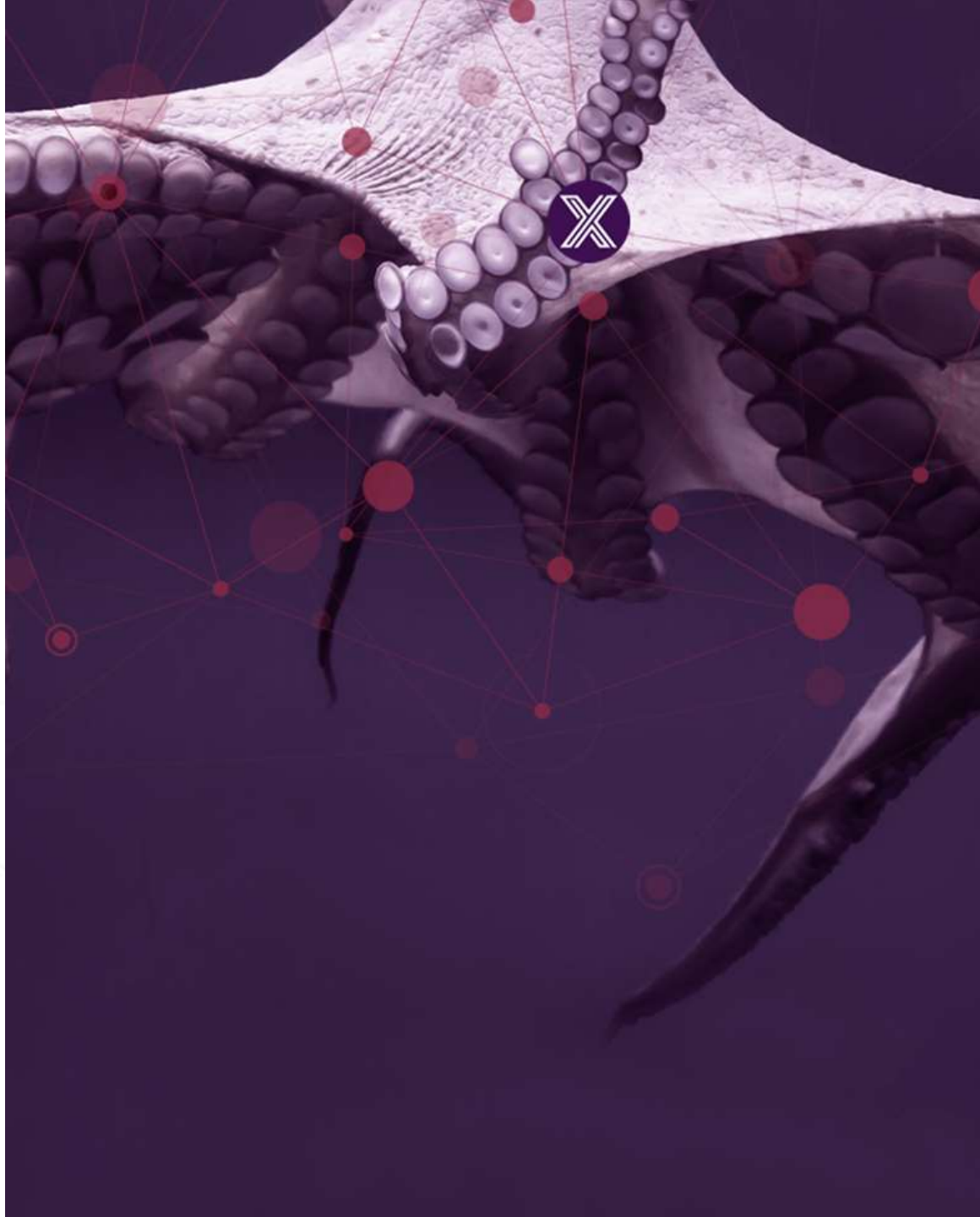
EDGEX FOUNDRY™

IIC / EdgeX Foundry Workshop

September 11, 2018

Chicago, IL



EDGEXFOUNDRY™

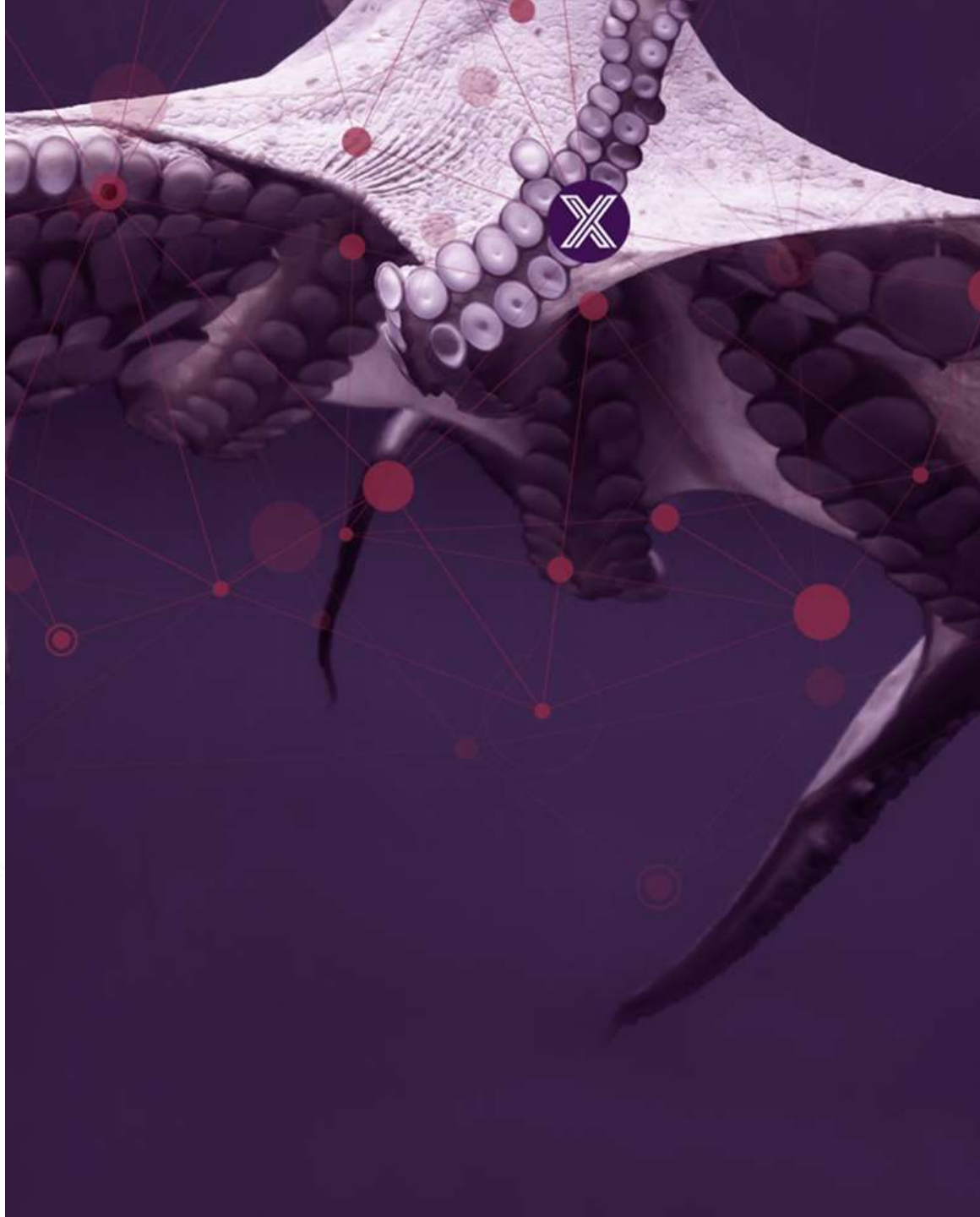
Agenda

Introductions and Workshop Overview	10 Minutes
Sven Schrecker, SWG Chair, Wael William Diab, Liaison WG Chair, Keith: EdgeX TSC Chair	
Overview of the IIC Liaison Working Group	10 Minutes
Wael Diab, Senior Director, Huawei Technologies	
Overview of EdgeX Foundry Initiatives/Activities	10 Minutes
Keith Steele	
IIC Security Initiatives	30 Minutes
EdgeX Architecture/Security Overview	30 Minutes
Jim White: EdgeX TSC Vice Chair, David Ferriera EdgeX Security WG Chair	
Break	15 Minutes
EdgeX System Management	15 Minutes
Salim AbiEzzi: VMWare, Director R&D, IoT OCTO at VMware	
Panel Discussion & Q/A on Collaboration/Way Forward	55 Minutes
Sven Schrecker Moderator and Panelists all presenters	
Closing	5 Minutes

EDGEX FOUNDRY™

Overview of EdgeX Foundry Initiatives/Activities

Keith Steele



The IoT market is inherently heterogeneous…

Domain Expertise

ANALYTICS | DATA MGMT | SECURITY | SYSTEM MGMT | SERVICES

Many different tools and skill sets are required to address myriad industry verticals and use cases

Connectivity

OPEN CONNECTIVITY FOUNDATION, AllJoyn, EAVE, OPC UA, MQTT, ZigBee, Z-WAVE, enocean, WirelessHART, Modbus, BACnet, EEBUS, KNX, genbus, DNP3, EtherNet/IP, IO-Link

IoT standards work is progressing, but there will always be widespread fragmentation in connectivity

Application Environments

Java, python, JS, Microsoft .NET, GO, C/C++

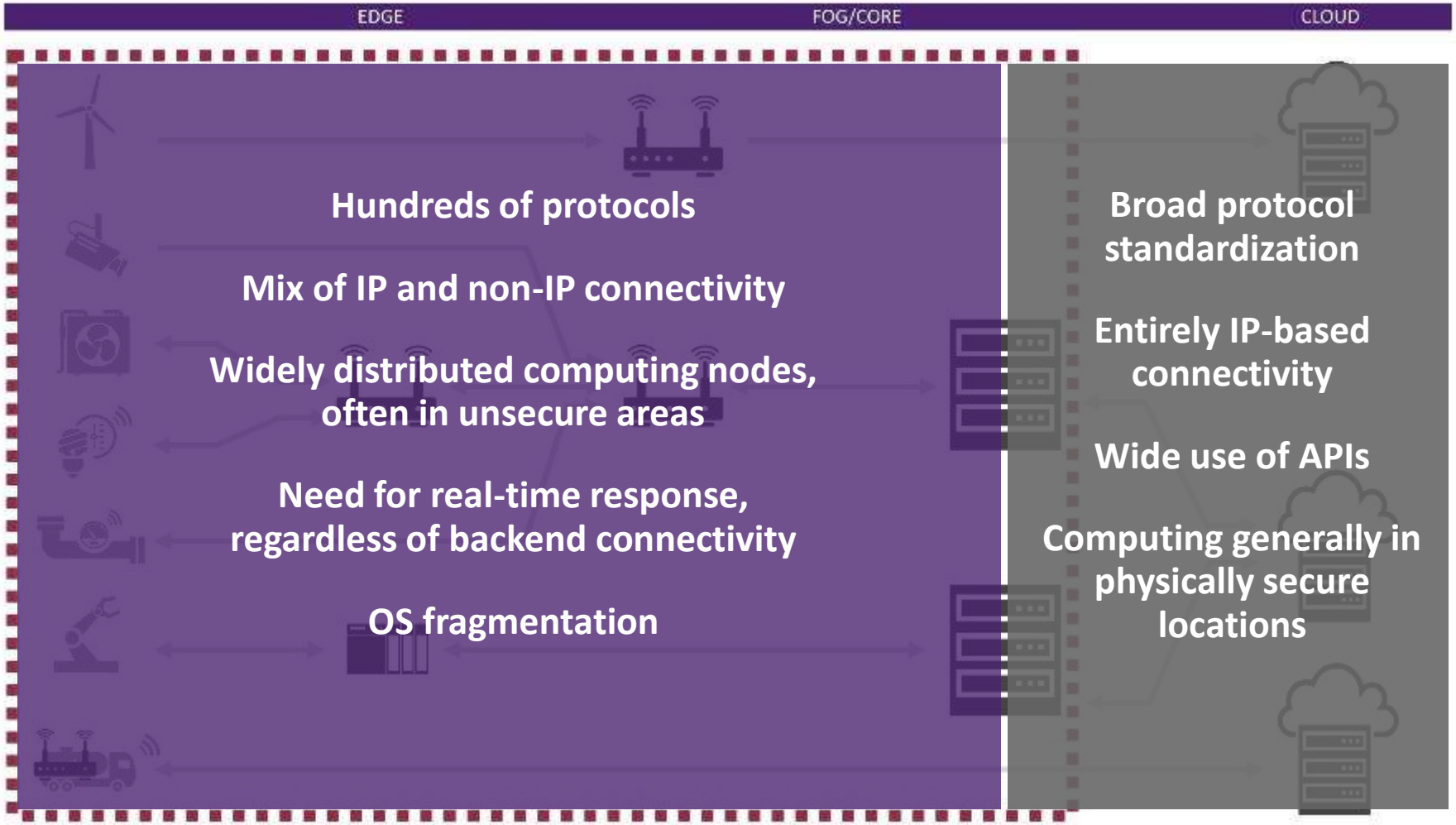
Variable preferences for coding and application environments among developers

