



EDGE X FOUNDRY™

# Project Introduction

*Austin IoT Meetup – November 28, 2017*

# About Us




- **Jim White of Dell**
  - Dell Distinguished Engineer & Senior Software Architect
  - Project Fuse Chief Architect, now EdgeX TSC member and Core WG Chair
- **Alan Hansford of RFMicron**
  - VP Marketing & Business Development, RFMicron
- **David Rosado of AMD**
  - Segment Marketing Manager, AMD Embedded Solutions Team
- **Matt Locke of Linaro**
  - IoT and Embedded Director & former member of Linaro Board of Directors
- **Mike Woster of the Linux Foundation**
  - CRO/COO & Founding Executive Team Member

# About You

- With a raise of hands...
  - How many of you are actively engaged in IoT projects/work today?
  - How many of you are engineers/developers?
  - How many of you manage engineers/developers?
  - How many of you are involved in marketing/business development efforts?
  - How many come from the OT side of the business?
  - How many of you are familiar with the Linux Foundation & LF projects?
  - How many of you are familiar with EdgeX Foundry?

# 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



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

## Application Environments



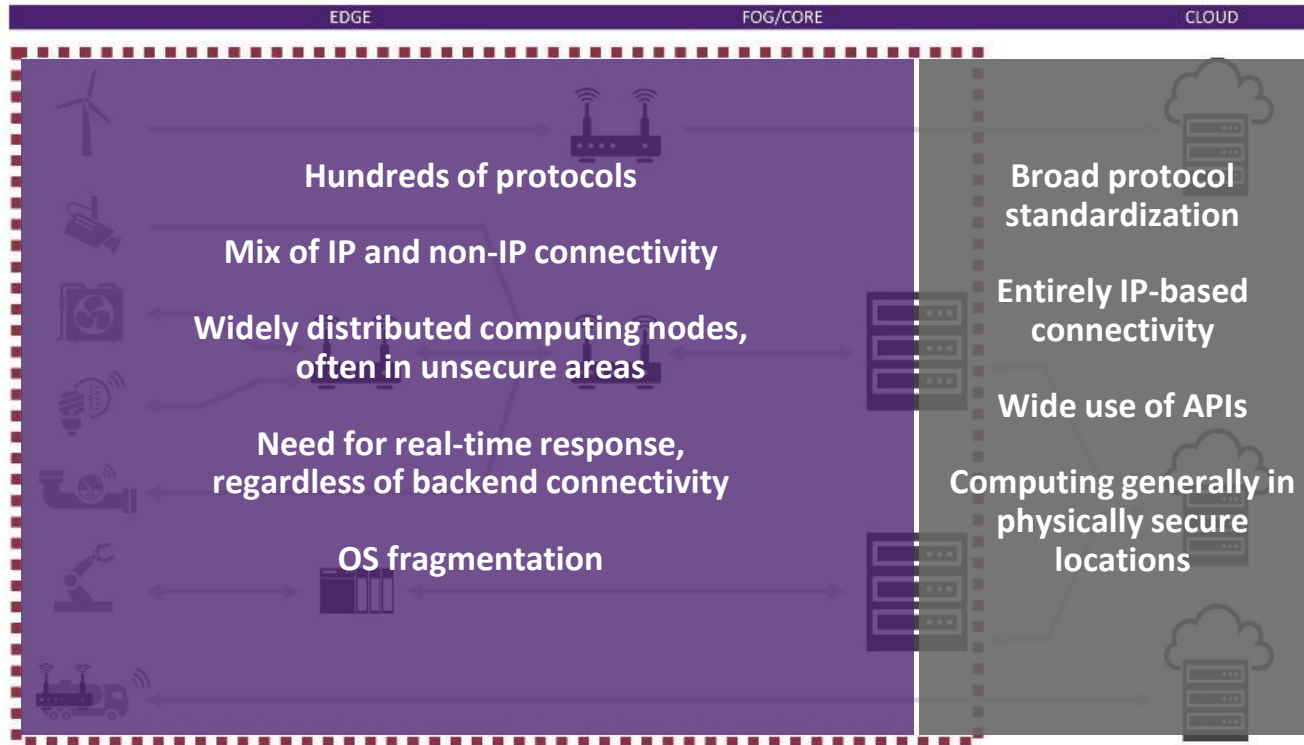
Variable preferences for coding and application environments among developers

## Operating Systems

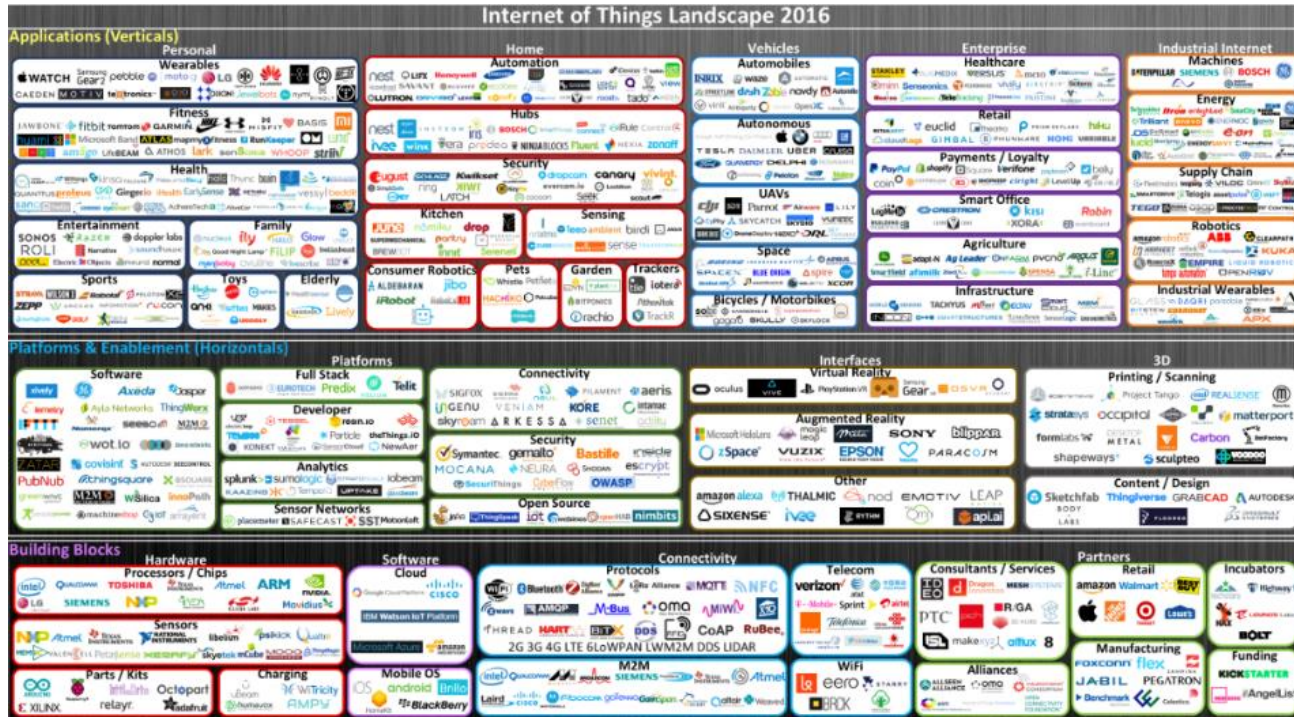


No line of sight to consistent choices across Linux, Windows and embedded/RTOS variants

