

Edge-Computing Framework (EdgeX)

J. Wagner, D. Gibietz, D. Helmer

Department of Computer Science and Engineering
Frankfurt University of Applied Sciences

Frankfurt am Main, 2021

Outline

- 1 Introduction
- 2 EdgeX architecture
- 3 How it works
- 4 Use case
- 5 Data processing
- 6 Live demonstration

Introduction

- EdgeX Foundry is an open source software framework
- It is serving between physical "things" and applications or clouds
- EdgeX provides multiple protocols to connect IoT devices, like MQTT, REST or BLE
- It allows to encrypt, transform, filter or format the data before forwarding it to an external source over different protocols like MQTT



Figure: The EdgeX gateway between the "things" and the IT-System.

Source: <https://docs.edgexfoundry.org/1.2/>

EdgeX architecture

EDGE X FOUNDRY™

Platform Architecture

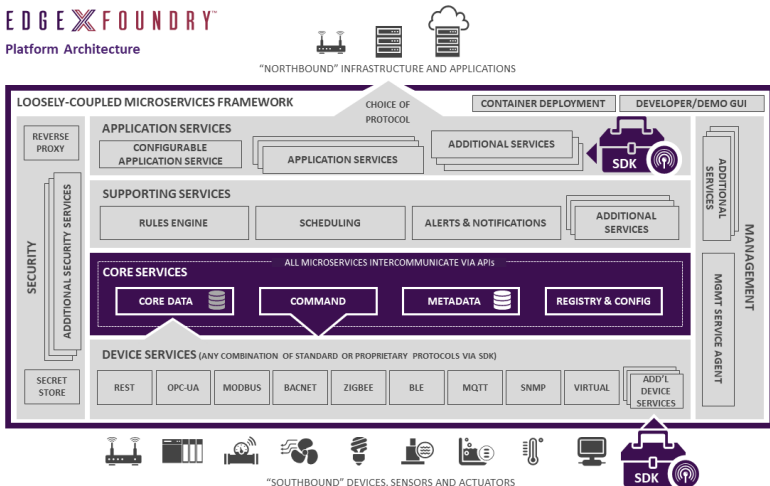


Figure: Visualisation of the EdgeX platform architecture.

Source: <https://docs.edgexfoundry.org/1.2/#edgex-foundry-service-layers>

- EdgeX data flow:
 - Sensor data of a 'thing' is collected
 - Sensor data is transferred to the Core Services
 - Data can be filtered, formatted, ... before sending them to a cloud
 - Data can be used to trigger other devices
 - Cloud can further process the data, analyze it, simply store it and more
- Using REST for communication
 - Device ↔ Gateway and Gateway ↔ Cloud
 - REST → **RE**presentational **S**tate **T**ransfer
 - Allows simple machine-to-machine communication
- The services are made available via Docker and Docker-Compose

Use case

- IoT gateway should be close to the edge devices
- Often needed system architecture: Multiple edge devices sending data to a cloud service via the IoT gateway
- Deployment of EdgeX on a SBC. Temperature values send by multiple nodes are forwarded to a data base in the cloud.

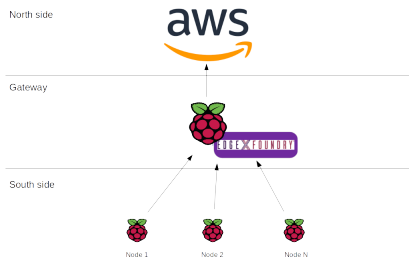


Figure: Data flow of overall system architecture

- The communication with the physical nodes is called "south side"
- South side means the south of the communication (device layer)
- This layer interacts with the devices and sensors
- Various protocols, such as REST, MQTT or even SNMP, are available for the south side connection
- After installation and running of EdgeX the REST service start by default
- The developed script sends a variety of real sensor data
- It is later presented in a live demo

Data processing

EdgeX device and value representation

- Device profile: defines a 'device type'
- Create digital representation of a device
 - Multiple devices can be created with one profile
- Create valuedescriptors / variables
 - Data sent by devices must be predefined

Live demo on how to add devices later in the presentation

- Exporting data to the north side is possible by using either Application service or rule engine
 - For deployment rule engine was chosen
- Rule engine since Geneva release: EMQ's Kuiper
- Service consists of three components: Source, SQL and Sink
 - Source: Data stream that is accessed by kuiper
 - SQL: Statements to filter the data of the source
 - Sink: Data sink that uses a defined protocol for forwarding messages

Data processing

North side

- An exemplary northside use case is implemented with AWS
- AWS was selected since all needed services were included in the free-trier

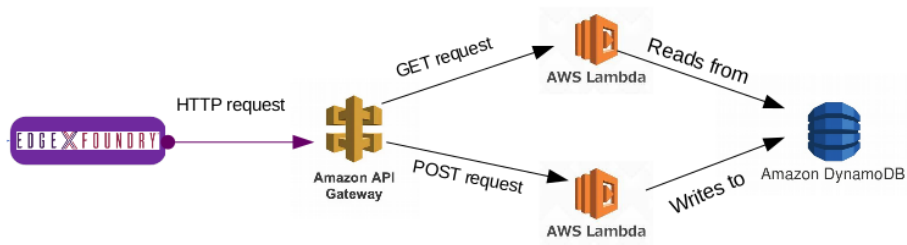


Figure: Data flow within AWS northside

Live demonstration

Thank you for your attention!