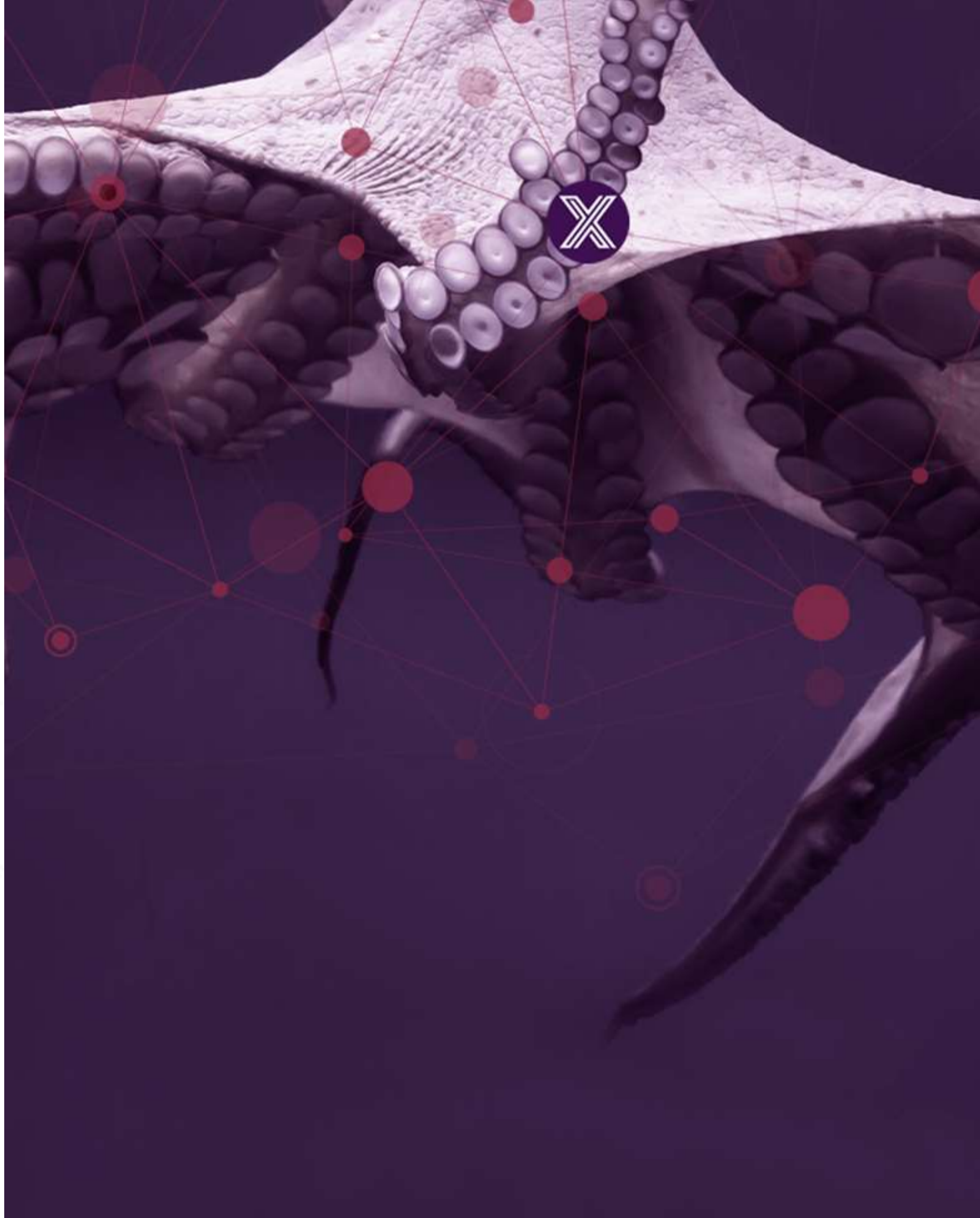
EDGEX FOUNDRY™

Introduction, Status & Roadmap

6/30/18



Agenda

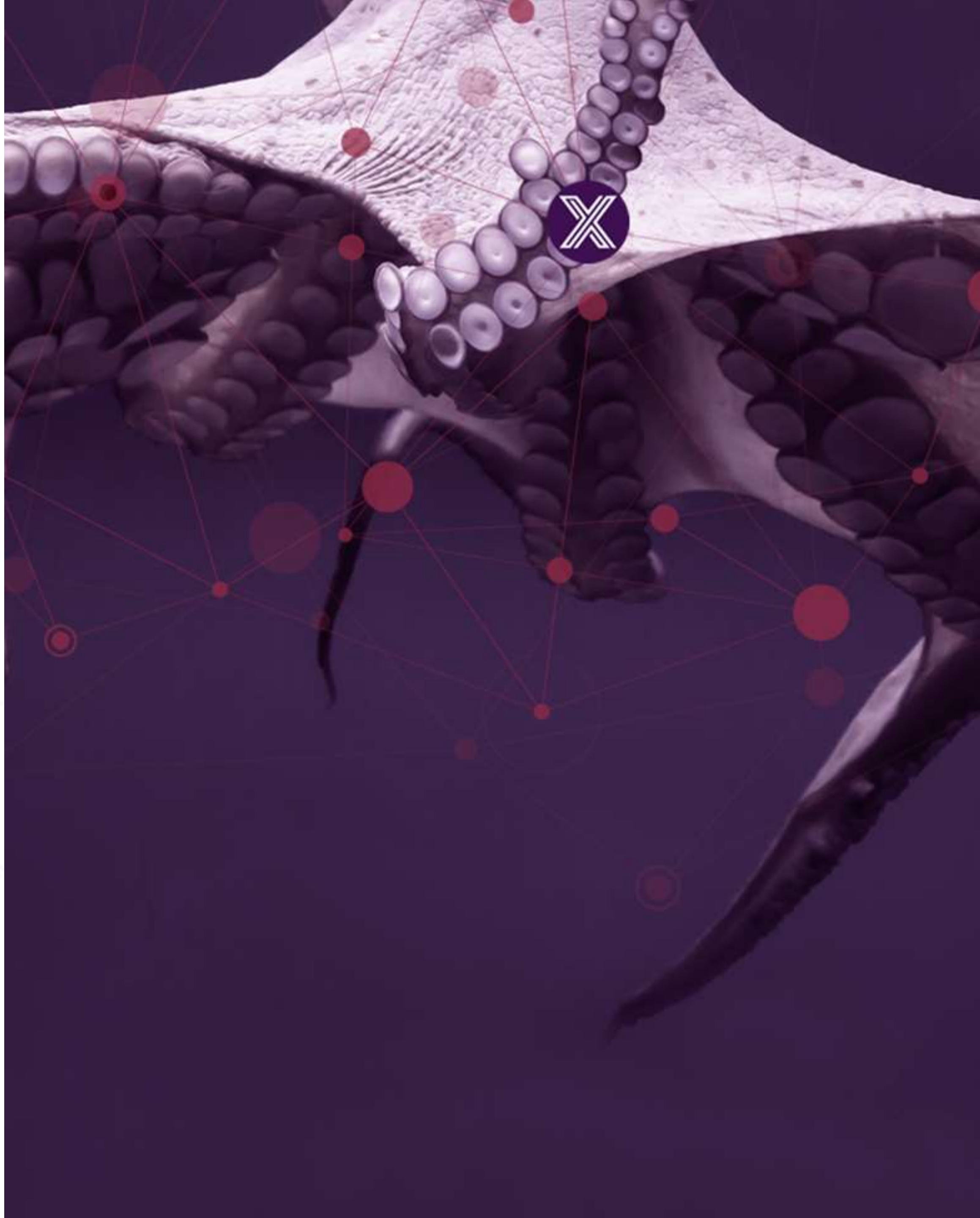
- History, rationale, & background on EdgeX
- How EdgeX works
  - Architecture and technology
- Current Status
  - Ecosystem and releases
  - Roadmap
- Dell Technologies Investment

