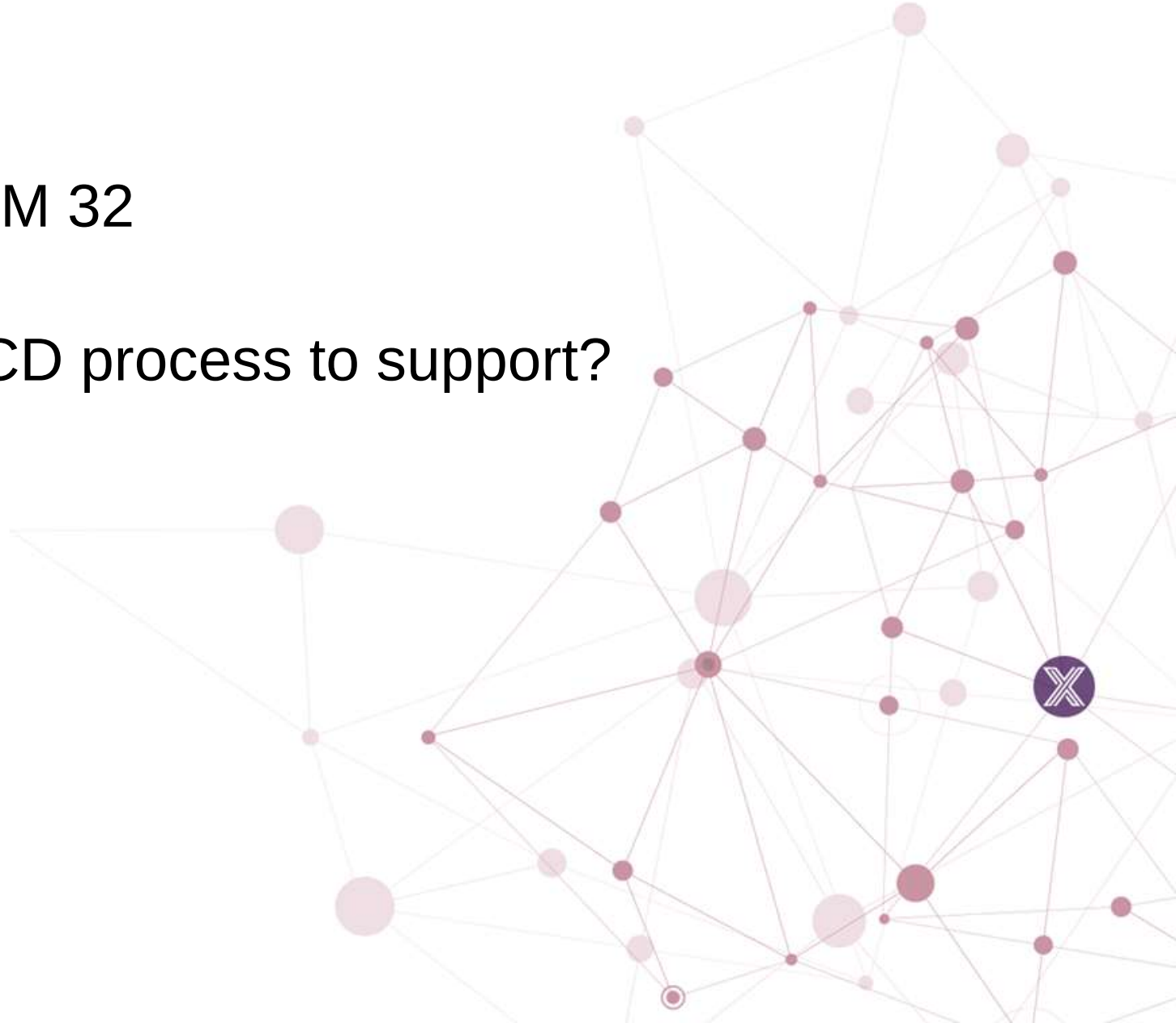
Windows Support

- Full Windows development support - ZeroMQ libraries do not allow (easily) the compiling and development of all of EdgeX on a Windows platform
  - Are we ok with this still?
  - How can we mitigate?
  - Alternatives to 0MQ?



ARM 32 support

- Mongo not supported on ARM 32
  - Requires older version
- Do we want yet another CI/CD process to support?



Device Hierarchy

- Devices managing other devices
  - Fuse had partially implemented idea of Device Manager
- Some protocols have the concept of devices managing or controlling other devices
  - BACNet
  - BLE
  - Mesh networks
- Some use cases/device schema/customers require
  - Schneider Electric FORUM

API Documentation

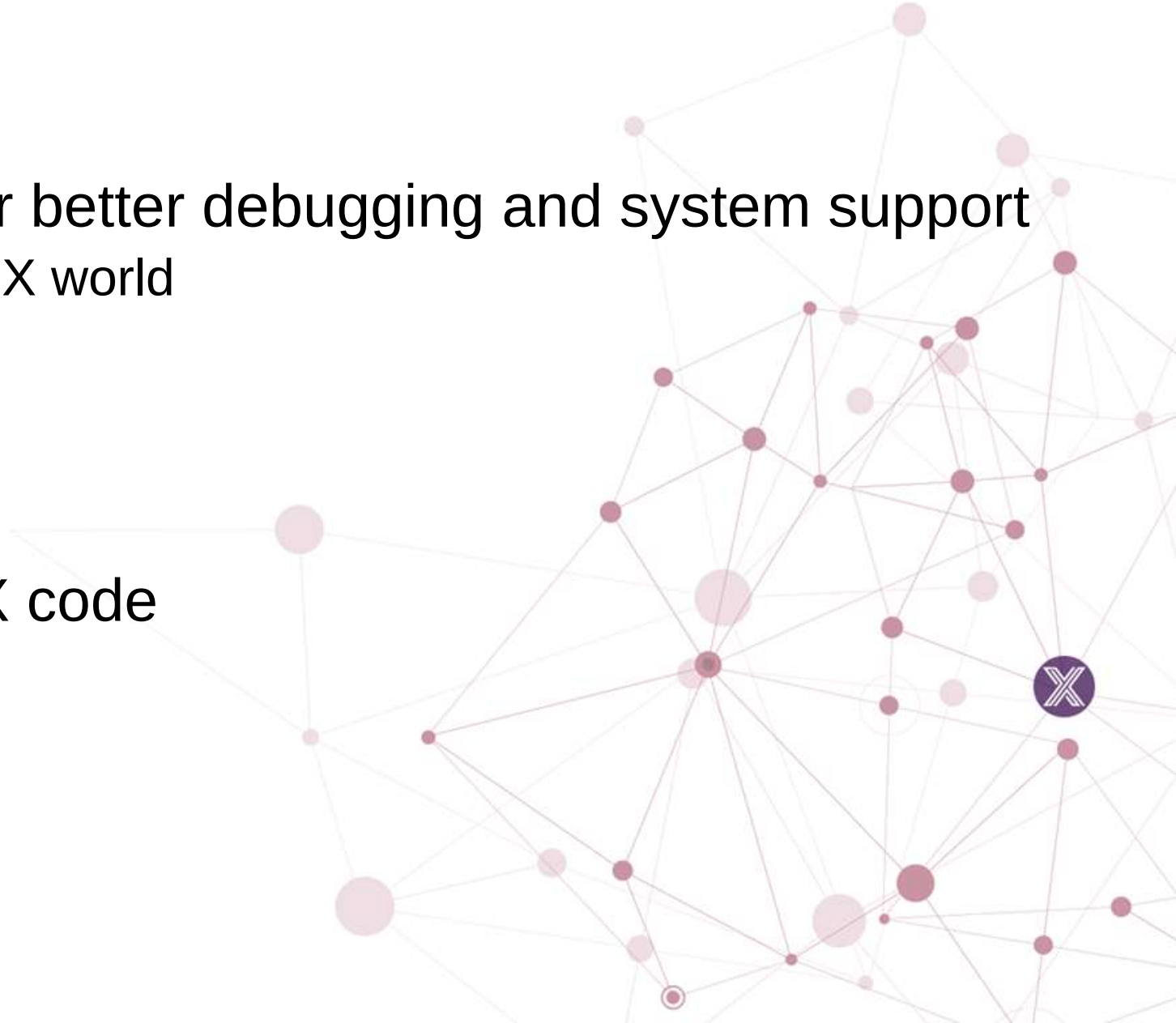
- Documentation has been greatly improved
- Reviews of the API have been made to make it more consistent
- Can we automate the API documentation creation so it more accurately depicts the actual services?
- Should we continue to support RAML
  - Do we consider replacing RAML or adding alternative t RAML (Swagger)?