Operating Systems

ubuntu, Windows, Zephyr, WIND RIVER PULSAR LINUX, VxWorks, yocto, mynewt, debian, redhat, suse

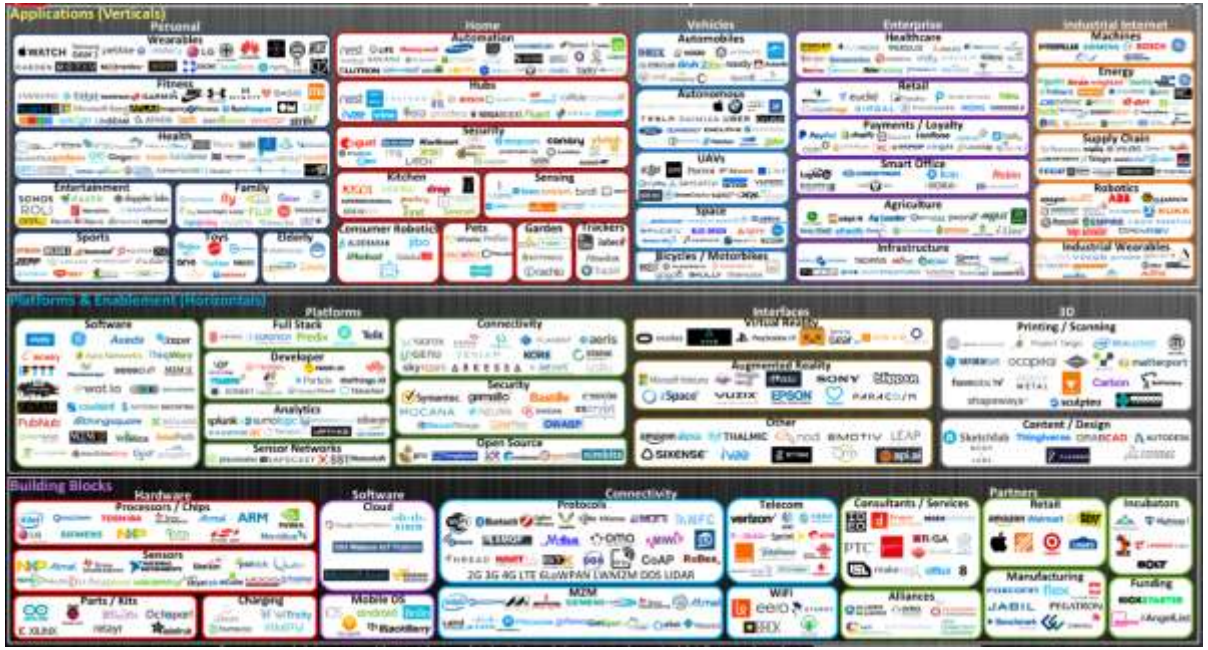
No line of sight to consistent choices across Linux, Windows and embedded/RTOS variants

… and the majority of the challenges are at the Edge.

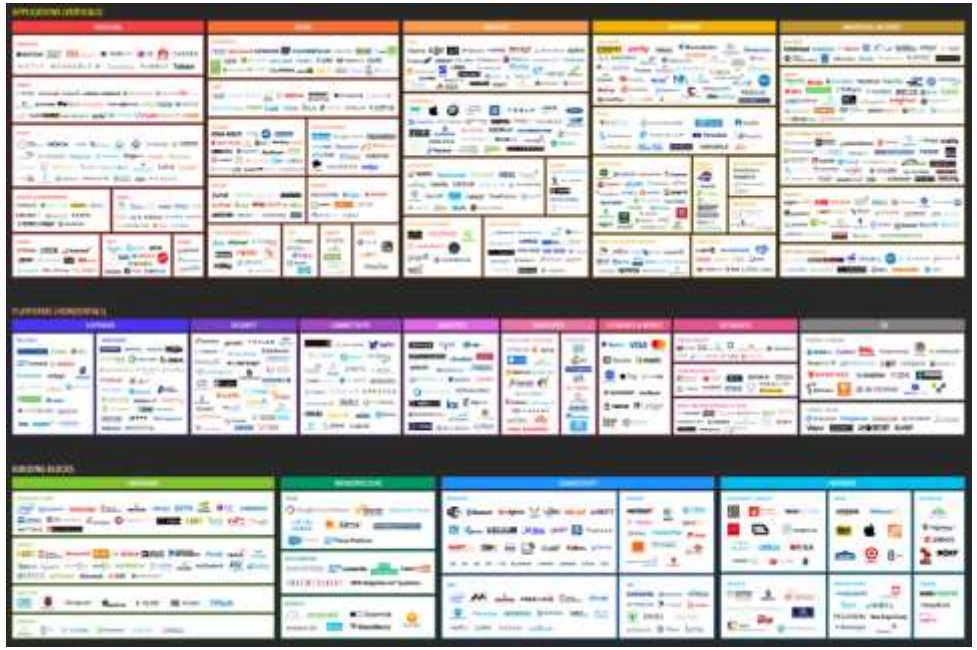


The Fragmented IoT Ecosystem

2016 IoT Landscape



2018 IoT Landscape



Source: Matt Turck, Demi Obayomi and FirstMark Capital

The IoT landscape is characterized by many software platforms reinventing the same foundational elements that also tend to lock end users in to one cloud. In order to scale, **the market needs a common foundation to brings together innovative applications, domain knowedge, and services**

EDGEXFOUNDRY™

EdgeX Foundry™ is a vendor-neutral open source project hosted by The Linux Foundation building a common open framework for IoT edge computing.

At the heart of the project is an interoperability framework hosted within a full hardware- and OS-agnostic reference software platform to enable an ecosystem of plug-and-play components that unifies the marketplace and accelerates the deployment of IoT solutions.

Architected to be agnostic to protocol, silicon (*e.g.*, x86, ARM), OS (*e.g.*, Linux, Windows, Mac OS), and application environment (*e.g.*, Java, JavaScript, Python, Go Lang, C/C++) to support customer preferences for differentiation

Use Case Scope

	Through Community Extensions	Open Source Baseline
	Real Time	**Soft Real Time ("Relevant Time")**
Response Time	High bandwidth, QoS, sub-millisecond, deterministic	Milliseconds+
OS	RTOS	Traditional Linux or Windows
Example Use Cases	Smart Building, Energy Management, Factory Optimization, Predictive Maintenance, Quality Control, Supply Chain Management, Remote Asset Management, Fleet Management, Logistics, Environmental Monitoring	
	High-speed Process Control, Robotics, Safety Systems, Autonomous Vehicles	

Backed by Industry Leaders

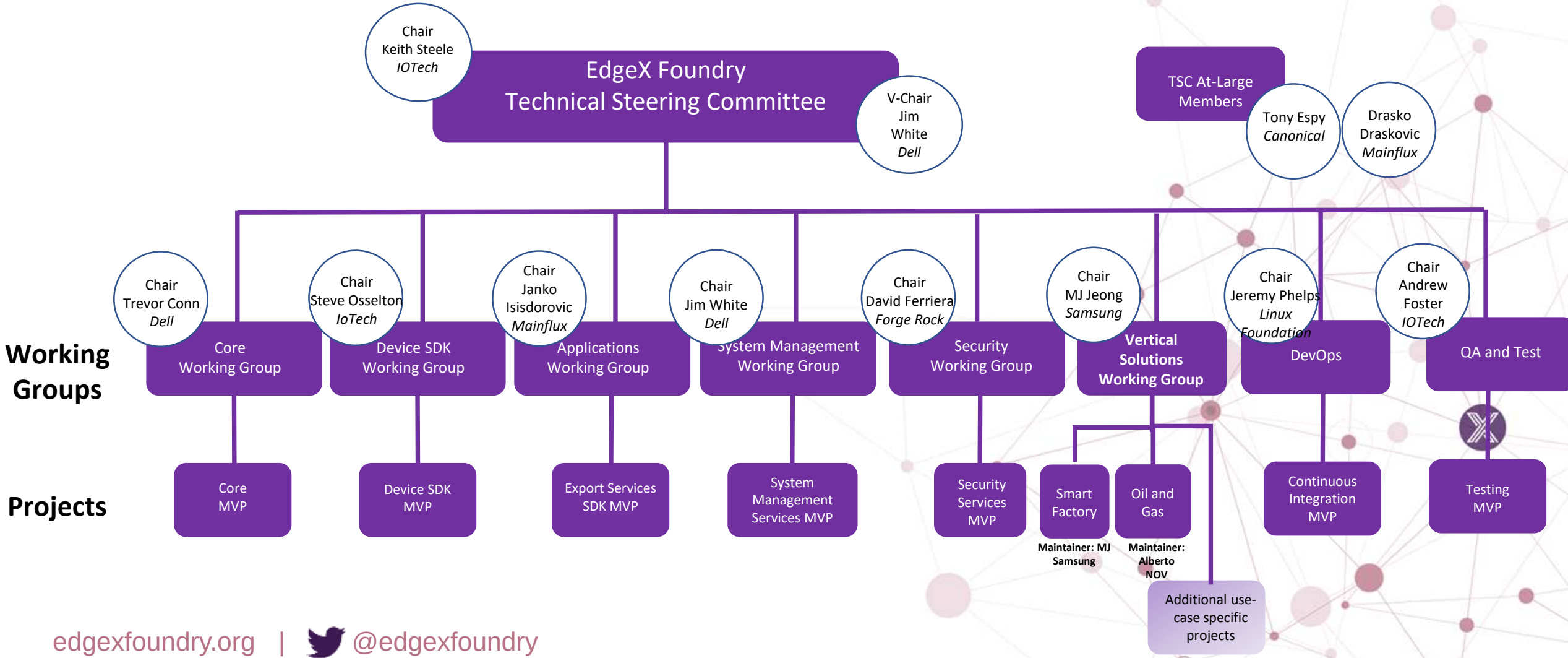
EDGEXFOUNDRY™

ABSOLUTE
aicas realtime
Alleantia
AMD
ANALOG DEVICES
AXZON
BASKING
Beechwoods
ubuntu Delivered by Canonical
CAVIUM
CertusNet
CIVIL INFRASTRUCTURE PLATFORM
CLOUD FOUNDRY
CloudPlugs
Cumulocity A SOFTWARE AG COMPANY
DELL EMC
DEVICE AUTHORITY
enigmedia
enocean
EPISENSOR
ETRI Electronics and Telecommunications Research Institute
FIWARE FOUNDATION
FOGHORN
FORGEROCK
IMS EVOLVE
industrial internet CONSORTIUM
inwinSTACK
IOTech
IoTium www.iotium.io
ISSAT Mateur
KMC CONTROLS
Linaro
MAINFLUX
MOCANA
modius
NETFOUNDRY
neustar
OMG OBJECT MANAGEMENT GROUP
OPTO 22
Project Haystack
Queen's University Belfast
redislabs home of redis
REPLY CONCEPT
RIOT
Rubicon
SAMSUNG
SAMSUNG SDS
Sixgill Govern IoE
striim
SWITCH AUTOMATION
THALES
twobulls
ULE ALLIANCE
V5 SYSTEMS
vmware
Volterra EDGE SERVICES
WANXIANG GROUP
xage SECURITY
ZEDEDA
Zephyr
ZingBox

edgexfoundry.org | @edgexfoundry

With more in process!

