

# Proposed Barcelona Goals and MVP

## Barcelona Goals

- Stabilize the platform
- Build community understanding of the platform
- General majority agreement on the architecture
- General majority agreement on micro services and APIs
- General majority agreement on the development of an open, platform independent, technology agnostic platform for the IoT/edge
- General agreement on temporary performance targets

## Minimum Viable Product (MVP)

### General

- Review and agree/adjust micro service APIs
- Review and agree/adjust object models
- Improve/increase documentation especially for areas of extension (DS SDK, Export Services)
- Harden/stabilize the platform
  - o Works properly for the intended use case (it may not be 100% complete implementations for all use cases or parts of a protocol for example, but it provides enough implementation to sustain the demo use cases for Barcelona and could support extension to the full needs or protocol in the future)
  - o Handles errors and exceptions gracefully
  - o Contains proper unit and integration tests (lacking in DS, supporting services and others)
  - o Follows good coding standards, and is well documented (following some prescribed standard per programming paradigm)
  - o Performs within the target measures established for Barcelona
    - § Examine Kura in detail and adjust performance targets accordingly
- Deliver Docker containers that run on Intel chip for Linux, Window, Mac

### Stretch goals

- Deliver Docker containers that run on Arm chip for Linux

## Device Service / DS SDK

- Deliver initial set of 7 Device Services based on Dell contributions (Modbus, BACnet, Bluetooth, SNMP, MQTT, Serial (Fischertechnik), Virtual)
- Clean up SDK (and DSs)
  - o Improve documentation
  - o Merge device-sdk into SDK tools
  - o Improve tooling (Eclipse Plugin)
  - o Cleanup scheduler

### Stretch goals

- Remove redundant code from Device Services/SDK to shared libraries
- Redo/refactor Bluetooth and BACnet DS to be single micro service
- Additional DS provided by additional community participation

## Core & Supporting Services

Clean up some minor issues

- o Logging OOM, Remove Device Manager, etc.

### **Stretch goals**

Implement configuration callbacks (allowing for configuration changes dynamically without service restart

Provide first Go replacements for Data, MetaData, Command (Dell has a start to these already)

## **Applications (including Export Services, Rules Engine/Analytics)**

Pick and provide at least one cloud connector (Azure IoT Hub has been prototyped by Dell)

Offer MQTT, REST, ZeroMQ export

Offer JSON, XML, CSV (not done yet) formats

Improve module for encryption options

Deal with potential number of readings, number of client scale problem

### **Stretch goals**

Implement 2<sup>nd</sup> cloud connector (ex: Amazon Greengrass, Watson, ???)

Add additional format offering (ex: Haystack, etc.)

Add hyperledger export option

## **Security**

In general, define the EdgeX security story but postpone a lot of implementation to California

WG to agree on requirements

WG to agree on what security features are going to be in EdgeX and what's going to be provided by the platform that EdgeX runs on (example: the underlying platform must have a keystore)

WG to define what EdgeX security service(s) need to be eventually implemented

WG to define what security hooks need to be added to the existing micro services

- o How and to which services would APIs communicate with

WG to define what standards, protocols, etc. are going to be adhered to and followed by EdgeX (ex: IIC specs, OAuth tokens, etc.)

WG to provide guidance on how security features can/should be tested

### **Stretch goals**

Add stubbed hooks into micro service code

## **System Management**

In general, define the EdgeX system management story but postpone a lot of implementation to California

WG to agree on requirements

WG to agree on what features are going to be in EdgeX and what is reserved for OS, 3<sup>rd</sup> party systems, other open source systems, etc.

WG to define what system management services need to be implemented as part of EdgeX (if any)

WG to define what system management hooks need to be implemented

WG to define any system management standards that will be followed/used in system management implementations (ex: LWM2M)

### **Stretch goals**

Add some simple system manage hooks/capability into BaseService of EdgeX micro services (Dell has already done some POC work with things like start, stop, ...)

## Testing/QA

Insure part of code review/code check is to insure proper unit/integration tests for the code are provided (backed up by code coverage statistics)

Create a blackbox testing framework to insure the APIs between services are not broken and to be able to measure performance of a micro service and across multiple micro services to insure targets are achieved

Automate blackbox testing on all micro services with each build

## Build/CI

Agree on a base set of policies and procedures around code check in, code approval, governance, etc.

Utilize the existing LF build process with some additions noted below.

Create Docker containers and push them to Docker Hub via the build process

Anoint a Bug Czar to setup a bug management process and monitor, track, and address incoming bugs (along with general support issues across media channels)

### Stretch goals

Automate creation of ARM build/containers.

## Event Demo

Create a working group to define the use case and demo presentation

Work with the community to get hardware, sensors, etc. donated for the demo

Create a minimal user interface for EdgeX (could be based on Dell Fuse UI)

## California Release Goals

Implement first security and system management services and tie to existing micro services

Improve performance

Introduce replacement services as appropriate (ex: Go Core)

Demonstrate EdgeX in real world POC/Test Bed