

EdgeX Foundry Performance Report

Hanoi Release

January 2021

EdgeX Foundry is a highly flexible and scalable open source software framework that facilitates interoperability between devices and applications at the IoT Edge.

Hanoi (version 1.3) marks the 7th community release of EdgeX and was formally released in November 2020. The key new features added in Hanoi are described on the [EdgeX Community Wiki here](#).

This report aims to provide EdgeX users with important performance information that can guide solution development and deployment strategies. The information also helps the EdgeX development community to ensure the platform remains suitable for lightweight edge deployments and can help to identify future performance targets.

The performance metrics described in this report relate to data obtained on the following hardware platform:

HP MP9 G4 Desktop Mini PC

- Intel Core i7-8700T processor @2.4GHz
- 16GB RAM
- Ubuntu 20.04 LTS

www.edgexfoundry.org

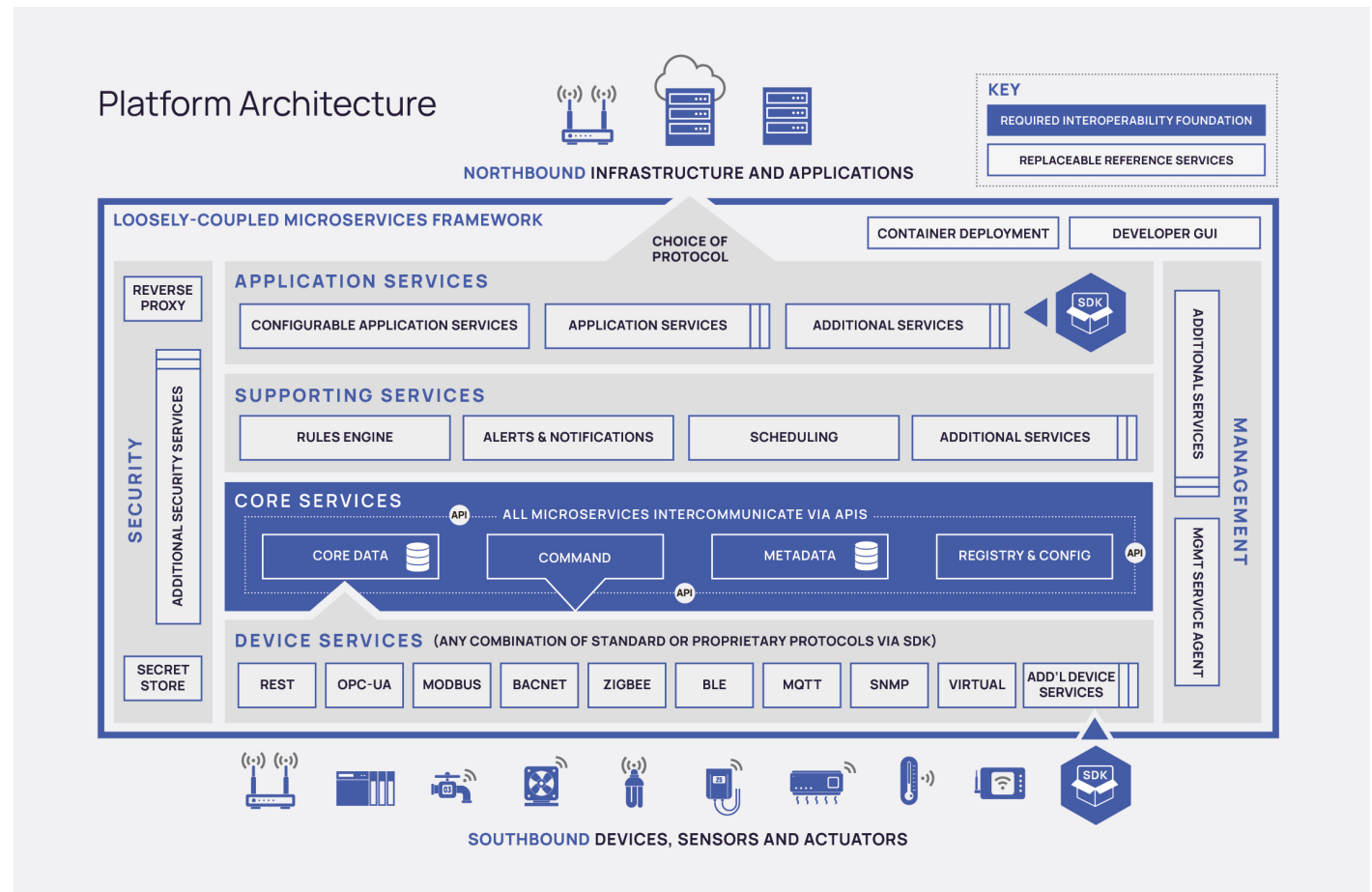
EdgeX Foundry Deployment Options

The EdgeX Foundry platform is a collection of modular microservices that each perform a specific role at the IoT Edge. While some microservices are used in almost all deployments of the platform, the specific collection of microservices used are dependent on the requirements of the target use case.