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



# Today's Fragmented IoT Ecosystem

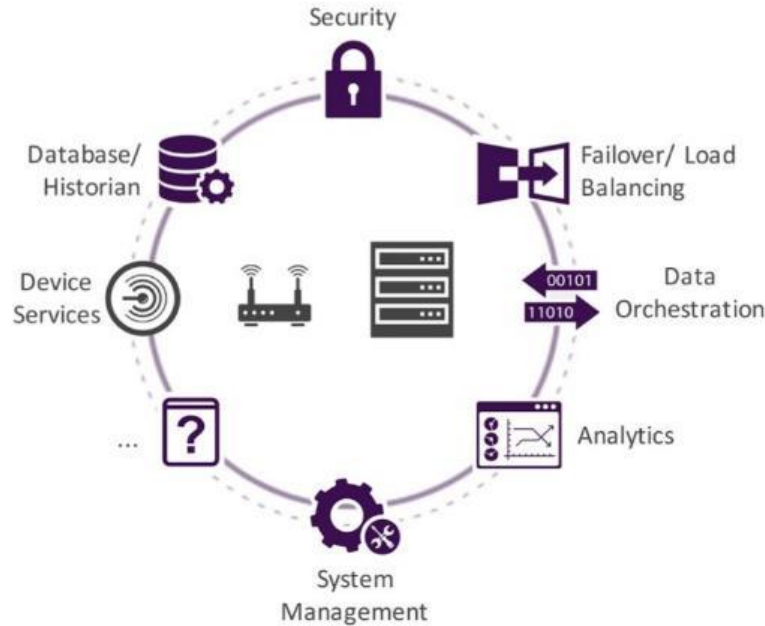


# Bridging Standards with An Ecosystem of Applications

Any Combination of Standards

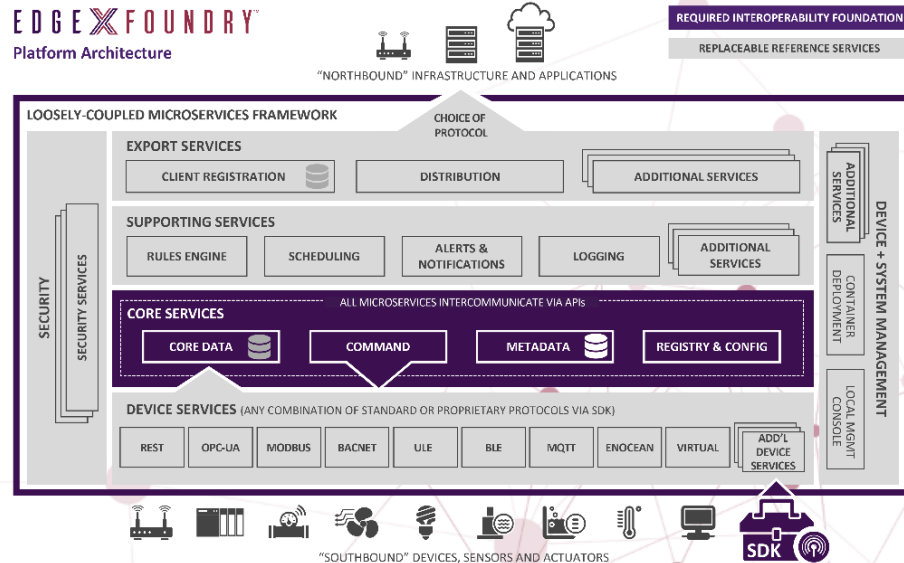
Interoperable Edge Applications

Choice of Backend Applications



# It's All About the APIs

- Loosely-coupled microservices bound by common APIs established through vendor-neutral collaboration in Linux Foundation
- HW- and OS- agnostic
- Polyglot: microservices can be written in any programming language (e.g. Java, Python, Go Lang, C) and deployed in containers or VMs
- Curating in open source community as a full reference platform surrounding the core interop framework allows granularity in API definition for key functions
- **Once key APIs are established, entire subsections can be replaced, combined, etc. with proprietary, differentiated “EdgeX-compliant” versions, even Core Services**

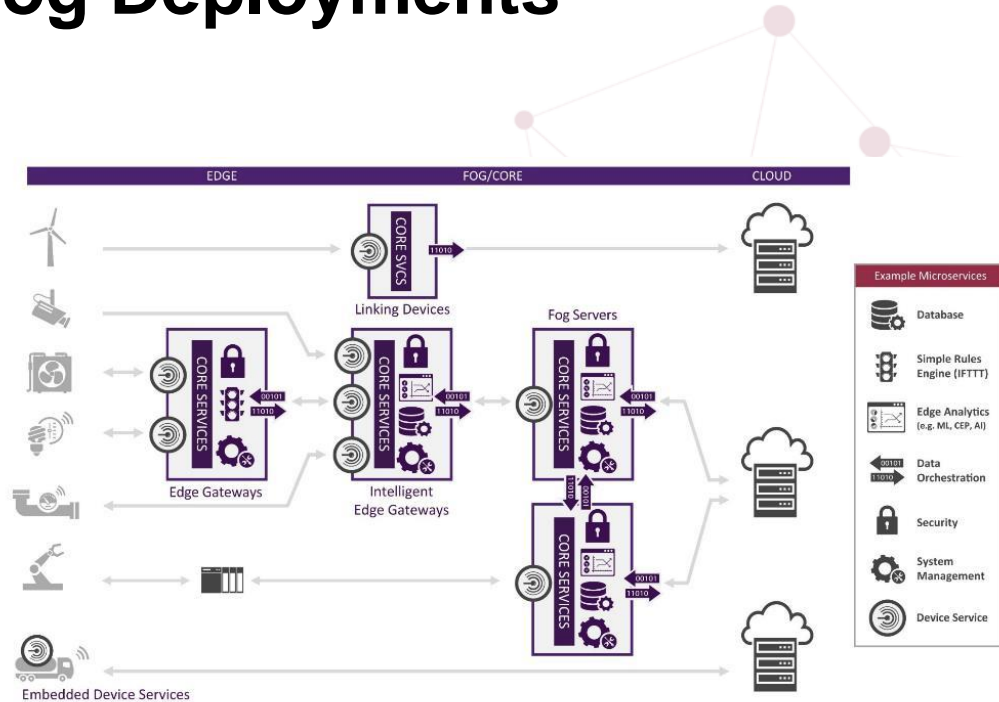


EdgeX is architected to enable commercial value-add around a lowest common denominator interoperability framework.



# EdgeX Enables Tiered Fog Deployments

- 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 microservices deployed on a given node depends on the use case and capability of hardware
- Discrete Device Services can be run on capable smart sensors/systems and communicate directly with other backend systems, including the cloud



# Basics of how EdgeX works

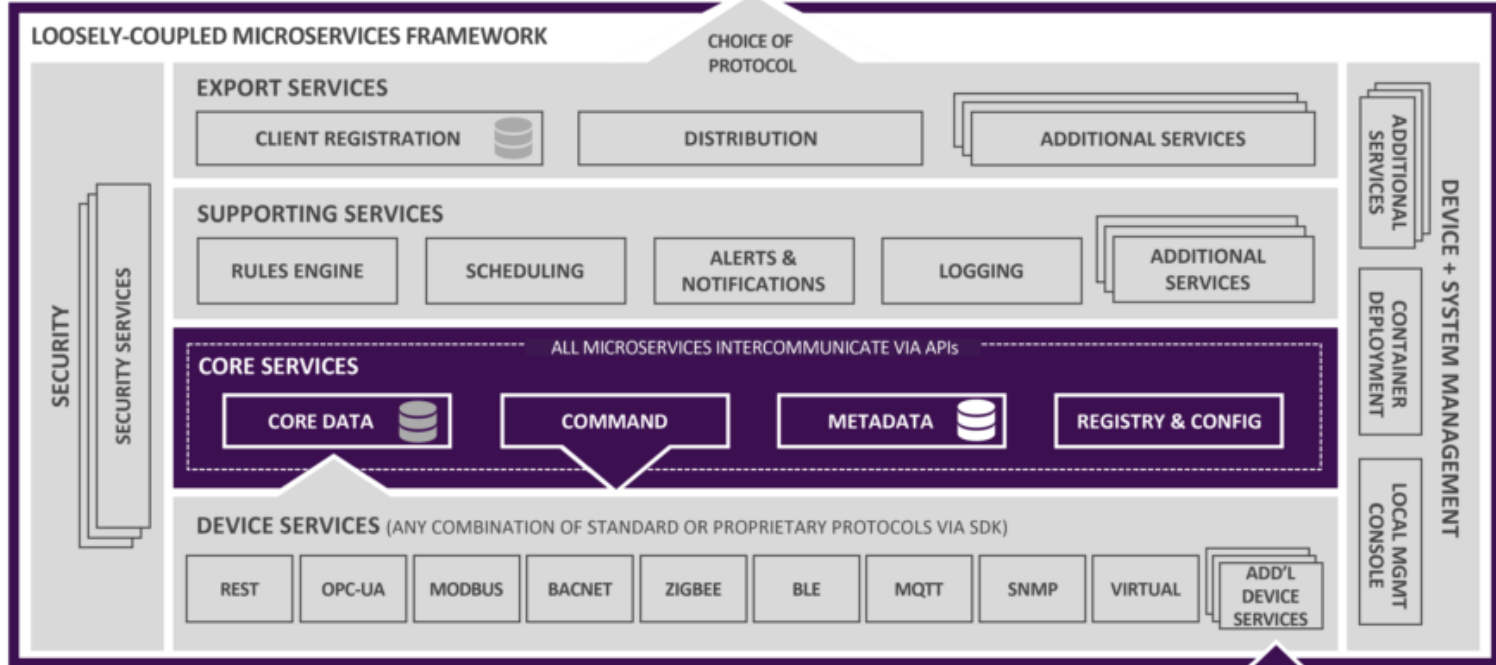
- A collection of a dozen+ micro services
  - More than a dozen micro services contributed to open source today
  - Several commonly used library projects (common domain objects, client libraries, etc.)
  - Automated build process
    - To include micro service containerization
    - Unit and integration tests per service
    - Black box testing to check the service level APIs
- EdgeX data flow:
  - Sensor data is collected by a device service from a thing
  - Data is passed to Core Data service 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
- 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