EdgeX Project Organization



EDGEXFOUNDRY™

Target Bi-Annual Release Roadmap

In order to provide EdgeX consumers with a predictable foundation to base their commercial offerings on it is the goal of the TSC to outline key release themes at least 12 months in advance and to plan features to be delivered in a given release 6 months in advance. As with any open source software project, delivery of planned features is based on priority and available developer bandwidth.

2017			2018												2019											
Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec

'Barcelona' Release

(Released Oct 20 2017)

- Improved fit and finish, formalized Core Service APIs, additional Device and Export Services, test apparatus

'California' Preview'

(Made available Jan 2018)

- Drop-in Go Lang microservice replacements demonstrating reduced footprint and higher performance

'California' Release

(Released June 2018)

- First integration of security
- Run in < 256MB RAM, come up in < 10 sec

'Delhi' Release

(Oct 2018)

- Additional security tools and first manageability capabilities
- Go / C Device Service SDKs
- Sample device services (e.g. Modbus, MQTT, OPC-UA)
- Community and Commercial Dev kits
- EdgeX reference UI

'Edinburgh' Release

(Apr 2019)

- Certification program
- Improved and more scalable northbound connectors / application services
- Additional southbound connectors to common protocols and devices
- ARM 32 support

'Fuji' Release

(Oct 2019)

- Load balancing
- Multi-host EdgeX
- Additional security and system management capability

First three releases were delivered by the community per the committed schedule

Key EdgeX project accomplishments since April 2017 launch

- **Bi-annual release roadmap established** and first two release dates met
- Now **64 individual code contributors**, 5X increase from January 2018
- **Refactored entire code base to Go Lang**
  - Full seed platform was ~2.5GB memory, booted in minutes; now ~128MB and boots in ~5 seconds
- Established **security + management plan**, 1st features in July + October releases
- **IIC alliance formed and first IIC test bed in process** from Wanxiang Group
- **Entire documentation base refreshed** @ https://docs.edgexfoundry.org/
- Now at 63 project members with **numerous marquee names joining in at SWC**
- **Increasing number of end customer PoCs** in various industries
- Numerous tech providers **integrating into commercial offers**
  - IOTech announced Edge Xpert (Red Hat model) and xRT as a licensed real-time variant

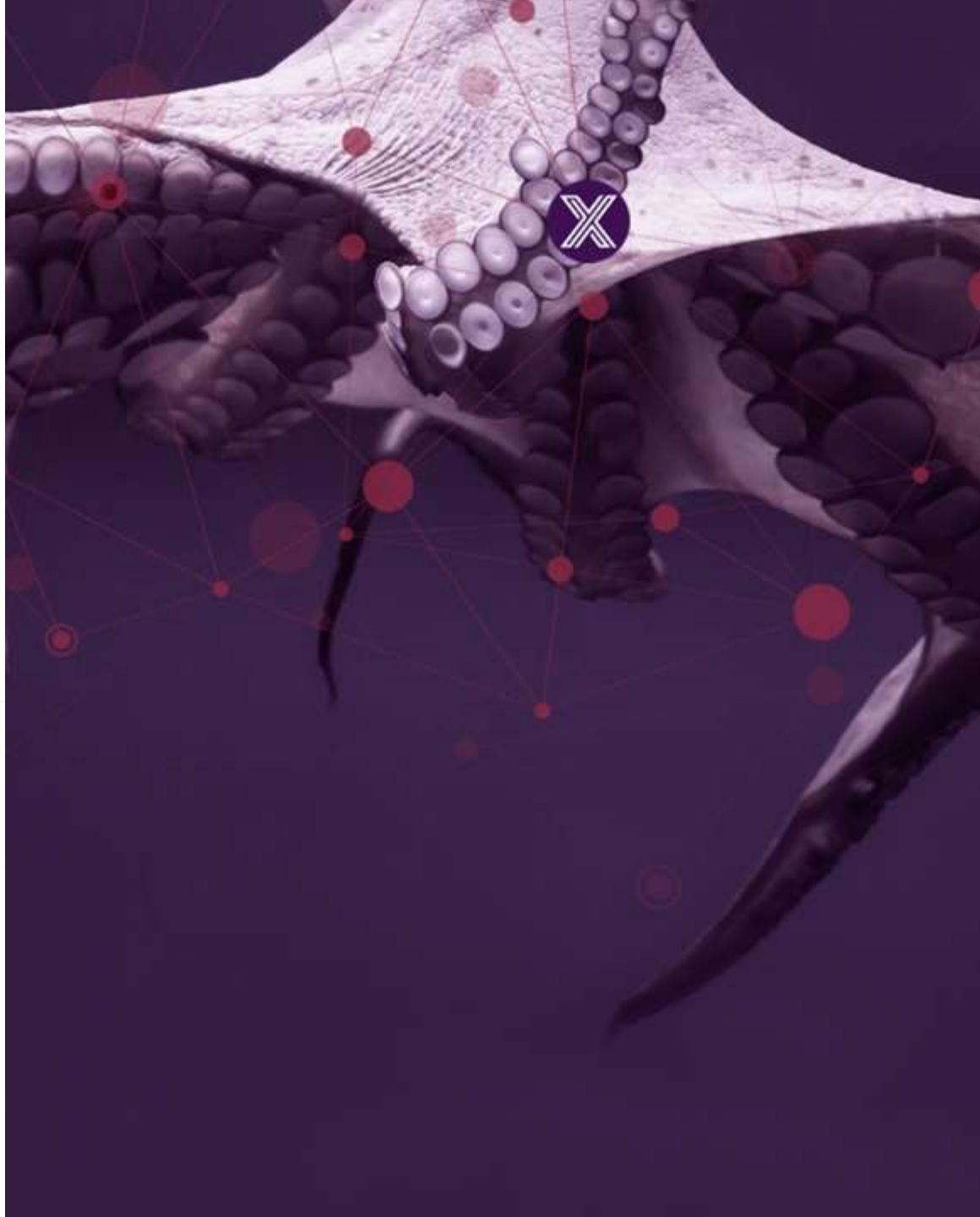
State of the Union Blog from January 2018

