EdgeX Foundry
Introduction

Janko Isidorovic
June 2019
About Me

Janko Isidorovic
Mainflux COO & Co-founder

Janko is the co-founder of Mainflux IoT open source project.

He is also chair of the Application Work Group of Linux Foundation EdgeX project.

Janko has a 10+ years background in Project Management, IT and Software integrations.
He holds MSc. In Telecommunications.
Why the Internet Of Things
It's Because

The adoption of IoT is off the charts. IoT projects will take 2x as much as expected. Things are.

by David Houghton, May 12, 2016

DZone

IoT Protocols

When entering the world of IoT, your way around the protocols will be able to communicate

by Candace Letizia, April 15, 2016

PCW

Look before you leap

Data errors, security holes and rapidly changing applications.

By Stephen Lawson, March 10, 2016

Wired

7 Reasons Why the Internet of Things Is Doomed

PARTNER CONTENT JASON BLOOMBERG, INTELIX

7 REASONS WHY THE INTERNET OF THINGS IS DOOMED

Image: Kyle Slattery/Flickr
Why is IoT hard to do?

- Heterogeneity of platforms
  - Diverse collection of OS and OS variants
    - Linux, Unix, Windows, VxWorks, embedded and RTOS, …
  - Various Hardware (Intel, AMD, ARM,…)
  - Cloud, gateway, smart thing (the “Fog continuum”)

- Thing protocol soup
  - Industrial: BACNet, Modbus, OPC-UA,…
  - Wireless: BLE, Z-Wave, Zigbee,…
  - Message: MQTT, AMQP, DDS, …

- Variety of cloud platforms
  - Azure IoT Hub, AWS IoT Platform, Google IoT Core, IBM Watson IoT Platform, …

- Add your favorite selection of…
  - Applications, edge analytics/intelligence, security, system management, …

- Difficulties in determining where to start

IoT is a post doctorate in all we know and have done in computing for the last 30-40 years

- Networks/protocols
- Mobile computing
- Distributed compute
- Cloud compute
- AI/Machine learning
- …
Introducing EdgeX Foundry

An open source, vendor neutral project (and ecosystem)

A **micro service**, loosely coupled software framework for IoT edge computing

Hardware and OS agnostic

Linux Foundation, Apache 2 project

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 Foundry Goals

- Build and promote EdgeX as the common open platform unifying edge computing
- Enable and encourage the rapidly growing community of IoT solutions providers to create an ecosystem of interoperable plug-and-play components
- Certify EdgeX components to ensure interoperability and compatibility
- Provide tools to quickly create EdgeX-based IoT edge solutions
- Collaborate with relevant open source projects, standards groups, and industry alliances to ensure consistency and interoperability across the IoT
A Brief EdgeX History

• Chartered by Dell IoT marketing in July 2015
  • A Dell Client CTO incubation project (Project Fuse)

• Designed to meet interoperable and connectivity concerns at the IoT edge

• Started with over 125,000 lines of Dell code

• Entered into open source through the Linux Foundation on April 24, 2017
  • Started with nearly 50 founding member organizations; today we have more than 75

• Release Cadence: 2 formal releases a year
  • Barcelona – Oct 2017
  • California – Jun 2018
  • Delhi – Oct 2018
  • Edinburgh – June 2019
  • Fuji – Oct 2019
  • Geneva – April 2020
  • Hanoi – Oct 2020
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
It's 102°F

Stop the machine
EdgeX Micro Service Layers

- Contextually, EdgeX micro services are divided into 4 layers
- Crudely speaking, the layers of EdgeX provide a dual transformation engine
  - 1x - Translating information coming from sensors and devices via hundreds of protocols and thousands of formats into EdgeX
  - 2x - Delivering data to applications, enterprises and cloud systems over TCP/IP based protocols in formats and structures of customer choice
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 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
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 10 seconds 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 not officially supported yet but has been done)
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,…
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