“NORTHBOUND” INFRASTRUCTURE AND APPLICATIONS

REQUIRED INTEROPERABILITY FOUNDATION

REPLACEABLE REFERENCE SERVICES



“SOUTHBOUND” DEVICES, SENSORS AND ACTUATORS



# Microservices

- Export and application services
  - Transport edge data to the enterprise and cloud systems in a manner they request
- Supporting services
  - Normal software application duties such as logging, scheduling, and data clean up
- Core services
  - Offers temporary persistence of edge data and facilitates actuation of things through common API
  - Knowledge warehouse and management of connected things
- Device services (and SDK)
  - Communicate with the devices, sensors, actuators, and other IoT “things” and objects through protocols native to the IoT object
  - Converts the data produced and communicated by the IoT object, into a common EdgeX Foundry data structure

# Benefits to Key IoT Stakeholders

## Hardware OEMs

(Examples: Controllers, Hubs, Routers, Gateways, Servers)



Scale faster with an interoperable partner ecosystem and more robust security and system management

## ISVs



Interoperate with 3rd party applications and hardware without reinventing connectivity

## Sensor/Device Manufacturers



Write a device driver with your selected protocol once using the SDK and get pull from all Solution Providers

## System Integrators

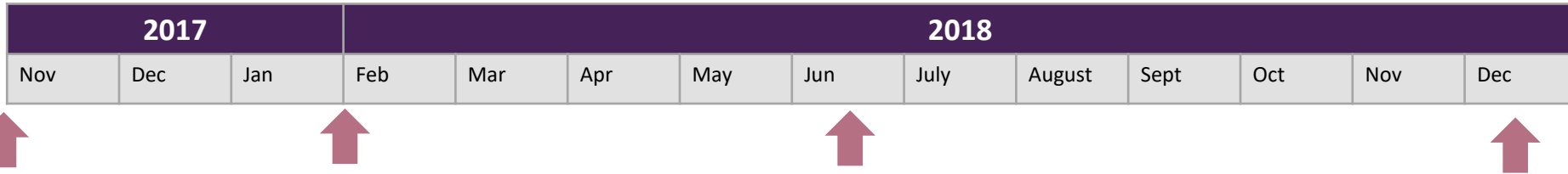


Get to market faster with plug-and-play ingredients combined with your own innovations

**End Customers: Less confusion and faster time to ROI**

# 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.



## 'Barcelona' Release

(Released Oct 20 2017)

- Improved fit and finish, formalized Core Service APIs, additional Device and Export Services, test apparatus
- Intended to ramp developer engagement and PoCs with end customers for feedback and further hardening

## 'California Preview'

(Jan 2018)

- Drop-in Go Lang microservice replacements demonstrating reduced footprint and higher performance
- Stretch goal: C-based Device Service SDK

## 'California' Release

(June 2018)

- First integration of security and manageability APIs
- Improved fit and finish, more Export and Device Services
- Intended as first product-quality OSS foundation for commercial differentiation and field deployments

## 'Delhi' Release

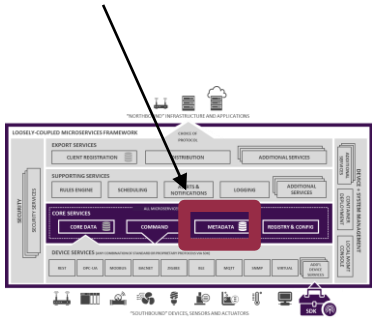
(Dec 2018)

- Additional security and manageability extensions
- High performance message bus option
- Extensions for distributed/ fog computing
- Beginning of EdgeX certification program

# Example of Lightweight Microservice Alternative

Preliminary performance and footprint numbers from first full Go Lang-based microservice alternative for the January 'California' preview are indicative of a major reduction in the overall baseline EdgeX footprint as the code base evolves. Microservice drops in with no changes to the rest of the architecture or APIs.

Drop-in replacement  
for Core Metadata  
microservice, written  
in Go Lang language

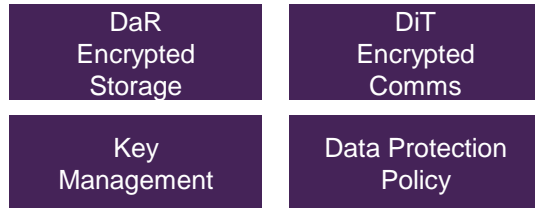


<i>Measure</i>	<i>Go Lang Alternative</i>	<i>Current Java Implementation</i>
Executable/JAR Footprint	11.7 MB	42.4 MB
<b>Container Footprint</b>	<b>17.2 MB</b>	<b>153 MB</b>
<b>Memory Usage (On Startup)</b>	<b>4.3 MB</b>	<b>221 MB</b>
<b>Memory Usage (Under Load)</b>	<b>9.2 MB</b>	<b>230 MB</b>
CPU Usage (Steady State)	0.15%	0.30%
CPU Usage (Under Load)	5.0% - 15.0%	6.0% - 15.0% (90% spike at heavy load)
<b>Startup Speed</b>	<b>0.14 Seconds</b>	<b>12.55 Seconds</b>
Response Speed (Ping)	0.0011 Seconds	0.0022 Seconds
Response Speed (Post)	0.0091 Seconds	0.0137 Seconds
Response Speed (Get)	0.0038 Seconds	0.0062 Seconds

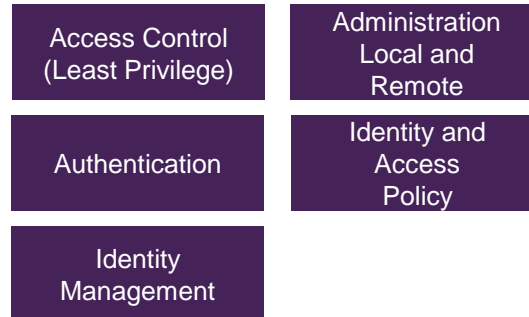
# Planned EdgeX Security Modules

- Modules in three main categories to be built as part of the EdgeX foundation
- Phased development approach, based on priority for securing overall stack
  1. Define standards to leverage and API requirements for each module
  2. Develop lean reference implementations for the open source code
- As with rest of platform, security reference modules will be replaceable with proprietary value-add