Introduction

- Jim White
  - Dell Technologies IoT Solutions Division – Distinguished Engineer
  - Team Lead of the IoT Platform Development Team
  - Chief architect and lead developer of Project Fuse
    - Dell’s original IoT platform project that became EdgeX Foundry
    - Yes – I wrote the first line(s) of code for EdgeX (apologies in advance)
  - EdgeX Foundry …
    - Technical Steering Committee member
    - Ad hoc and unofficial lead architect

EDGEXFOUNDRY™

Architecture & Technology

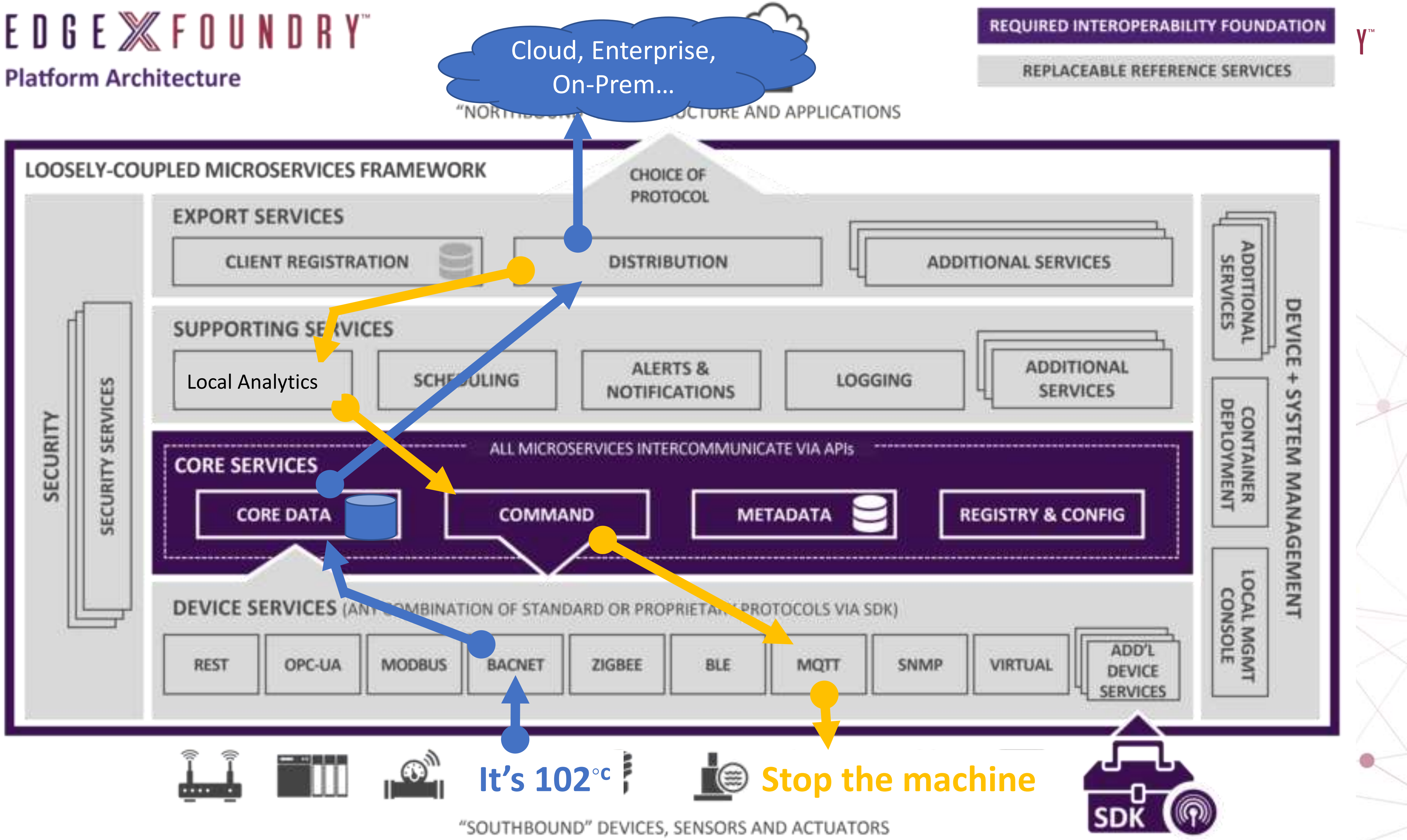


Introducing EdgeX Foundry

- An open source, vendor neutral project (and ecosystem)
- A **microservice**, loosely coupled software framework for IoT edge computing
- Hardware and OS agnostic
- Goal: enable and encourage growth in IoT solutions
  - The community builds and maintains common building blocks and APIs
  - Plenty of room for adding value and getting a return on investment
  - Allowing best-of-breed solutions

EdgeX Primer - How it works

- A collection of a dozen+ microservices
  - Written in multiple languages (Java, Go, C, … we are polyglot believers!!)
  - Several commonly used library projects (common domain objects, client libraries, etc.)
- 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
- REST communications between the service
  - Some services exchange data via message bus (core data to export services and rules engine)
- Microservices are deployed via Docker and Docker Compose



EDGE X FOUNDRY™
Platform Architecture
Cloud, Enterprise, On-Prem…
"NORTHBOUND" INFRASTRUCTURE AND APPLICATIONS
REQUIRED INTEROPERABILITY FOUNDATION
REPLACEABLE REFERENCE SERVICES
LOOSELY-COUPLED MICROSERVICES FRAMEWORK
CHOICE OF PROTOCOL
EXPORT SERVICES
CLIENT REGISTRATION
DISTRIBUTION
ADDITIONAL SERVICES
SUPPORTING SERVICES
Local Analytics
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
SECURITY
SECURITY SERVICES
ADDITIONAL SERVICES
CONTAINER DEPLOYMENT
LOCAL MGMT CONSOLE
DEVICE + SYSTEM MANAGEMENT
It's 102°C
Stop the machine
SDK
"SOUTHBOUND" DEVICES, SENSORS AND ACTUATORS

EdgeX Micro Service Layers

- Contextually, EdgeX micro services are divided into 4 layers
- Device Services (device services for various protocols like Modbus, BACnet, … and SDK to create new device services)
  - Communicate in native sensor/device protocol to the physical – that is to the IoT “things”
  - Transform sensor data to common format
  - Translate command requests to actuate devices (in native protocol/format)
- Core Services (core data, metadata, command & configuration/registration)
  - Offers temporary persistence of edge data and facilitates actuation of things through common API
  - Collect sensor data
  - Understand what sensors/devices are connected how to communicate with them (metadata)
  - Provision facility for new sensors/devices (and device services)
  - Manage device actuation requests to device services/devices
  - Provide micro service registry
  - Provide micro service configuration
- Supporting Services (logging, notifications, scheduler, rules engine)
  - Normal software application duties plus “edge intelligence”
  - Logging
  - Notifications and alerting
  - Scheduling and clean up
  - Rules engine
- Export Services (client, distribution)
  - On or off box client registration of data
  - Distribution center of sensor data to clients
  - Transport edge data to the enterprise and cloud systems in a manner they request