Go and versioning

- Go 1.11 – when/what's needed
  - CI/CD impact
- When to move from Glide to Modules and vgo
  - What are the considerations/tasks

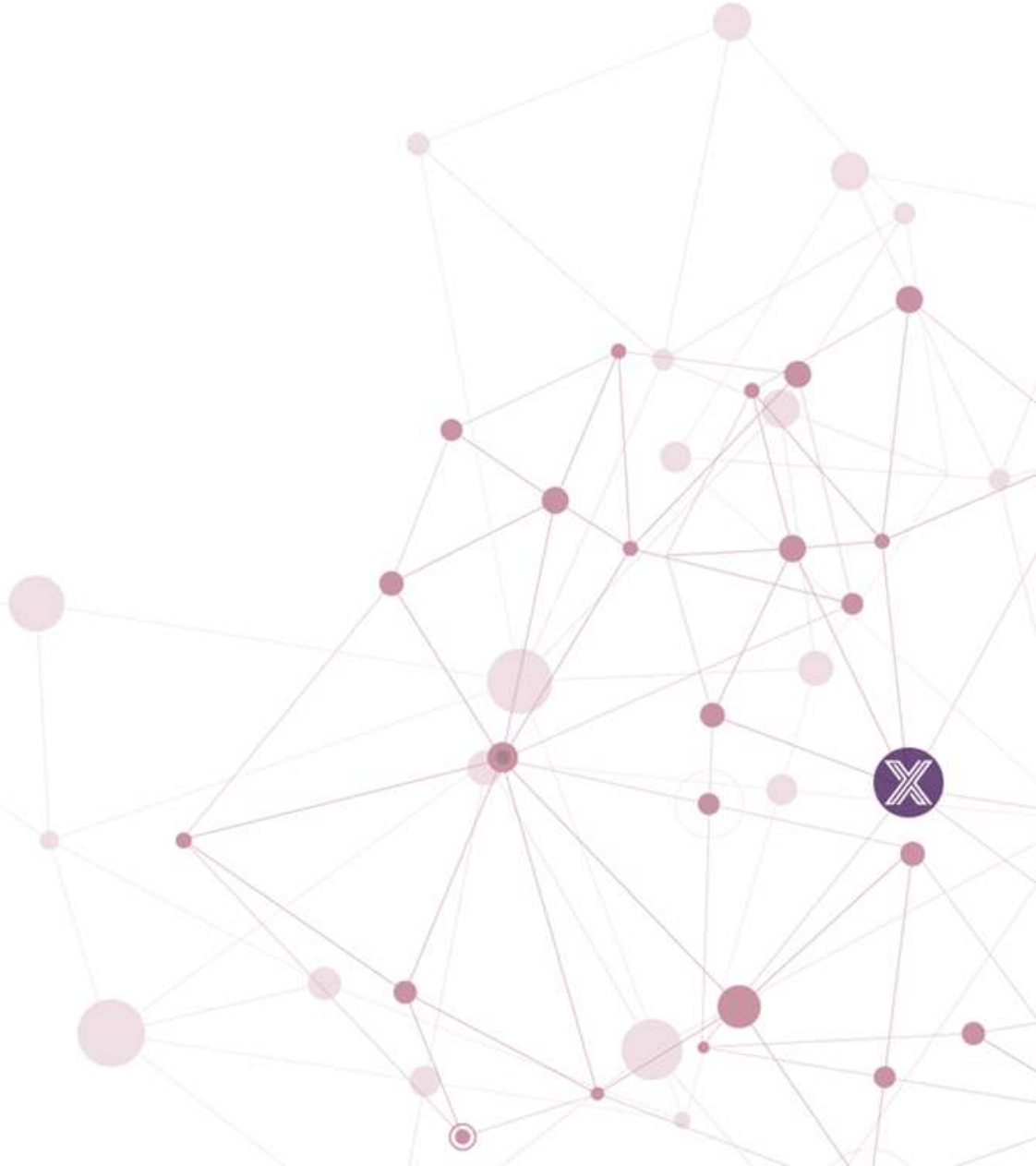
Tracing

- Suggested this is needed for better debugging and system support
  - Especially in distributed EdgeX world
- What is traced and how?
- What can provide tracing?
  - GoKit
- What is the impact to EdgeX code



Performance Testing

- Andy and Team have presented
- What needs performance testing
- How are we performance testing
- What's the impact on CI/CD
- How often are we performance testing
- How about load/stress testing
- How do we crawl, walk, run