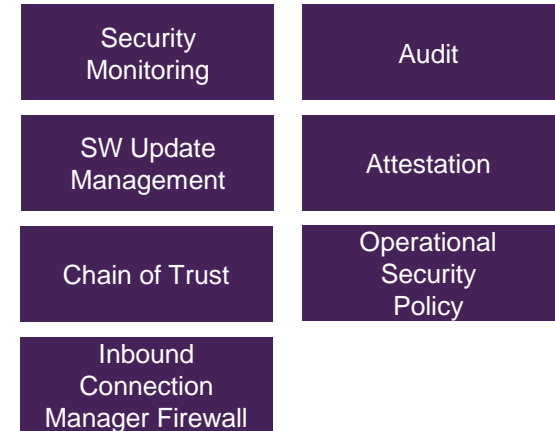
## Data Protection



## Identity and Access Management

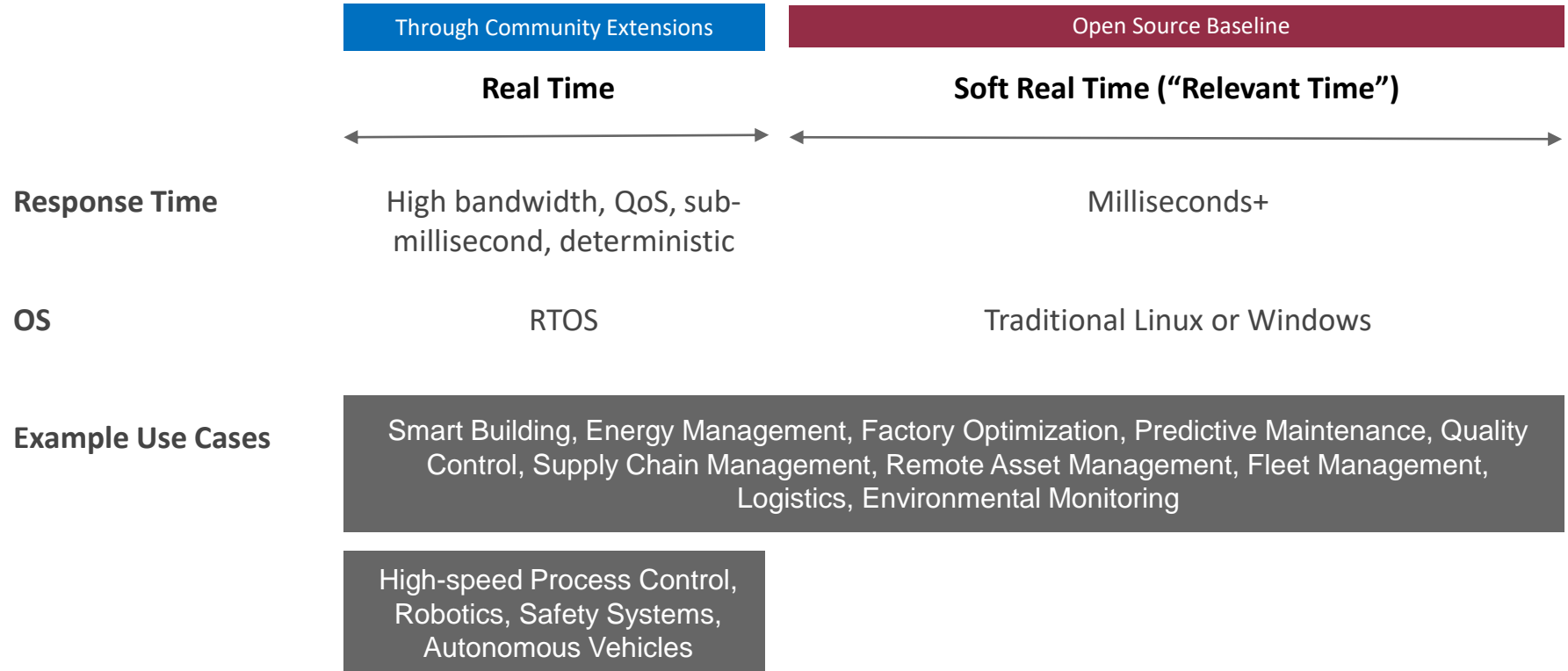


## Operational Security Management





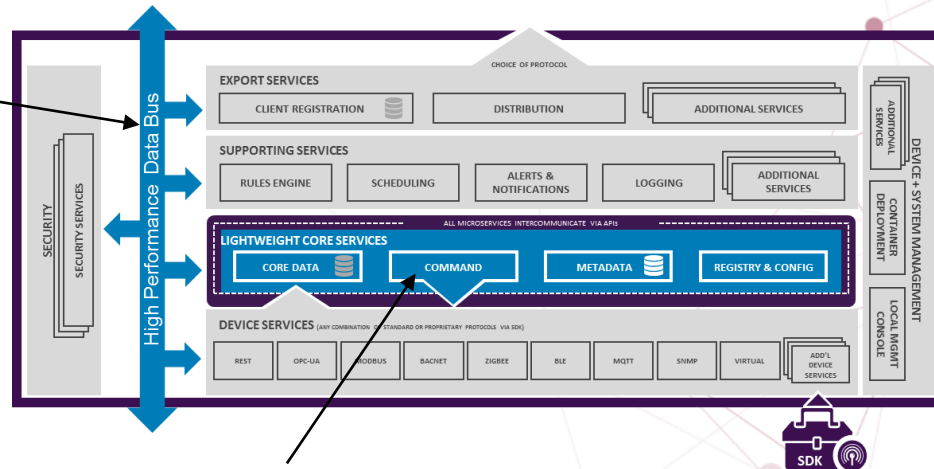
# Real Time Enabled Via Code Extensions



# Opportunities for Proprietary Performance Extensions

- Targeted at real time use cases requiring low footprint (e.g. <10MB), high streaming bandwidth, and predictable low latency cycle times (e.g. <10ms)
- Enabled via lightweight, EdgeX-compliant instances of Core Services and pluggable high performance data/message bus for intercommunication between microservices
- Fully compatible with baseline version of EdgeX to benefit from broader ecosystem

EdgeX-compliant high-performance data bus as an alternative to baseline REST intercommunication between services



Lightweight, high performance version of Core Services leveraging the key EdgeX APIs at the perimeter. Entire core could even be compressed to a proprietary C-based binary and still benefit from plug-in EdgeX-compliant services from the community.

# What's with that 'X'?

Fundamental goal of the EdgeX project is to provide a stable, product-quality open source foundation for interoperable commercial offers

- The 'X' in EdgeX allows the project name to be trademarked for use as a certification mark
- A certification program will be established in the project for commercial offerings to verify that key EdgeX interoperability APIs were maintained alongside proprietary value-add
  - Initial program launch targeted for 'Delhi' release at the end of 2018 with ramp in 2019
- Stability for key elements (e.g. core APIs and certification process) is maintained through the EdgeX Technical Steering Committee (TSC) and clear versioning system
- Licensed under Apache 2.0, anyone can leverage the EdgeX code base as a foundation for their commercial offerings
  - Can be a full EdgeX-compliant IoT platform, value-added plug-in microservice(s) or a services model

# Backed by 60+ Members



With more in process!

# EdgeX Foundry Governance Structure

EDGEX FOUNDRY MEMBER COMPANIES (60+)

## GOVERNING BOARD (GB)

*Composed of appointed and elected individuals; manages the business of the EdgeX Foundry.*

## TECHNICAL STEERING COMMITTEE (TSC)

*Leads the technical work of EdgeX Foundry. Oversees and aligns working groups.*

WG

WG

WG

WG

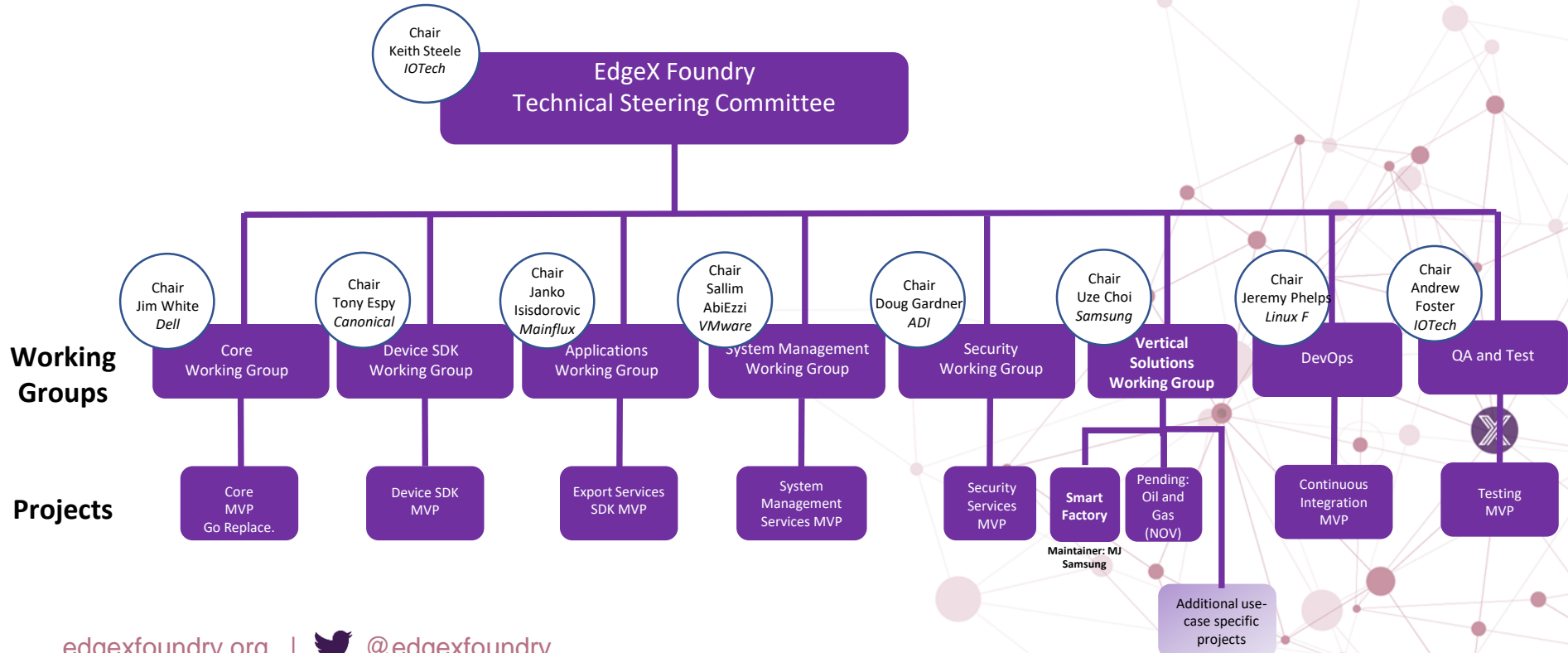
WG

## CERTIFICATION COMMITTEE

*Develops and oversees the certification program for EdgeX Certified components.*