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

Join the Slack Channels: https://slack.edgexfoundry.org/

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
What’s with the ‘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 (~ Oct 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 micro service(s) or a services model
The Linux Foundation Launches New LF Edge to Establish a Unified Open Source Framework for the Edge

More than 60 global founding members across enterprise, IoT, telecom and cloud collaborate on open source framework for edge computing and future of IoT

SAN FRANCISCO, January 24, 2019 – The Linux Foundation, the nonprofit organization enabling mass innovation through open source, today announced the launch of LF Edge, an umbrella organization to establish an open, interoperable framework for edge computing independent of hardware, silicon, cloud, or operating system. LF Edge is initially comprised of five projects that will support emerging edge applications in the area of non-traditional video and connected things that require lower latency, faster processing and mobility.

LF Edge includes Akaino Edge Stack, EdgeX Foundry, and Open Glossary of Edge Computing, formerly stand-alone projects at The Linux Foundation. The initiative also includes a new project contributed by Samsung Electronics, which will create a hub for real-time data collected through smart home devices, and another project from ZEDEDA, which is contributing a new agnostic standard edge architecture.
LF Anchor projects for Edge

Akraino and EdgeX Foundry are complementary open source projects addressing Telecom, Enterprise and IOT edge.

On Prem Edge Virtualization Engine

IOT Interoperability framework

Telco Edge Use Case

EDGE
VNFs, vEPC, MEC, distributed RAN, vRAN, BBU hotel, PMC, vCPE, AI/ML, IoT

PARTIAL EDGE
85% of operators plan VNF execution in DC Near CO

NOT EDGE
70% of operators plan VNF execution in DC Not Near CO

Data Center Near CO [Regional DC]

Data Center Not Near CO [Central DC]

19 January 2019
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
Release Roadmap

‘Barcelona’
Improved fit and finish, formalized Core Service APIs, additional Device and Export Services, test apparatus

‘California’
First integration of security, Java to Go code base, run in < 256MB RAM, come up in < 10 sec

‘Delhi’
First manageability capabilities, Go / C device service SDKs & sample device services, EdgeX UI

‘Edinburgh’
Support binary data, certification program, improved and more scalable northbound connectors / application services, additional southbound connectors to common protocols and devices

‘Fuji’
More security, more management capability, scalable north side connectors to the clouds, self-assessment/certification process

Released Oct 2017
Released Jun 2018
Released Oct 2018
Jun 2019
Oct 2019
Edinburgh Release – Major Themes & Objectives

- Releasing June 2019
- Ratified during EdgeX TSC Face-to-Face meetings in UK, Nov 2018
- High level scope
  - Improved on-boarding for EdgeX Users (docs, tutorials, dev kits, etc.)
  - Support of ingestion, use, export of binary data (via CBOR format)
  - Automate performance testing, automate security testing
  - Add many device services (improving out-of-the-box southbound connectivity)
  - Provide application services – a more scalable and flexible exportation capability (eventually replacing the existing export services)
  - Refactor database-using services to be more loosely coupled to the persistence mechanism (allow for use of alternative persistence stores and technologies in future releases)
  - Outline a certification program for micro service drop in replacements
Fuji Release – Anticipated Major Themes & Objectives

• Releasing October 2019
• Ratified during EdgeX TSC Face-to-Face meetings in Seoul, April 2019
• Anticipated high level scope
  • Improved performance testing and measuring.
  • Improved security to include better PKI management, storing service secrets in separate secret stores, and ensuring services running are those expected/trusted.
  • In the first step toward offering 3rd party service certification, we will offer a self-assessment program (delivered by EOY 2019)
  • Complete the EdgeX store and forward capability.
  • Finish implementation of application services and archive the existing export services
  • Provide connectivity to the top cloud providers (Azure and Amazon IoT).
  • Improve unit testing and overall harden the services and SDKs.
• Restart the EdgeX marketing work group – which we lost when LF Edge was formed
Thank You

janko@mainflux.com