Performance Targets

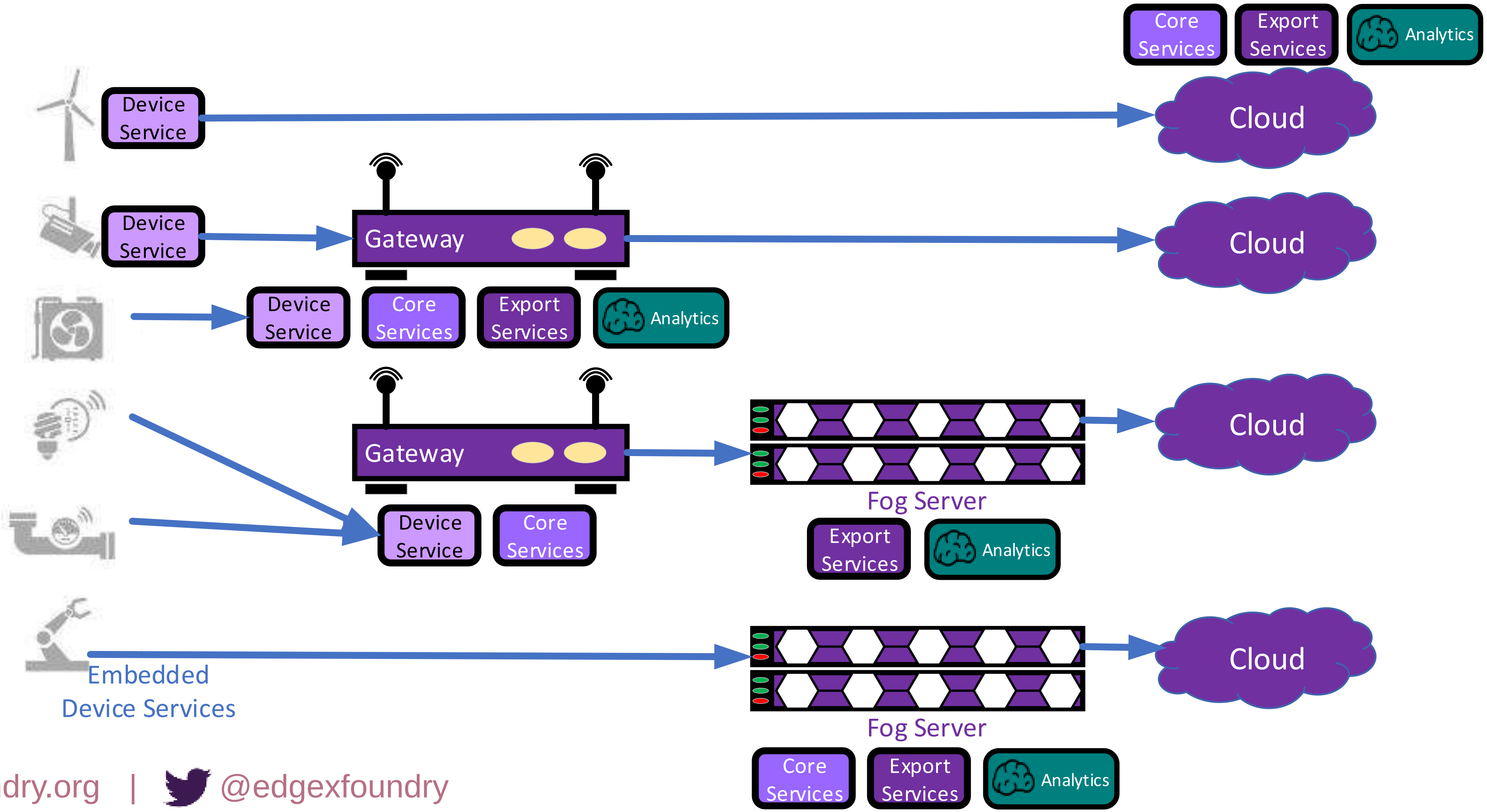
- The target is to run all of EdgeX on a Raspberry Pi 3 type of device
  - 1 GB RAM, 64bit CPU, at least 32GB storage space
- Additional “developer community” targets
  - Startup in 1 minute or less (post OS boot)
  - Latency for one piece of data from data ingestion to actuation will be < 1 second
- Remaining OS and Hardware agnostic
  - Windows, Linux, *nix, …
  - Intel/Arm 64/Arm 32
- **Indications are that these targets are met or exceeded with the California release**

Current #'s	
Footprint	76 MB
Footprint with container	113 MB
Memory (idle)	26 MB
Memory with 100 devices	40 MB
Startup time	< 10 sec
without DB or device services	

Microservice Distribution

- Microservices can live where they can get the resources they need
- With a tendency to push to the south
  - Latency needs
  - Storage and transportation costs
  - Disconnected modes
- Allow the microservices to adapt to the use case
- Requires extremely loose coupling
- In some uses, microservices might be collapsed or combined

EdgeX Flexible Deployment Possibilities



Supported Export (Northbound) Interfaces today

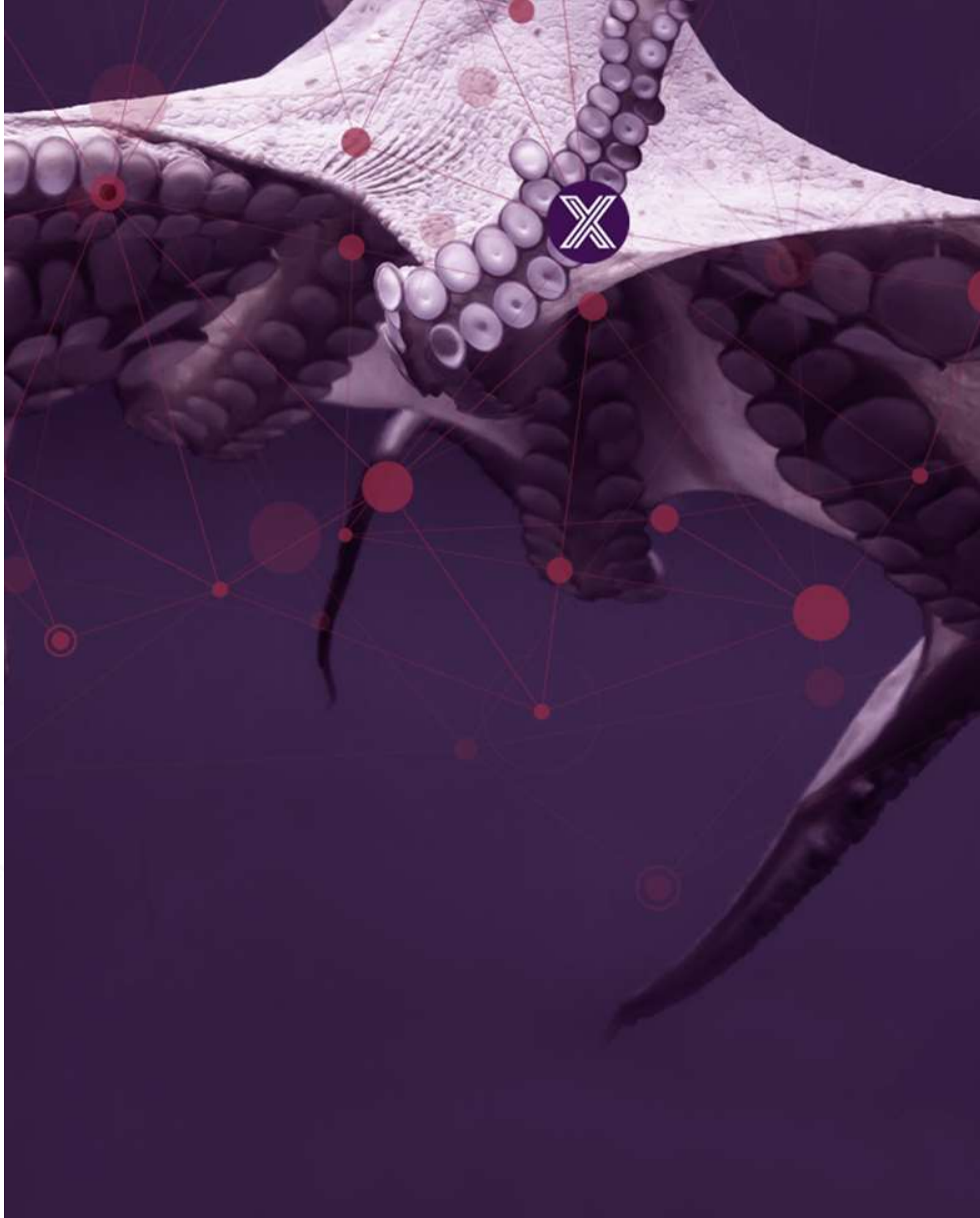
- HTTP/HTTPS
- MQTT/MQTTS
- Google IoT Core
- Azure IoT Hub
- *Coming with California Release*
  - XMPP
  - ThingsBoard IoT
  - Brightics IoT
- *WIP*
  - AWS IoT
  - IBM Watson