The EdgeX microservices are designed to be location independent so can be distributed between computing nodes where necessary, however the EdgeX Core Services are usually expected to be deployed onto all nodes running EdgeX. In very austere environments there may only be a requirement to collect data from a small set of edge devices and simply stream the data to another endpoint. A reduced number of microservices are required in those use cases.

This report provides performance data and resource usage based on both a “typical” and “minimal” EdgeX deployment:

- A typical EdgeX deployment provides all of the services as envisioned by the original architects of the system and includes services for ingesting different edge data types, edge decision making, device actuation at the edge, and streaming to a cloud
- A minimal EdgeX deployment provides only the services needed to ingest data for a single edge protocol and capability to stream to a northbound endpoint



The EdgeX Microservices

Core Services - Where most of the knowledge as to what “things” are connected, what data is flowing and how EdgeX is configured in a given deployment

Microservice	EdgeX Deployment Name	Description
Core Data	edgex-core-data	Data store for readings collected by devices and sensors. Works in conjunction with the Database service
Core Metadata	edgex-core-metadata	Used by other services for knowledge about the devices and how to communicate with them
Core Command	edgex-core-command	Enables commands to devices on behalf of other services, applications or external systems. Optional
Database	edgex-redis	The default database for EdgeX, implemented via Redis
Configuration & Registry	edgex-core-consul	Centralized service configuration for EdgeX implemented via Consul. Optional

Device Services - The connectors that interact directly with the edge sensors and devices. An EdgeX use case would typically run at least one Device Service. This report provides statistics for the REST and the Virtual Device Service but many more Device Services are provided both by the Foundry and the commercial EdgeX ecosystem.

Microservice	EdgeX Deployment Name	Description
REST Device Service	edgex-device-rest	Device Service for interacting with edge devices that provide a REST-based API
Virtual Device Service	edgex-device-virtual	Device Service for simulating device data

Supporting Services - Includes edge rules, analytics and typical software application duties such as logging, scheduling and data clean up.

Microservice	EdgeX Deployment Name	Description
Rules Engine	edgex-core-data	Data store for readings collected by devices and sensors. Works in conjunction with the Database service
Support Notifications	edgex-core-metadata	Used by other services for knowledge about the devices and how to communicate with them
Support Scheduler	edgex-core-command	Enables commands to devices on behalf of other services, applications or external systems. Optional

Application Services - The means to extract, process and send EdgeX data to an endpoint or application of your choice. A full SDK is provided, but for convenience, EdgeX provides a base application service that can be configured to execute a number of built-in functions and deliver data to northbound endpoints.

Microservice	EdgeX Deployment Name	Description
App Service Configurable	edgex-app-service-configurable-rules	Application Service that can be configured to execute built-in transform and export functions

EdgeX Footprint and Container Sizes

Each EdgeX microservice is typically implemented in either Go or C and then compiled into an executable which has a size or footprint as it sits on disk. For convenient deployment and orchestration, each microservice executable can also be built into a container image so it can be run as a standalone service with less dependencies on the host. This greatly improves portability and platform independence. A container image will be larger than the base executable because it may also contain a minimal operating system and any required configuration and supporting infrastructure. EdgeX provides container images for both Docker and Ubuntu Snaps.

Both containerized Docker image footprint data and non-containerized executable footprint data is measured.