EDGEXFOUNDRY™

EdgeX Architecture/Security Overview

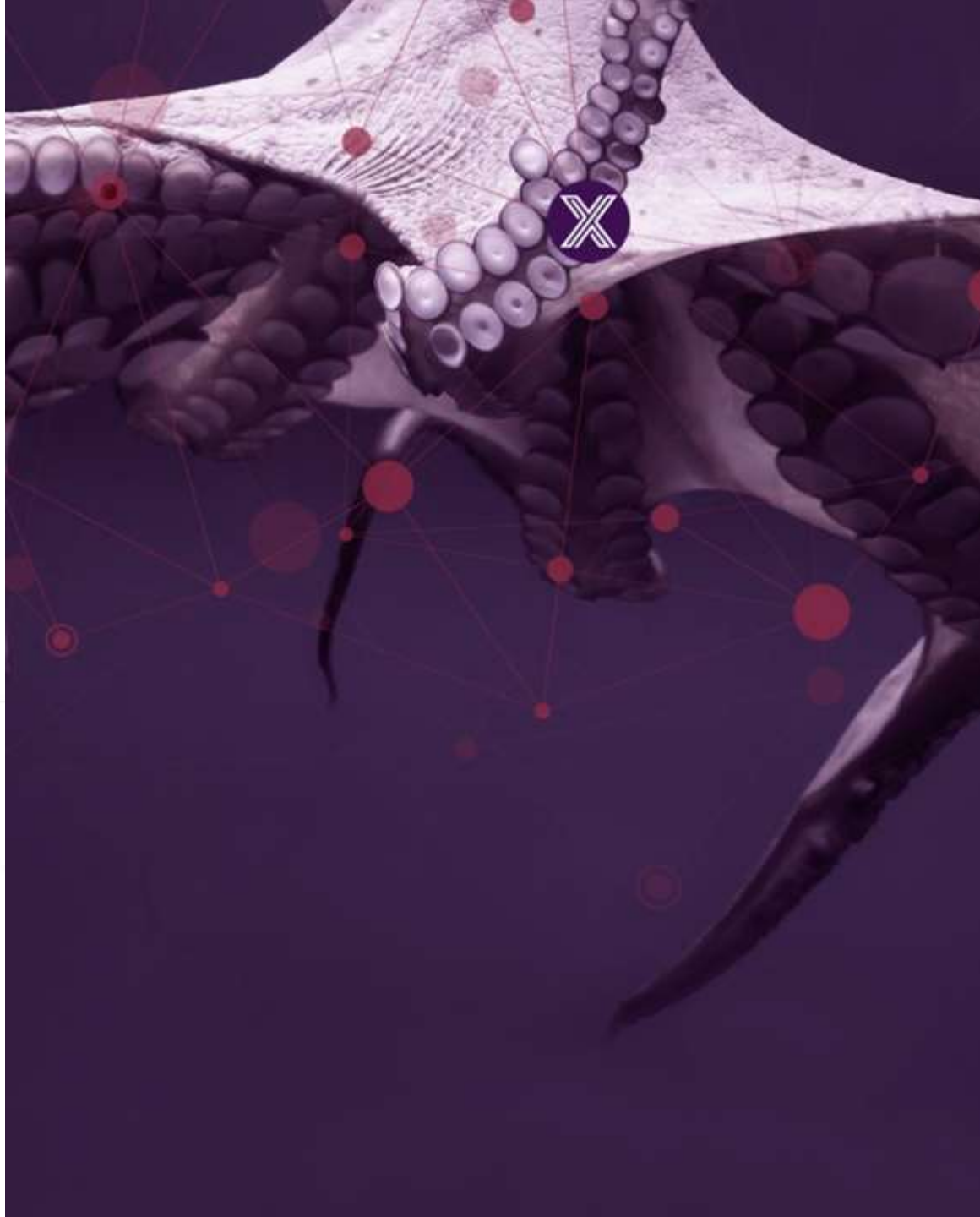
Jim White

David Ferriera



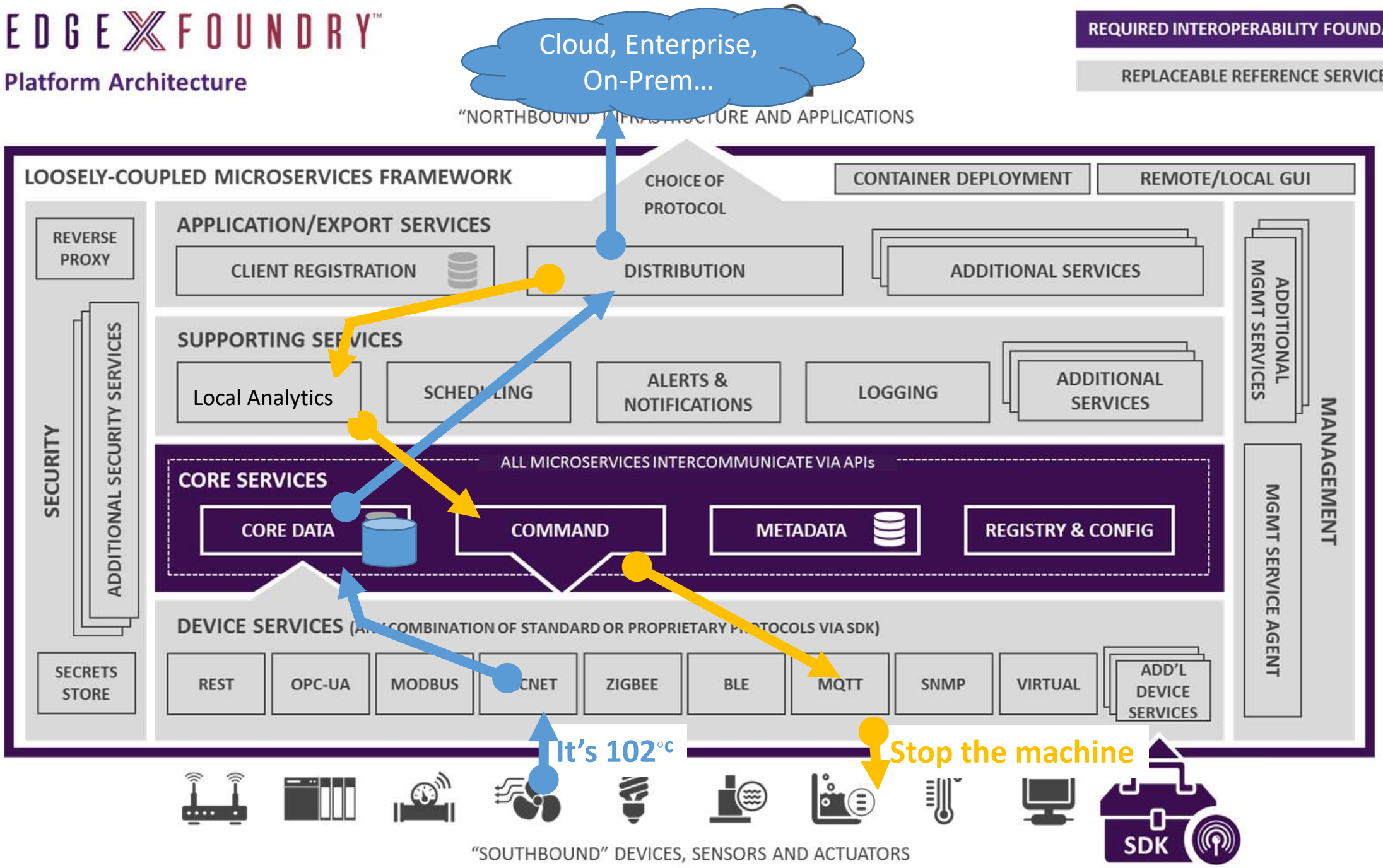
EDGEXFOUNDRY™

Architecture



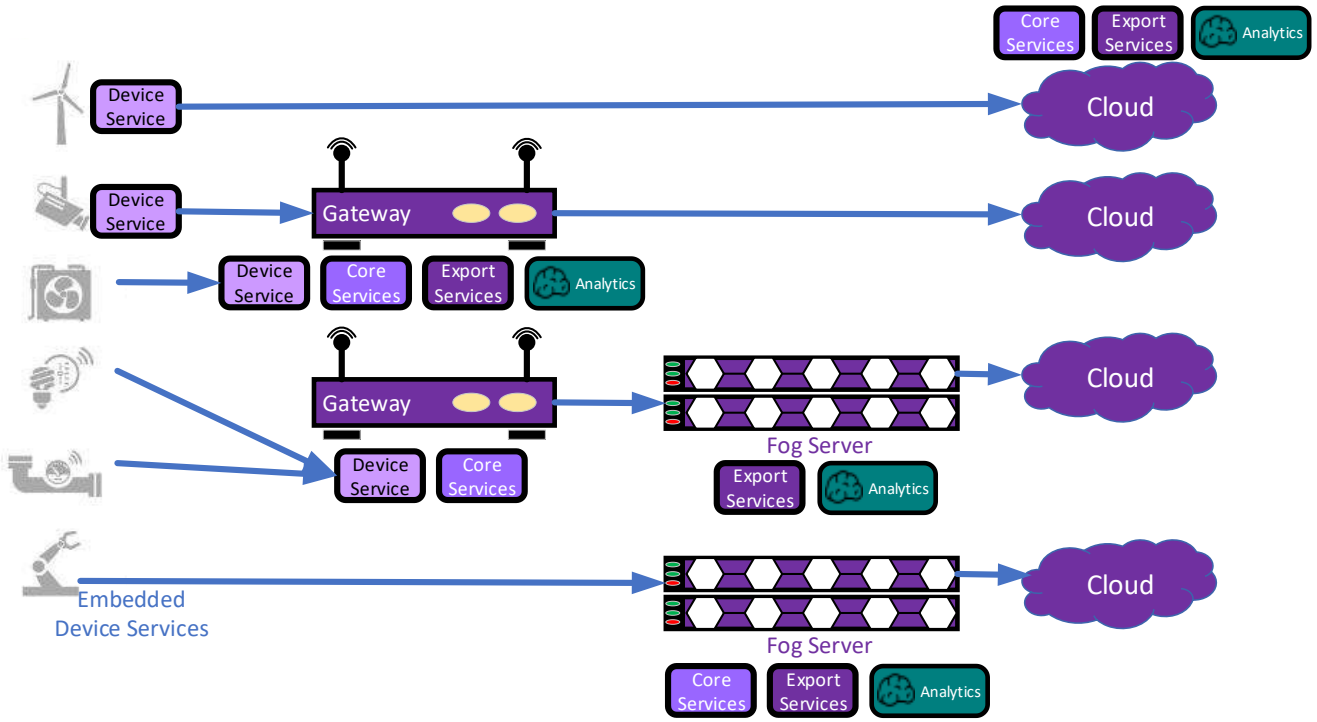
EdgeX Primer - How it works

- A collection of a dozen+ micro services
  - Written in multiple languages (Java, Go, C, … we are polyglot believers!!)
- EdgeX data flow:
  - Sensor data is collected by a **Device Service** from a thing
  - Data is passed to the **Core Services** for local persistence
  - Data is then passed to **Export Services** for transformation, formatting, filtering and can then be sent "north" to enterprise/cloud systems
  - Data is then available for edge analysis and can trigger device actuation through Command service
  - Many others services provide the supporting capability that drives this flow
- REST communications between the service
  - Some services exchange data via message bus (core data to export services and rules engine)
- Micro services are deployed via Docker and Docker Compose


EDGEXFOUNDRY™
Platform Architecture
Cloud, Enterprise, On-Prem…
REQUIRED INTEROPERABILITY FOUNDATION
REPLACEABLE REFERENCE SERVICES
"NORTHBOUND" INFRASTRUCTURE AND APPLICATIONS
LOOSELY-COUPLED MICROSERVICES FRAMEWORK
CHOICE OF PROTOCOL
CONTAINER DEPLOYMENT
REMOTE/LOCAL GUI
REVERSE PROXY
APPLICATION/EXPORT SERVICES
CLIENT REGISTRATION
DISTRIBUTION
ADDITIONAL SERVICES
SUPPORTING SERVICES
Local Analytics
SCHEDULING
ALERTS & NOTIFICATIONS
LOGGING
ADDITIONAL SERVICES
ADDITIONAL MGMT SERVICES
MANAGEMENT
SECURITY
ADDITIONAL SECURITY SERVICES
ALL MICROSERVICES INTERCOMMUNICATE VIA APIs
CORE SERVICES
CORE DATA
COMMAND
METADATA
REGISTRY & CONFIG
MGMT SERVICE AGENT
DEVICE SERVICES (ANY COMBINATION OF STANDARD OR PROPRIETARY PROTOCOLS VIA SDK)
SECRETS STORE
REST
OPC-UA
MODBUS
ZIGBEE
BLE
MQTT
SNMP
VIRTUAL
ADD'L DEVICE SERVICES
It's 102°c
Stop the machine
SDK
"SOUTHBOUND" DEVICES, SENSORS AND ACTUATORS

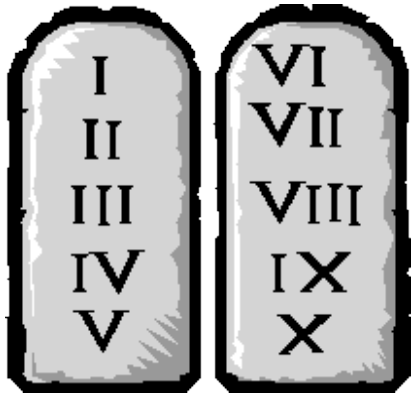
EdgeX Enables Tiered Fog Deployments

- In today's IoT landscape, it is imperative to leverage compute, storage, network resources where every they live
- Loosely-coupled architecture enables distribution across nodes to enable tiered edge/fog computing
- Scope includes embedded sensors to controllers, edge gateways and servers
- Quantity and function of micro services deployed on a given node depends on the use case and capability of hardware



EdgeX Architectural Tenets

- EdgeX Foundry must be **platform agnostic** with regard to hardware, OS, distribution/deployment, protocols/sensors
- EdgeX Foundry must be **extremely flexible**
  - Any part of the platform may be upgraded, replaced or augmented by other micro services or software components
  - Allow services to scale up and down based on device capability and use case
- EdgeX Foundry should provide “reference implementation” services but **encourages best of breed solutions**
- EdgeX Foundry must provide for **store and forward** capability (to support disconnected/remote edge systems)
- EdgeX Foundry must support and **facilitate “intelligence” moving closer to the edge** in order to address
  - Actuation latency concerns
  - Bandwidth and storage concerns
  - Operating remotely concerns
- EdgeX Foundry must **support brown and green device/sensor** field deployments
- EdgeX Foundry **must be secure and easily managed**