Supported Device Services (South) Interfaces today

- Modbus
- BACNet
- MQTT
- OPC-UA
- SNMP
- BLE
- Device Service SDK’s in Go and C coming this summer

EDGE X FOUNDRY™

Ecosystem & Current Status



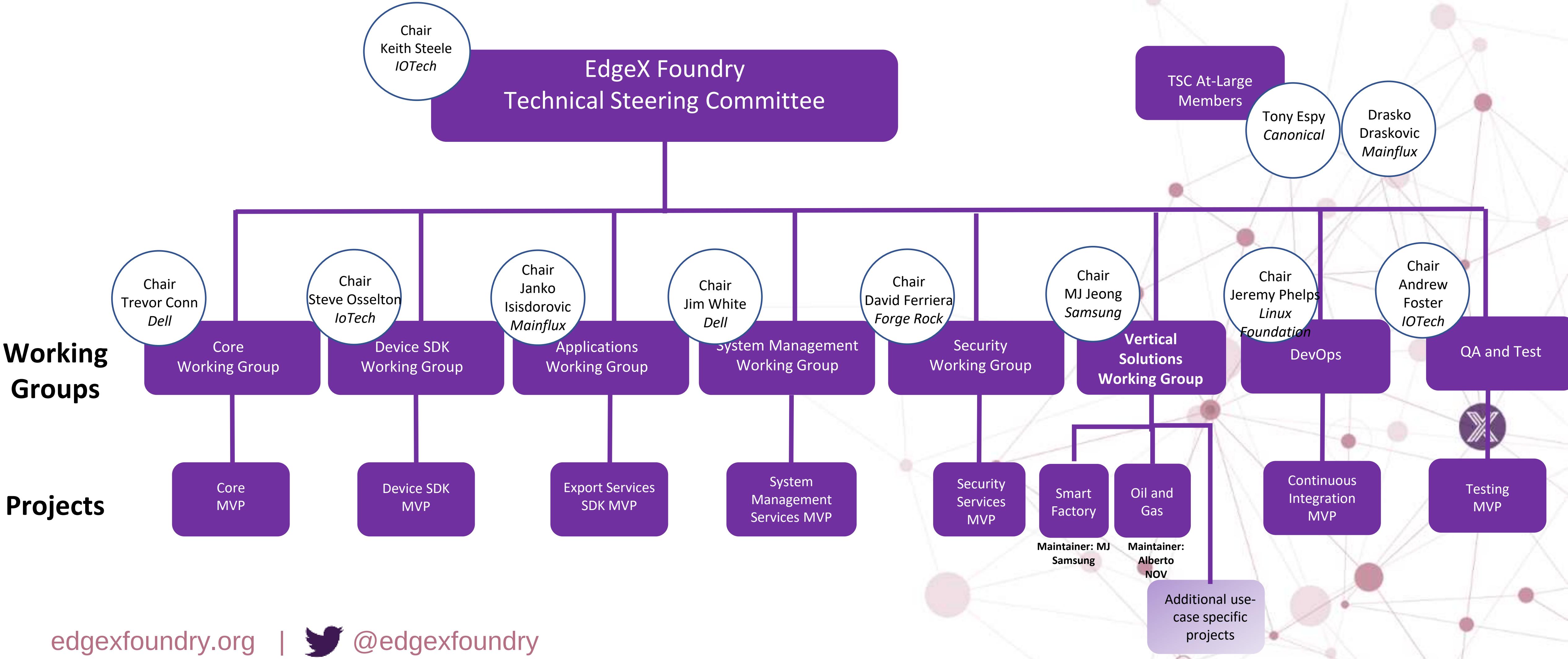
Now Backed by 70+ Members

EDGEXFOUNDRY™

With more in process!

EDGEXFOUNDRY

EdgeX Project Organization



Current Status

- EdgeX California Release on track for release at the end of June 2018. Key features include:
  - Initial security building blocks (reverse proxy, secure store)
  - Most services transitioned from Java to Go (exception: device services and SDK)
  - Dramatically improved performance, resource usage, and footprint (~7x reduction in size)
    - Already hitting our system performance targets
  - Additional “northbound” connectors
  - Improved documentation (documentation treated more like code in its management)
  - Arm 64 support
  - Blackbox testing for all services
  - Improved continuous integration
- Technical Steering Committee meet in Palo Alto, June 4-6
  - Scoped next release (code named Delhi) due Oct 2018
  - Roadmapped future releases (Edinburgh – Apr 2019, Fuji – Oct 2019)
  - Potential new members in attendance (Hitachi, Redis)
- Current membership: ~70 companies/organizations
  - Code contributions from ~40 developers

EdgeX Releases

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 < 1 GB RAM, come up in < 30 sec, < 1 second actuation latency

‘Delhi’ Release

(Oct 2018)

- Additional security and first manageability capabilities
- Go / C device service SDKs
- EdgeX UI

‘Edinburgh’ Release

(Apr 2019)

- Certification Program
- Improved and more scalable northbound connectors
- Southbound connectors to common protocol devices
- ARM 32 support

‘Fuji’ Release

(Oct 2019)

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

Delhi Release - Major Themes & Objectives

- Smaller development cycle (due to California length) so scope has to match
- High level scope
  - Initial System Management APIs and agent
  - Device Service SDKs (Go/C) & at least one example device service
  - The next wave of security features
    - Access control lists to grant access to appropriate services, and improved security service bootstrapping
  - Improve testing
    - Better/more unit, complete black box and add performance testing
  - Refactored and improved Go Lang microservices
  - Design and architecture work in advance of Edinburgh release
    - Options and implementation plan for database replacement
    - Design and implementation plans for export service replacement with application services
  - An EdgeX UI suitable for demos and smaller installations

Dell Technologies & EdgeX

Dell Technologies Commitment

- Dell invested ~7 man years of effort in Project Fuse (the precursor to EdgeX)
- Dell Technologies, IoT Solution Division announced in October 2017
- Dell Technologies, IoT Solution Division organized in February 2018
- IoT Platform Development Team in charge of Dell Technologies open source contributions
  - 6 full time employees dedicated to the development of EdgeX
  - Integration of EdgeX to DT products and solutions
- DT leadership
  - EdgeX Foundry, President, Governing Board (Jason Shepherd)
  - Two members of the Technical Steering Committee
  - Core Working Group Chairman (Trevor Conn)
  - System Management Working Group (Jim White)

Dell Technologies IoT Offer

Bridge PCF to the edge for E2E IoT application development and workload orchestration

IaaS

iPaaS data integration

Enable distributed Core, analytics via Project Nautilus and World Wide Herd and IoT Services

