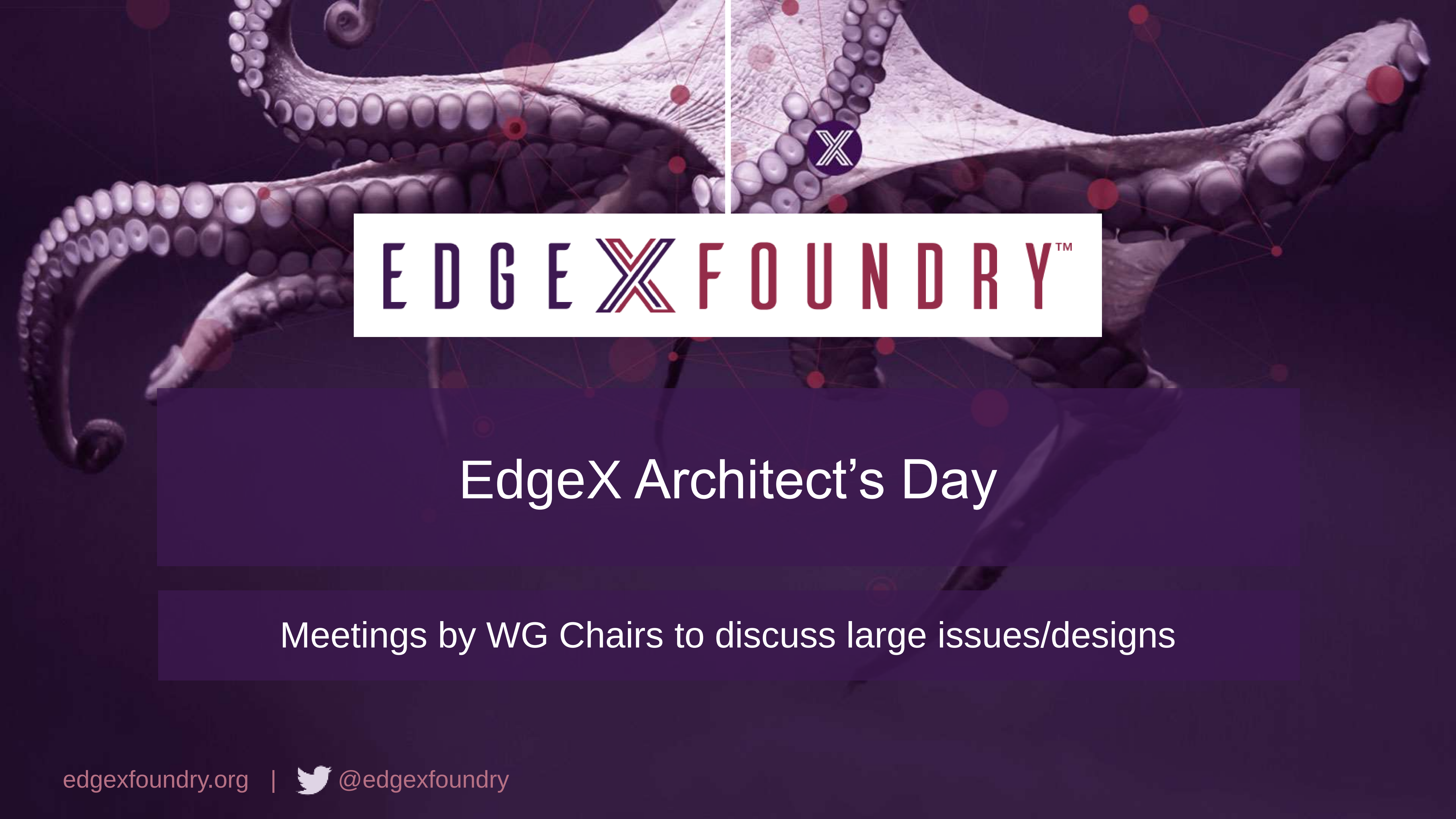
EDGEX FOUNDRY™

Edinburgh TSC F2F Working Agenda and Deck

Edinburgh, UK

Oct 23-25, 2018



EDGEXFOUNDRY™

EdgeX Architect’s Day

Meetings by WG Chairs to discuss large issues/designs

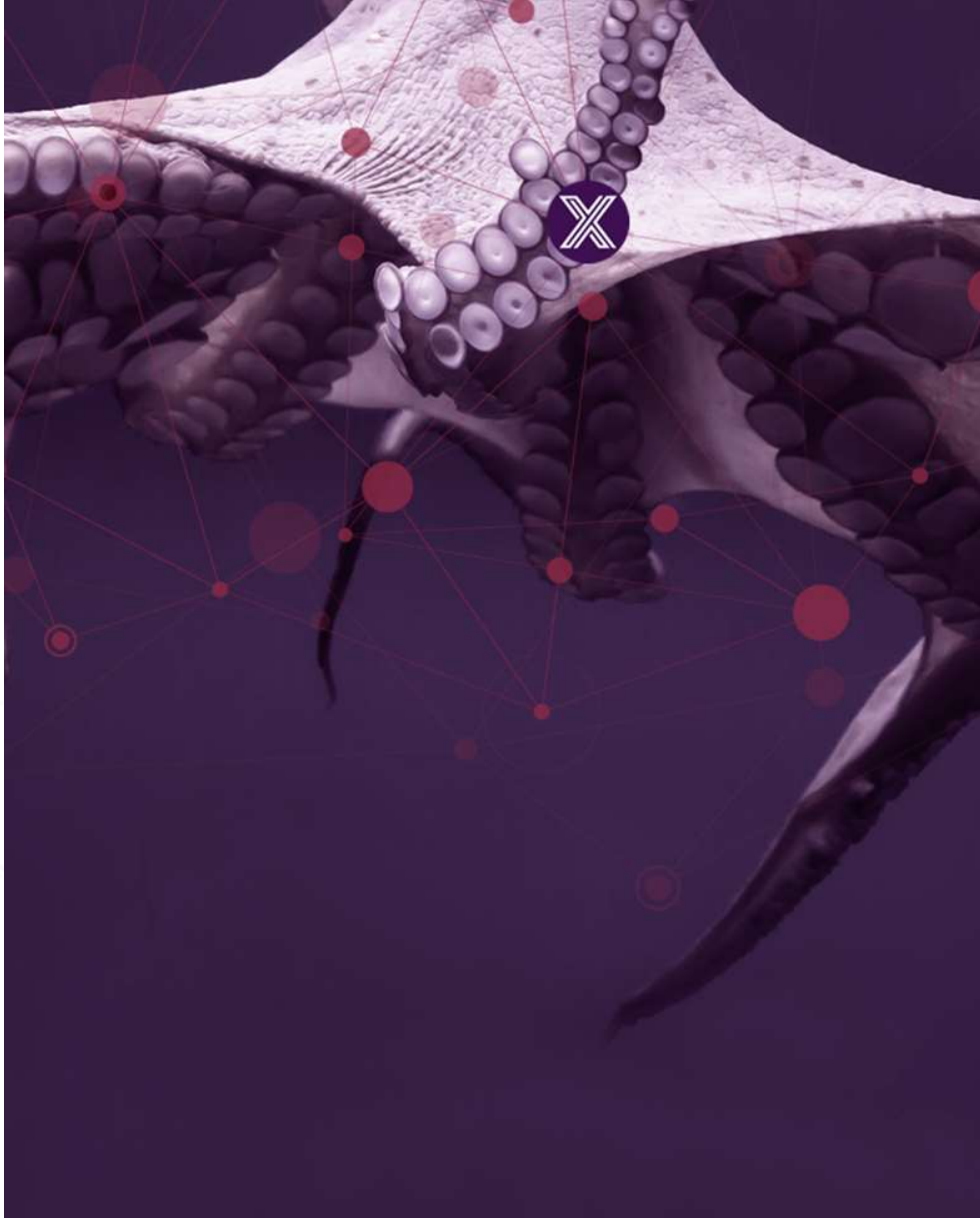
EDGEXFOUNDRY™

New Member Welcome & Architecture Tutorial

7:30am – 9am, Oct 23

(optional to existing members)

Presented by Keith Steele & Jim White



edgexfoundry.org | @edgexfoundry

Edinburgh Architect’s Meeting (pre-F2F) – Oct 23 9am - 5pm

- Morning Topics *(IoTech arriving at ~ 10am)*
  A. System Management WG
  - What is EdgeX system management versus gateway management
  - Scoping future system management capabilities

  B. Security WG
  - HW based security discussion
  - Where are the system management/security overlaps?

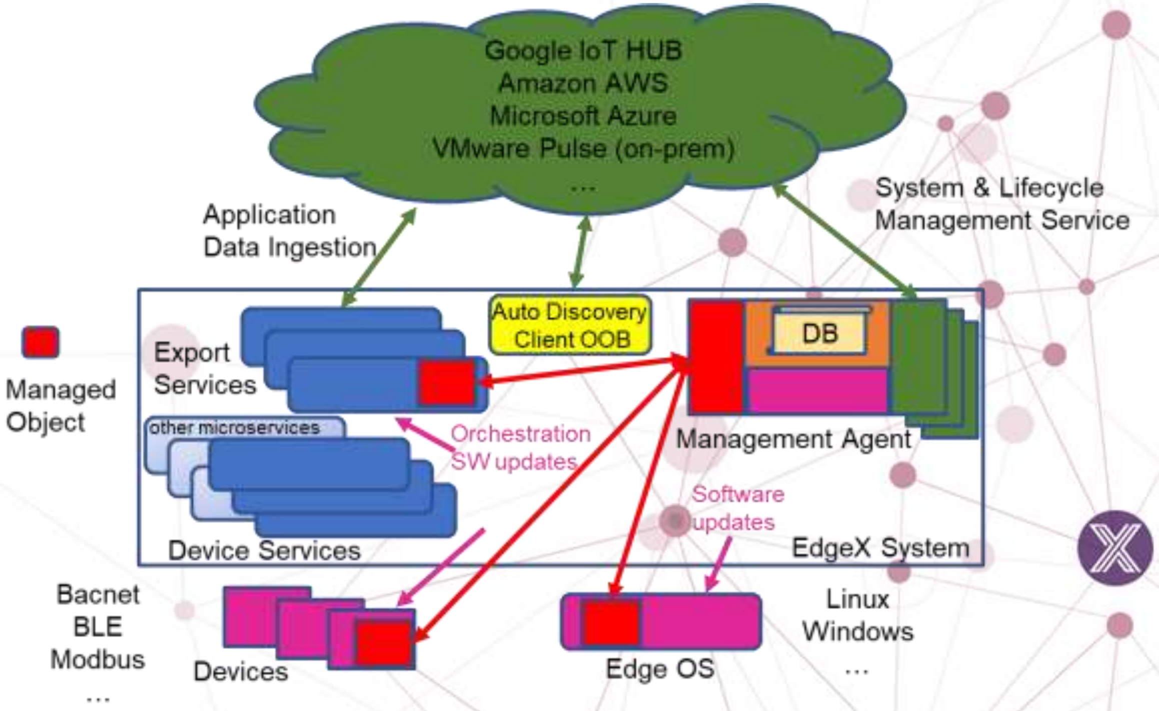
  C. Application Services Design
  - The new export services; more scalable; more configurable & loosely coupled

- Afternoon Topics
  D. Device Service/SDK
  - Device onboarding; DS callbacks; adding device through metadata and next iteration of SDK

  E. Test/QA
  - Performance testing strategy
  - Test with multiple configuration(s) strategy (edge case testing)

A – System Management Topics

- What is the scope of system management in EdgeX?
  - Already done (or in the roadmap)
    - Allow for start/stop/restart of services (when something like Kubernetes is not there)
    - Get / set configuration for services
    - Get service metrics
    - Notify on a change to service status, config or metric
  - Monitor and take action?
    - Restart a service if it stops
    - Stop a service if it uses too much memory
  - Provision a device?
  - Initialize a database (or something like Consul or Vault)?
  - Update or deploy EdgeX services?
- If EdgeX is not gateway or edge platform management … what is?
  - Who do we need to work with for providing our edge platform management requirements?
  - Are there going to be integration points with the edge platform management?