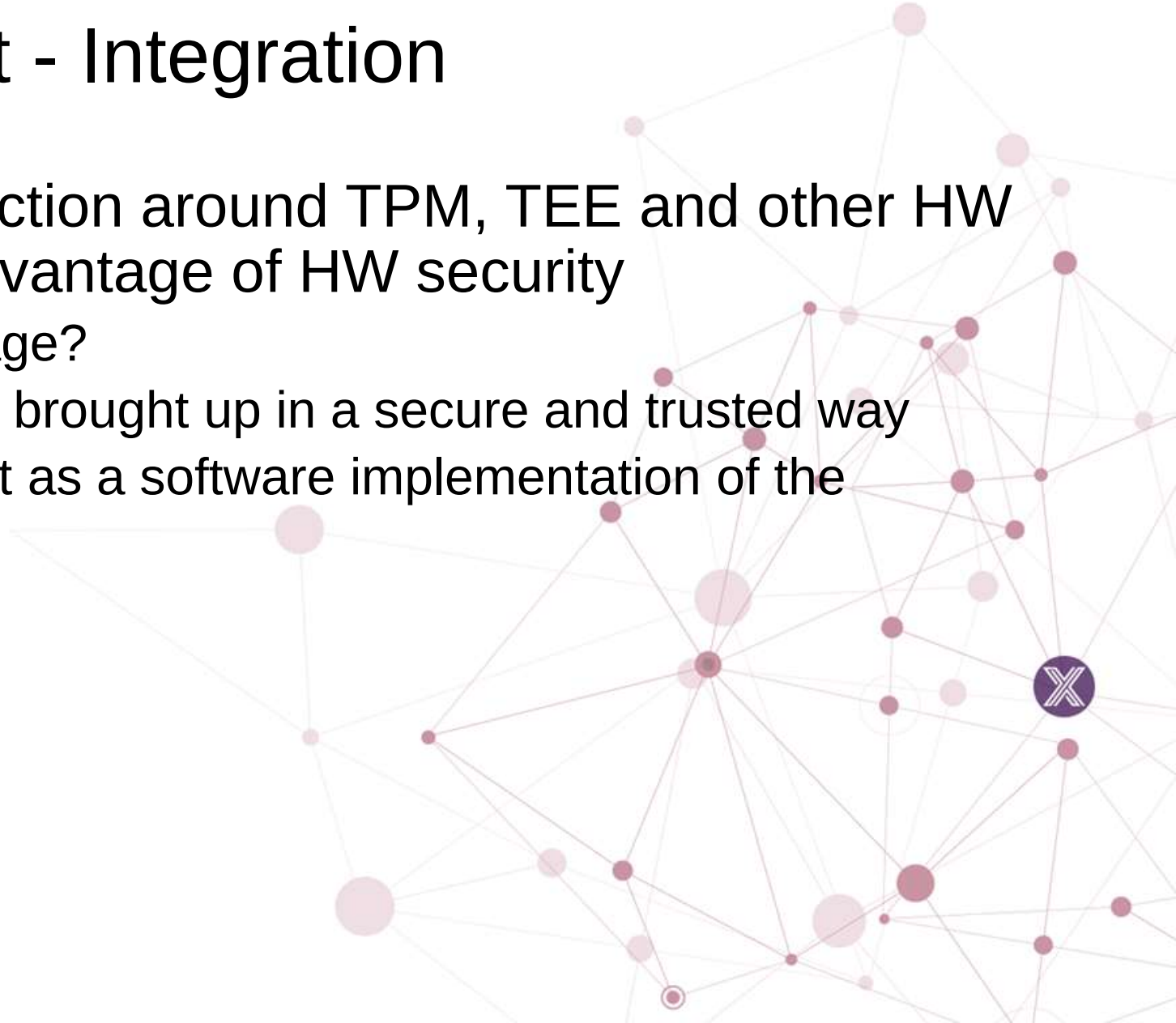


Security Testing

- What is tested?
  - Blackbox tests fail/pass with security in place (Kong, Vault)
  - Check that direct access to services is denied when Kong is on and firewall protection setup
  - Check secrets in Vault after initialization
  - Port scanning (to ensure something hasn't been accidentally left open)
  - Check for weak passwords
  - Test positive and negative access based on access control lists
- How or what parts can be automated?

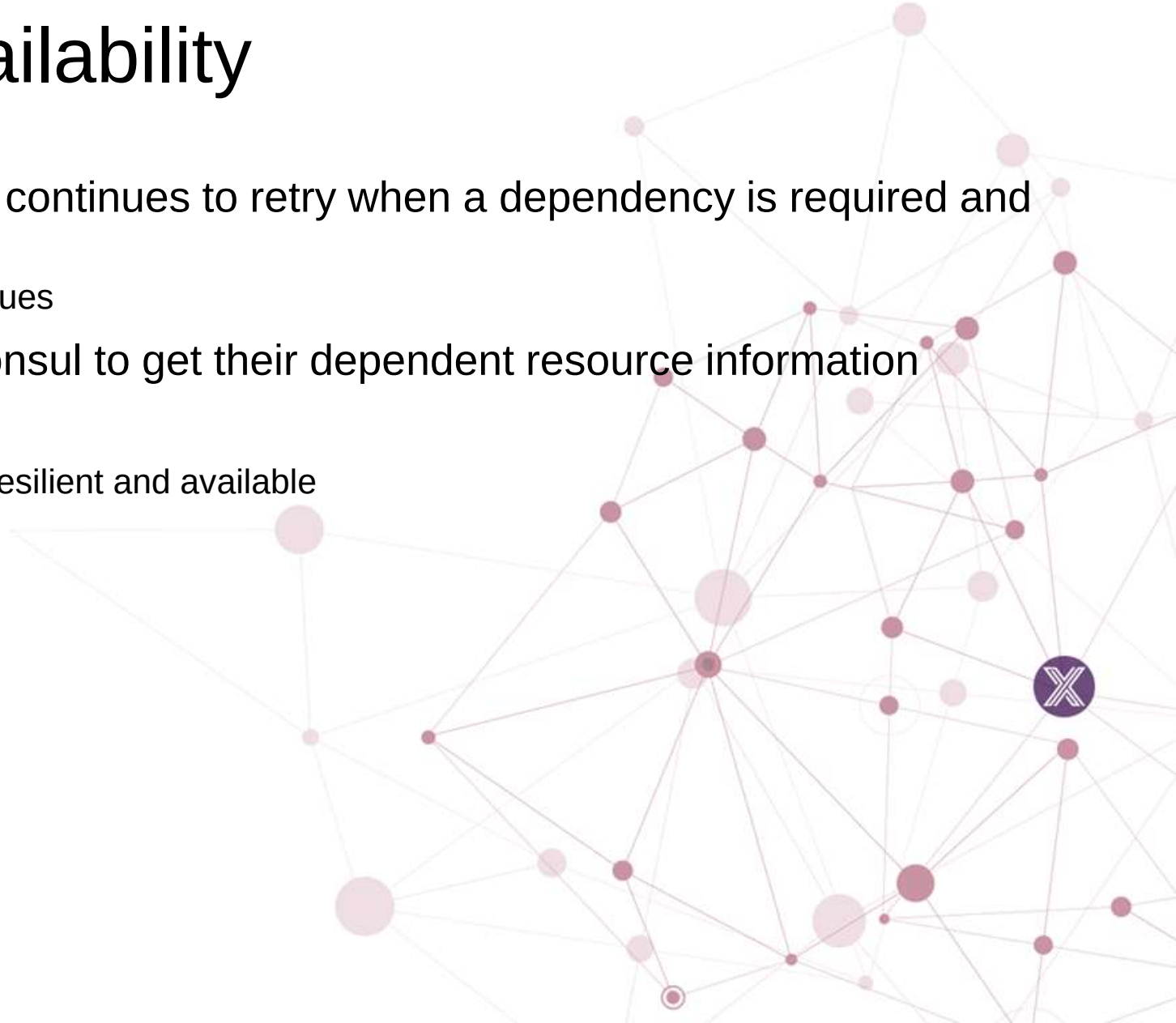
Hardware Root of Trust - Integration

- Can EdgeX offer and abstraction around TPM, TEE and other HW RoT mechanisms to take advantage of HW security
  - What gets stored in HW storage?
  - How is the EdgeX application brought up in a secure and trusted way
  - How / what can we implement as a software implementation of the abstraction



Improve Resiliency/Availability

- We added code to make sure a service continues to retry when a dependency is required and not yet up
  - Services now are more resilient to timing issues
- We added code for services to go to Consul to get their dependent resource information
- What do we need to do next?
  - How can we make our services even more resilient and available



Distributed EdgeX

- What is needed to allow every/all EdgeX services and infrastructure operate on a different host?
- Allow device from EdgeX instance on A to trigger action on device on EdgeX instance B
- What is needed to allow load balances to operate in front of every/all EdgeX services and infrastructure?
- What prohibits truly distributed/scaled out EdgeX today?
  - Kong and how we address access security across hosts?
  - How to assign or address a device to a single instance of the device service?
- Are there parts of EdgeX that cannot be distributed or cannot be scaled?