IoT Solutions Partner Program

Extend security tools and expertise to the IoT edge

EDGE X FOUNDRY™

Manage hardware and EdgeX ecosystem with Pulse IoT Center

Software-define edge hardware with EdgeX-compliant value-add for analytics, security, manageability and data ingestion

DT Goals with EdgeX

- Help accelerate and drive the sale of IoT hardware, software and services
- Allow interoperability with partners
- Center of DT software solutions at the edge
  - Providing the base platform to leverage DT software portfolio at the edge (Pulse, RSA, PCF, etc.)
  - A distributable software solution that can be delivered and used on DT platforms (PCF, PKS) and hardware (gateways, vSphere, etc.)
  - Facilitate the transportation to DT cloud/data platforms and services
- Provide a total DT edge solution
  - PhotonOS (VMWare)
  - EdgeX
  - Pulse IoT Center/LIOTA
  - WWH
  - Dell Gateway
  - Dell Core servers

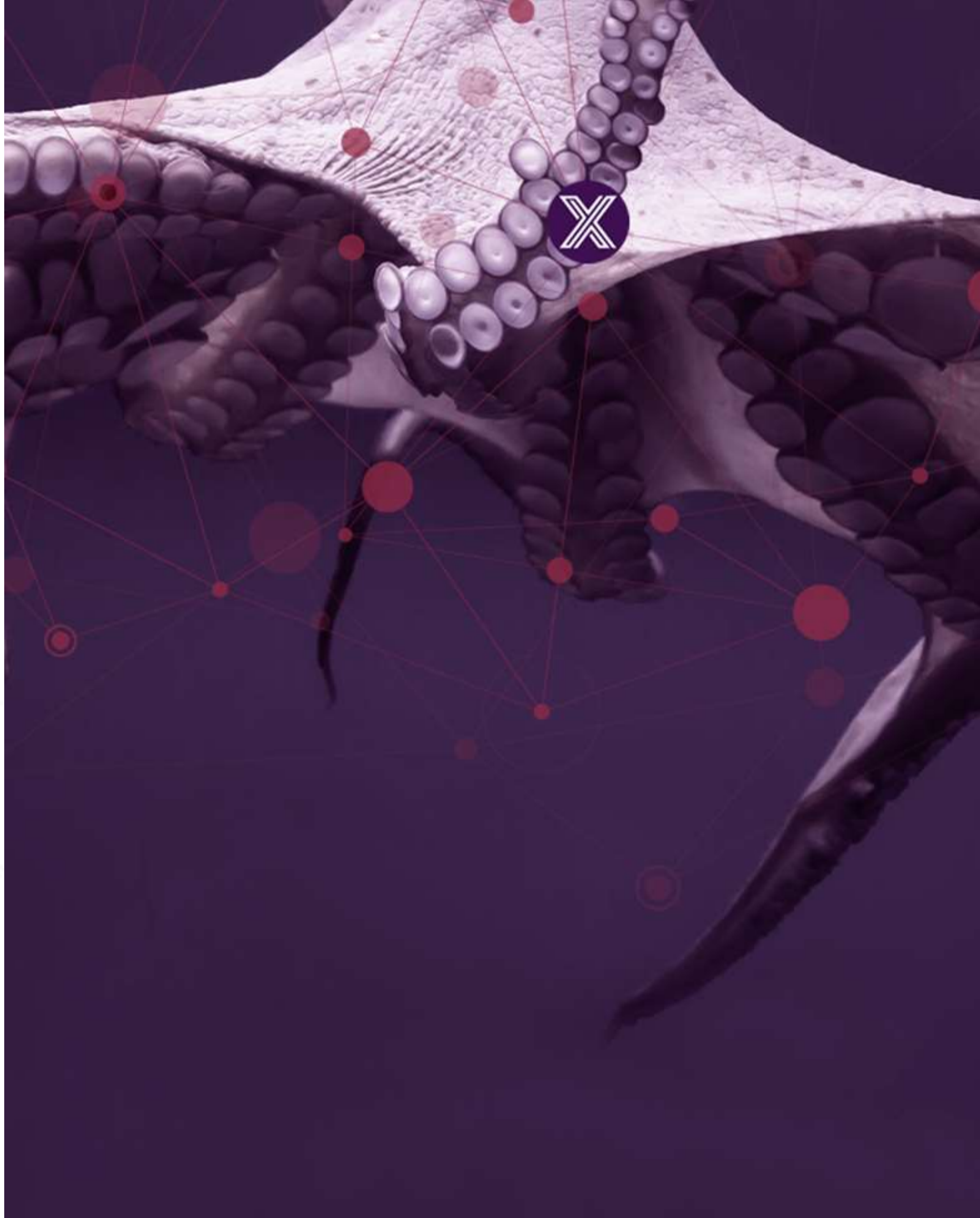
Key Project Links

- Access the code:
  - https://github.com/edgexfoundry
- Access the technical documentation:
  - https://wiki.edgexfoundry.org
- 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
- Roadmaps & Backlog
  - https://wiki.edgexfoundry.org/display/FA/Roadmap

BACKUP

Use Cases & POCs

Sample Proof-of-Concepts



Industrial Automation POC

- Large industrial automation provider
  - Working with EdgeX to bridge legacy and new OT infrastructure to SCADA and proprietary cloud environments
  - Software stack/platform will be deployed in different operational configurations
  - Need the capability/flexibility to provide common software functions independent of the hardware configuration
  - Example: deploy the stack on a standalone very low footprint micro gateway connected directly to the cloud or distribute entire stack to a larger on-prem node
- Need the platform independence and small footprint EdgeX offers to run on their gateways
- Conducting gap analysis between existing data models to EdgeX model; exploring options for model changes or extensions
- Exploring 3rd party integration for system management