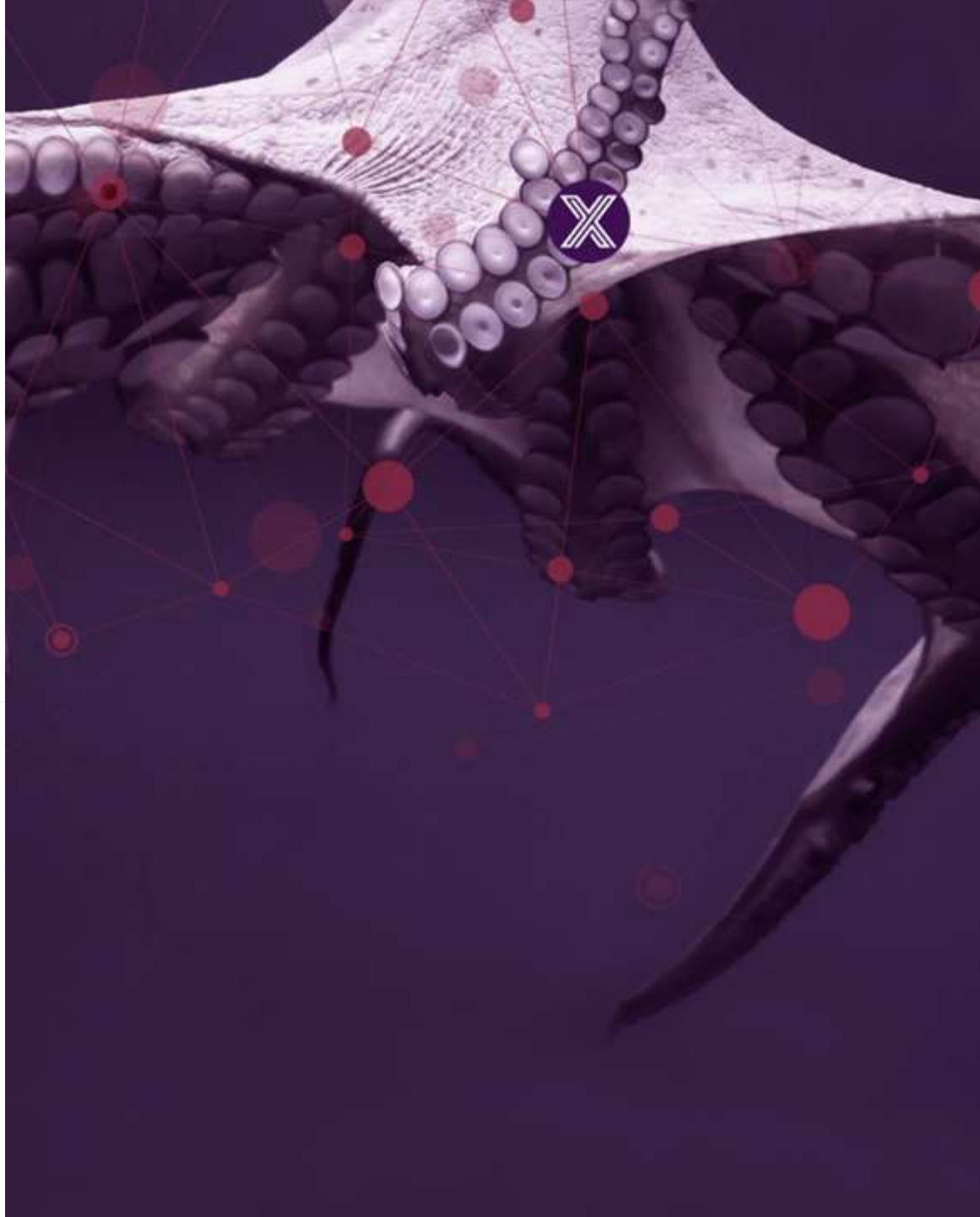


EdgeX Technology

- A majority of the micro services are written in Go Lang
  - Previously written in Java
  - Some Device Services written in C/C++
  - A user interface is provided in JavaScript
  - Polyglot belief – use the language and tools that make sense for each service
- Each service has a REST API for communicating with it
- Uses MongoDB to persist sensor data at the edge
  - Also stores application relevant information
  - Allows for alternate persistence storage (and has been done in the past)
- A message pipe connects Core Data to Export Services and/or Rules Engine
  - Uses ZeroMQ by default
  - Allow use of MQTT as alternate if broker is provided
- Uses open source technology where appropriate
  - Ex: Consul for configuration/registry, Kong for reverse proxy, Drools for rules engine,…

EDGEXFOUNDRY™

Security



EDGE X FOUNDRY™

Platform Architecture

REQUIRED INTEROPERABILITY FOUNDATION

REPLACEABLE REFERENCE SERVICES

"NORTHBOUND" INFRASTRUCTURE AND APPLICATIONS

LOOSELY-COUPLED MICROSERVICES FRAMEWORK

CHOICE OF PROTOCOL

SECURITY

In the beginning /dev/null

EXPORT SERVICES

- CLIENT REGISTRATION
- DISTRIBUTION
- ADDITIONAL SERVICES

SUPPORTING SERVICES

- RULES ENGINE
- SCHEDULING
- ALERTS & NOTIFICATIONS
- LOGGING
- ADDITIONAL SERVICES

ALL MICROSERVICES INTERCOMMUNICATE VIA APIs

CORE SERVICES

- CORE DATA
- COMMAND
- METADATA
- REGISTRY & CONFIG

DEVICE SERVICES (ANY COMBINATION OF STANDARD OR PROPRIETARY PROTOCOLS VIA SDK)