Config and Metadata Changes

- Changes to configuration in Consul are already immediately made available to applications.
- Applications must implement a “watcher” to see a configuration change and then call to get that change
- Further, even if micro services are made to be more dynamic in watching for and using new/updated configuration, some of the configuration changes would only work after a restart (ex: the REST endpoint port number of the micro service).
- Is there a way to signify which configuration is allowed to be changed at runtime and which can only take affect after a restart of the service.
- How should we handle changes in Metadata which also impact the operations of a service?
  - Example: addition of a device through Metadata API

Alternative Deployment and Orchestration Options

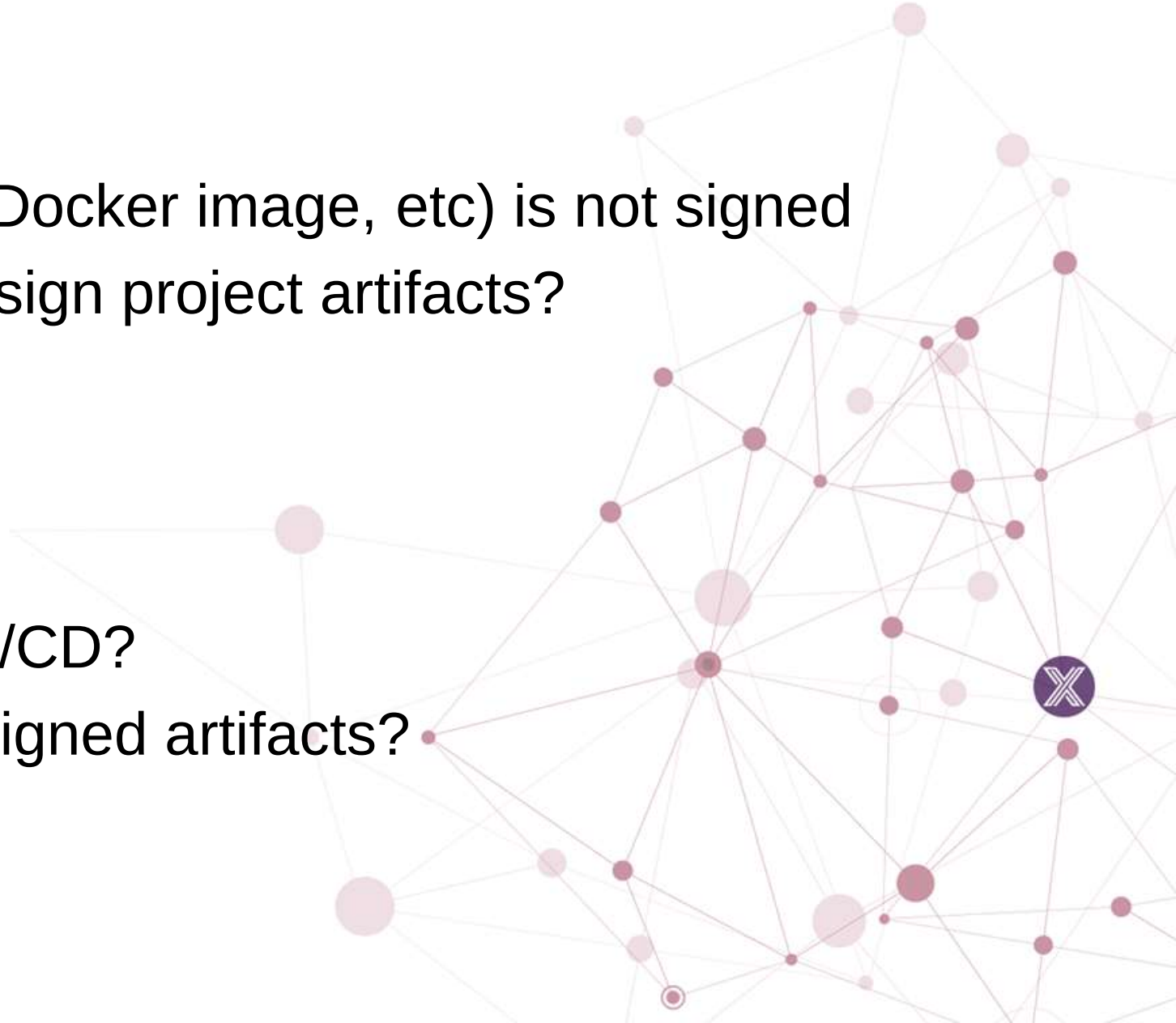
- Apart from Docker/Docker Compose & Snappy deployment and orchestration options for EdgeX
- The community and users of EdgeX are free to deploy EdgeX as they see fit based on their use case/needs
- Going forward should we look to support (by reference implementation) Kubernetes, Swarm, Mesos, Nomad, to name a few.
- Should we only support a single reference implementation for demonstration and allow 3rd parties/marketplace address?

Facilitate Commands from the North

- The command API allows anything to call and actuate a device (GET or PUT)
- The API is not supplied to any north side system as part of export.
- The northside system would have to know how to call and get the device actuating APIs
- How can we facilitate command information to be supplied and known to the northern edge systems.
  - Example, how would we provide Azure IoT with commands that it or a cloud solution could use to actuate on devices?
- How should this be secured?

Artifact Signing

- Today, artifacts (EXE, JAR, Docker image, etc) is not signed
- Can we/ should we digitally sign project artifacts?
  - Go executables
  - JAR files
  - Docker images
  - Etc.
- How would this effect our CI/CD?
- Who/what would verify the signed artifacts?



Downsampling

- The device service may receive from the device new unattended readings (e.g. in a pub/sub type of scenario).
- There should be a setting to specify whether we accept all readings or we decide to downsample because the source is pumping data too fast.
- This is actually a very common scenario when you deal with high frequency sensor packages.