EDGE X FOUNDRY™

Building Management POC

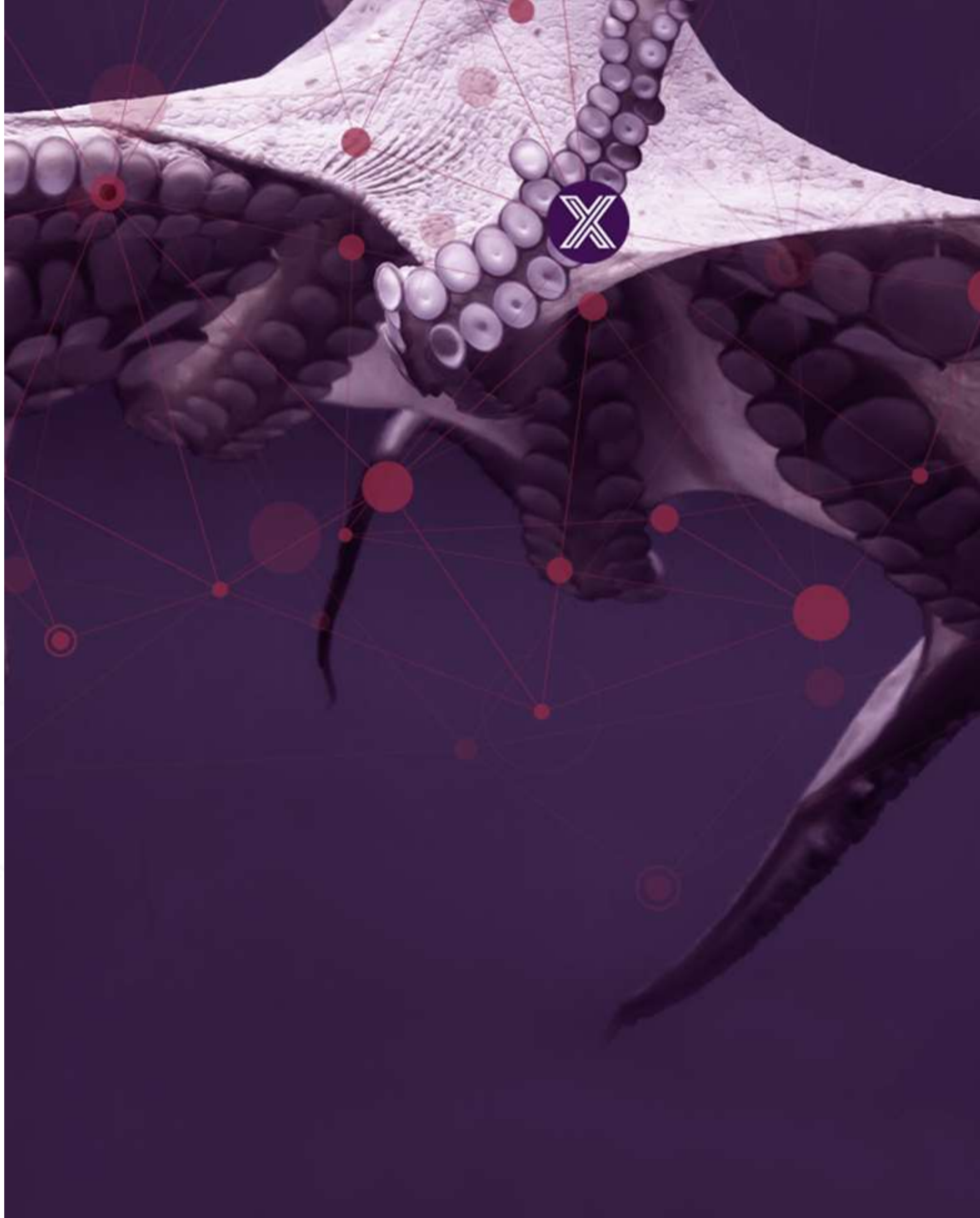
- A mid-sized building management company in the Germany needs to connect legacy systems to a central IoT system to unlock near real-time data on energy spend, space utilization and occupancy
  - Highlight resource usage discrepancies
  - Make informed cost saving decisions from data collected
- They want to build advanced analytics and visualization capability on top of a common/open platform
- Developing dynamic building automation by using EdgeX to integrate Lighting, Heating, Ventilation and AC
  - Will automatically respond to occupancy trends, people comfort and cost saving goals
- Plan to setup notifications and alerts to be notified when system performance falls outside of expected thresholds
- Given the size of their IT organization, they want to automate more tasks and use an open platform in order to leverage community assets where possible

Oil & Gas POC

- A global oil and gas supplier has the need to incorporate numerous sensors/devices/controllers through a real-time bus while also integrating various controller applications
- EdgeX would serve as the interoperable glue that brings the mix of devices/control applications together
- Real time needs are going to be provided by an EdgeX commercialization and specialization firm