- REST
- OPC-UA
- MODBUS
- BACNET
- ZIGBEE
- BLE
- MQTT
- SNMP
- VIRTUAL
- ADD'L DEVICE SERVICES

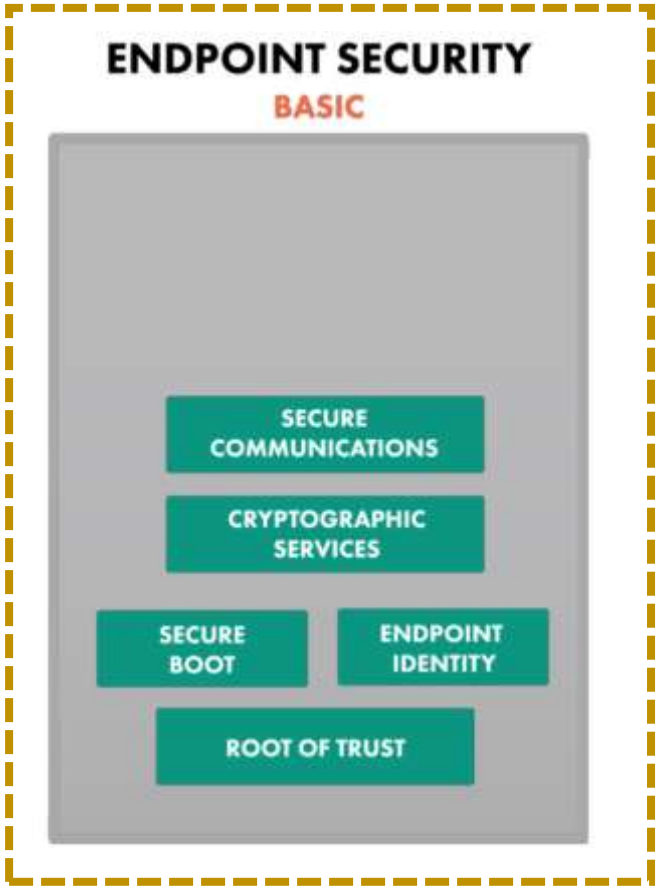
DEVICE + SYSTEM MANAGEMENT

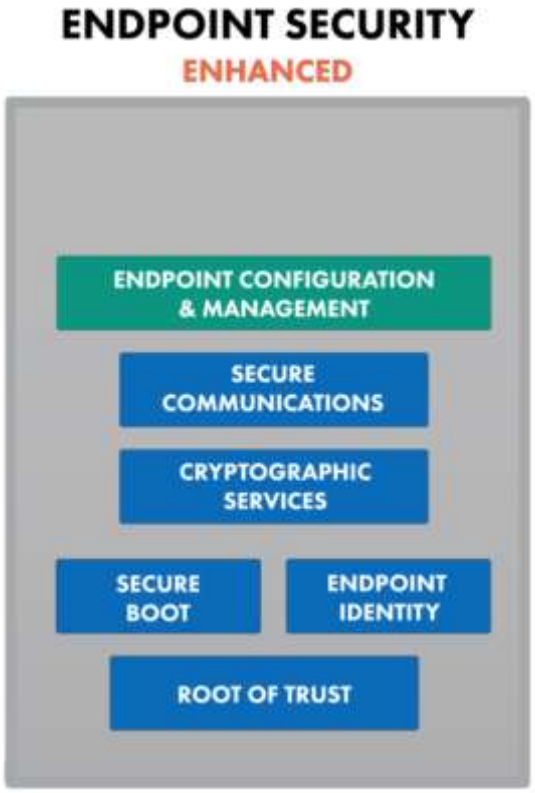
- ADDITIONAL SERVICES
- CONTAINER DEPLOYMENT
- LOCAL MGMT CONSOLE

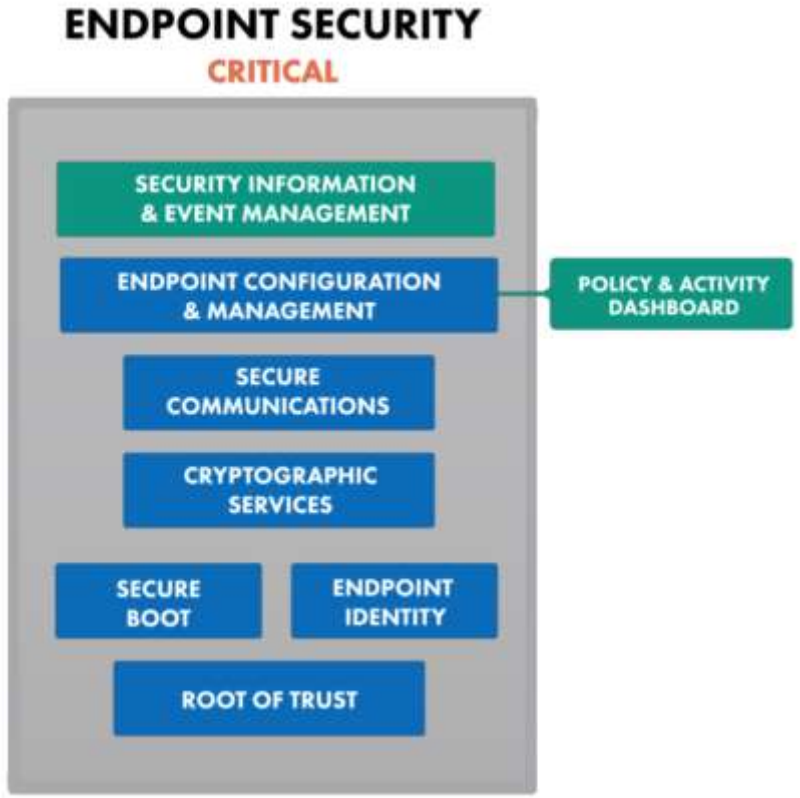
SDK

"SOUTHBOUND" DEVICES, SENSORS AND ACTUATORS

IIC Endpoint Security Best Practices and EdgeX

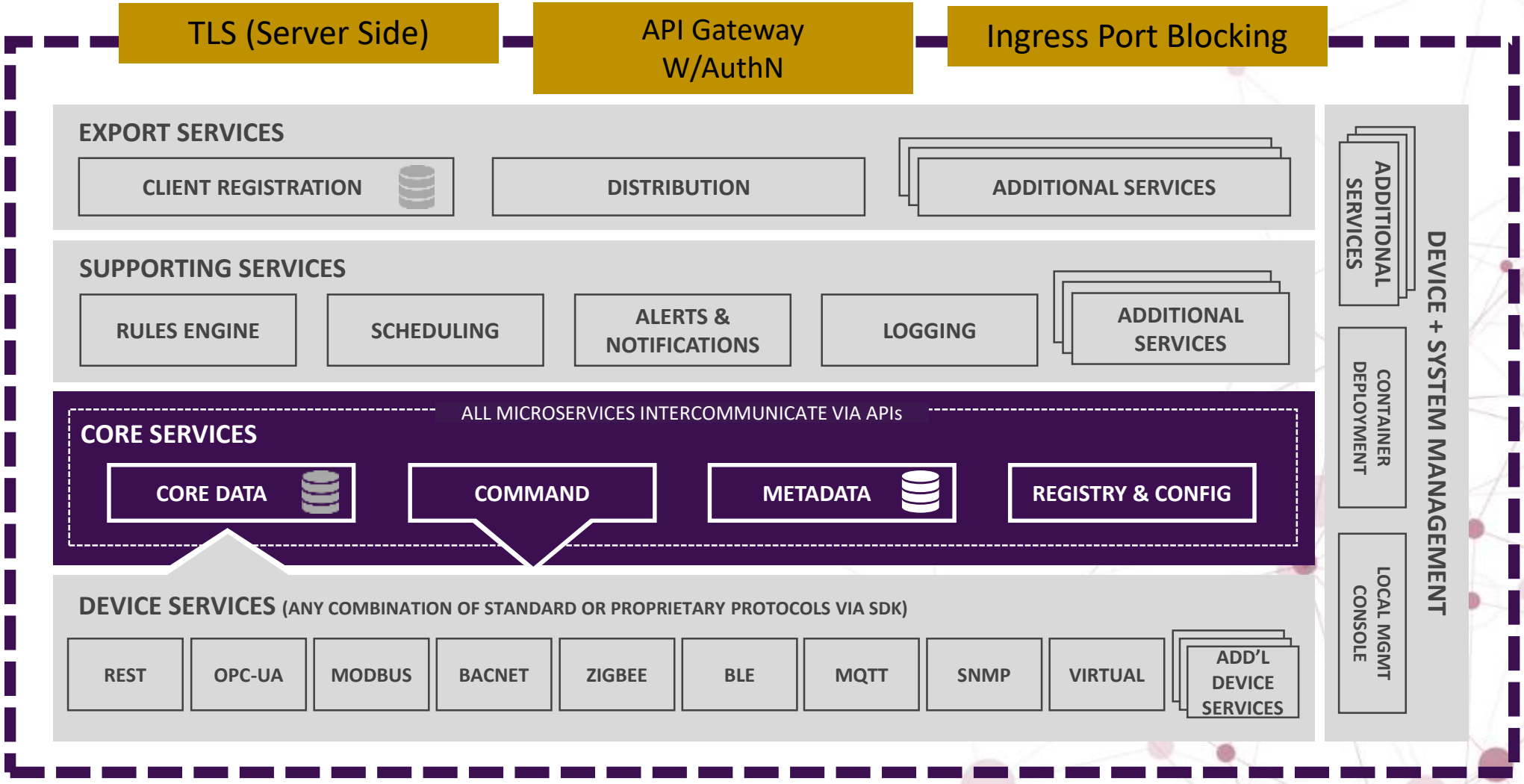






EdgeX will begin here

Start with the Basics: Protect Perimeter Ingress



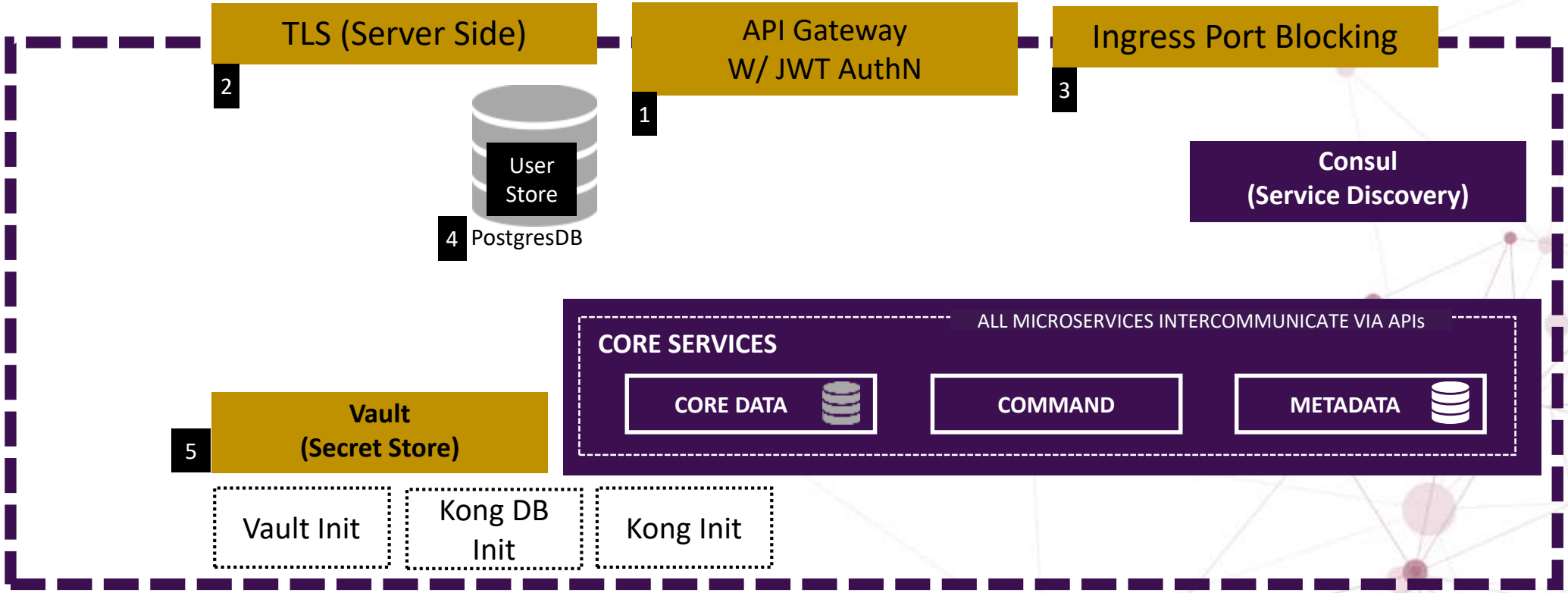
Protect Perimeter Ingress: Details and Roadmap

EDGEXFOUNDRY™

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

California Security Architecture



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

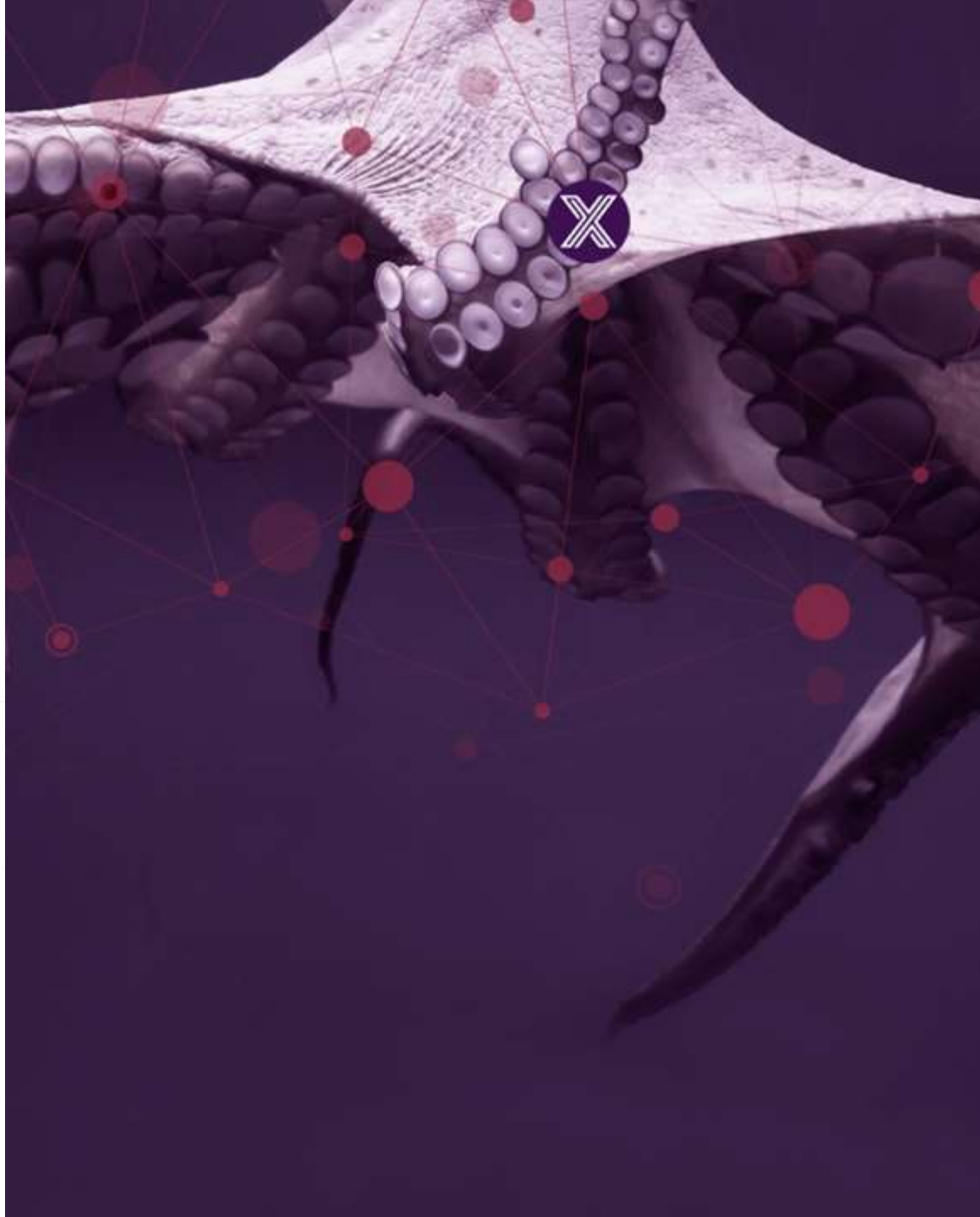
Operational Security

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

EDGEXFOUNDRY™

EdgeX System Management

Salim AbiEzzi



Scope

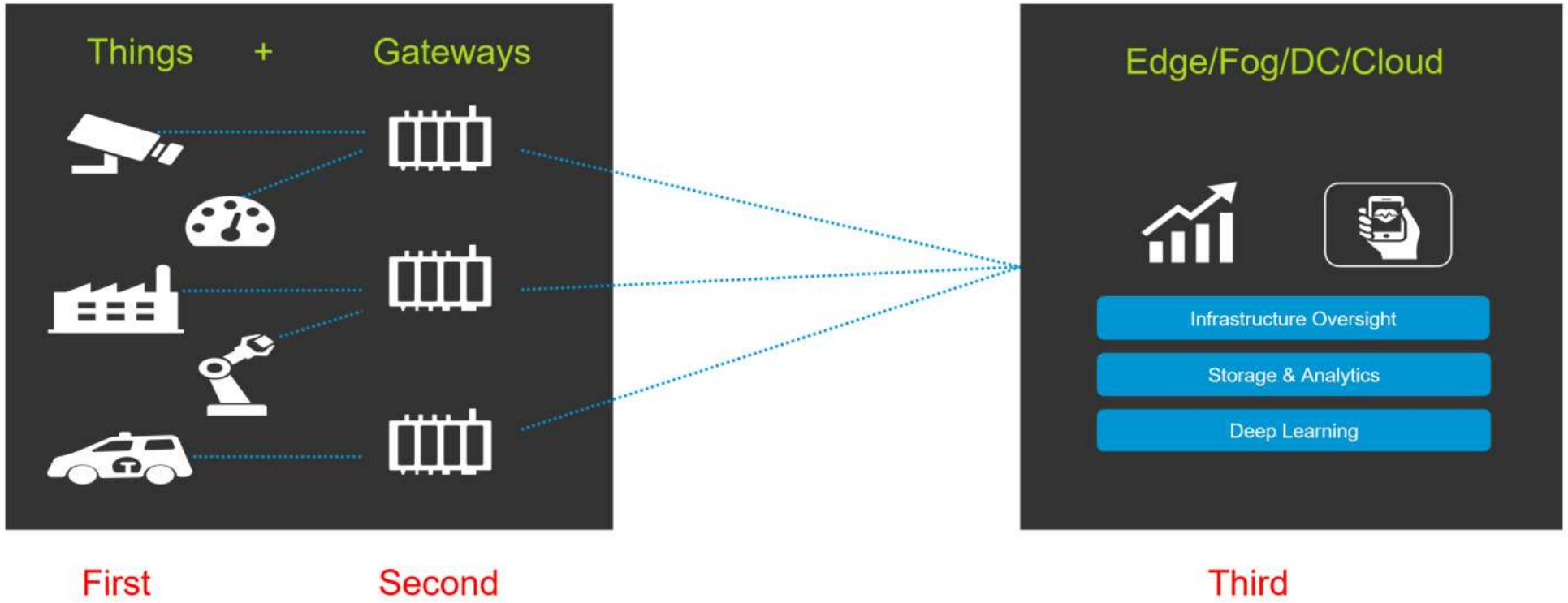
- Motivation
- Components
- Terminology
- Importance
- Architecture
- Prescriptive Guidance
- etc.