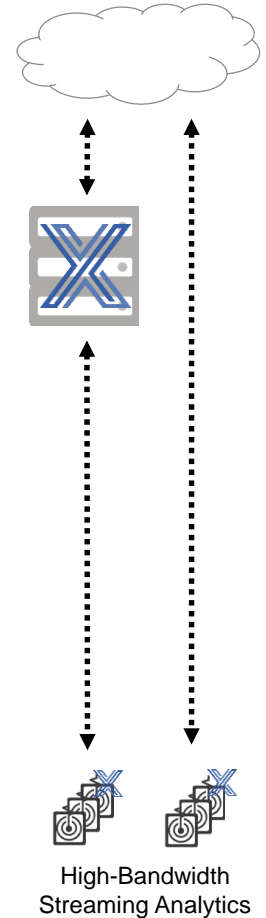
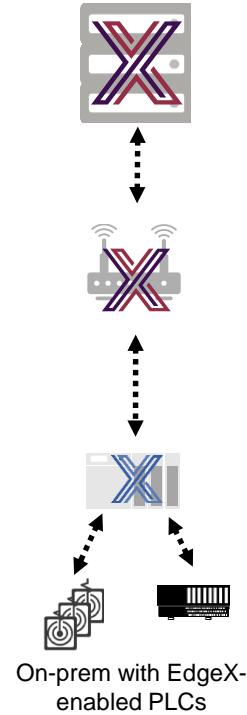
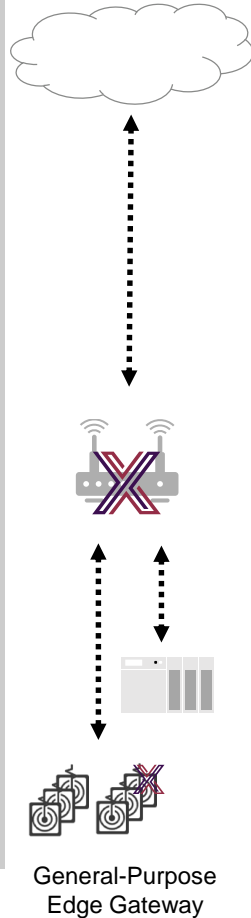
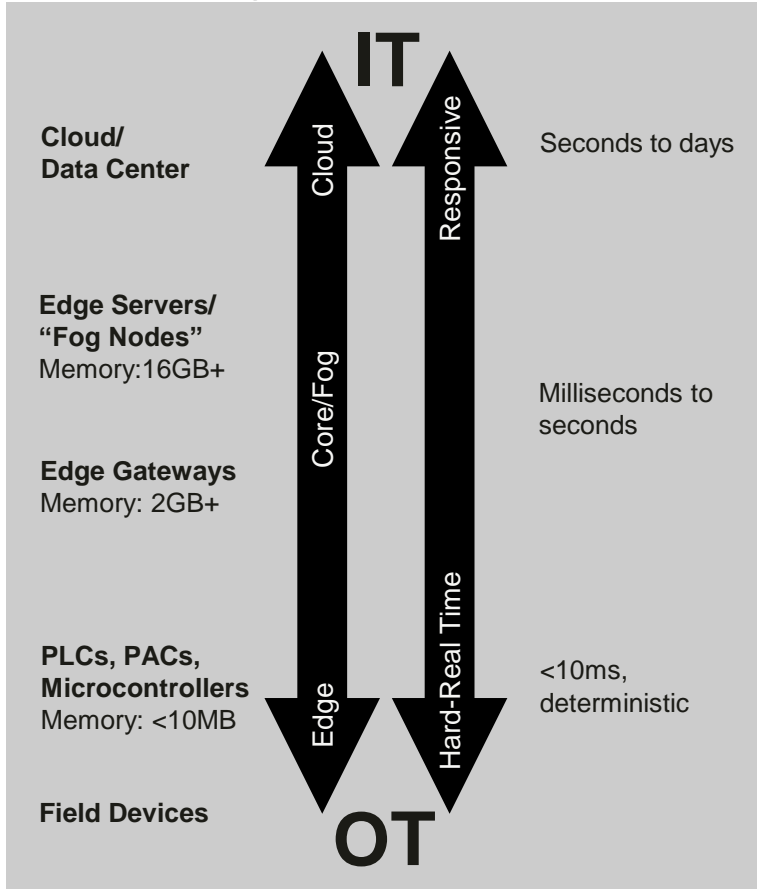
LF SUPPORT TEAM