Sys Mgmt Conclusions

- Out of scope of EdgeX
  - Orchestration of services
  - Install/update of software, firmware, OS, intra-structure (networks, etc.)
- SMA is single point of management for EdgeX deployment
  - 1 - SMA per deployment (could mean multiple copies of EdgeX services like core data)
  - We'll have to consider this with regard to service implementation
  - "It's a deployment of services not multiple EdgeX instances across the fog"
- Stop – done with option to call service directly or manifest specified "stop function"
- Config Set – done to Config service
  - Config service must persist change
  - Option to repopulate from config seed
  - Set properties by naming conf – but user bewaire
- Detect / health check is proxy to config/reg service
  - Allows one to restrict access to consul when necessary

B - Security Topics

- How is root of trust going to get accomplished?
  - What can we/should we categorically define?
  - What are the EdgeX design and implementation guide(s) when we can't provide the technology?
    - What gets store in HW storage?
  - What is EdgeX code in this area?
    - What are we dependent on?
  - How do we address both TPM and TEE?
  - What does it take to get to something like what Intel demonstrated – but more generically?

- Where do system management and security overlap?
- What coordination is needed?
  - Can system management provide a means to provide certs to security services?
  - Security needs a "kick off" service that bootstraps. Is this system management?
- Consideration: Intel's Open source Go wrappers around TPM utilization (Box Creek)

C – Application Services

- What should the design of the new Export Services (aka application services) look like?
  - What functions should it perform?
    - Essentially an EAI engine
    - Filter (only give me readings from device A; only give me readings regarding temperature, …)
    - Transformation (convert C to F values, convert CBOR to Protobuf, …)
    - Enrich (add device metadata to reading, …)
    - Format (JSON, XML, CSV, …)
    - Encrypt (really different kind of transformation)
    - Compress (really different kind of transformation)
    - Custom (black box that you define what you want to happen inside)
  - How do we order the functions?
  - How does the event data get from one function to the next?
    - Message Bus
    - Serverless/FAAS
  - Do we need a client?
  - What endpoints do we support (Azure IoT, Google IoT, MQTTS, HTTP/S, …)
  - Do we need an SDK to help arrange/order the functions, draw from existing functions, etc?
    - How do we create the executable(s)?
    - How are multiple application services seen by and used by EdgeX? Pub-sub from Core Data?

D – Device Services/SDK

- Device provisioning
  - Use cases
    - How does a new device get provisioned by the central management system (top/down approach)? Who else has to know when this happens?
    - How does a new device get provisioned when discovered by the device service (bottom up approach)? Who else has to know when this happens?
    - How does a device get provisioned securely (and what does that mean)?
  - Who owns / manages the device metadata?
    - Who/what populates the device metadata to Metadata service?
- Callbacks
  - What happens when a device or device service makes or detects change with regard to device?
  - What happens when a central management system or external system wants to make a change to a device?
- What additional features need to go into the SDK(s) that are not there today? What is scope for the next few releases?

DS SDK

- Metadata is the source of device knowledge
- Deleting a device in metadata requires removing all associated metadata info (provision watcher, etc.)
- Need blacklist manager in metadata
  - Need to add device to blacklist when deleted so it does not get automatically re-added
- Need to consider abstraction around provisioning
  - For possible separate service or external-to-device service implementation
- Need to consider unique identifier for devices (and other EdgeX objects)

E - Test/QA

- Performance / Scale testing
  - What do we need to know about performance of EdgeX?
    - Service performance
      - Amount of memory & CPU each service uses/needs (average, peak)
      - Start up time for the service
      - How big (on disk) is the service (or service and its container)
      - How quickly does the service perform its duties? Response time of particular APIs
      - How much load can the service take? Varies per service, but as example – how many log entries/second are required to cause logging to fail?
    - System performance
      - Overall RAM and CPU needs for all services (nice to have – same with or without specific services)
      - Overall storage and disk needs
      - Latency and throughput of data collection to export
      - Latency and throughput of data collection to analysis to device actuation
    - Device service performance
      - How many devices can be attached to each device service (and EdgeX as a whole) before the device service fails
      - DS throughput (number of events/readings per/s pushed to Core Data)
  - How do we determine these things?
    - What tools do we need?
    - What do we need to add to our CI/CD?
    - What instrumentation do we need in the code (or services like the database)?
    - How can we automate the collection of these statistics?
- Configuration testing
  - Blackbox tests today test under default Docker configuration profile?
  - How can we (automate) test all the profiles?
  - How can we test all the configuration options per profile?
  - How can we make blackbox test more loosely coupled? Test for Docker, Snaps, etc.

Use Cases

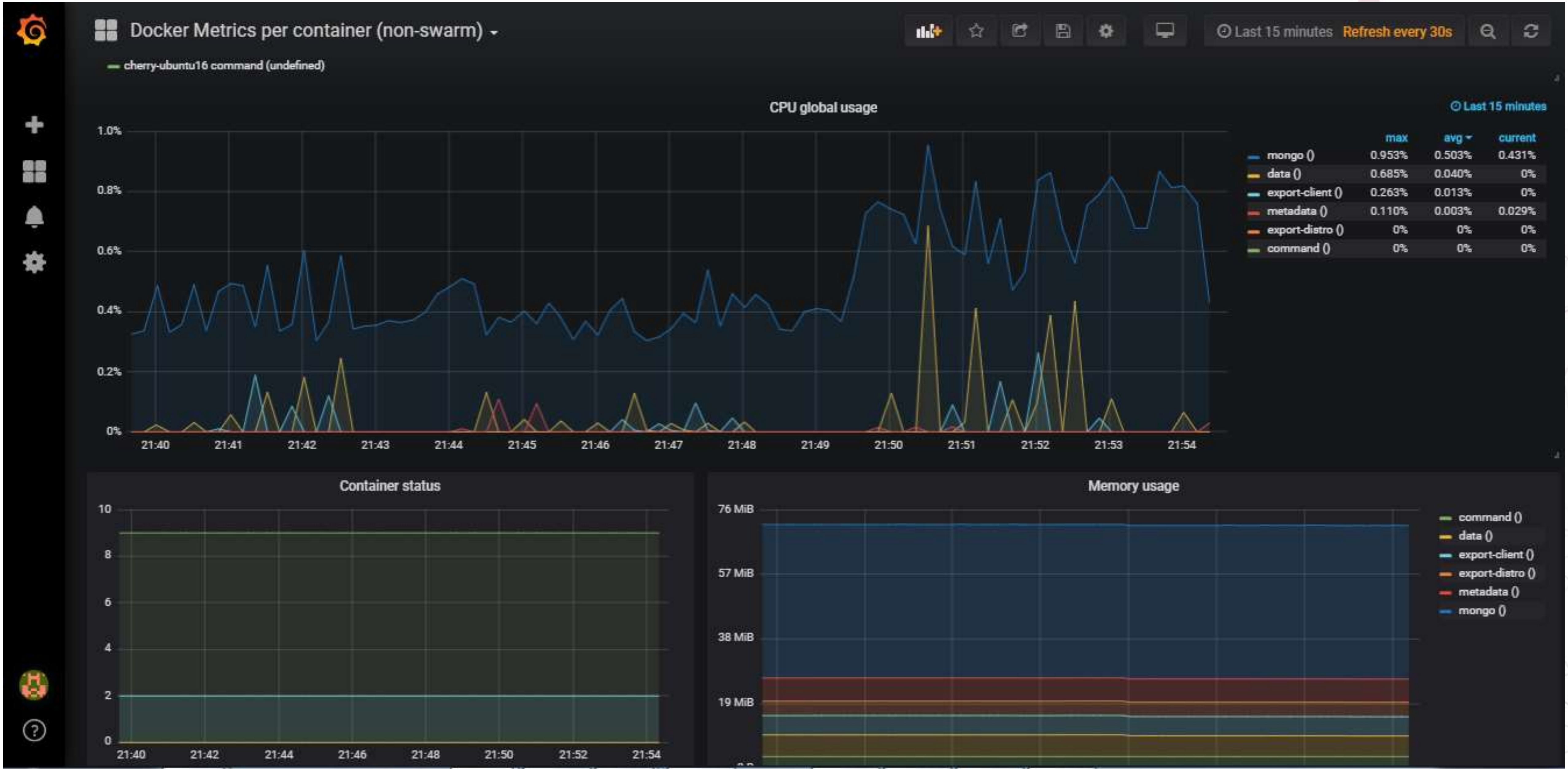
- Service performance
  - Amount of memory & CPU each service uses/needs (average, peak)
  - Start up time for the service
  - How big (on disk) is the service (or service and its container)
  - How quickly does the service perform its duties? Response time of particular APIs
  - How much load can the service take? Varies per service, but as example – how many log entries/second are required to cause logging to fail?
- System performance
  - Overall RAM and CPU needs for all services (nice to have – same with or without specific services)
  - Overall storage and disk needs
  - Latency and throughput of data collection to export
  - Latency and throughput of data collection to analysis to device actuation
- Device service performance
  - How many devices can be attached to each device service (and EdgeX as a whole) before the device service fails
  - DS throughput (number of events/readings per/s pushed to Core Data)

- Service performance (Tooling Required)
  - Monitoring
  - Instrumentation (write to log file)
  - OS and container utilities (e.g. size(), docker stats
  - API load testing
  - API load testing
- System performance (Tooling Required)
  - Monitoring
  - API load testing to write to Core Data and populate DB
  - Virtual Device + Virtual Export Service
  - Virtual Device (read/write) with instrumentation
- Device service performance (Tooling Required)
  - Scripts/setup application to configure/add devices (to virtual or specific device services) and setup scheduling + Virtual Export Service + dockerized simulator for each Device Service (e.g. Modbus)

Performance Testing Tools

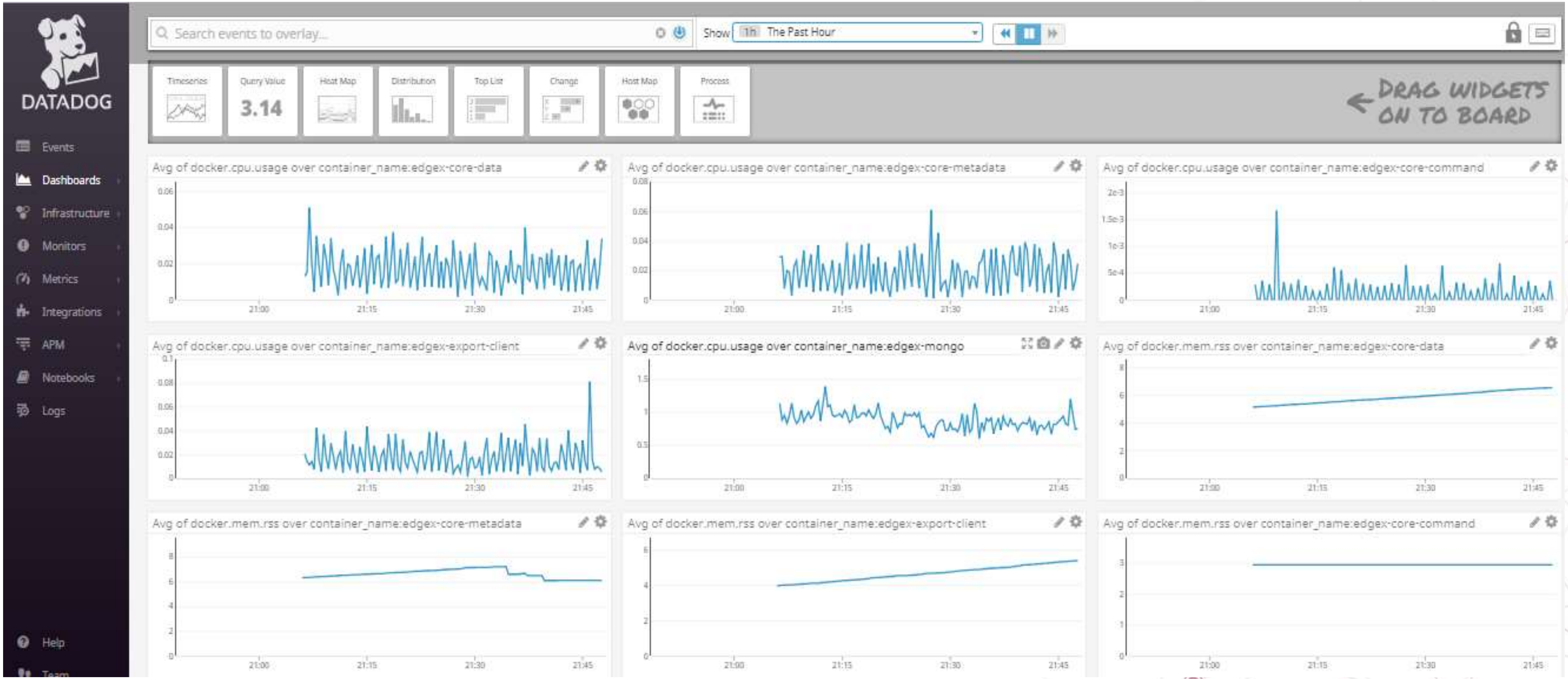
- API/Load Testing
  - Jmeter - load and performance test many different applications/server/protocol types
  - Load Impact – load testing tool that can be used with existing Postman REST API test collections
  - Bender – go-based API load testing tool, can be used with different applications/server/protocol types
- Monitoring & Reporting
  - Telegraf agent for collecting and reporting metrics, output plugins to send metrics to a variety of other datastores, services, and message queues, including InfluxDB (Influxdb + Grafana) also works with Graphite, OpenTSDB, Librato
  - Datadog – commercial monitoring tool, with many excellent features such as real-time dashboards and alerts

Telegraf + Influxdb + Grafana



	max	avg	current
mongo ()	0.953%	0.503%	0.431%
data ()	0.685%	0.040%	0%
export-client ()	0.263%	0.013%	0%
metadata ()	0.110%	0.003%	0.029%
export-distro ()	0%	0%	0%
command ()	0%	0%	0%

DataDog



Search events to overlay...

