



§ OPEN SOURCE SUMMIT

China 2019



Integration of Edge and Cloud IoT Platforms

Drasko DRASKOVIC

Janko ISIDOROVIC



Authors



**Drasko
Draskovic**
Mainflux CEO & Founder

Drasko is the founder of Mainflux IoT open source project and member of TSC in EdgeX Foundry Project

Drasko is an IoT expert with over 15 years of professional experience. He hacked embedded Linux SW and HW device drivers, designing complex wireless systems in telecom industry.

He holds MSc. In Telecommunications.



**Janko
Isidorovic**
Mainflux COO & Co-founder

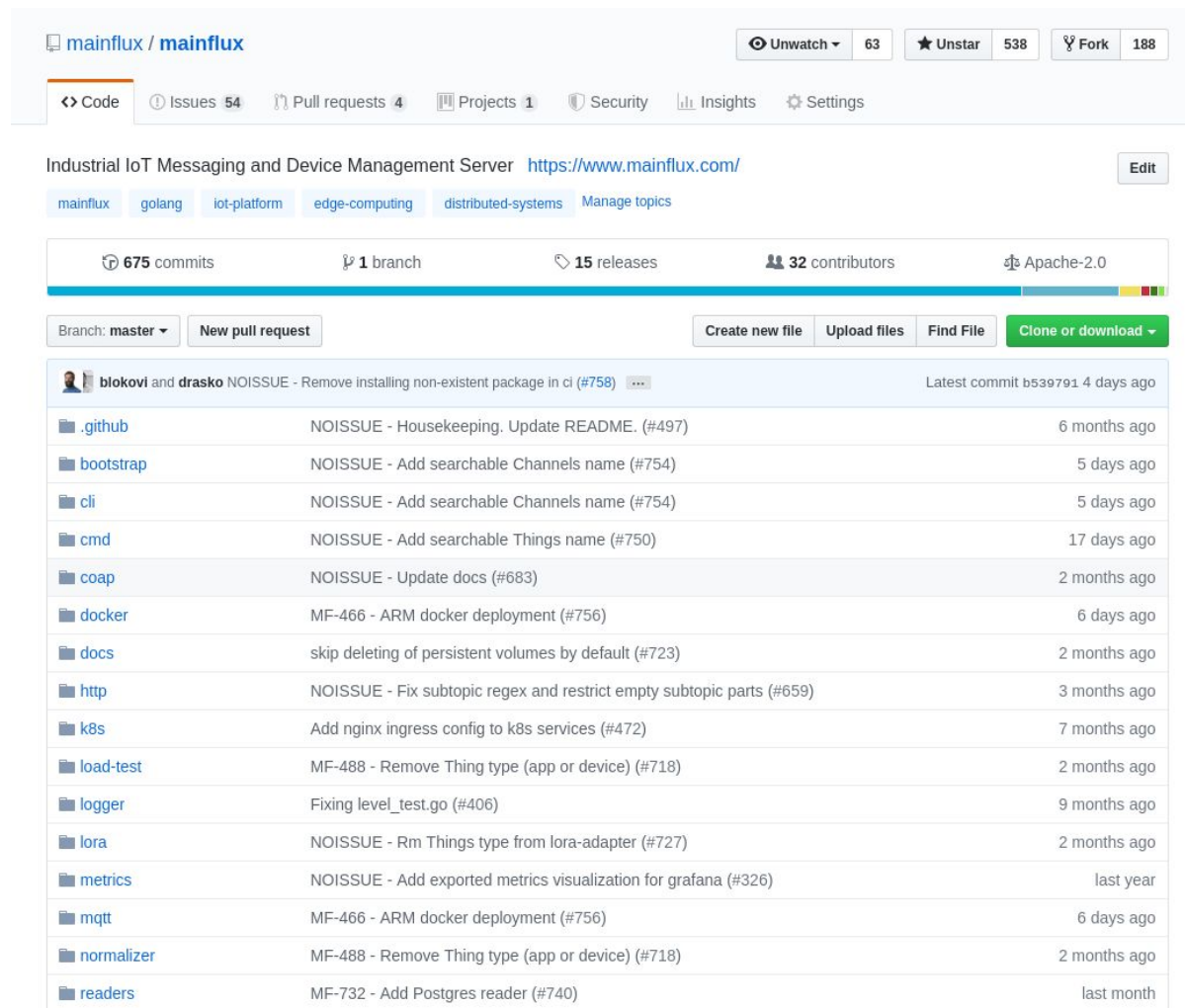
Janko is the co-founder of Mainflux IoT open source project.

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.