Typical EdgeX Deployment

EdgeX Microservice	Image Footprint (MB)	Executable Footprint (MB)
Core Services		
edgex-core-data	23.69	15.48
edgex-core-metadata	14.49	14.48
edgex-core-command	12.17	12.17
edgex-redis	31.15	N/A
edgex-core-consul	141.05	N/A
Device Services		
edgex-device-virtual	20.48	14.85
edgex-device-rest	17.09	11.50
Supporting Services		
edgex-support-notifications	13.63	12.30
edgex-support-scheduler	12.30	12.29
edgex-kuiper	20.24	N/A
Application Services		
edgex-app-service-configurable-rules	28.94	20.21
TOTAL	335.23	

Minimal EdgeX Deployment

EdgeX Microservice	Image Footprint (MB)	Executable Footprint (MB)
Core Services		
edgex-core-data	23.69	15.48
edgex-core-metadata	14.49	14.48
edgex-redis	31.15	N/A
Device Services		
edgex-device-rest	17.09	11.50
Application Services		
edgex-app-service-configurable-rules	28.94	20.21
TOTAL	127.53	

The Docker-based image footprint for a typical EdgeX deployment is approximately 335 MB and approximately 128 MB for a minimal deployment. The optional Configuration & Registry service (Consul) is the most significant contributor to the footprint.

Note: The executable footprint is not recorded for third-party open source services (Redis, Consul, Kuiper, etc) so N/A is listed for those services.

EdgeX CPU Consumption

Each EdgeX microservice has its CPU consumption measured as it is started as a Docker container. The CPU usage is reported by the Docker engine and is measured as a percentage of the available CPU on the machine. In general, the measure of usage at startup is a good indication on the upper bound for many of the services. Note that the characteristics of different chip architectures may have a significant effect on the percentage of CPU that is utilized.

Typical Deployment Memory Usage

EdgeX Microservice	Average (%)	Maximum (%)	Minimum (%)
Core Services			
edgex-core-data	1.51	1.56	1.44
edgex-core-metadata	1.56	1.68	1.39
edgex-core-command	1.44	1.49	1.35
edgex-redis	0.21	0.36	0.18
edgex-core-consul	1.36	1.87	1.12
Device Services			
edgex-device-virtual	0.08	0.11	0.00
edgex-device-rest	0.00	0.04	0.00
Supporting Services			
edgex-support-notifications	1.46	1.58	1.27
edgex-support-scheduler	1.42	1.54	1.27
edgex-kuiper	0.00	0.00	0.00
Application Services			
edgex-app-service-configurable-rules	0.00	0.00	0.00
TOTAL	9.04	10.23	8.02

Minimal EdgeX Deployment

EdgeX Microservice	Average (%)	Maximum (%)	Minimum (%)
Core Services			
edgex-core-data	1.51	1.56	1.44
edgex-core-metadata	1.56	1.68	1.39
edgex-redis	0.21	0.36	0.18
Device Services			
edgex-device-rest	0.00	0.04	0.00
Application Services			
edgex-app-service-configurable-rules	0.00	0.00	0.00
TOTAL	3.28	3.64	3.01

CPU usage recorded on the test hardware for a typical EdgeX deployment is approximately 10% and approximately 3% for a minimal deployment.

EdgeX Memory Consumption

Each EdgeX microservice has its memory consumption measured as it is started as a Docker container. The memory consumption is reported by the Docker engine and is measured in Megabytes (MB). In general, the measure of usage at startup is a good indication on the upper bound for many of the services.

Typical EdgeX Deployment

EdgeX Microservice	Average (MB)	Maximum (MB)	Minimum (MB)
Core Services			
edgex-core-data	12.92	13.61	12.30
edgex-core-metadata	13.15	13.53	12.91
edgex-core-command	9.20	9.99	8.08
edgex-redis	5.32	5.82	5.03
edgex-core-consul	48.89	49.09	48.67
Device Services			
edgex-device-virtual	13.96	14.41	13.44
edgex-device-rest	11.72	11.80	11.63
Supporting Services			
edgex-support-notifications	9.55	10.08	8.74
edgex-support-scheduler	9.91	10.35	9.24
edgex-kuiper	21.61	21.61	21.61
Application Services			
edgex-app-service-configurable-rules	11.86	12.08	11.26
TOTAL	168.09	172.34	162.90

Minimal EdgeX Deployment

EdgeX Microservice	Average (MB)	Maximum (MB)	Minimum (MB)
Core Services			
edgex-core-data	12.92	13.61	12.30
edgex-core-metadata	13.15	13.53	12.91
edgex-redis	5.32	5.82	5.03
Device Services			
edgex-device-rest	11.72	11.80	11.63
Application Services			
edgex-app-service-configurable-rules	11.86	12.08	11.26
TOTAL	54.97	56.84	53.13

Memory usage recorded on the test hardware for a typical EdgeX deployment is approximately 170 MBs and approximately 56 MB for a minimal deployment.

EdgeX Startup Times

The startup times are measured for each of the microservices developed by the EdgeX Foundry community.