Audience

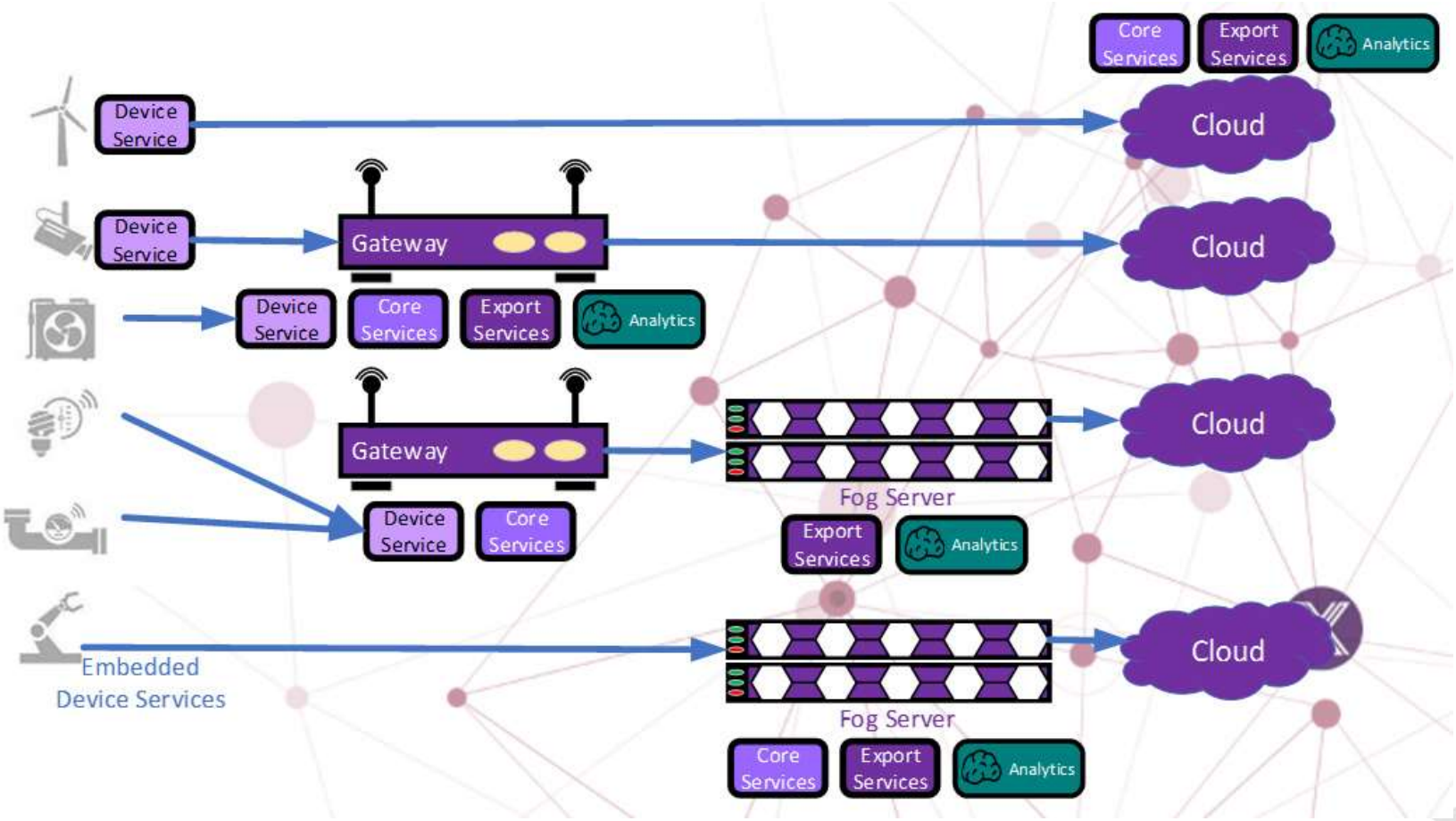
- Communities developing open source for IoT, such as the EdgeX Foundry
- System Integrators developing IoT solutions, such as IIC testbeds
- Developers of IoT management systems, such as VMware's Pulse IoT Center
- Official standards organizations, such as ISO-IEC/JTC1/SC41 on IoT
- Government legislators and regulatory agency concerned with IoT

Single Point of Oversight

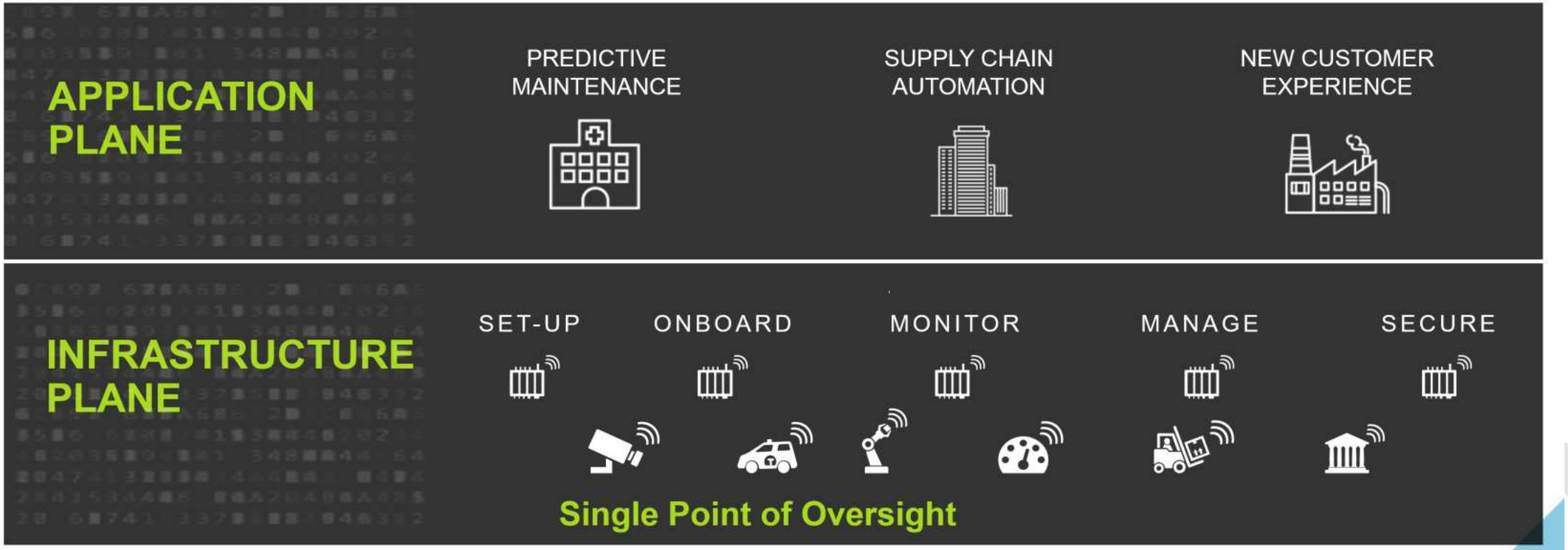
Three Tier Architecture, a useful Simplification



Practical Variations



Infrastructure versus Application Planes



Lifecycle, from Cradle to Grave

- Provisioning:
  - Secure on-boarding
  - Connectivity
  - Central UI
- Application initiation through container management
- Monitoring & alerting, both hardware and software
- Management:
  - Configuration
  - Software updates
  - Certificate management
- Shut down: end of life, forklift changes

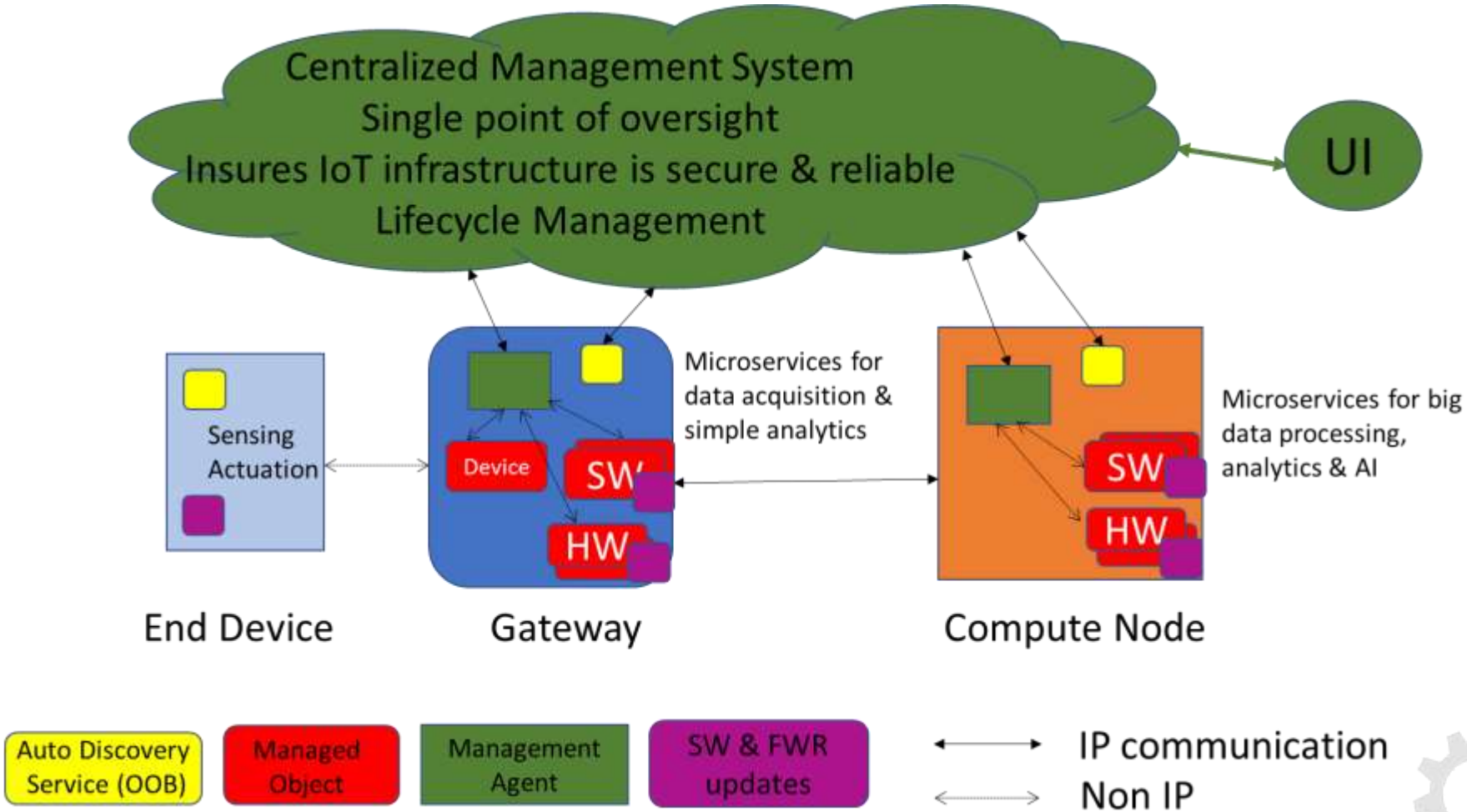
In Steady State

- Show system state visually
- Alerts; e.g., predicting failure, off-normal, overload detection
- Actuation/changing configuration; e.g., in response to alerts to avoid failure