Show 1h The Past Hour

DRAG WIDGETS ON TO BOARD

DATADOG
Events
Dashboards
Infrastructure
Monitors
Metrics
Integrations
APM
Notebooks
Logs
Help
Team
Timeseries
Query Value
3.14
Heat Map
Distribution
Top List
Change
Host Map
Process
Avg of docker.cpu.usage over container_name:edgex-core-data
Avg of docker.cpu.usage over container_name:edgex-core-metadata
Avg of docker.cpu.usage over container_name:edgex-core-command
Avg of docker.cpu.usage over container_name:edgex-export-client
Avg of docker.cpu.usage over container_name:edgex-mongo
Avg of docker.mem.rss over container_name:edgex-core-data
Avg of docker.mem.rss over container_name:edgex-core-metadata
Avg of docker.mem.rss over container_name:edgex-export-client
Avg of docker.mem.rss over container_name:edgex-core-command
21:00
21:15
21:30
21:45

Monitor Tool

	DataDog	Telegraf + Influxdb + Grafana
Cost	**$15** Per host, per month*	Free
Run Env.	Cloud	Local
Use	Easy	Easy
Setup	Easy	Complex
Jmeter	Not supported	Supported

Detail comparison, please reference url https://www.g2crowd.com/compare/datadog-vs-grafana

Performance Testing Tools

- Simulation
  - Dockerized Device Simulators (e.g. Modbus, OPC UA, BACnet etc.) required for Device Service testing (functional and performance)

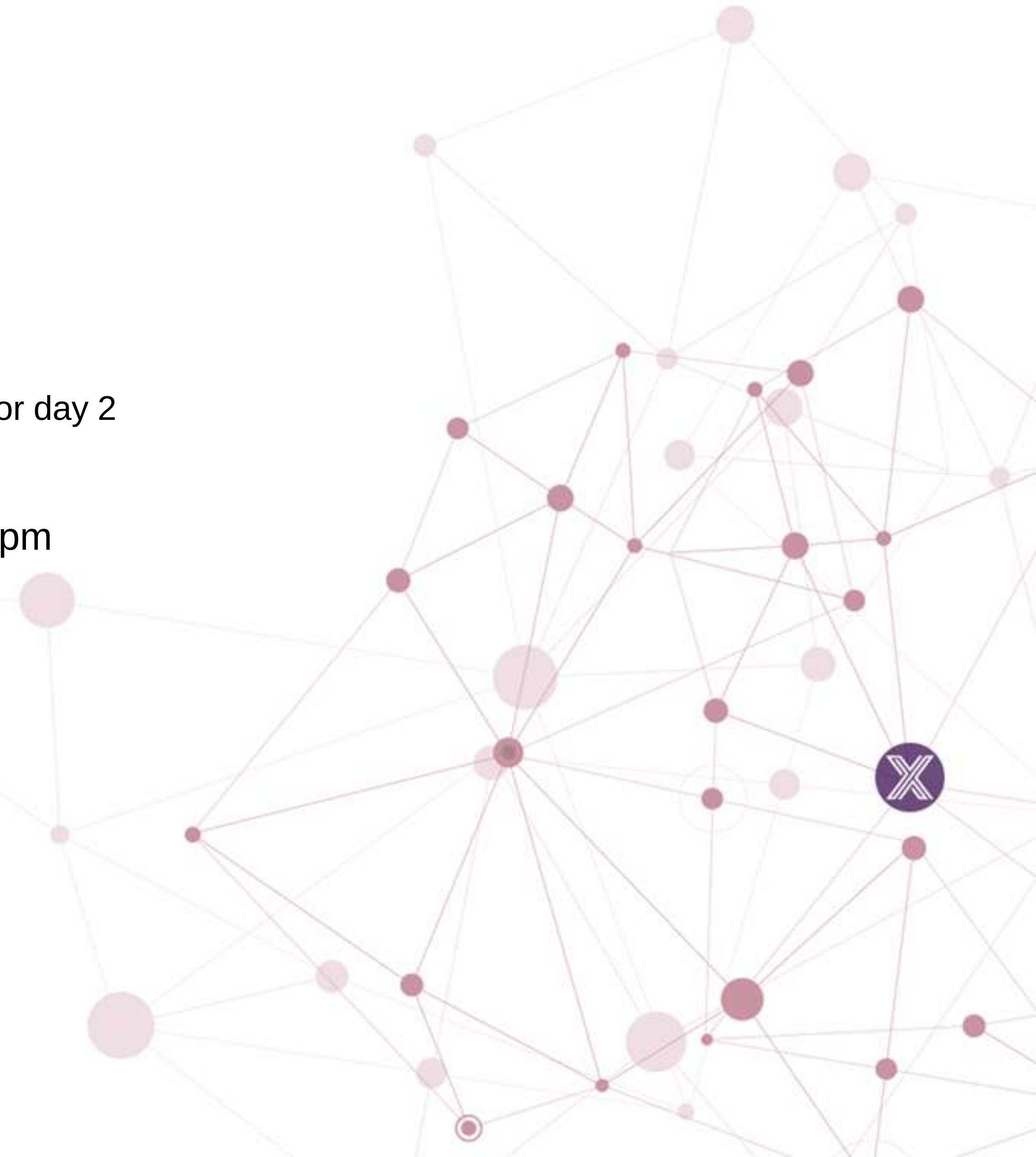
EDGEXFOUNDRY™

EdgeX TSC 2-day F2F

General Meeting

Edinburgh F2F Meeting

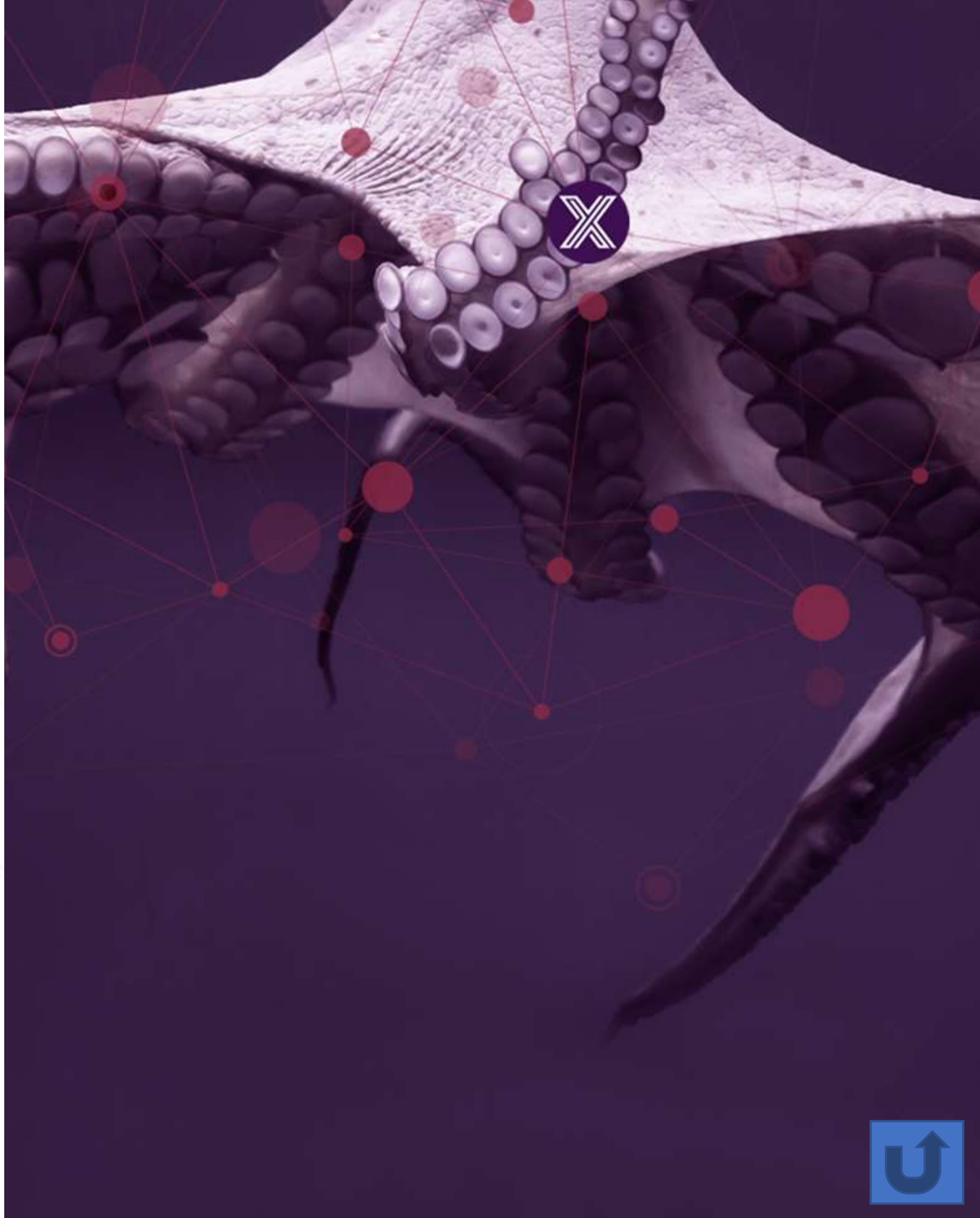
- Agenda
  - Day 1 – Edinburgh planning day
    - Welcome and intro by Keith Steele (IoTech): 9-9:30am
    - Architecture issues tee-up: 9:30-11am
      - Review and explanation of upcoming items of discussion for day 2
    - Edinburgh Planning – what's in/out: 11-2:30pm
      - Scope definition
   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



EDGEXFOUNDRY™

Architecture Issues Tee Up

9:30-11am Day 1



Architecture Issues

- Divided into 2 rough categories
  - Things we need to fix or address architecturally from an existing code base
  - Things we want to add (new features, functions, technology, etc.)
- Lists are further prioritized
  - What we think we want to address/cover in Edinburgh
  - What we think we want to address in future releases (roadmap items)
    - Covered time permitting

Architecture Issues Tee-up – Technical Debt

Edinburgh (Priority Topics)

1) Database Abstraction/Alternatives
   - Plugin architecture
   - Reference implementation (Mongo v. Redis)
   - EOL drivers
2) Device services and SDK in mono repo
3) Use of a Go framework
   - To improve isolation, help instrument new features (logging, authorization, tracing, etc.)
   - Pros/Cons
   - Costs & length of time needed to convert
   - Is now the time? Blocks what other development?
   - Options
     - GoKit
     - Macaron
     - https://go.libhunt.com/kit-alternatives

Fuji or later (Secondary Topics)

10) Windows developer support
    - The 0MQ issue
11) ARM 32 support
12) Device Hierarchy
    - Supporting devices owning / managing other devices
    - Mesh networks of devices
13) API Documentation
    - Automate generation of API documentation (RAML)
    - Replacement of RAML or alternate to RAML (Swagger)

Architecture Issues Tee-up - Enhancements

Edinburgh (Priority Topics)

4) Upgrades?
   - Move to Go 1.11
   - Consul to 1.2.3
   - MongoDB to 4.0
   - Vault or Kong?