Startup times include any overhead associated with creating the Docker container in which the microservices runs. Note that the metrics are obtained while starting all microservices at the same time so any dependencies between the services starting is included in the data. The total time recorded is the time taken for all of the services to be started, rather than a sum of all individual startup times.

Note also that services can be started and stopped as EdgeX runs. In this case it is not necessary to create a new Docker container for each service. Restarting an already-created Docker container reduces the startup overhead slightly

EdgeX Startup Times

EdgeX Microservice	Average (s)	Maximum (s)	Minimum (s)
Core Services			
edgex-core-data	9.10	11.34	7.82
edgex-core-metadata	9.15	11.19	8.03
edgex-core-command	9.10	11.15	7.69
Device Services			
edgex-device-virtual	17.64	18.51	16.46
edgex-device-rest	17.40	18.35	16.13
Supporting Services			
edgex-support-notifications	9.09	11.23	8.00
edgex-support-scheduler	9.17	11.29	8.06
Application Services			
edgex-app-service-configurable-rules	9.37	11.47	8.27
TOTAL	17.64	18.51	16.46

The startup time for the EdgeX microservices on the test hardware is approximately 18 seconds.



EdgeX Operational Latency

Ping and Data Export Latency

The ping responses times are measured for each of the microservices developed by the EdgeX Foundry community and represent the reactivity of each service when it receives an HTTP/REST request

Ping Response Times

EdgeX Microservice	Average (ms)	Maximum (ms)	Minimum (ms)
Core Services			
edgex-core-data	0.98	4.35	0.79
edgex-core-metadata	0.98	2.03	0.79
edgex-core-command	1.16	1.74	0.78
Device Services			
edgex-device-virtual	1.12	1.57	0.88
edgex-device-rest	1.12	1.52	0.87
Supporting Services			
edgex-support-notifications	1.14	1.59	0.92
edgex-support-scheduler	1.17	1.83	0.93
Application Services			
edgex-app-service-configurable-rules	1.23	6.03	0.93

Ping response times recorded on the test hardware are consistently around 1ms for each of the EdgeX microservices.

Data export latency represents the time it takes to collect EdgeX data from the “southside” devices & sensors and deliver it through the platform to the “northside” applications.

The measurement represents the time taken for an event reading to be created in a Device Service, delivered to the Core Data Service and then passed to an Application Service where it is read and prepared for northbound delivery. The event is extracted from the Core Data event message topic, sent to the designated endpoint and finally marked as ‘pushed’ in Core Data.

The latency reported here relates to the Virtual Device Service, which simulates 3 different data attributes. Each Device Service type (Modbus, BACnet, etc), will obviously have its own data rates associated with collecting the data from the physical edge devices or sensors, but the Virtual Device provides useful indicative performance data. It is planned to provide performance relating to more Device Service types in future versions.

Data Export Latency

Virtual Device Name	Average (ms)	Maximum (ms)	Minimum (ms)
Core Services			
Random-Integer-Device	3.5	7.0	2.0
Random-UnsignedInteger-Device	4.8	8.0	3.0
Random-Boolean-Device	6.0	9.0	4.0

The Virtual Device data is delivered through the EdgeX platform on the test hardware at an average latency of approximately 5 ms.

EDGE X FOUNDRY™

Test Methodology

All EdgeX microservices were run with their default configurations. In order to provide accurate and reliable statistics the tests were ran as follows:

- The CPU and memory consumption metrics were gathered by obtaining the data 10 times at an interval of 7 seconds between each measurement
- The startup time data was obtained by starting the EdgeX microservices 5 different times
- The ping response test was performed 100 times for each EdgeX microservice
- The data export latency metric was performed 10 times for each of the Virtual Device attributes

The full raw performance metrics, as well as data for other hardware platforms, is provided on the [EdgeX Community Wiki](#) [here](#)

Contact Us

For general information about Edge X, EdgeX Foundry, or membership inquiries, please email info@lfdedge.org

Visit our website at www.edgexfoundry.org

Have you got any questions?

This document was produced by IOtech Systems – the Edge Software Company whose products include Edge Xpert which is an enhanced, productized and commercially-supported implementation of EdgeX Foundry.

If you have questions about this report or if you need more information on IOtech or Edge Xpert, email info@iotechsys.com or visit the website www.iotechsys.com

The IOtech logo features the word "IOtech" in a white, sans-serif font. The letter "I" is stylized with a grid of small squares, resembling a digital or pixelated effect.