# EdgeX Project Organization



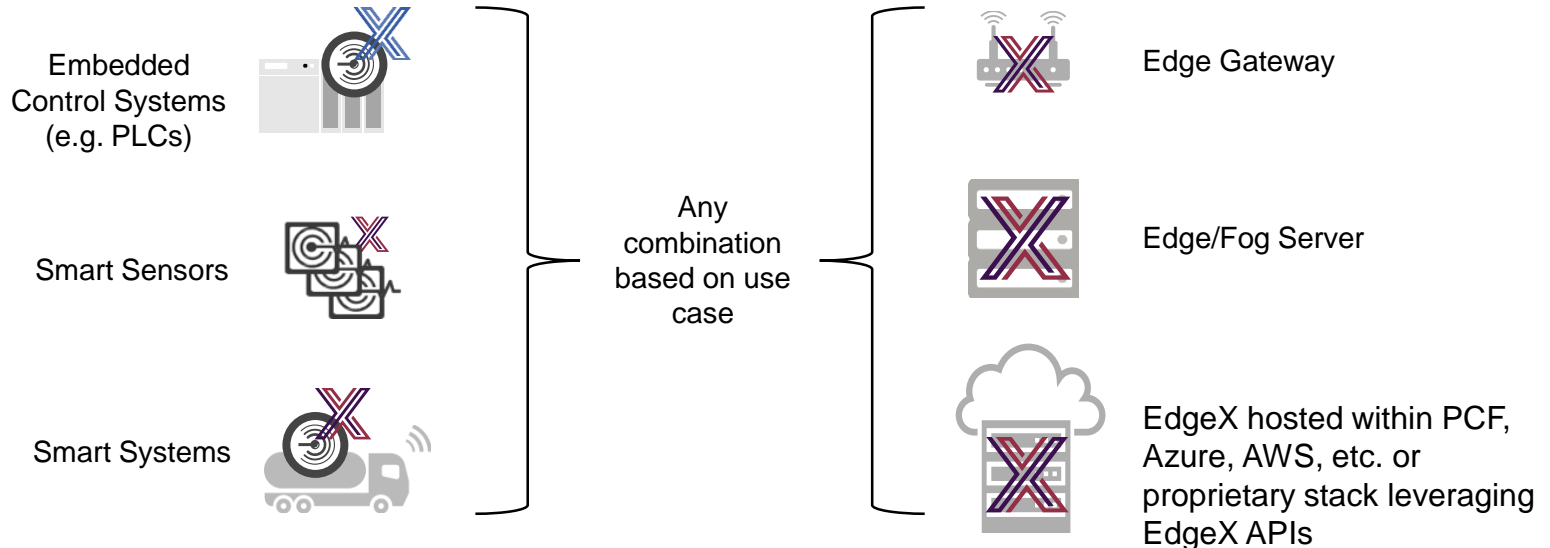
# Summary of Example Use Cases

Open Source Baseline  
 Proprietary EdgeX-compliant Extensions



# Embedded Device Services

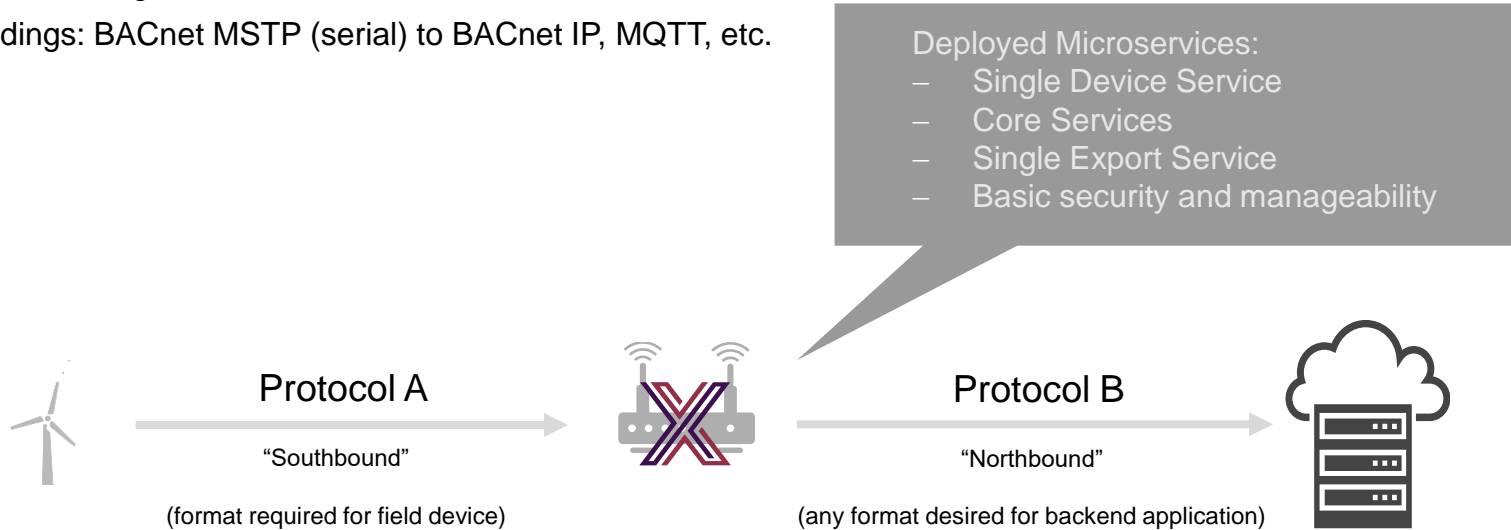
- Planned work will enable C-based Device Services to be embedded in constrained microcontrollers running a RTOS for real-time use cases (e.g. within a smart sensor or PLC)
- Due to loosely-coupled architecture, baseline EdgeX-compliant Device Services can be deployed directly on smart sensors or systems capable of hosting a microservice (via container or VM)
- IP-capable sensors with an EdgeX Device Service / APIs can communicate directly with Core Services running on any other compute node such as a gateway, server or directly to the cloud





# Simple Linking Device

- A minimal deployment of EdgeX can function as a linking device which simply converts one protocol into another
- Typical protocol combinations vary by vertical and installation, some typical examples:
  - Energy: DNP3 to MQTT, Modbus to REST
  - Manufacturing: Profibus to OPC-UA
  - Buildings: BACnet MSTP (serial) to BACnet IP, MQTT, etc.

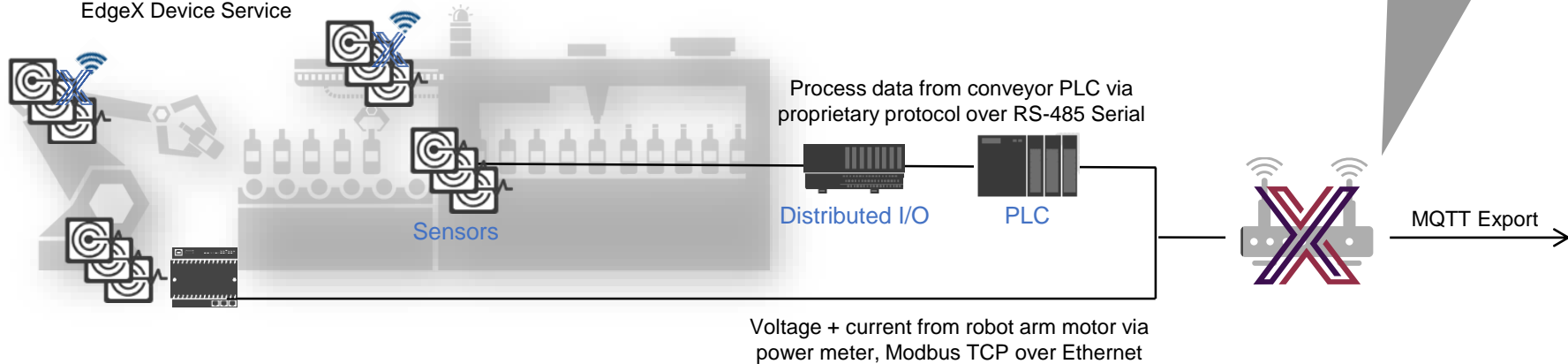


# Full Edge Gateway Stack in Manufacturing

## Deployed Microservices:

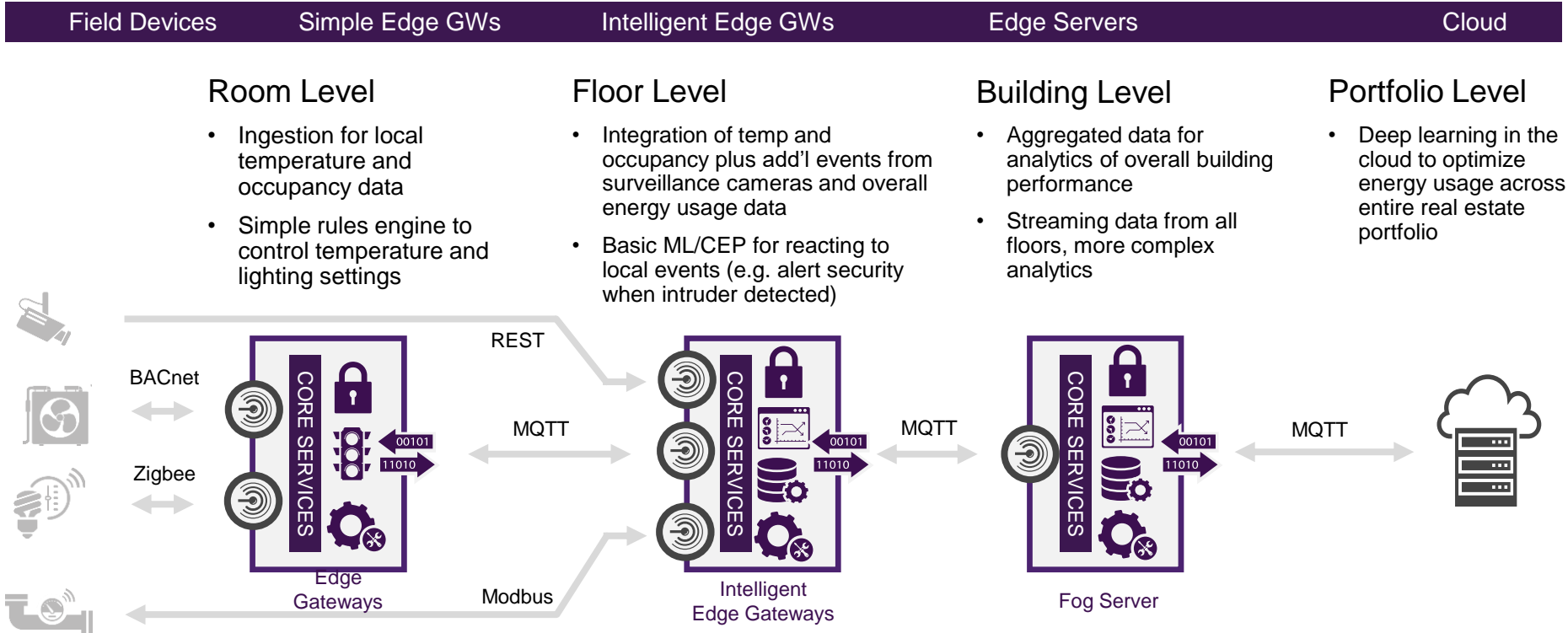
- Multiple Device Services for data ingestion and control across heterogeneous protocols
- Local database for buffering during periods of lost connectivity
- 3<sup>rd</sup> party CEP for edge analytics
- Various security services
- 3<sup>rd</sup> party remote management console
- MQTT Export Service