EDGEXFOUNDRY™

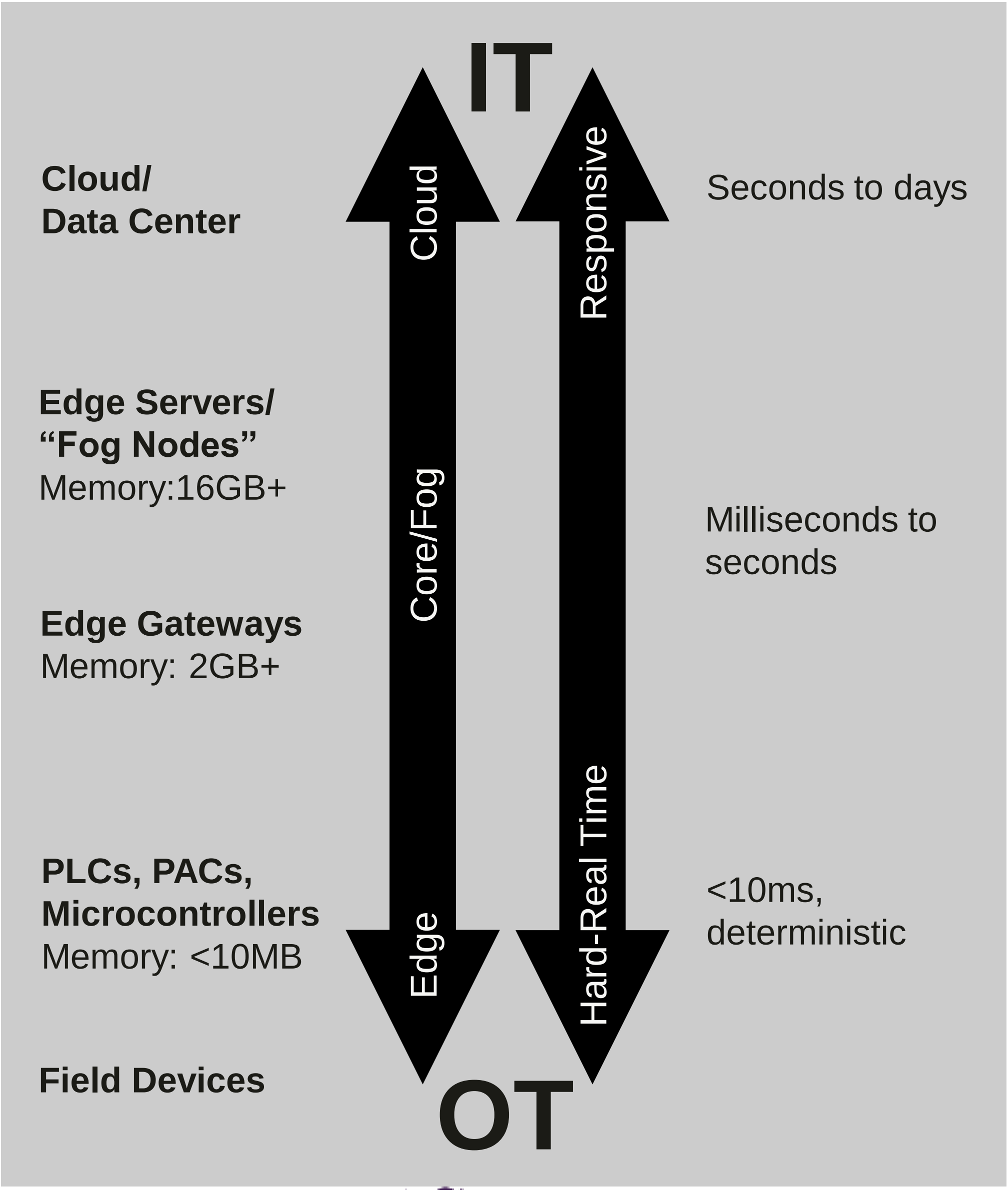
Use Cases

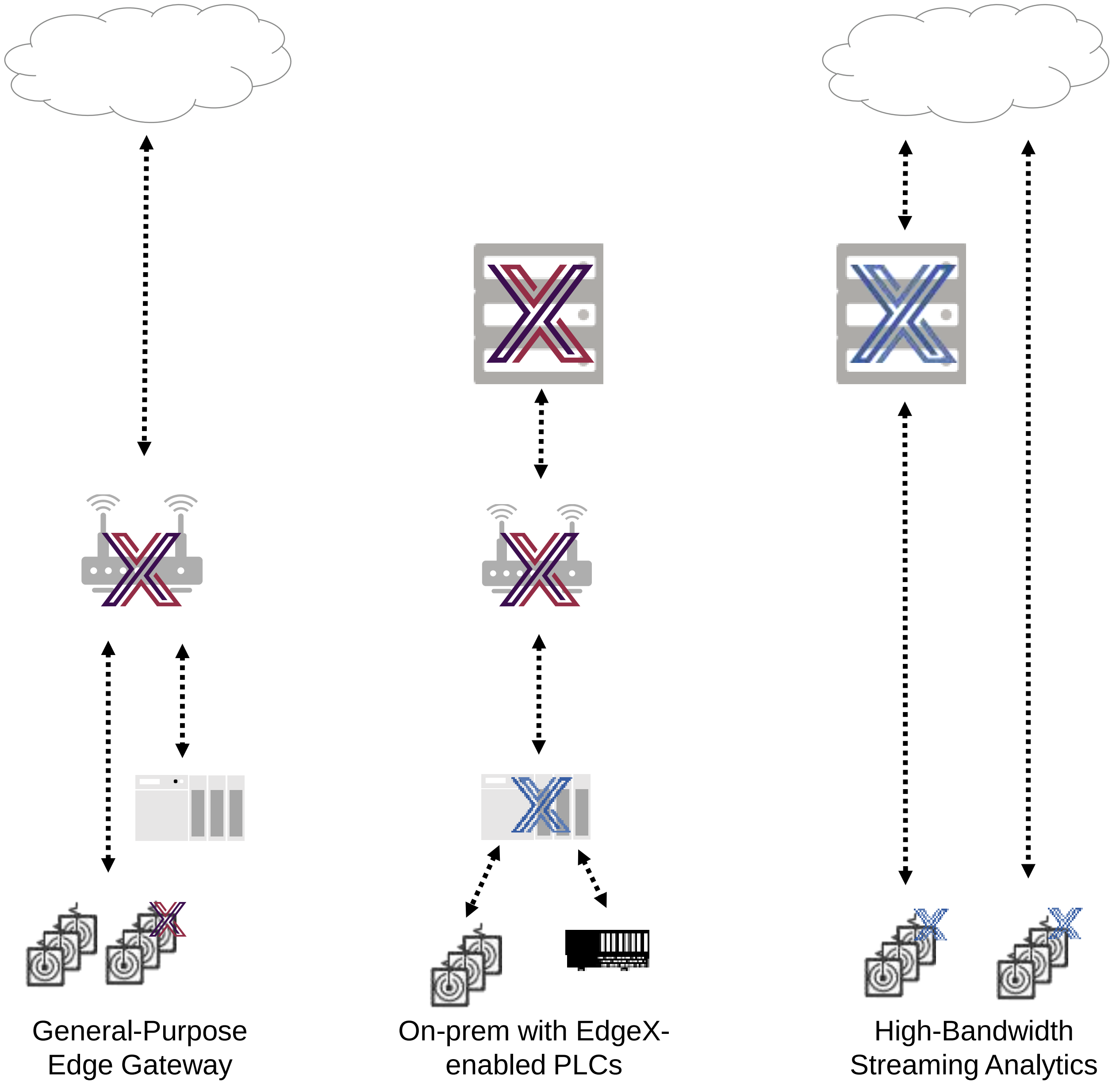


Summary of Example Use Cases

Open Source Baseline

Proprietary EdgeX-compliant Extensions



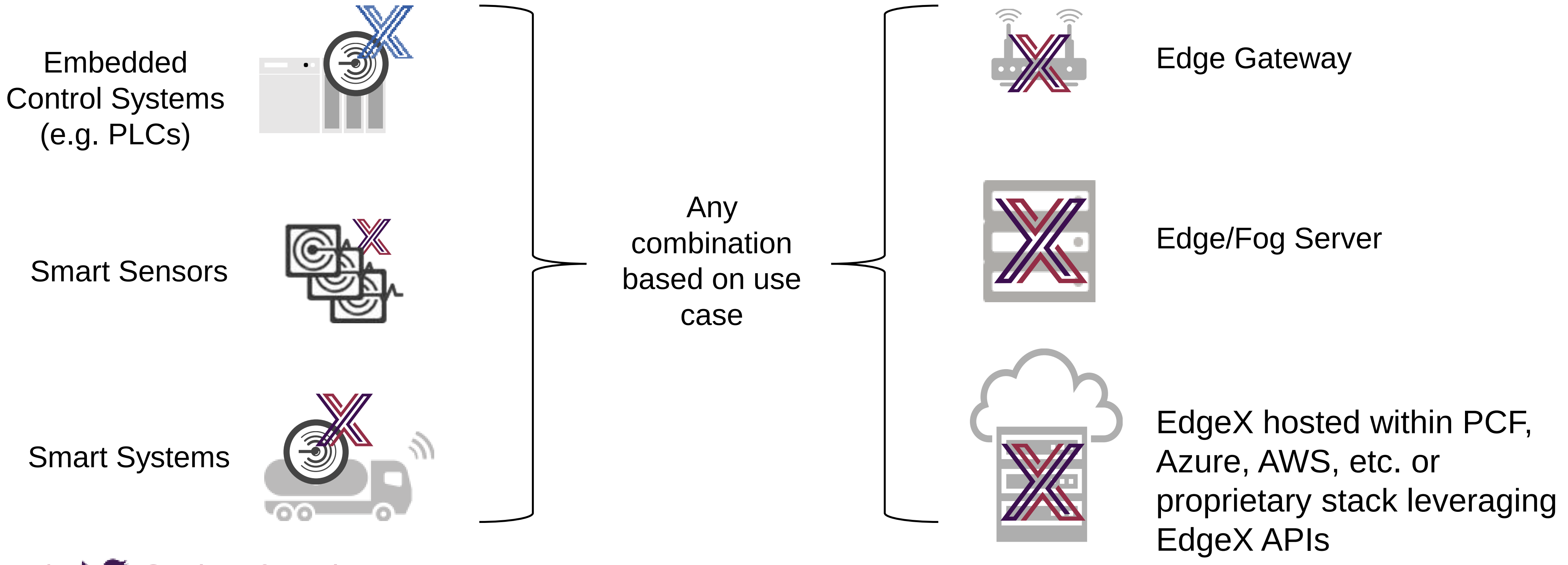


Real Time Enabled Via Code Extensions

	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	

Embedded Device Services

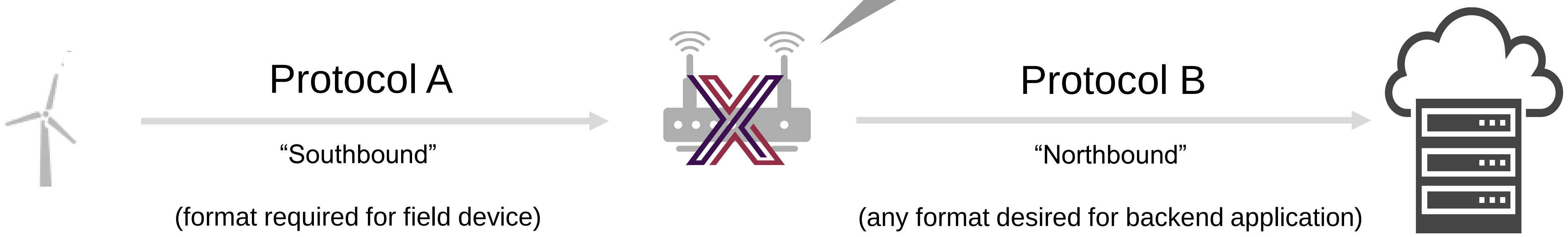
- Planned work will enable C-based Device Services to be embedded in constrained microcontrollers running a RTOS for real-time use cases (e.g. within a smart sensor or PLC)
- Due to loosely-coupled architecture, baseline EdgeX-compliant Device Services can be deployed directly on smart sensors or systems 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.