5) Versioning
   - Modules / vgo

6) Tracing
   - Following an event or API request through all services to allow better debugging and support

7) Automated security testing

8) Improved resiliency
   - What do we need to do next? Services now are more resilient to timing issues

9) Support for distribution
   - What do we need to do to better support truly distributed EdgeX

Fuji or later (Secondary Topics)

14) Config and Metadata changes
    - Using callback/watchers

15) Alternate deployment / orchestration
    - ex: adding Kubernetes support

16) Facilitating commands from the North
    - How to supply command information to the north side systems
    - Ex: how to give Azure IoT the ability to command devices?

17) Artifact signing
    - Exe/JAR/etc. artifacts
    - Docker containers

18) Downsampling @ device service level

19) Command Parameter Check
    - Min/Max values (or other checks) on command parameters

20) Message infrastructure
    - Time to consider message bus intercommunications between more micro services.

21) Configuration versioning

Carry over issues from Architect’s Day

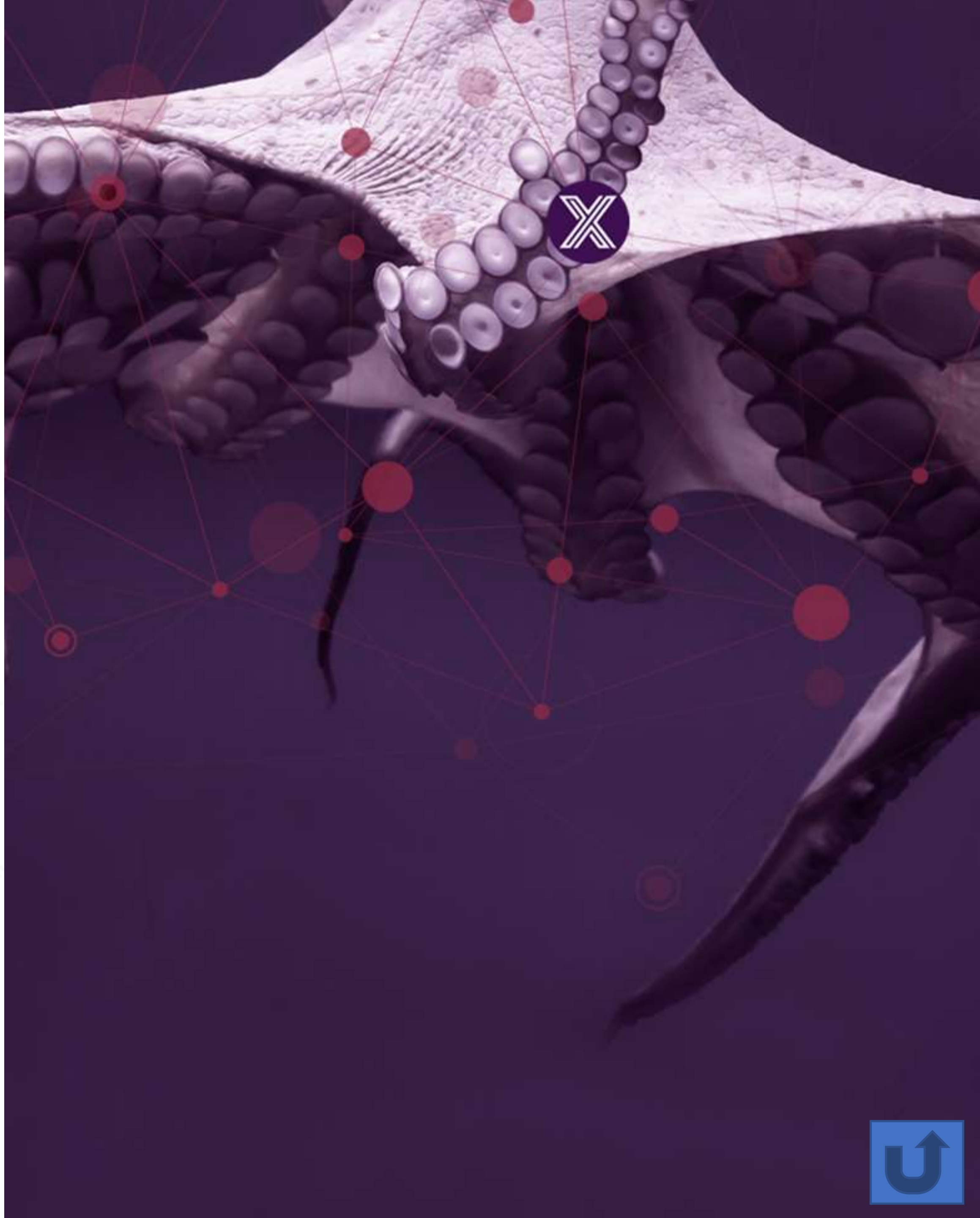
- System management issues
  - Scoping
- Security issues
  - HWRoT
  - System management/Security overlap
- Application Services
- Device Services and SDK
  - Device provisioning
  - Callbacks
- Test/QA
  - Performance/Scale Testing
  - Configuration Testing

COMING SOON!

This area will get filled in after Architect’s Day on 22nd

EDGEX FOUNDRY™

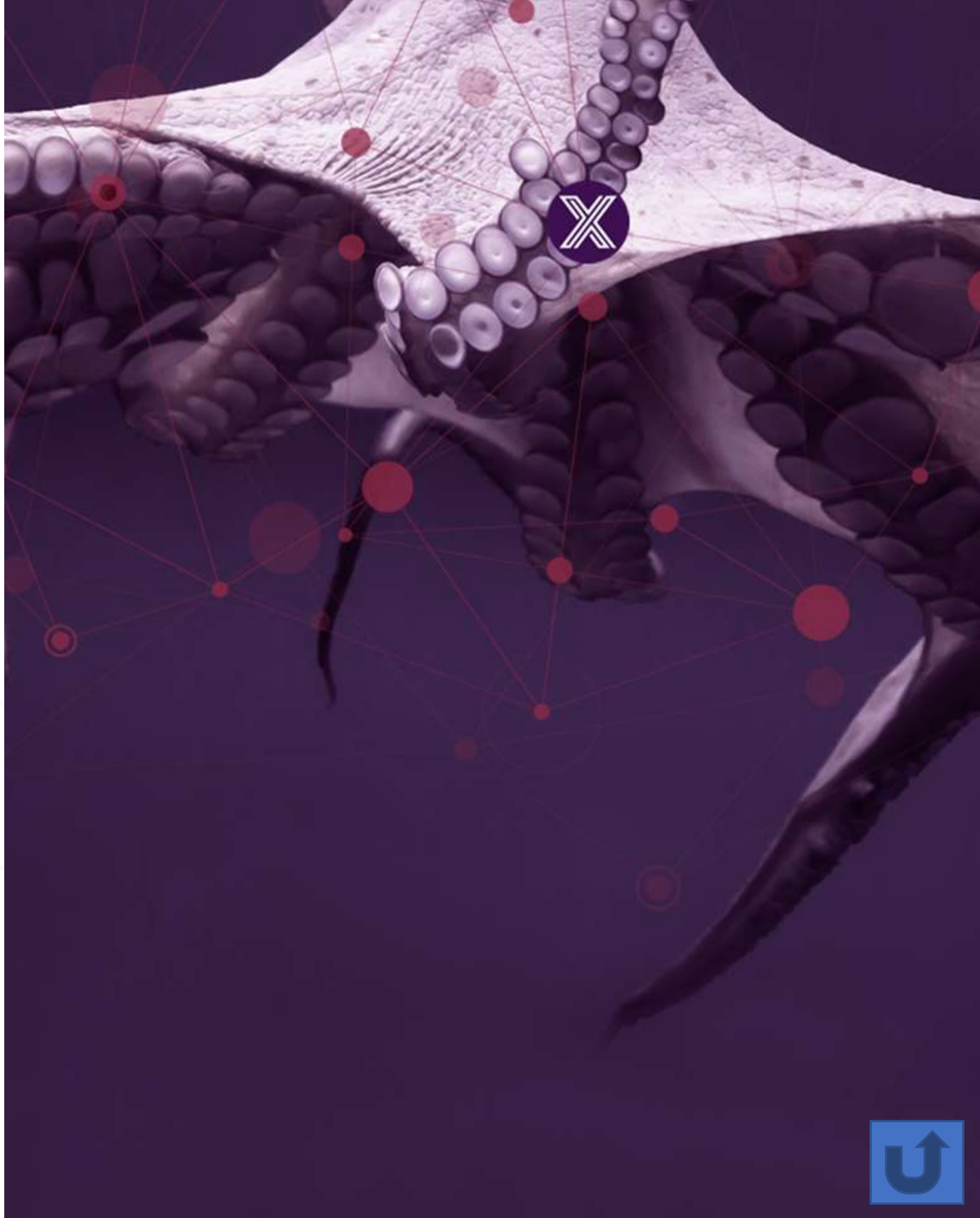
EdgeX Release Cadence



Cadence Check

- April & Oct remain target release months
  - Edinburgh – April 2019
  - Fuji – Oct 2019
  - Geneva – April 2020
  - “H” release – Oct 2020
- F2F planning around time of completion of each release
  - Korea – April 2019
  - ??? – Oct 2019 (venue to be selected – nominations??)
- 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)

EDGEX FOUNDRY™



Edinburgh Planning

What is in/out of scope for Edinburgh release?

11-2:30pm Day 1

Edinburgh Planning

- Considerations
  - Full release cycle this time – 6 months
  - Intel plans to provide many developers and other contributors to assist on various aspects of EdgeX
  - We have other new members looking to contribute as well
  - Appetites and needs still need to be carefully considered
    - We need to recognize that new members are going to have some of their own needs
- Addressing technical debt…
  - We need to strike a balance
  - Refactoring alone will not satisfy the needs of our use cases
  - Not addressing the worst of our past sins will result in a product that does not scale or work long term
- The IN/OUT scoping discussions were setup under the premise that resource contributions were going to be the same
  - New organizations joining EdgeX or others contributing more may negate this premise

Edinburgh Scope

- Major themes
  - Certification program
  - Binary data support (CBOR)
  - Export Services -> Application Services
  - Database abstraction/replacement(s)
  - Device Services galore
- Other efforts
  - Initial Performance testing
  - HW Root of Trust roadmap/document

- Healthy debate required here!
- We need to be careful of too much scope
- We need to address some technical debt but also start to add some new features

Edinburgh Planning – General (or cross area)

In

- Support binary data with CBOR
  - DS -> Core Data -> Export Distro (& Appl Services)
  - Intel has lessons learned and CV framework worth exploring
  - Give to Core WG to lead design/impl
- Versioning with modules/vgo
  - How do we version multiple micro services from a single repo
- Improving the developer experience
  - From 0 to real IoT edge solution with little effort

Out

- ARM 32 support
- Windows development support
  - 0MQ issue
- East/West support
  - Failover
  - RAS
  - Intel's dynamically allocates workloads to clustered hardware
- Tracing for performance and system management
- Use of GoKit for all Go Services
  - Provide better layers of abstraction & separation of concerns
  - Allow for better instrumentation/tracing
  - Continue to explore technology
  - Isolate the business logic in services to better support a transition in the future
    - Support/encourage more unit testing to help drive this
  - Allow working groups to explore use as they see fit
    - Requires TSC support to use in ultimate code base

Edinburgh Planning – Export (Application WG)

In

- Application Services first iteration (crawl)
  - The future replacement of export services
  - Potential use of plugin architecture
  - Exploration of technologies like GoKit & FAAS
- Lightweight rules engine option or other “edge analytics”
  - Drools
  - NodeRed