Command Parameter Check

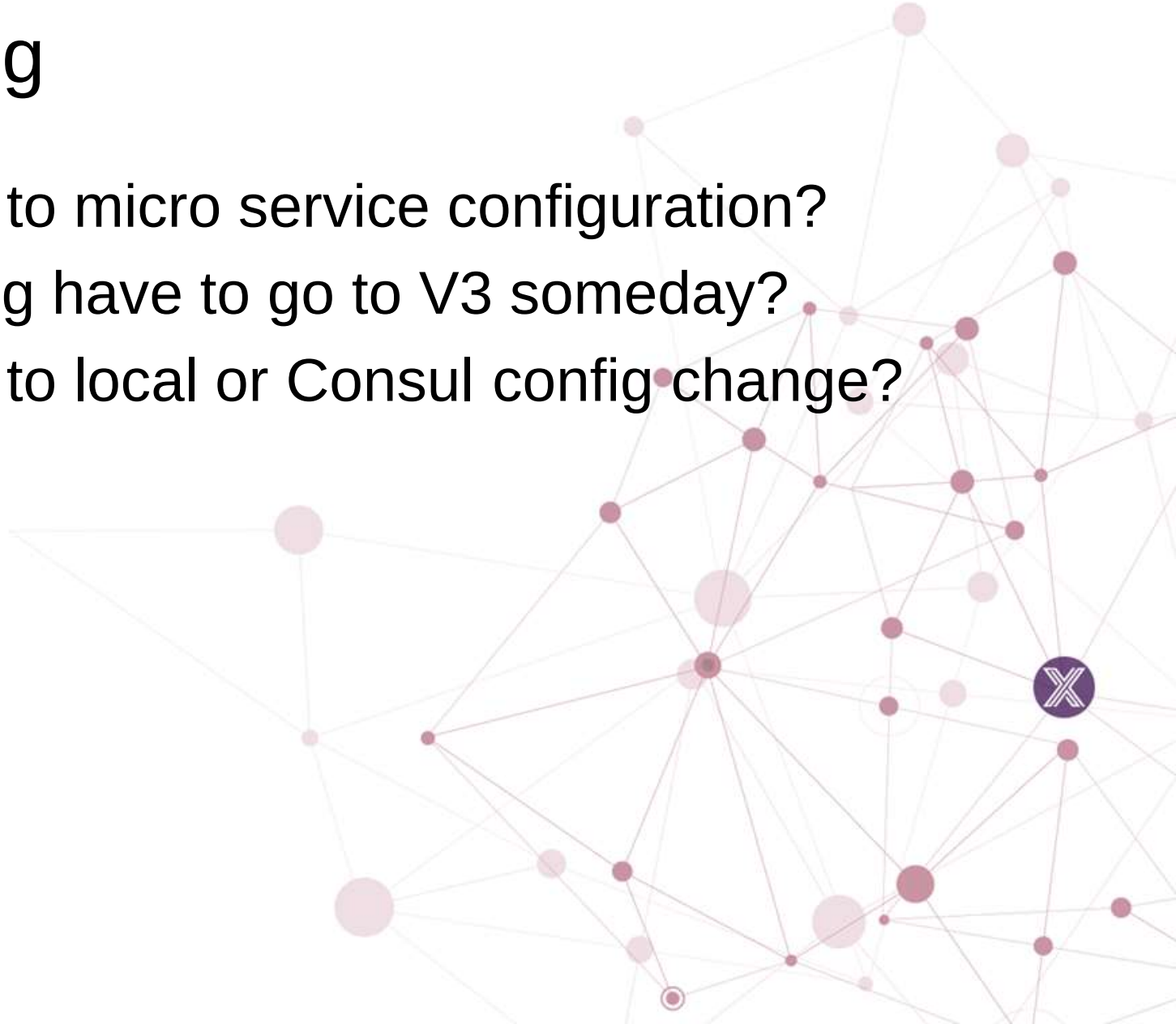
- In order to protect the device from harmful commands, there should be the possibility to set a Min and Max limit for the value that is accepted on every single command.
- The command service today is rather a hollow simple proxy, but in the future we very much envisioned adding additional security, caching to avoid having to hit the DS when unnecessary, and even grouping command requests for better resource conservation (especially for devices like BLE that get woken up when you hit them).

Message Infrastructure

- Implementation of message bus alternative for intercommunication between microservices as an alternative to REST.
- While REST will not go away (a REST API will still exist around each micro service), there may be a need to implement point-to-point messaging between select services or to adopt some type of message bus unilaterally across all of EdgeX to support messaging among services.
- Messaging provides for more asynchronous communications, typically lower latency communications, and better (or more finely tuned) communication quality of service (QoS). Are there places where messaging might be more appropriate (like between core data and export distro today). Would a use case dictate the use of an alternate messaging infrastructure be used among all services with an underlying message bus to support it? Would alternate protocols (SNMP, WebSockets, etc.) be desired in some use cases?

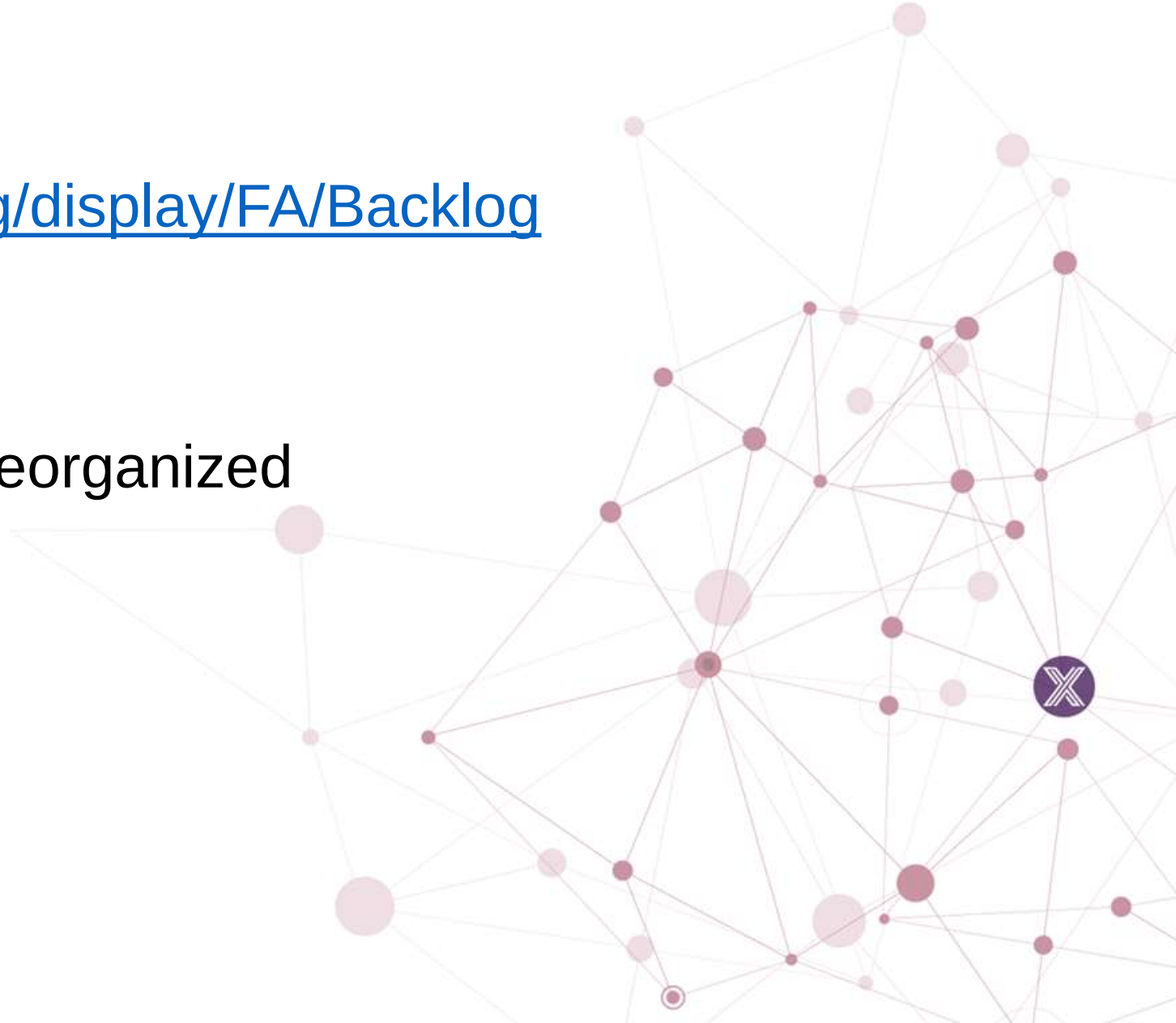
Configuration versioning

- How do we handle changes to micro service configuration?
- We have V2, does everything have to go to V3 someday?
- How do we handle changes to local or Consul config change?