Contract of Interoperability, Zero-Config & Auto-Discovery

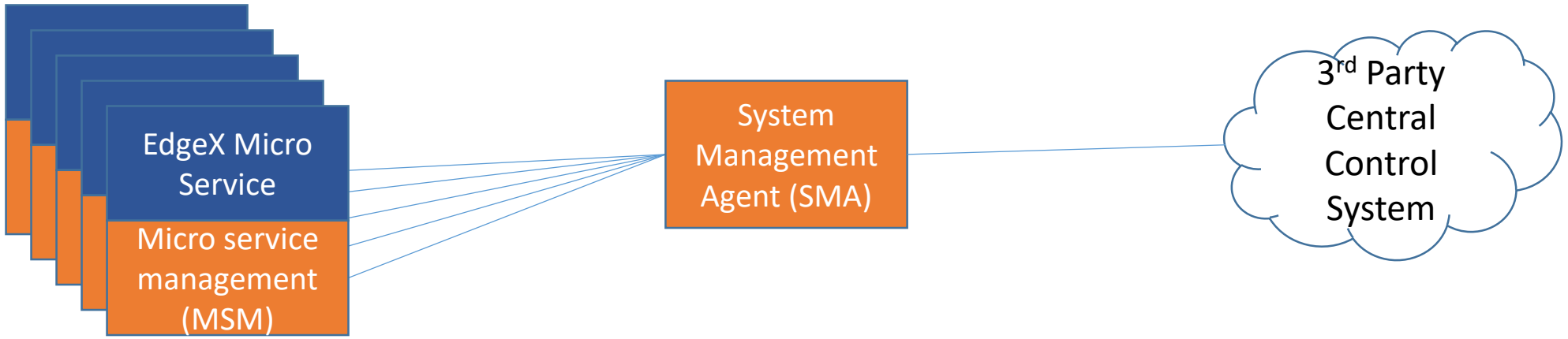
- Handshake between the management system and the device
- Devices out of the box, to auto-configure, be auto-discovered & on-boarded
- Self describing managed objects fronting both hardware & software components
- How to do it securely

How it looks when you put it together



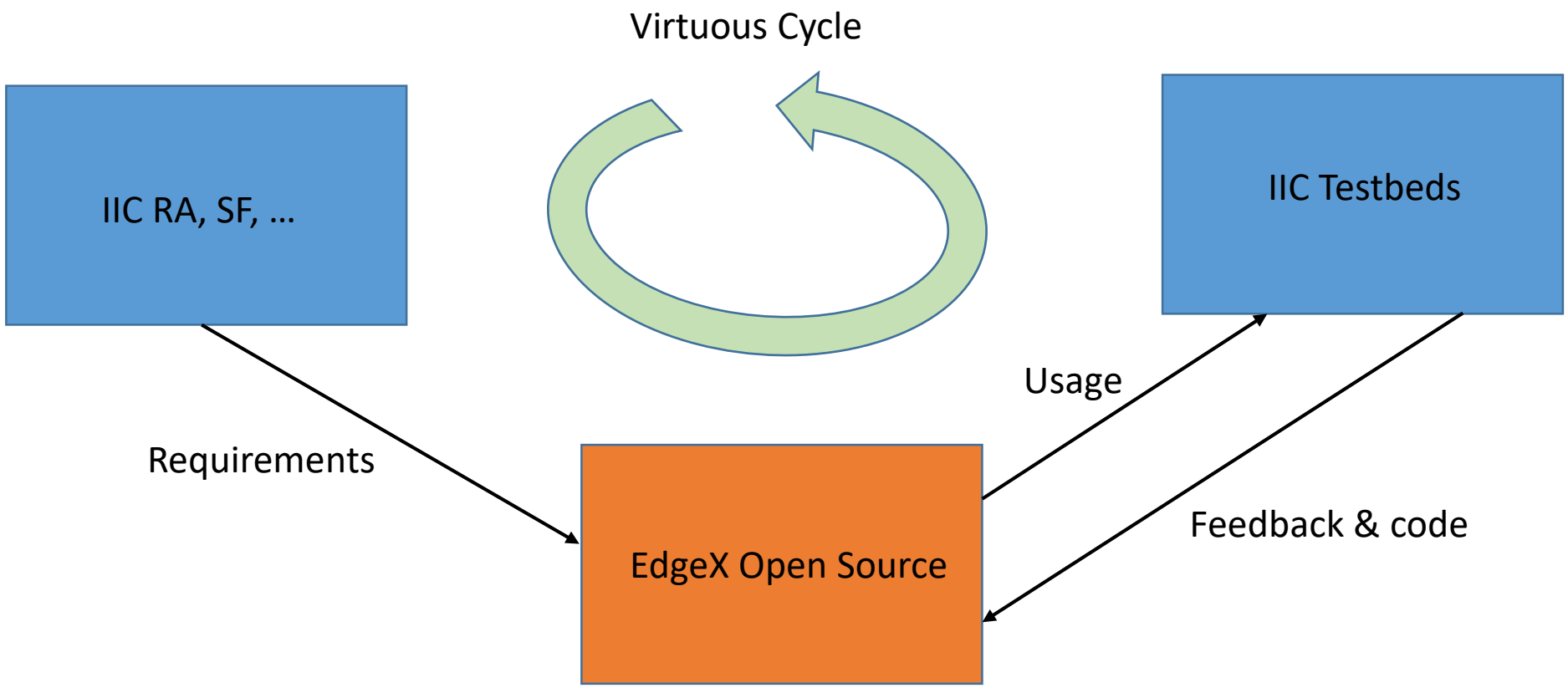
EdgeX System Management

Adding System Management Agent & System Management API



- Start, stop, restart all EdgeX microservices
- Get the configuration settings (aka properties) for a microservice
- Get the memory usage for microservices
- Provide a REST API for 3rd parties to manage EdgeX

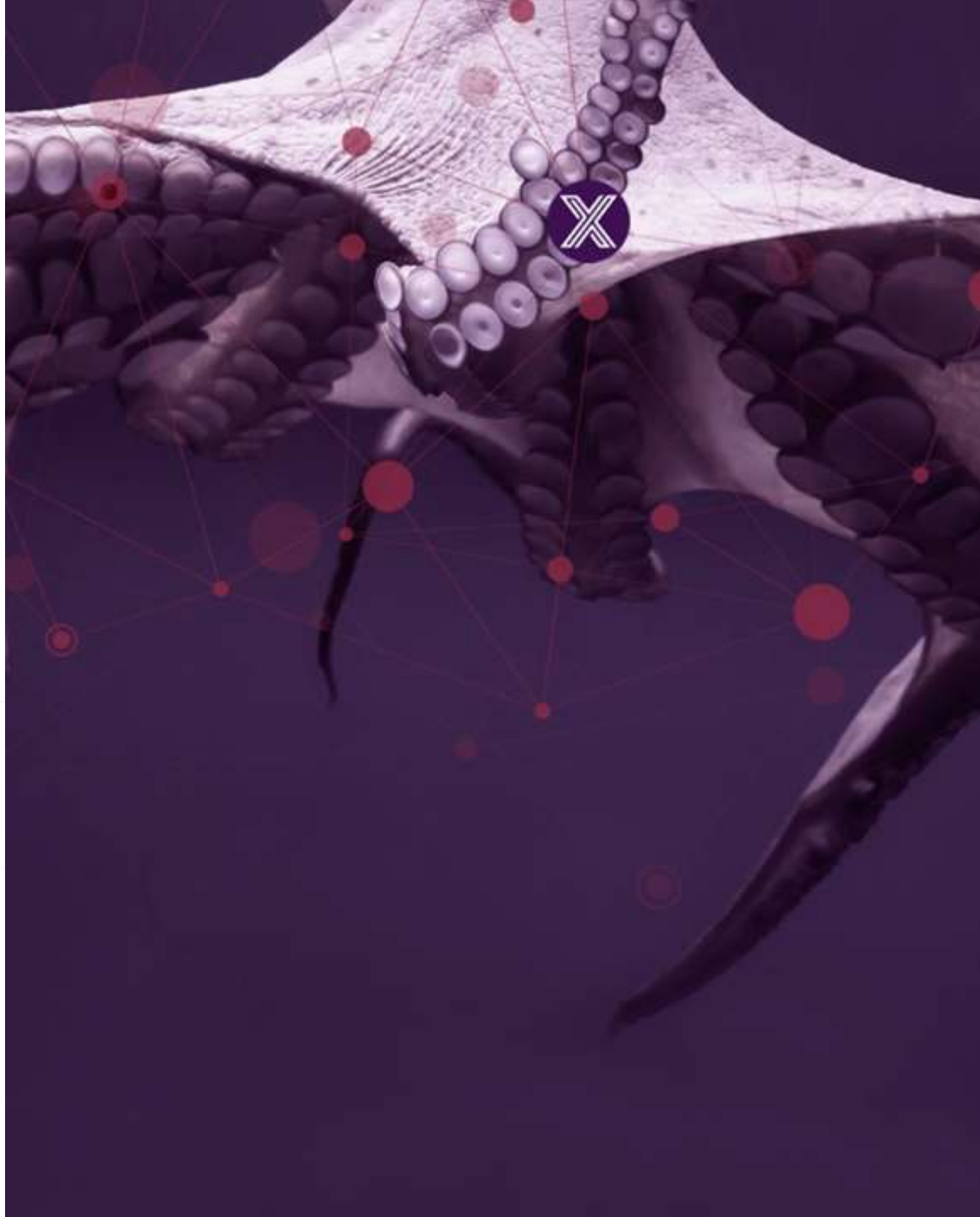
EdgeX & IIC



California release is ready for prime time

EDGEXFOUNDRY™

Next Steps



Key Project Links

Access the code:
https://github.com/edgexfoundry

Access the technical documentation:
https://docs.edgexfoundry.org/

Access technical video tutorials:
https://wiki.edgexfoundry.org/display/FA/EdgeX+Tech+Talks

EdgeX Blog:
https://www.edgexfoundry.org/news/blog/

Join an email distribution:
https://lists.edgexfoundry.org/mailman/listinfo

Join the Rocket Chat:
https://chat.edgexfoundry.org/home

Become a project member:
https://www.edgexfoundry.org/about/members/join/

LinkedIn:
https://www.linkedin.com/company/edgexfoundry/

Twitter:
https://twitter.com/EdgeXFoundry

Youtube:
https://www.youtube.com/edgexfoundry

IIC & EdgeX Next Steps

- Continue to find events like these to educate / exchange on the consortia and products
  - Guidance on how to get a regular cadance
- Test beds
  - Wanxiang Group led IIC test bed with EdgeX
  - Open to and looking for additional participation
- Promote the creation of (or participation in existing) IIC Contribution Groups or Task Groups
  - HW Root of Trust
  - System Management
- Participate in current or upcoming best practices guides
  - Providing implementers feedback

EDGEXFOUNDRY™

Thank You