Out

- Additional Northbound endpoints - Fuji
  - IBM Watson
  - IoTivity
  - DDS
  - AMQP
- Intel’s OpenVino integration for CV analytics at DS level
  - TensorFlow, Caffe, Etc
- Support additional northbound formats
  - Haystack
- Integrate to edge software/agents
  - AWS Greengrass
  - Microsoft IoT Edge

Edinburgh Planning – Core (and Supporting)

In

- Better database abstraction architecture
  - Core data, Metadata, Exports, Notifications, Logging
  - Removing the BSON
  - Hiding domain IDs
  - Implementation of Core Services Using Redis
  - Certification/marketplace for alternatives
- Scheduling service rework
  - Owner vs where exercised

Out

- Replace driver
  - The new MongoDB driver is not fit for use
  - MongoDB may not be used long term so this may be OBE
- 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
- Logging service rework
  - Intel has developed a Logging/Telemetry as a Service – Replace/Augmenting service
- Support for alternate logging format
  - XML & CSV in addition to JSON
- Intel has done work in alerting and may have some suggestions for additions/improvements

Edinburgh Planning – DS & SDK

In

- Management of Devices
  - Metadata is the source of truth for device info
  - Device delete requires delete of all associated data (provision watcher, etc)
  - Device deleted gets added to Metadata blacklist so it is not re-added
- 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
  - Cache of readings
  - Incorporate SMA features in SDK and DS

Out

- Additional Device Services – Fuji
  - Some of these may be in Edinburgh at contributors discretion
  - 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
- Intel Camera device integration (stretch for Edinburgh at Intel discretion)
  - Camera Discovery Components (SAF Bus)
- Intel RFID device services (stretch for Edinburgh at Intel discretion)

EDGE X FOUNDRY™

Edinburgh Planning – System Management

In

- Add Metrics
  - CPU
  - Others as system wide tracing and service to service collection permit
- SMA Translation layer (stretch)
  - Pick one protocol to start (LWM2M)

Out

- Storing metrics collected locally - Fuji
- Setting configuration - Fuji
- Callbacks (alert on changes to config/metric) - Fuji
- SMA translations to other protocols
  - Redfish
  - OMADM
- Actuation based on metric change
  - “rules engine” for control plane data
- Consider use of QoS and blockchain to prioritize resource usage by certain services.
- ~~Software updates/deployments~~
- ~~OOB support~~
  - ~~Intel’s Vpro Manageability project~~

Sys Mgmt Conclusions

- Out of scope of EdgeX
  - Orchestration of services
  - Install/update of software, firmware, OS, intra-structure (networks, etc.)
- SMA is single point of management for EdgeX deployment
  - 1 - SMA per deployment (could mean multiple copies of EdgeX services like core data)
  - We'll have to consider this with regard to service implementation
  - "It's a deployment of services not multiple EdgeX instances across the fog"
- Stop – done with option to call service directly or manifest specified "stop function"
- Config Set – done to Config service
  - Config service must persist change
  - Option to repopulate from config seed
  - Set properties by naming conf – but user bewaire
- Detect / health check is proxy to config/reg service
  - Allows one to restrict access to consul when necessary

Edinburgh Release Feature Candidates

- IN
  - Delhi Rollover
    - Testing Automation
    - Possibly Vault namespaces
  - Secure Storage abstraction layer – code
    - Includes a sample that talks to TPM simulator
  - Encrypt initial secrets on file system using the secure storage abstraction layer
  - Begin secrets migration in to vault – at least one service
    - Includes specification for shared code to use vault
  - Service to Service AuthN/AuthZ requirements and design

Edinburgh Planning - Security

- 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

Edinburgh Test/QA

- IN
  1. Performance Testing
     1. First step better Visualization/Dashboarding
        - For existing blackbox tests to start with
        - Support historical test result data
        - Candidate tools: Telegraf+Granfana+InfluxDB, DataDog, Prometheus
     2. Capture resource metrics
        - Memory, CPU consumption etc.
        - Candidate tools (as above)
     3. Performance/Load testing
        - Can be used to drive EdgeX APIs in different load scenarios
        - Candidate tools: Bender, Jmeter, Load Impact (can be used with existing Postman tests)
  2. Security Testing – implementation driven by Security WG

- OUT (or at least a stretch)
  - Tracing
    - During testing, configuration
    - Candidate tools/technology based on OpenTracing standard: Zipkin, Jaeger
  - Service instrumentation
    - To support metrics such as Service startup times
    - Need to consider EdgeX performance with Security enabled
  - Need to consider EdgeX performance with Security enabled
  - Configuration testing
    - As of Delhi we have the ability to change EdgeX configurations dynamically
    - Existing testing uses a single static configuration
    - Need to identify and add additional testing configurations to automated blackbox testing
  - Data import
    - DB agnostic
    - Scenario independent
    - Test chaining
  - Automated system level latency and throughput testing (e.g. device read to export or device read to analytics to device actuation)
  - Automated Device Service testing (functional and performance for each Device Service e.g. Modbus DS)

- Other issues/considerations:
  - Performance Testing
    - Should we baseline performance of service binaries (no container)
  - General
    - Should we run blackbox tests against service binaries independent of container technology
    - Or, additional blackbox test runs against other container technologies supported (e.g. snaps)

Edinburgh Planning – DevOps

In

- On-board a new chairperson
- Go 1.11
- Vgo/modules
- Support testing efforts

Out

- Alternate deployment/orchestration
  - Beyond Docker/Snaps
  - Kubernetes
  - Kata Containers
  - …
- Static code analysis
- Automatic go code formatting

Edinburgh Planning - Vertical Solutions

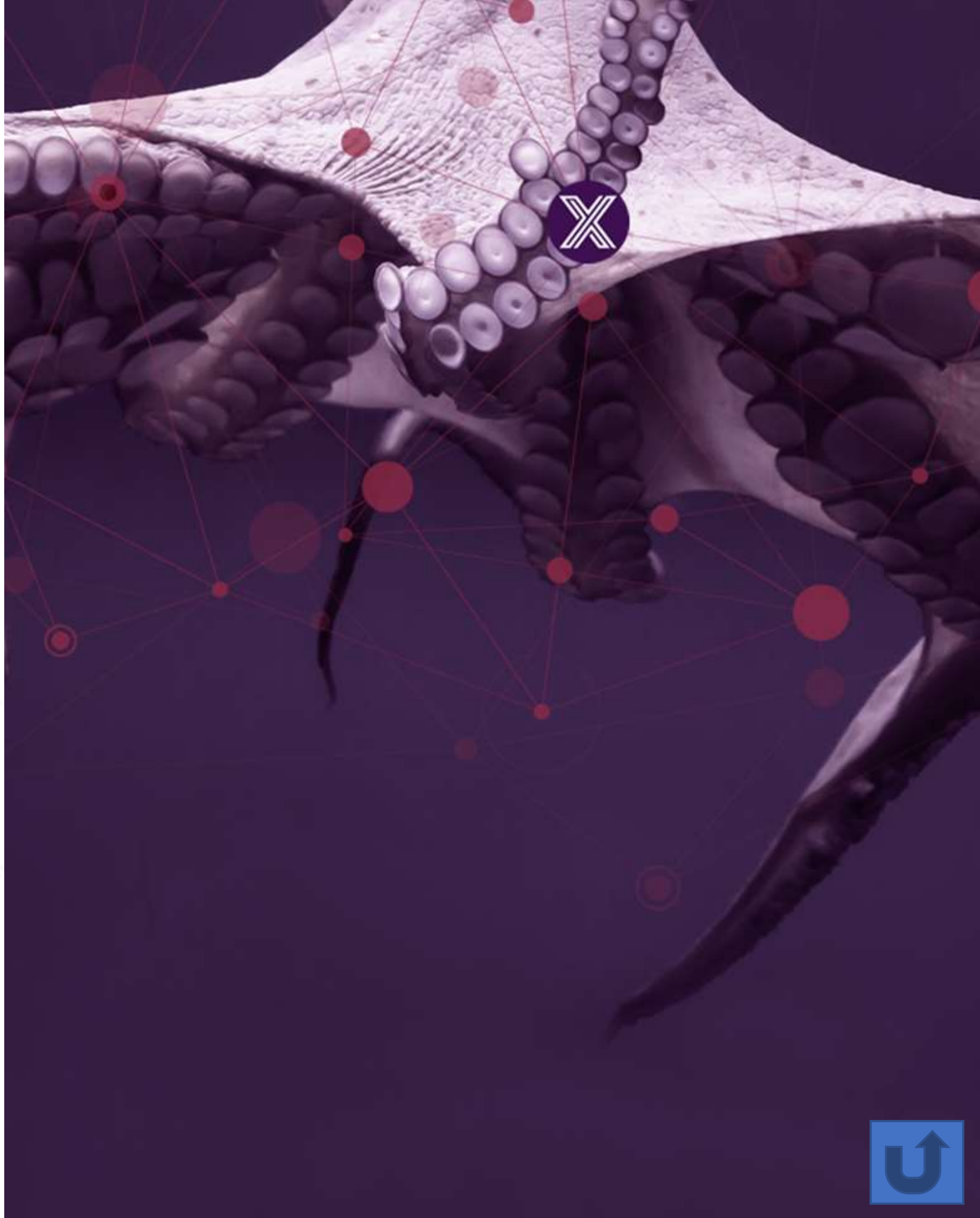
- Vote on retail vertical project group under Vertical Solutions WG
  - Approved unanimously
- Requirements definition by each project group
  - Oil/Gas, Smart Factory, Retail, …
- Gap analysis by each project group
  - What is needed by the vertical that is not provided by EdgeX today?
- Proposed EdgeX roadmap additions
  - High level architectural needs/changes
  - High level designs
  - Technological suggestions/input

EDGEXFOUNDRY™

Future Release Roadmaps

Fuji, Genvea, and beyond roadmapping

2:30-3:00pm Day 1



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

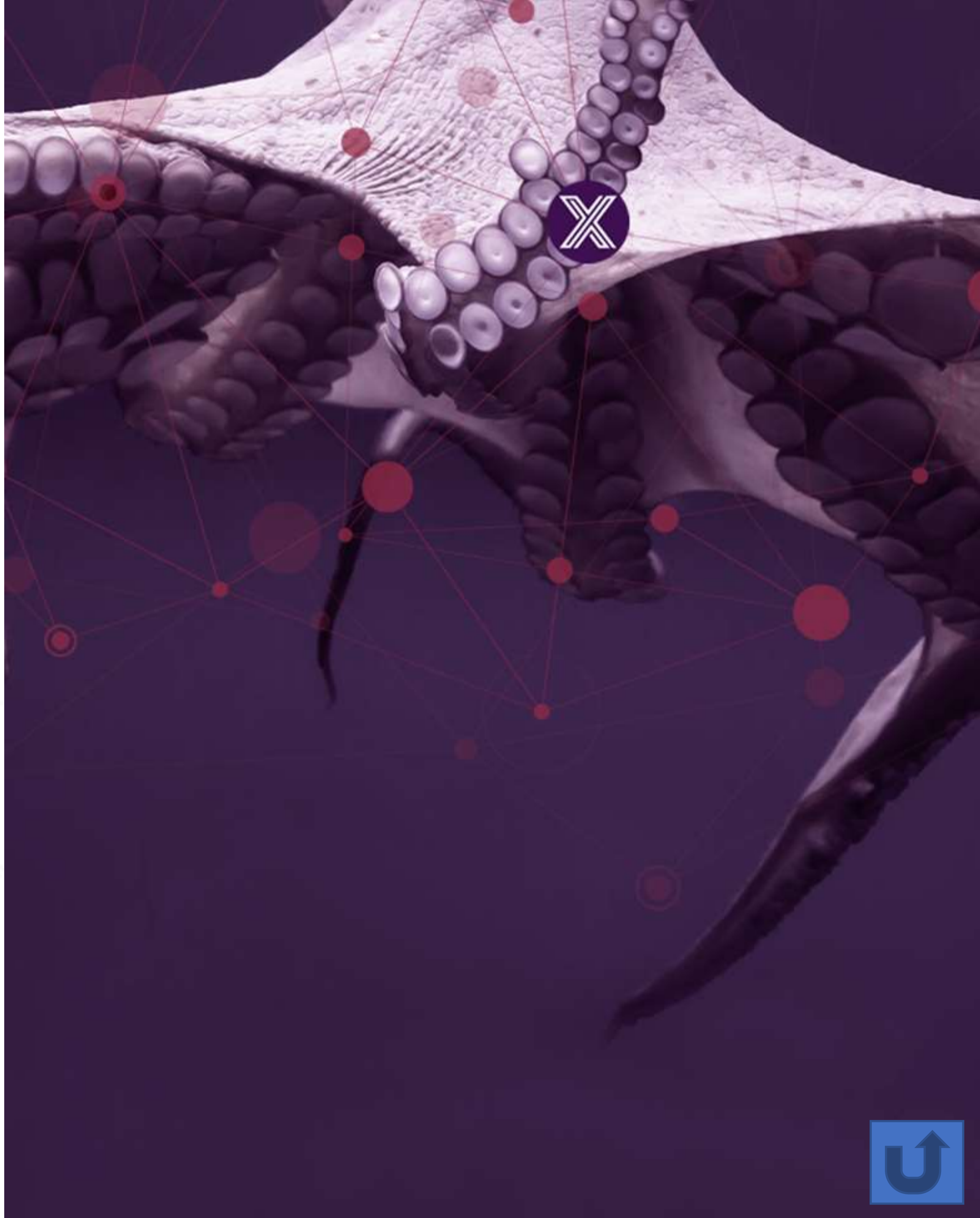
Geneva Planning (Apr 2020)