Review of Backlog

- https://wiki.edgexfoundry.org/display/FA/Backlog
- What needs to be removed
- What needs to be added
- What needs to be updated/reorganized

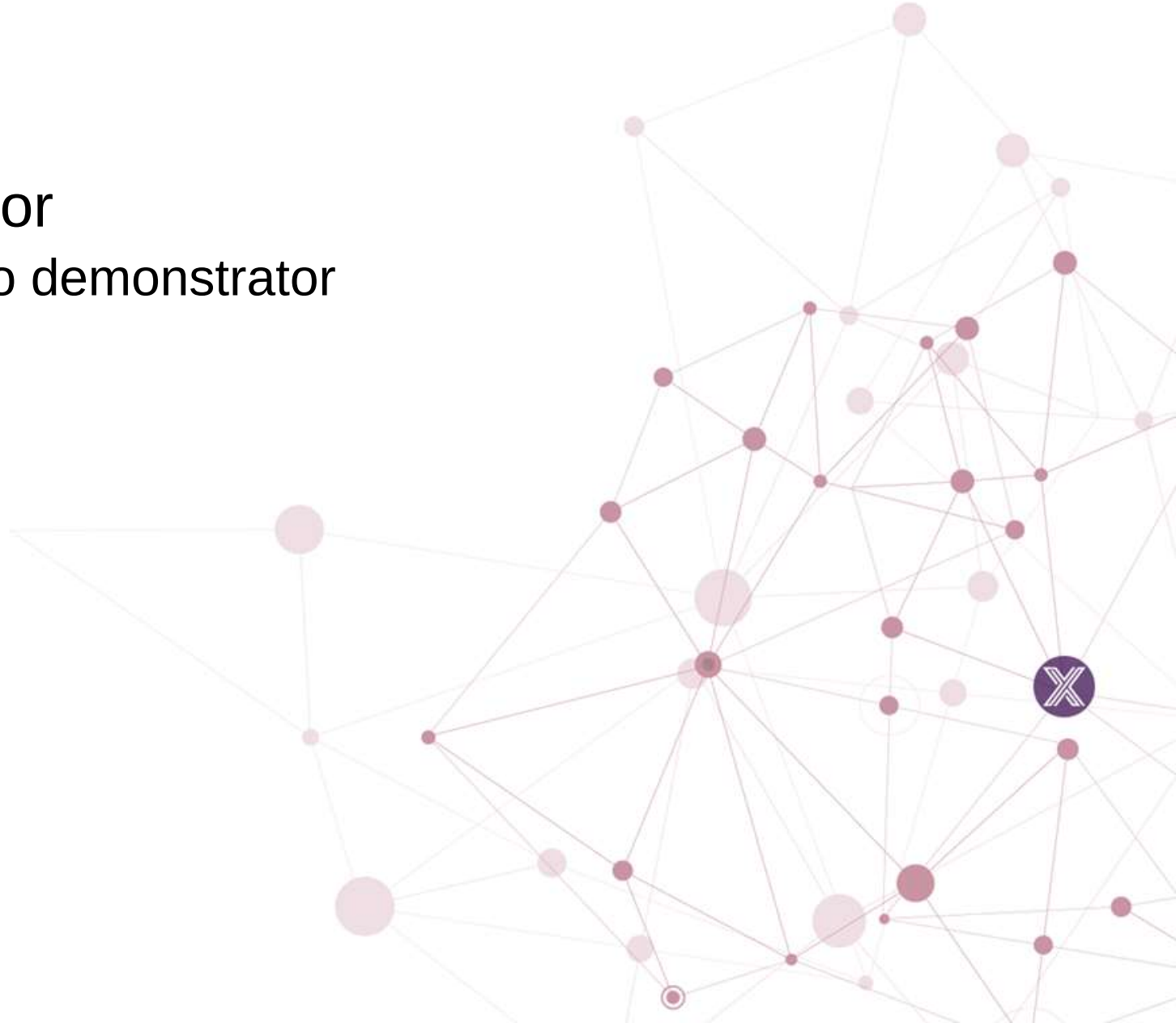


Business Issues

- May be moved to a different day/time
- Discussion items
  - Demonstrator
  - Dev kits
  - IIC and other liaison efforts
  - Which events we use to show case and announce
  - Certification/Marketplace offerings
  - Marketing and messaging efforts (like those raised by Moonki at the last meeting)
  - Developer Advocate and other role

Demonstrator

- Status of IoTech demonstrator
  - Potential additions/changes to demonstrator



Architecture

Basking Demo will show analytics deployed on both the Cloud and at the edge.

Basking IOT Backend

Data and control

Demo will show EdgeX deployment on both x86 and ARM GW hardware. All gateways shown already supported by IOTech.

Dell 3000 or 5000 (x86)

Cavium GW (ARM, 64bit)

Mainflux MFX-100 Edge GW (ARM, 32 bit)