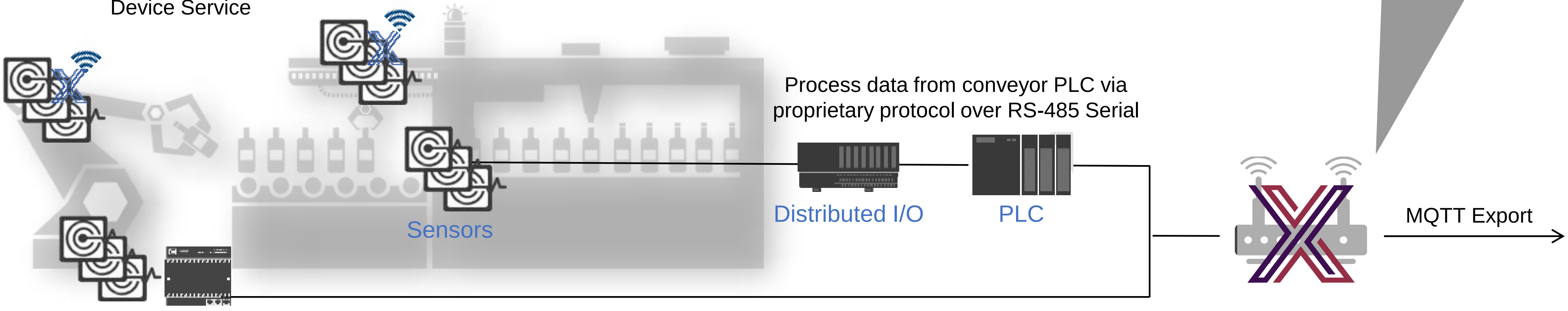
Deployed Microservices:
- Single Device Service
- Core Services
- Single Export Service
- Basic security and manageability



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
- 3rd party CEP for edge analytics
- Various security services
- 3rd party remote management console
- MQTT Export Service



Tiered Deployment in Smart Buildings

Number of deployed microservices and functionality increases higher in tier

Field Devices	Simple Edge GWs	Intelligent Edge GWs	Edge Servers	Cloud

Room Level

- Ingestion for local temperature and occupancy data
- Simple rules engine to control temperature and lighting settings

Floor Level

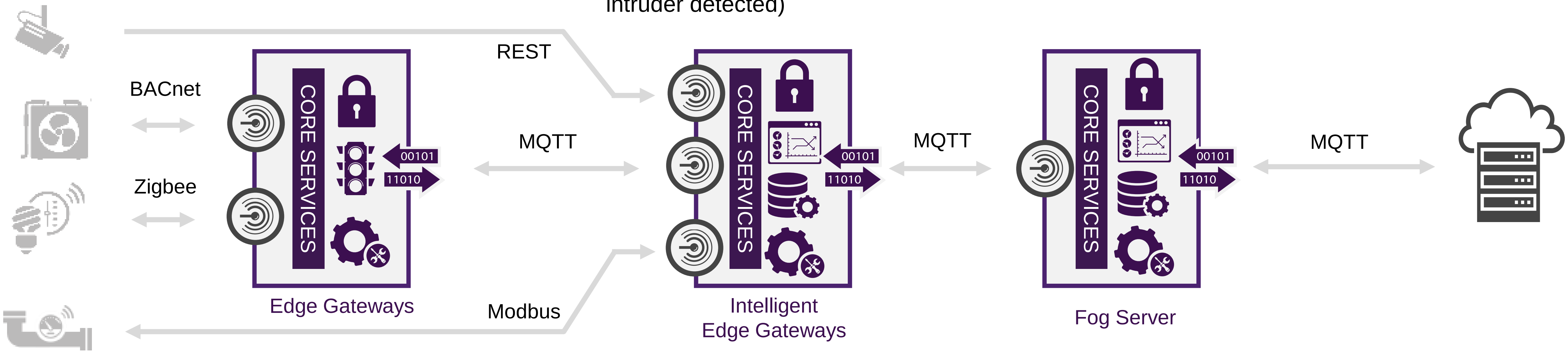
- Integration of temp and occupancy plus add'l events from surveillance cameras and overall energy usage data
- Basic ML/CEP for reacting to local events (e.g. alert security when intruder detected)

Building Level

- Aggregated data for analytics of overall building performance
- Streaming data from all floors, more complex analytics

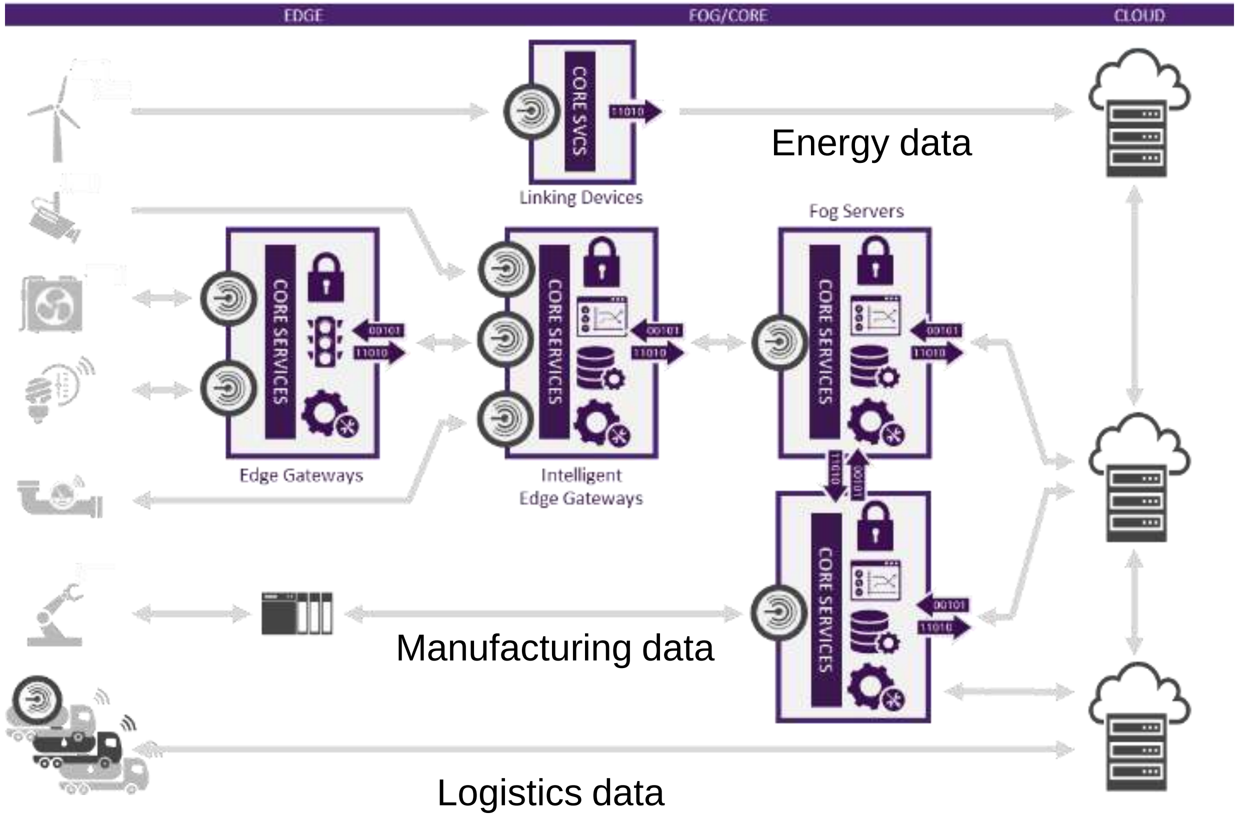
Portfolio Level

- Deep learning in the cloud to optimize energy usage across entire real estate portfolio



Distributed (e.g. ‘Fog’) Computing

- Introducing specific microservices to address QoS, failover between nodes, redundancy and “east-west” communication
- 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.



EDGE X FOUNDRY™

Thank You