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

EDGE X FOUNDRY™

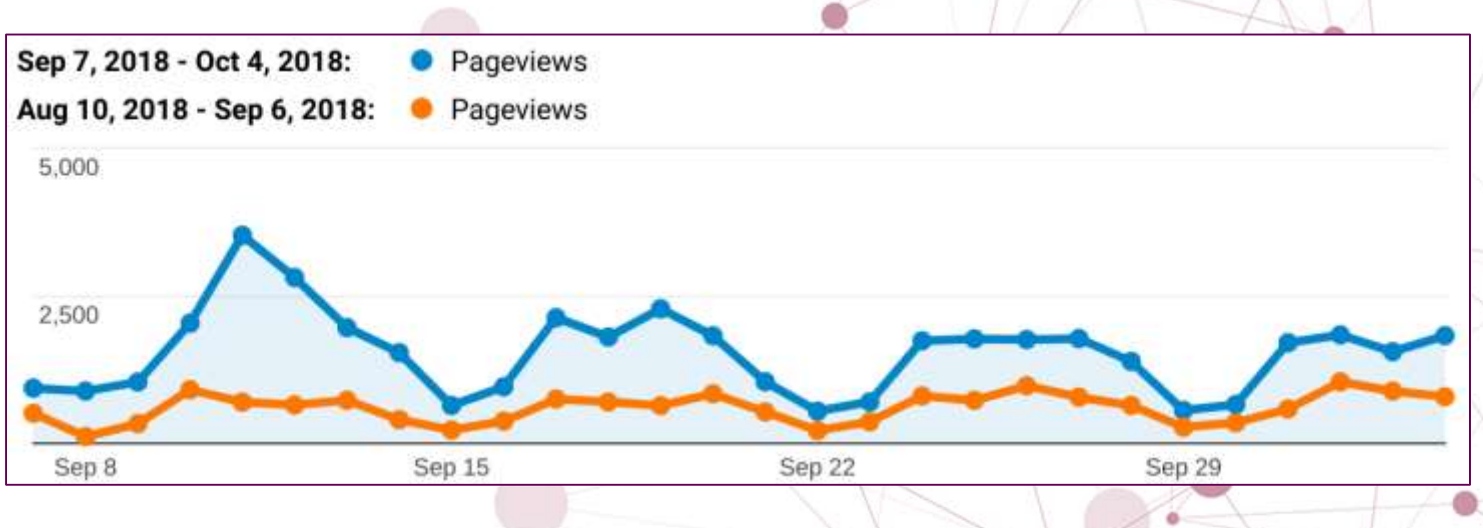
Developer Advocate Perspective

3:00-3:30pm Day 1

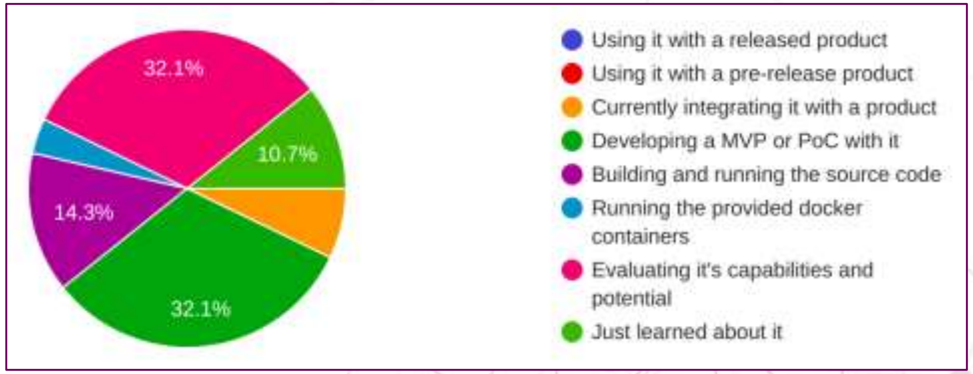


Developer Advocate – Accomplishments

- Website & Documentation
  - 80% increase in traffic to Getting Started
  - New documentation website with > 20,000 pageviews
  - Promoted API walk through has > 7,500 pageviews
  - Wiki cleanup
  - Analytics on everything*
- Raspberry Pi demo
  - Precursor to devkits
  - 4,000 views on Hackster.io



Developer Advocate – EdgeX Survey



- Everybody targets Linux
- Java and Python are relevant languages
- 38% are at the PoC or integration stage
- PoC interested in DS, evaluators interested in south bound
  - 44% interested in DS or South Bound
- 45% regularly use Rocket.Chat (75% of those building on EdgeX)
- Wiki, Docs & GitHub each account for about 1/3 of most used sites
- Engineers almost all learn about new tech from technology websites

EDGE X FOUNDRY™

Developer Advocate – Eclipse Survey

- Python is #3 for constrained devices, #2 for gateways, #3 for cloud
- Java is #4 for constrained devices, #1 for gateways, #1 for cloud
- Security is primary concern, encryption is #1, auth is #2, updates #3
  - Only 10% use secure boot/TPM/HRoT
- MQTT dominates, followed by HTTP and Websockets
- 85% have used open hardware, 50% use it at work, only 8% never
    - 30% Arduino IDE, #2 most used IDE

Developer Advocate – Onboarding

- Discovery
  - Proactively seek articles in news sites
  - Increase speaker presence at tech events
- Documentation
  - More examples, tutorials & walk-throughs
  - Make better use of the Wiki
- Communication
  - Rocket.Chat is underutilized
  - Need a Forum and/or Q&A site
- Promotion
  - Social Media
  - Release Notes

Developer Advocate – Events

- Meetups 2,300 groups with > 1 million members
  - Target Architects and Researchers
  - Provide presentations & an example video of them
  - Provide workshop with setup, steps and FAQ
  - Send a Community Kit or Demo
  - Need member companies to encourage their employees to participate
- Conferences
  - Extend Speaker's Bureau to cover conference travel
    - ~$5,000 USD will cover 3-5 conferences
  - Provide presentations & an example video of them

Developer Advocate – Device Services

- Resources:
  - Provide Device Service project templates for popular IDEs
  - Write step-by-step walk-throughs for each language
- SDK:
  - Maintain the Java SDK
  - Potentially add a Python SDK
- Platforms:
  - Ubuntu Core
  - Raspbian
  - Arduino

Developer Advocate – Demos

- For Meetups (~ $200 each)
  - Small gateway device (ARTIK/RPi3)
  - Battery-powered sensor board (Reel board)
  - Connect via WiFi or Bluetooth, live view of data on the projector
- For Conference booths (~ $100-$500)
  - Give away $0.99 programmable, WiFi capable boards
  - Develop an interactive contest involving those boards
  - Announce the winner by calling a command on the winning board
  - Have “take home” instructions for connecting to a local EdgeX instance

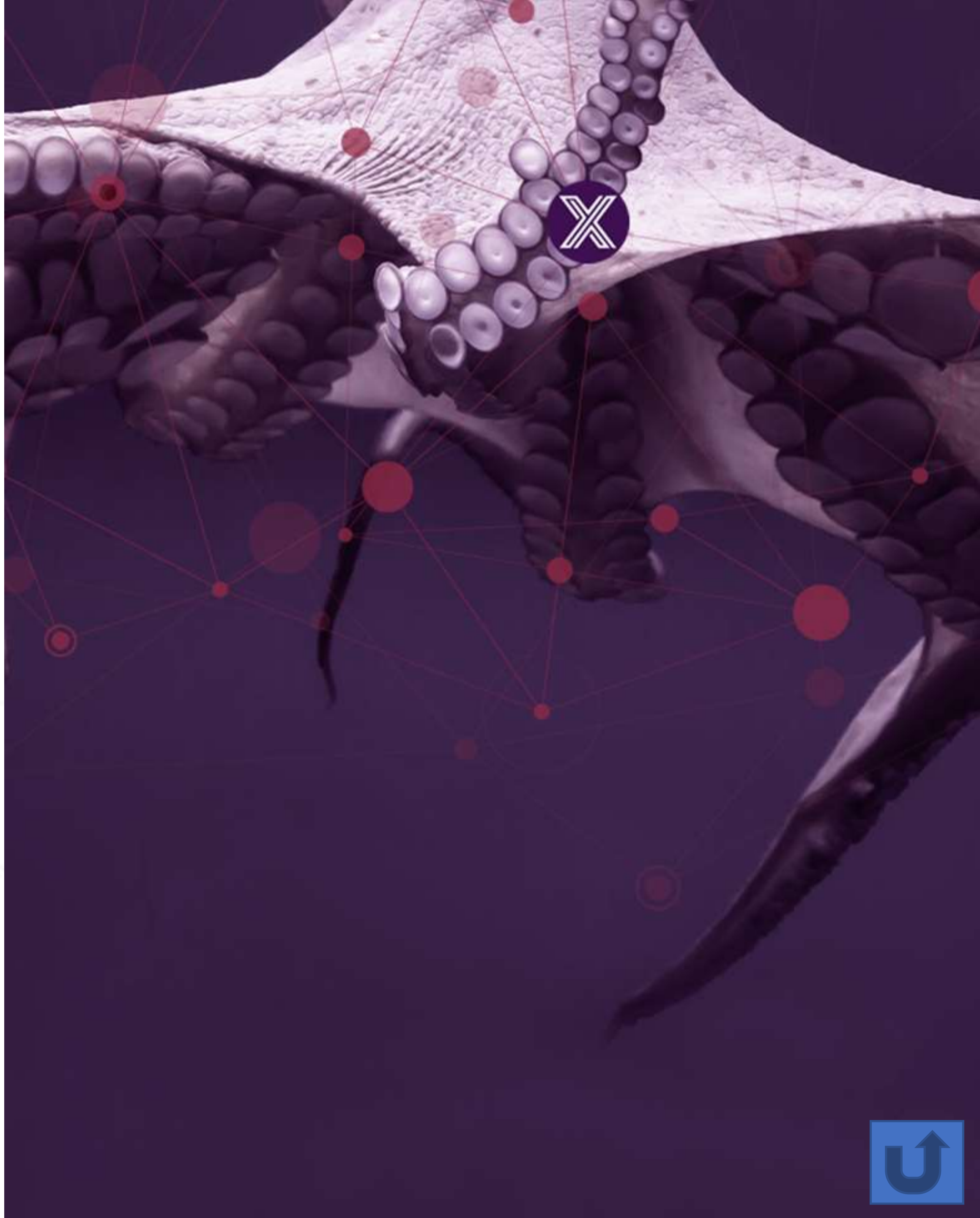
Release Manager

- Release Manager – volunteer position (interested party needed!! ☺)
  - Maintain the release process documentation
  - Freeze dates (what they mean)
  - Feature Freeze exception process
  - Branching/tagging process
  - Maintenance releases
  - Gather list of features/issues from each WG targeted to the release
  - Publish the release schedule (freeze dates, release dates, EOL, etc)
  - Send reminder announcements as freeze dates approach
  - Coordinate freeze-breaking changes with TSC and relevant WGs
  - Trigger branching/tagging of repos
  - Trigger publishing new stable release artifacts
  - Write & publish release notes & changelog
  - Provide regular updates on the TSC meeting on the progress of a release
- https://docs.google.com/document/d/1M84vy23wmfe0COwK7JxvCf8lEEmYxN6bswUq00NlcZc/edit