Data via API and MQTT connection directly to the Cloud

Google IoT Core

Data and control

Data

Data and control

Data

Wired & wireless protocol devices

CoolMasterNet **Modbus TCP** HVAC controller

EnOcean Temperature sensor

Deuta **EnOcean** Lighting Controller

EnOcean Light Level sensor

Salto Klay **API enabled** Access management

API enabled Voltaware Energy monitor

Climate control

Lighting control

Occupancy tracking

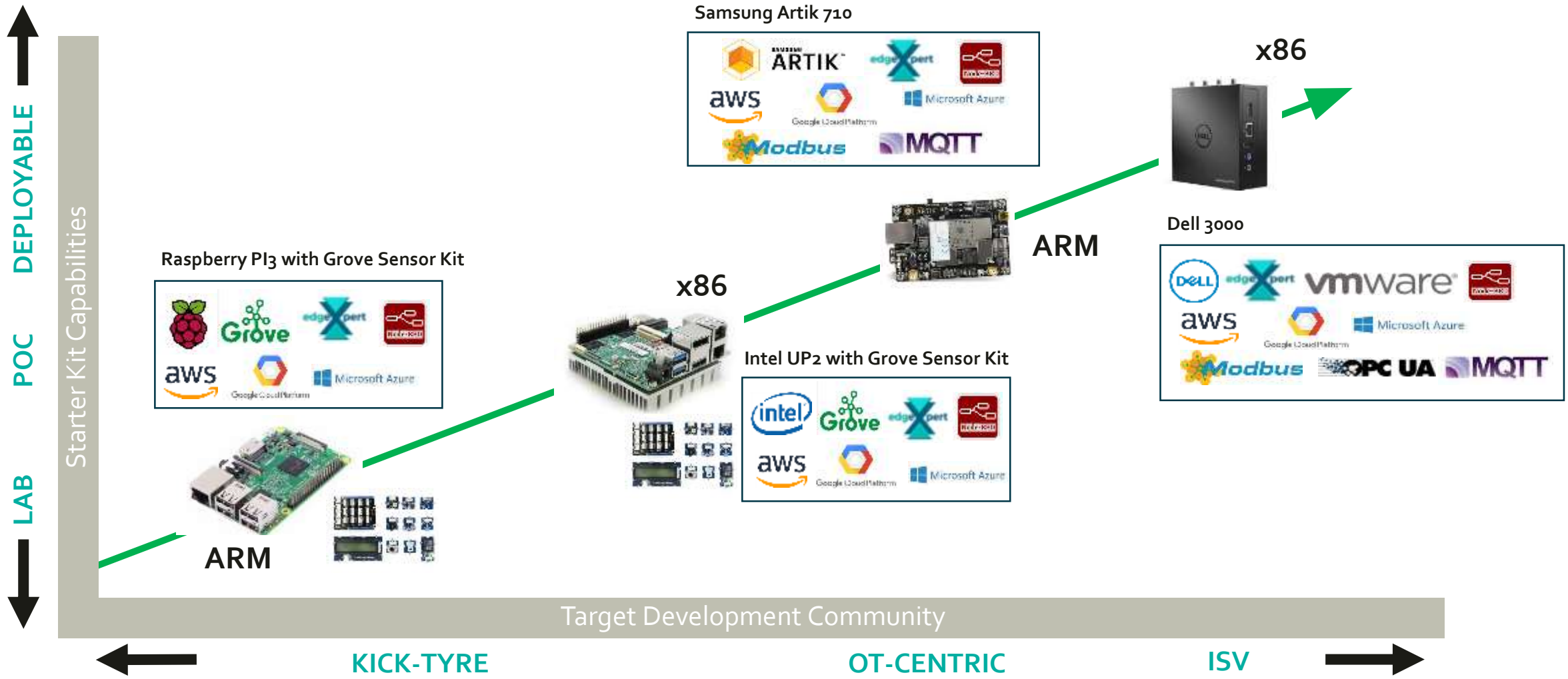
Energy Management

IOTech

DevKits

- TBD

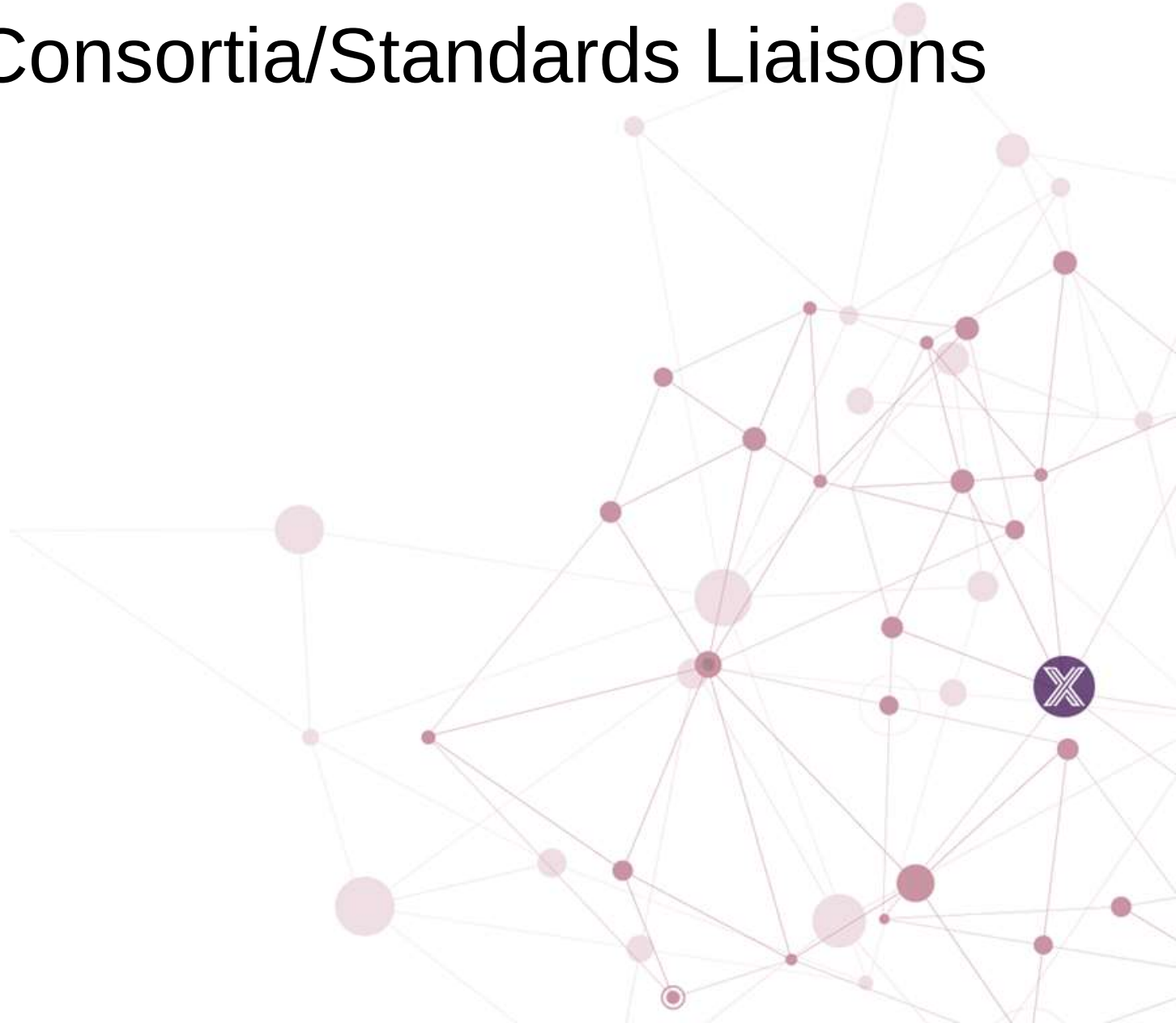
Edge X IoT Dev Kits (Available November)