Temperature + vibration  
via BLE Sensors with  
vendor-embedded  
EdgeX Device Service



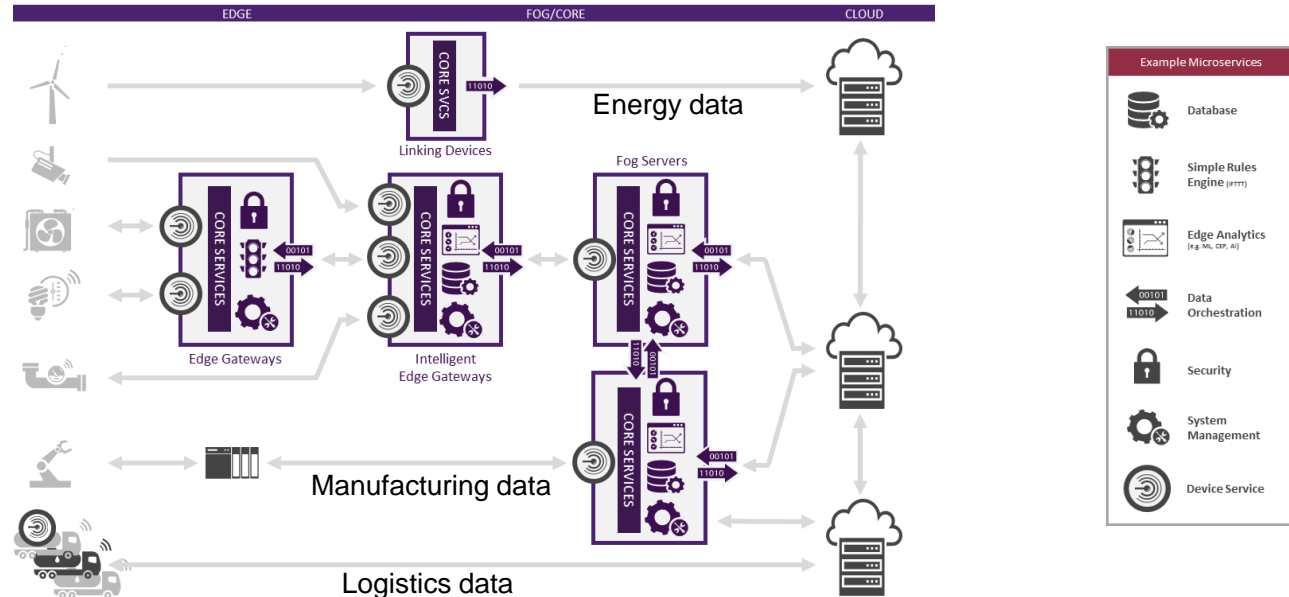
# Tiered Deployment in Smart Buildings

Number of deployed microservices and functionality increases higher in tier



# Distributed (e.g. ‘Fog’) Computing

- Introducing specific microservices to address QoS, failover between nodes, redundancy, “east-west” communication, multi-tenancy, etc.
- Workloads deployed dynamically at different tiers to optimize performance and results
- In a manufacturing example, data can be coordinated for manufacturing process, building performance energy usage and logistics across various buildings, plants and trucks
- Multi-tenancy for different service providers addressing unique business needs with the same infrastructure



# Key Project Links

Access the code:

<https://github.com/edgexfoundry>

Access the technical documentation:

<https://wiki.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:

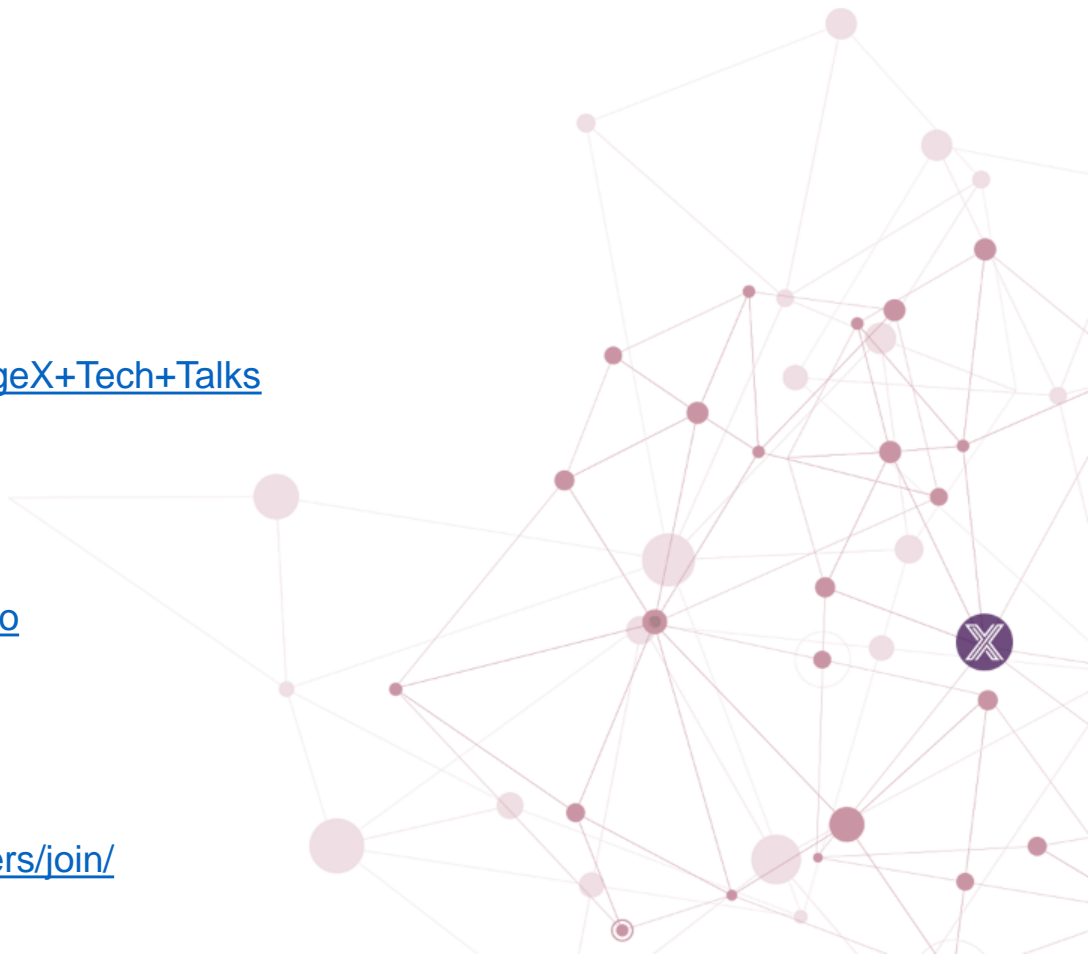
<https://lists.edgexfoundry.org/mailman/listinfo>

Join the Rocket Chat:

<https://chat.edgexfoundry.org/home>

Become a project member:


<https://www.edgexfoundry.org/about/members/join/>





# Panel Discussion

# Our Panel

- Alan Hansford of RFMicron → 
  - VP Marketing & Business Development, RFMicron
- David Rosado of AMD → 
  - Segment Marketing Manager, AMD Embedded Solutions Team
- Matt Locke of Linaro → 
  - IoT and Embedded Director & former member of Linaro Board of Directors
- Mike Woster of the Linux Foundation → 
  - CRO/COO & Founding Executive Team Member