EDGEXFOUNDRY™

DevOps Changes

4-5pm Day 1



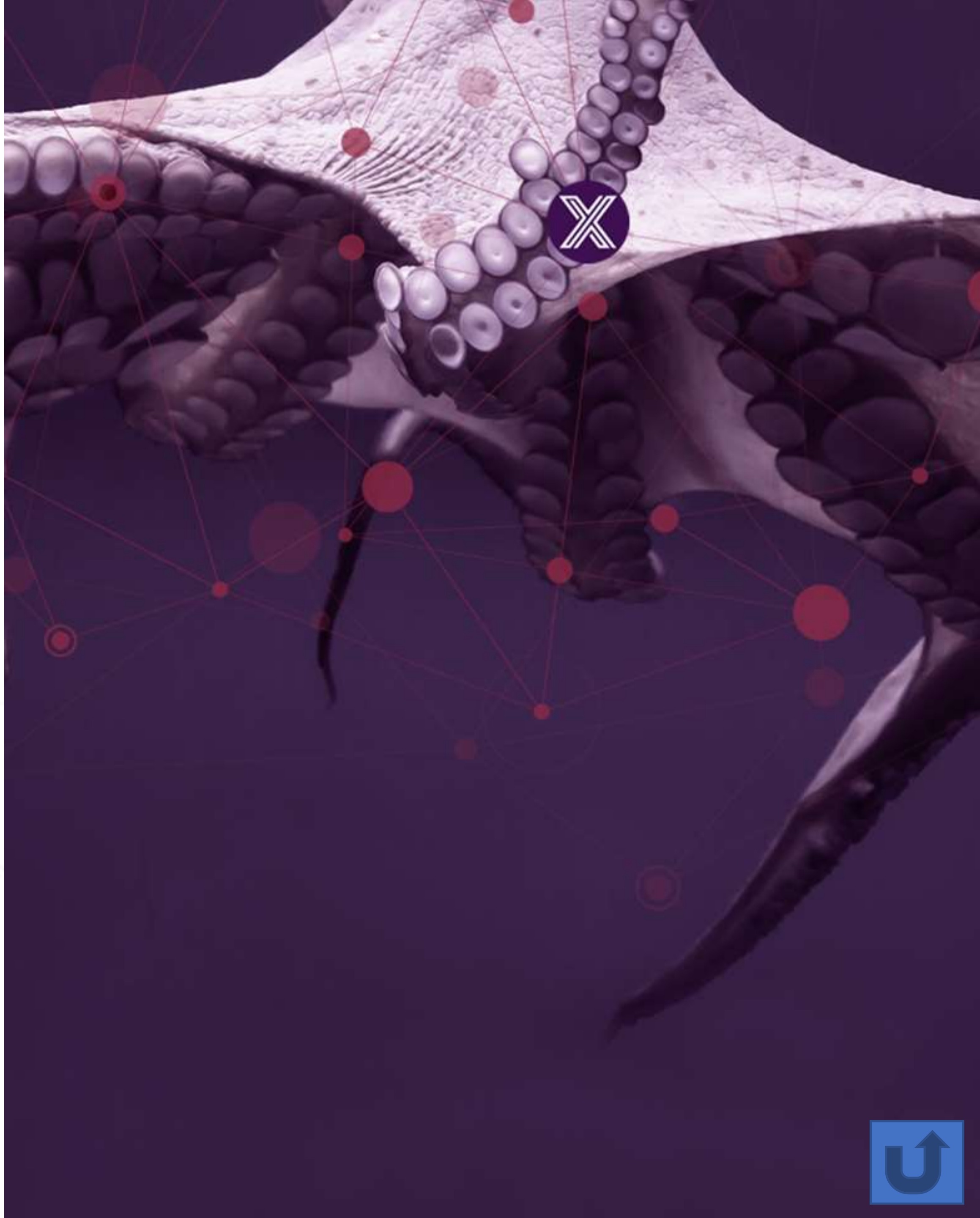
DevOps Issues

- Procedures for tracking release content
- Developer participation
- Need a better way of reporting results from our daily CI/CD jobs
- Tracking PR numbers, developers shown on Github, etc.
  - Automate the process
- Support policy
  - How long do we support a release?
  - Is there a separate policy for individual services some day?
  - What’s our official statement on this and where should it live in the documentation?
- DevOp Chair
  - Volunteers (elections if needed)
  - Options without WG chair?

EDGE X FOUNDRY™

Architectural Issues/Discussion

9:00-2:30 Day 2



1) 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
- Replace MongoDB driver with official driver
- Priority and sequencing of EdgeX model types from Mongo types
- Denormalization of persistence layer(less relational)

2) Device services and SDK in mono repo

- Where should the Go-based DS & SDK live
  - In Mono repo?
    - Advantage – more easily reference other EdgeX 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
    - Disadvantage – additional releases of the SDK can be done outside of the larger project cadence when the DS and SDK remain independent
    - Disadvantage – unless there is a way to use vgo/modules/git to handle, putting the SDK and DS in the mono repo ties them to the specific release of the other work
- Are there other repos we want to move to edgex-go?
- Are there items we want to break away from edgex-go to separate repo?

3) Use of a Go framework

- Why?
  - Better isolation of cross cutting concerns
  - Easier to hang new cross cutting concerns (logging, tracing, multi-tenancy, content negotiation, authorization, etc.)
  - Pro/Con discussion
    - Impact to other efforts
    - Refactor vs. new features
    - Impact to overall schedule
    - How to get done while other work ongoing
  - Options
    - GoKit
    - Macaron
    - https://go.libhunt.com/kit-alternatives

4) & 5) Upgrades and versioning

- Glide is deprecated
- Need (or is there??) to release services (or SDK or DS) independent from each other (all in edgex-go)
  - Do we wan independent versioning of core services from support services for example
- Which upgrades do we want to consider for the project this release?
  - Move to Go 1.11
  - Consul to 1.2.3
  - MongoDB to 4.0
  - Vault or Kong?
- Go 1.11 – when/what’s needed
  - CI/CD impact
- When to move from Glide to Modules and vgo
  - What are the considerations/tasks?
  - How to we apply it (crawl walk run)?

6) Tracing

- Suggested this is needed for better debugging, performance monitoring, and system support
  - Especially in distributed EdgeX world
- What is traced and how?
  - Conform to opentracing.io API specs.
- What can provide tracing?
  - The top two options are Zipkin and Jaeger.
  - GoKit is relevant in that it provides middleware to intercept the request pipeline, and through that we'd hook into a tracing solution.
- What is the impact to EdgeX code

7) Automated 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?

8) Improve Resiliency/Availability

- Per Palo Alto F2F, we wanted next couple of releases to improve resiliency and 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?
  - How would adding a load balancer around each service work/not work?

9) Support for Distribution

- What is needed to allow every/all EdgeX services and infrastructure operate on a different host?
  - If I moved a service today, what would happen? What changes would I need to make?
  - Can we move dynamically and would the system react? Do we want to be able to do that?
- What is needed to 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?

10) Windows Developer upport

- Full Windows development support - ZeroMQ libraries do not allow 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?

11) ARM 32 support

- Mongo latest release not supported on ARM 32
  - Requires older version
  - Redis supports ARM 32
- Are there other elements of EdgeX that are not 32bit compliant?
- Do we want / need another CI/CD process to support?
  - Who owns and manages?

12) 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

13) 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)?

14) 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

15) Alternative Deployment and Orchestration

- Apart from Docker/Docker Compose & Snappy deployment and orchestration options for EdgeX what do we want to explore?
- 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, Kata Containers to name a few.
  - Kubernetes
    - CNCF / Kubernetes IoT Edge WG
    - https://github.com/kubernetes/community/tree/master/wg-iot-edge
  - Intel working on Kubernetes / Helm –create organize and manage manifests for Kubernetes
- Should we only support a single reference implementation for demonstration and allow 3rd parties/marketplace address?
  - At what point does supporting these get to be a scaling problem, a CI/CD problem

16) 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?

17) Artifact Signing

- Today, artifacts (EXE, JAR, Docker image, etc.) are 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?

18) 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.

19) 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).

20) 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.
  - Indications from IoT experts is that REST will not support high scale/volume
- 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?

21) 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 vs. Consul config change?