IOTech

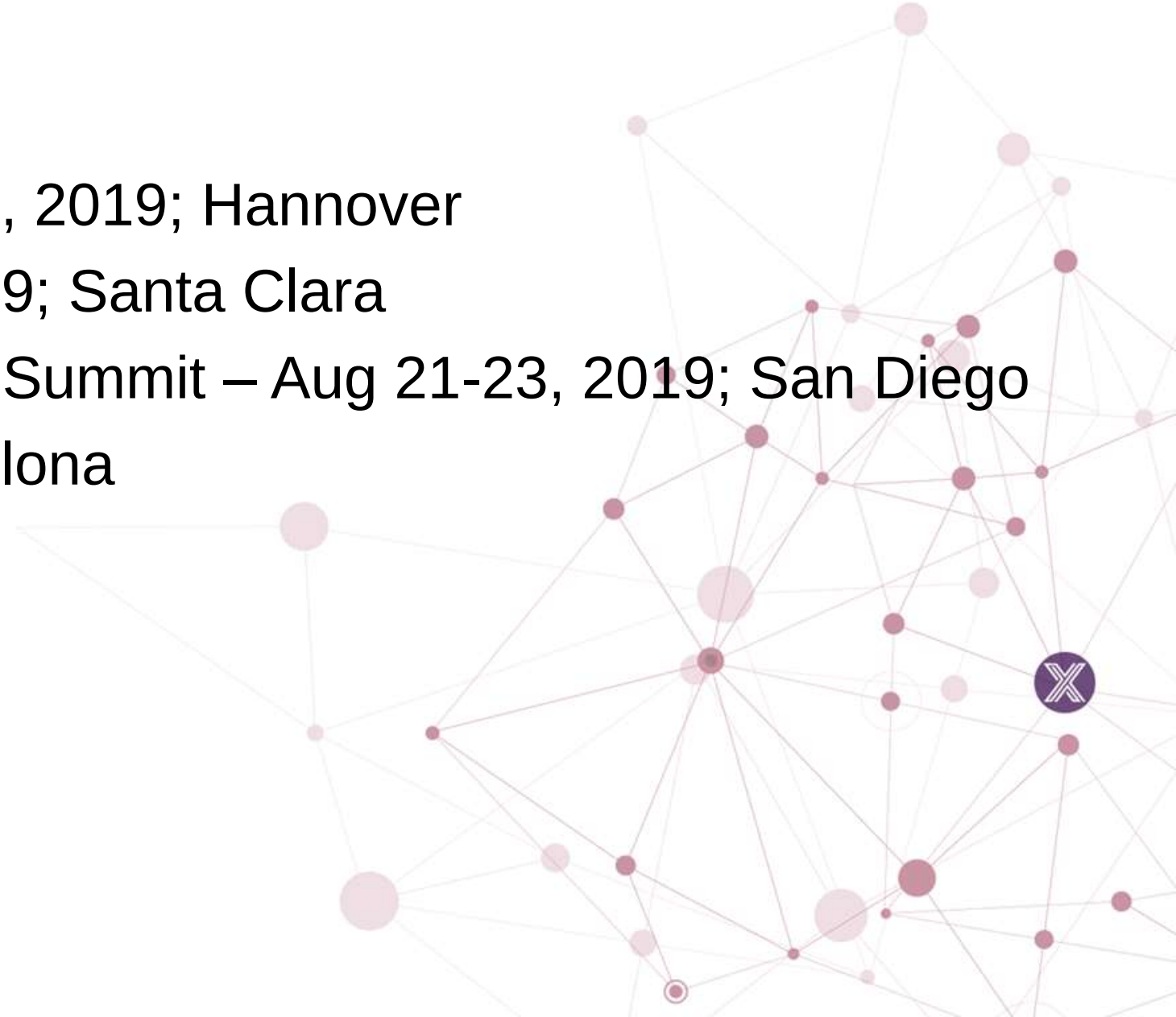
Edinburgh Planning – Consortia/Standards Liaisons

- TBD



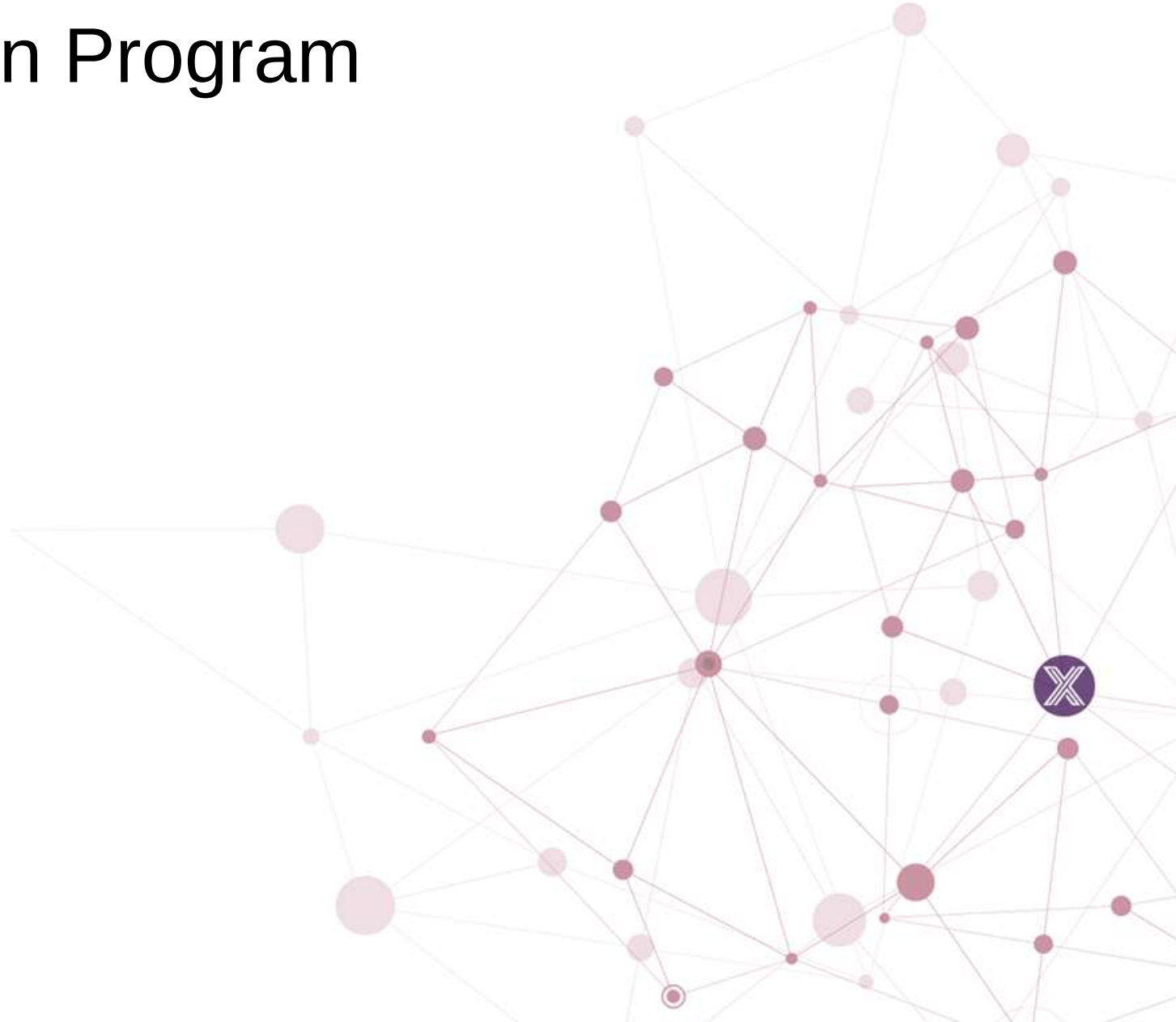
Future Big Events

- Hannover Messe – April 1-5, 2019; Hannover
- IoT World – May 13-16, 2019; Santa Clara
- Linux Foundation Open IoT Summit – Aug 21-23, 2019; San Diego
- IoT SWC – Oct 2019, Barcelona



Edinburgh - Certification Program

- TBD
  - What are we certifying
  - What does it look like
  - What resources does it need



Developer Advocate Role

- Michael Hall's contract expires at the end of the year
- Is the position still required?
- If so, do we retain Michael?
- How should the role change?
- Are there tasks/priorities we want to shift with Michael?
- Are there other roles we need to consider?