Demos

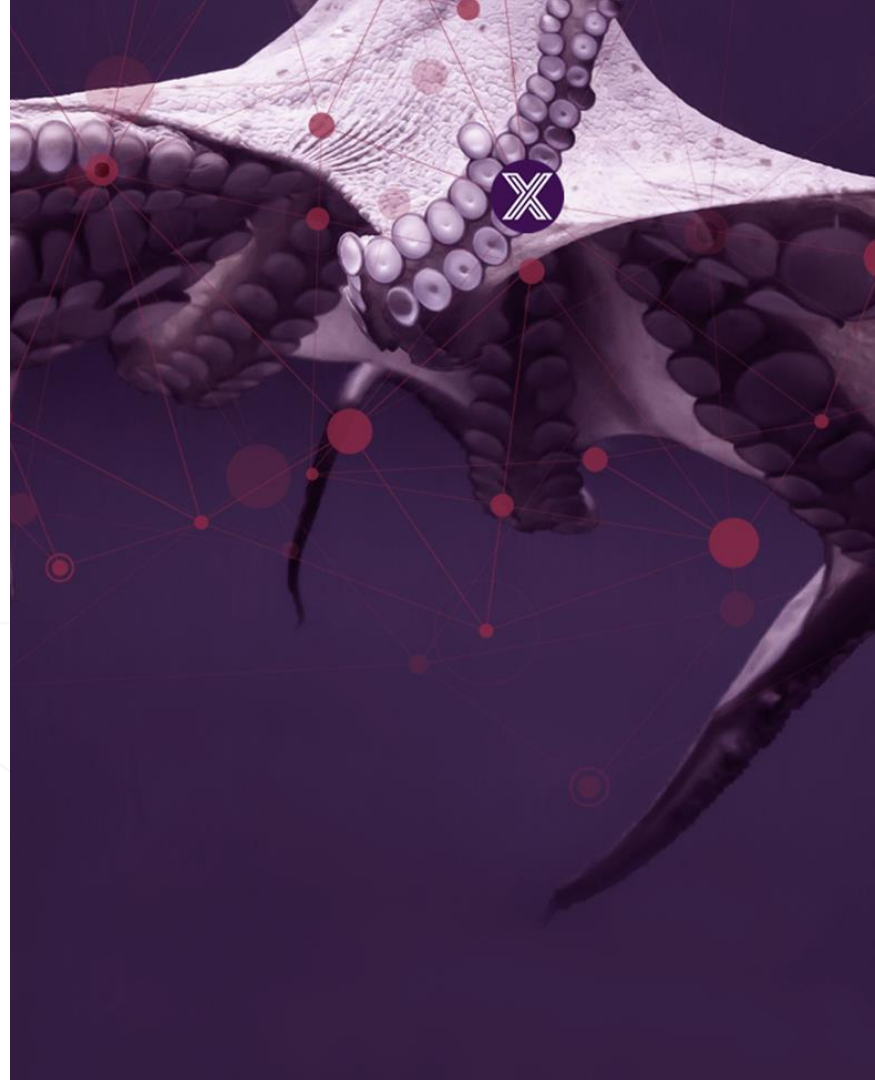




Q&A

EDGE X FOUNDRY™

Thank You!



Backup

# EDGE X FOUNDRY™

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 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

# Engagement Options

- Project is a technical meritocracy. Anyone can contribute to or use the EdgeX Foundry code for free.
- Technical Steering Committee (TSC) and Working Group (WG) meetings are open to the public
- TSC and WG Chairs in addition to code committers and maintainers are voted in based on technical acumen and alignment to project tenets. This ensures robustness and stability in the architecture, technology choices, roadmap and code base.
- Joining as a paid project member affords maximum influence over project direction

# EdgeX Foundry Goals

- Build and promote EdgeX as the **common open platform** unifying Internet of Things (IoT) edge computing.
- Enable and encourage the rapidly growing community of IoT solutions providers to **create an ecosystem of interoperable plug-and-play components** around the EdgeX platform architecture.
- **Certify EdgeX components** to ensure interoperability and compatibility.
- Provide tools to quickly create EdgeX-based IoT edge solutions that can **easily adapt to changing business needs**.
- Collaborate with relevant open source projects, standards groups, and industry alliances to **ensure consistency and interoperability across the IoT**.

# Member Benefits

- Additional influence to shape the overall platform architecture to enable commercialization needs
- Recognition for Industry thought and technology leadership
- Marketing and networking within the EdgeX Project for business opportunities (effectively a vendor-neutral partner program)
- Discounted sponsorships at Linux Foundation and EdgeX Foundry-produced events (e.g. trade shows, hackathons, etc.)
- Learning and engagement

# Project Membership Options

- **Platinum**

- Appoint one (1) representative to the EdgeX Governing Board (GB)
- Appoint one (1) representative as a voting member in any subcommittees or activities of the GB
- Appoint one member to the start-up TSC (6 month position)
- Enjoy most prominent placement in displays of membership
- Access to LF's invitation-only Open Source Leadership Summit
- Ongoing, individual engagement with EdgeX executive director and staff
- \$150,000 annually

- **Silver**

- Participate as one of three (3) Silver representatives to the EdgeX Governing Board
- Enjoy prominent placement in displays of membership
- Ongoing engagement with EdgeX executive director and staff
- \$2,500 to \$50,000 annually, depending on employee count

- **Associate (non-profits)**

- Limited to pre-approved non-profits, open source projects, and government entities
- Entitled to identify their organization as members supporting the mission of EdgeX and any other rights or benefits as determined by the Governing Board



# Goals for EdgeX Open Source Baseline

- Baseline code for a variety of general purpose use cases to drive common APIs
- Core Services available in various language implementations (e.g. Java, Go Lang)
- Foundational security and system management APIs to enable proprietary value-add, reference services for encryption, certificate management, container deployment, etc.
- Reference services for local database (e.g. MongoDB), simple edge analytics (e.g. Drools rules engine) and “housekeeping” functionality (e.g. logging, alerts)
- Device and Export Service SDKs in several supported languages (e.g. Java, Go Lang)
- Reference device and export services for popular OT and IT connectivity (e.g. OPC-UA, BACnet, Modbus, CAN bus, Zigbee, DDS, MQTT, REST/HTTP) and Clouds (e.g. Microsoft Azure, AWS, Google IoT Core, SAP HANA, IBM Watson IoT)
- Validated on multiple hardware and OS combinations
- Development tools (e.g. dev kits and management GUIs)
- Test and certification framework
- Potential industry-specific extensions

# Trajectory of Baseline EdgeX Footprint

- Initial EdgeX incubation was focused on driving architectural alignment, not performance
- Microservices currently implemented mostly in Java with some Javascript and Python
- Overall non-optimized footprint ranges from 2.5-4GB memory if all microservices (~18) are deployed
- Architecture supports lightweight alternatives for key microservices that leverage the same APIs
- Alternative implementations for Core Services, Device Service SDK and Export SDK based in Go Lang in process (to be released in January 'California Preview'). See example of benefits on next page.
- Alternative baseline code implementations will reduce footprint of full deployment to <512MB memory utilization which will draw in general ARM community and makers
- Ultimate footprint is dependent on number of deployed microservices for a given use case
- Further footprint reductions can be enabled by licensed embedded versions (EdgeX members are already targeting C-based implementations for real-time response in PLCs in <10MB of memory)

Footprint for average deployment of basic gateway functionality is trending to <512MB memory by mid 2018 if not <256MB

# Project Progress since April launch

- Considerable Momentum, now over 60 ecosystem members with most recent additions of Samsung at Platinum and Thales and Cavium at Silver. More in flight.
- Technical Steering Committee (TSC), work groups and bi-annual release roadmap established
- First community release (dubbed 'Barcelona') delivered Oct 20
- Alliance recently announced with Industrial Internet Consortium (IIC) to collaborate on test beds, security and best practices
- Reference device services for Modbus, BACnet, SNMP, BLE, MQTT, and Serial. Samsung currently developing OPC-UA.
- Progress on foundational security and manageability APIs (first implementation in June 2018 'California' release)
- Work underway for Go Lang microservice alternatives which reduce the current footprint & boot times by order of magnitude. Some Go microservices already done or nearing completion. All will be released to open source in January 2017 'California Preview'.

# Project Progress since April launch (cont'd)

- Growing community interest with ramping contributions. For example, recent Schlumberger contribution of a Google IoT Core Export Service and Cavium ported the current Docker containers to ARM.
- Vertical Solution Working Group formed, chaired by Samsung. Samsung spinning up a ‘Smart Factory’ project and National Oilwell Varco one for Oil and Gas. Others will be added for key use cases to provide requirements back to project and develop and deploy test beds.
- Work started for a message bus option between services (compared to current REST), plus community members working on commercial hard real-time variants of the core
- Growing interest from end customers with multiple projects based on EdgeX targeting field pilot in mid-summer 2018 and production end of 2018

# Public Coverage on EdgeX

- [From Fuse to Foundry: The New Meritocracy in the Evolution of the IoT Edge](#)
- [EdgeX Foundry Is the Solution the IoT World Desperately Needs](#)
- [EdgeX Foundry and the Quest for Multivendor Interoperability](#)
- [EdgeX Foundry Unifies the IoT Marketplace to Accelerate Enterprise IoT Deployments](#)
- [Moor Insights White Paper](#)
- [Dell Blog: How to Bring Order to IoT's Spaghetti Junction?](#)

Much more public coverage of EdgeX can be found online