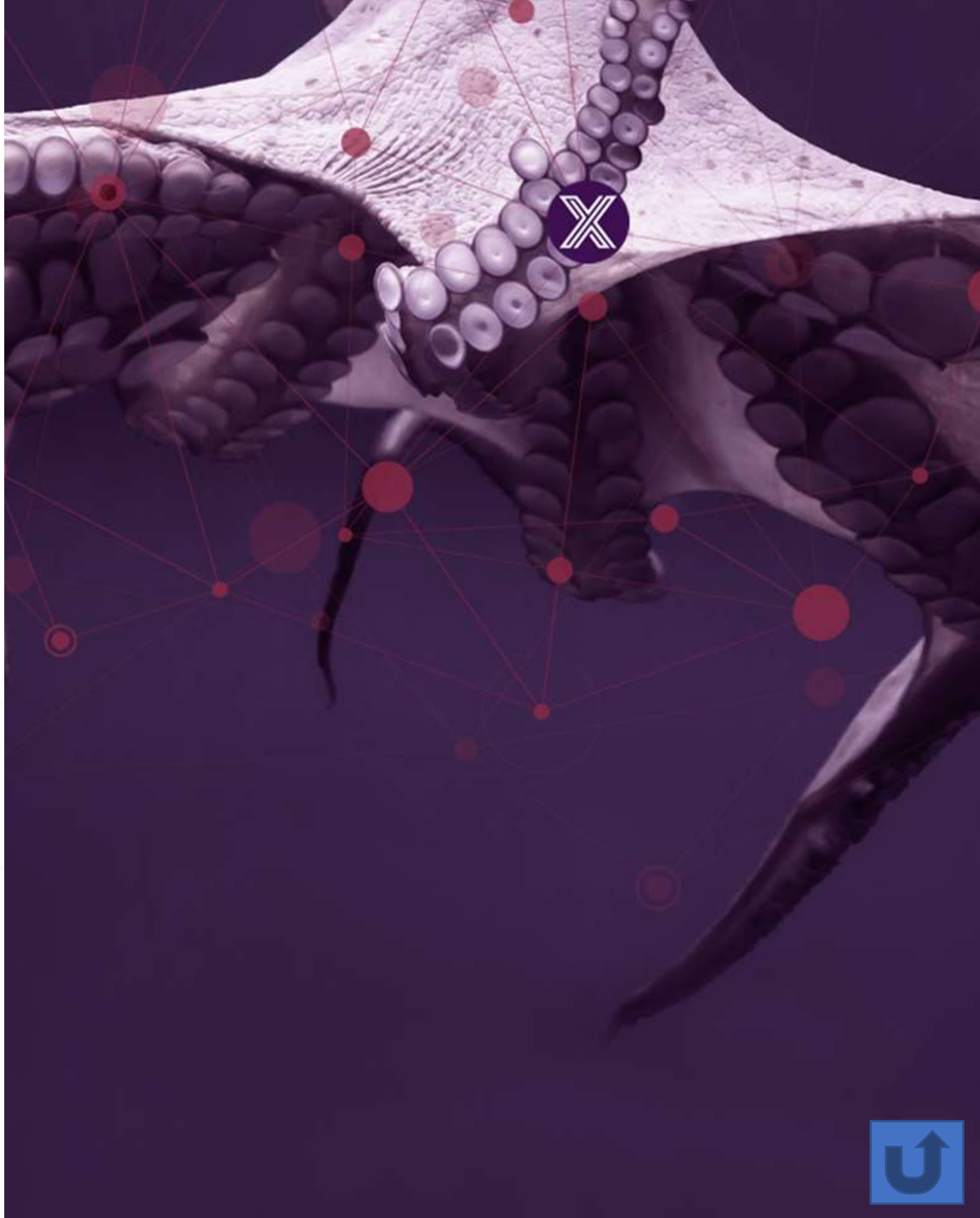
Review of Backlog

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

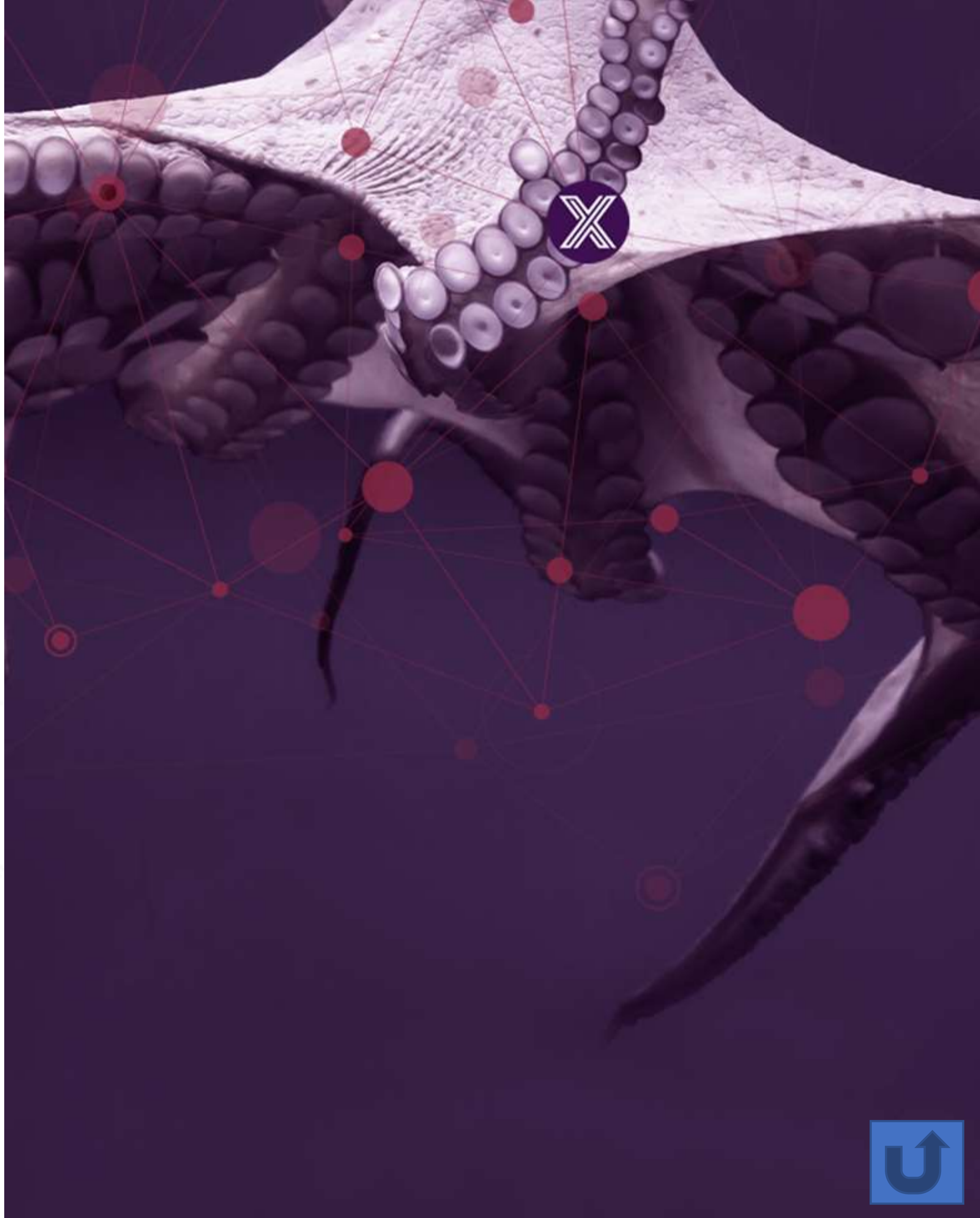
EDGE X FOUNDRY™

Action Item Review & Goodbyes

2:30-3pm Day 2



EDGE X FOUNDRY™



Business Issues

3-5pm Day 2

edgexfoundry.org | @edgexfoundry

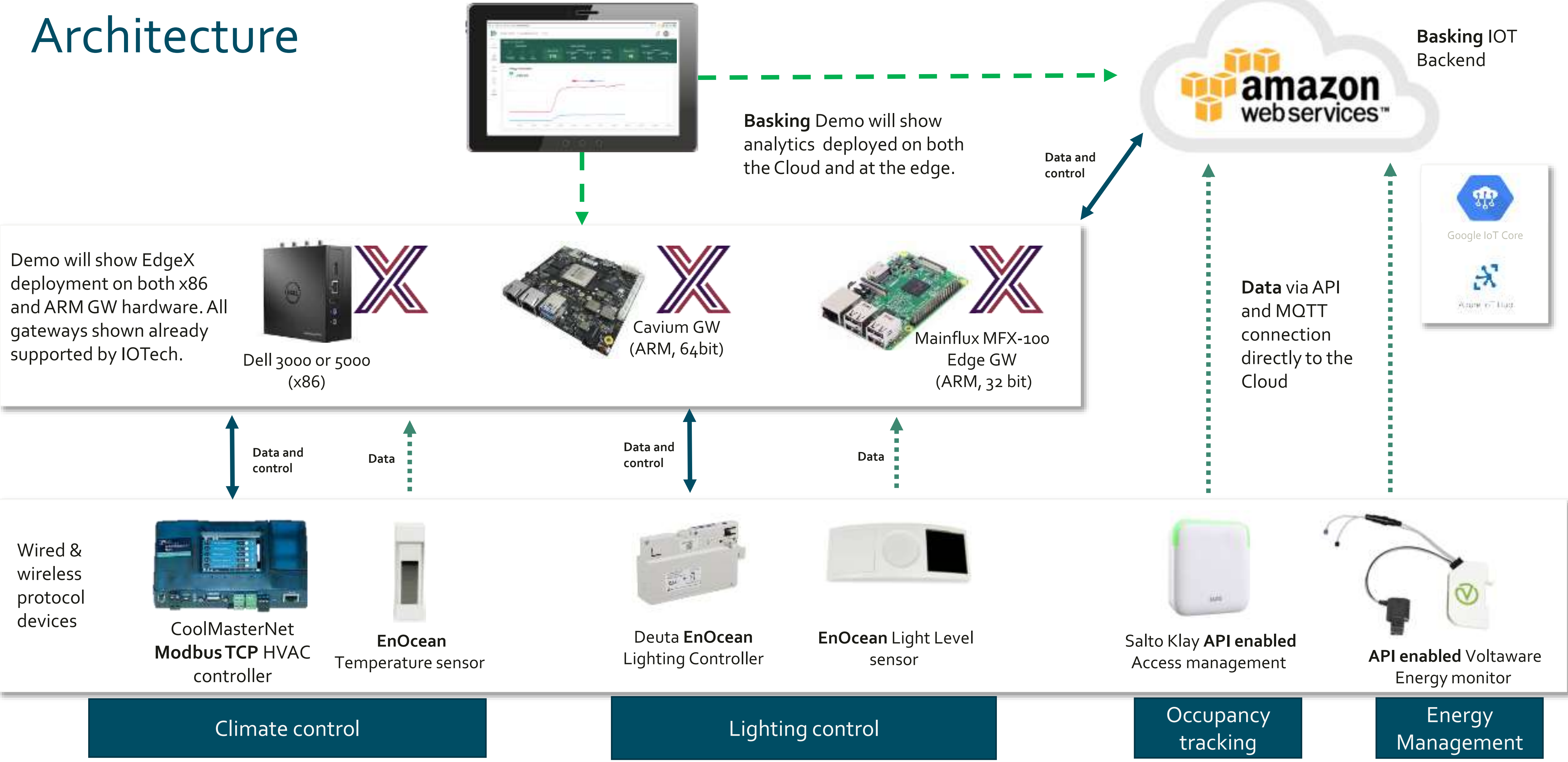
Business Issues

- Discussion items
  - Demonstrator
  - Dev kits
  - IIC and other liaison efforts
  - Which events we use to show case and announce
  - Certification/Marketplace offerings
  - Developer Advocate and other role

Demonstrator

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

Architecture



Basking IOT Backend

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

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

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

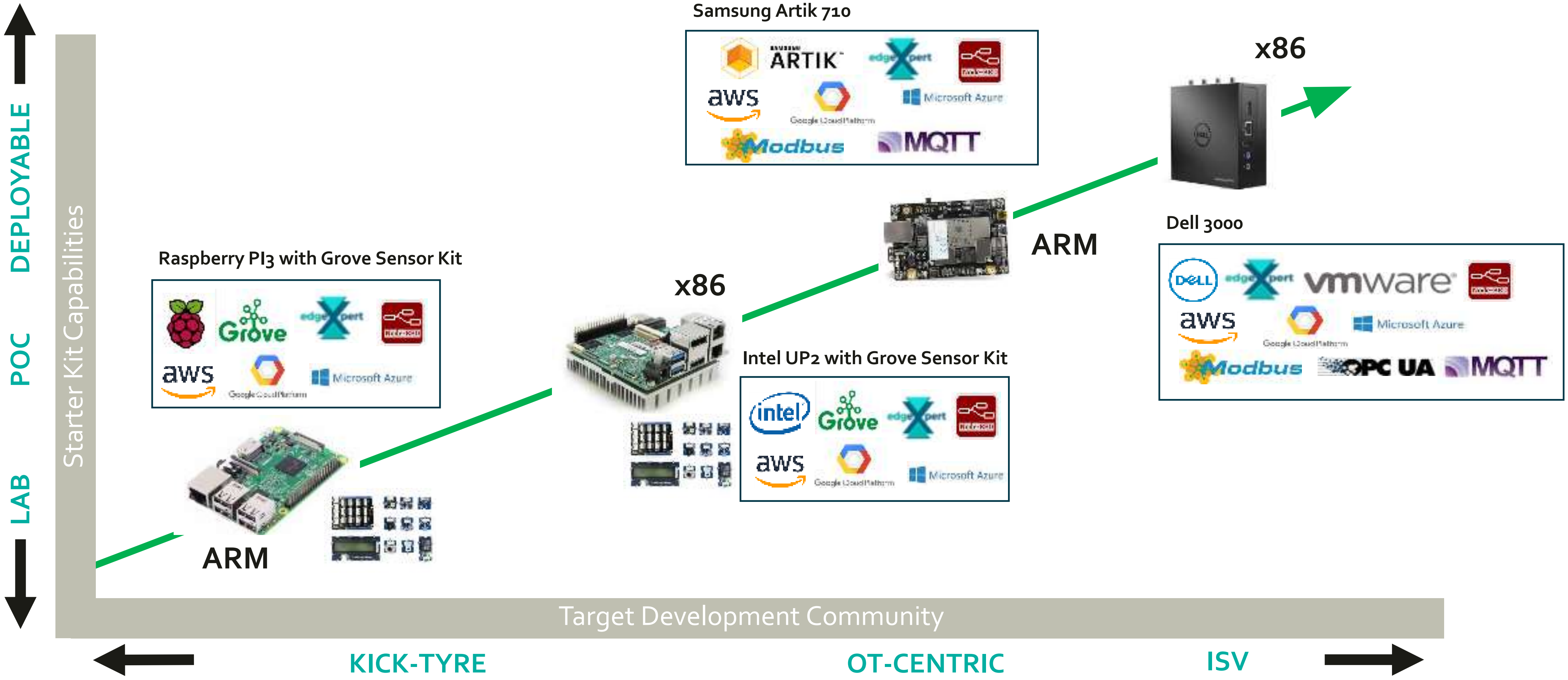
Energy Management

IOTech

DevKits

- TBD

Edge X IoT Dev Kits (Available November)



DEPLOYABLE

POC

LAB

Starter Kit Capabilities

Raspberry PI3 with Grove Sensor Kit

ARM

x86

Intel UP2 with Grove Sensor Kit

Samsung Artik 710

ARM

x86

Dell 3000

Target Development Community

KICK-TYRE

OT-CENTRIC

ISV

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?
- How should the role change?
- Are there tasks/priorities we want to shift with Michael?
- Are there other roles we need to consider?
  - Release Manager?