Mainflux Project

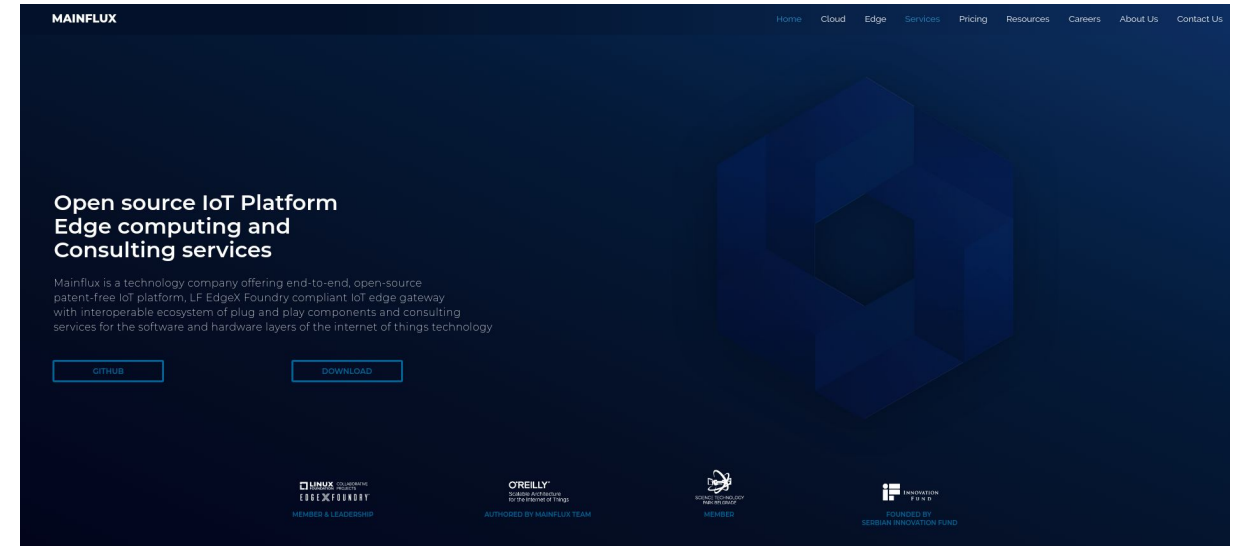
- <https://github.com/mainflux/mainflux>
- Open-Source, Apache-2.0 licensed, patent-free
- Over 30 contributors
- Written in Go
- Docker & Kubernetes
- Used in production



The screenshot shows the GitHub repository for the Mainflux project. At the top, the repository name 'mainflux / mainflux' is displayed, along with statistics: 63 Unwatch, 538 Unstar, and 188 Fork. Below this, navigation tabs for Code, Issues (54), Pull requests (4), Projects (1), Security, Insights, and Settings are visible. The repository description is 'Industrial IoT Messaging and Device Management Server' with a link to the website. A list of topics (mainflux, golang, iot-platform, edge-computing, distributed-systems) and a 'Manage topics' button are shown. Repository statistics include 675 commits, 1 branch, 15 releases, 32 contributors, and Apache-2.0 license. Action buttons for 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download' are present. A table of recent issues is displayed, showing the issue title, issue number, and the time since the latest commit.

| Issue Title | Issue Number | Latest commit |
|--|--------------|--------------------|
| NOISSUE - Remove installing non-existent package in ci | #758 | b539791 4 days ago |
| NOISSUE - Housekeeping. Update README. | #497 | 6 months ago |
| NOISSUE - Add searchable Channels name | #754 | 5 days ago |
| NOISSUE - Add searchable Channels name | #754 | 5 days ago |
| NOISSUE - Add searchable Things name | #750 | 17 days ago |
| NOISSUE - Update docs | #683 | 2 months ago |
| MF-466 - ARM docker deployment | #756 | 6 days ago |
| skip deleting of persistent volumes by default | #723 | 2 months ago |
| NOISSUE - Fix subtopic regex and restrict empty subtopic parts | #659 | 3 months ago |
| Add nginx ingress config to k8s services | #472 | 7 months ago |
| MF-488 - Remove Thing type (app or device) | #718 | 2 months ago |
| Fixing level_test.go | #406 | 9 months ago |
| NOISSUE - Rm Things type from lora-adapter | #727 | 2 months ago |
| NOISSUE - Add exported metrics visualization for grafana | #326 | last year |
| MF-466 - ARM docker deployment | #756 | 6 days ago |
| MF-488 - Remove Thing type (app or device) | #718 | 2 months ago |
| MF-732 - Add Postgres reader | #740 | last month |

- <https://www.mainflux.com/>
- Professional services around Mainflux IoT Platform
- Clients are big companies in the domain of Oil&Gas, Retail, Telecom, Semiconductor...
- MFX-1 IoT gateway with EdgeX Foundry SW
- Edgeflux gateway management SaaS
- Docker & Kubernetes
- Used in production



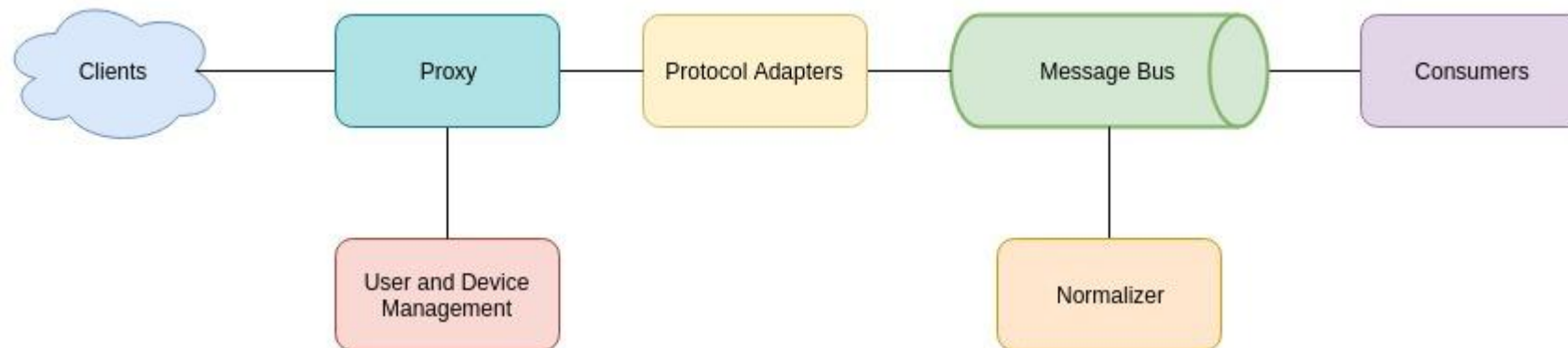


Cloud



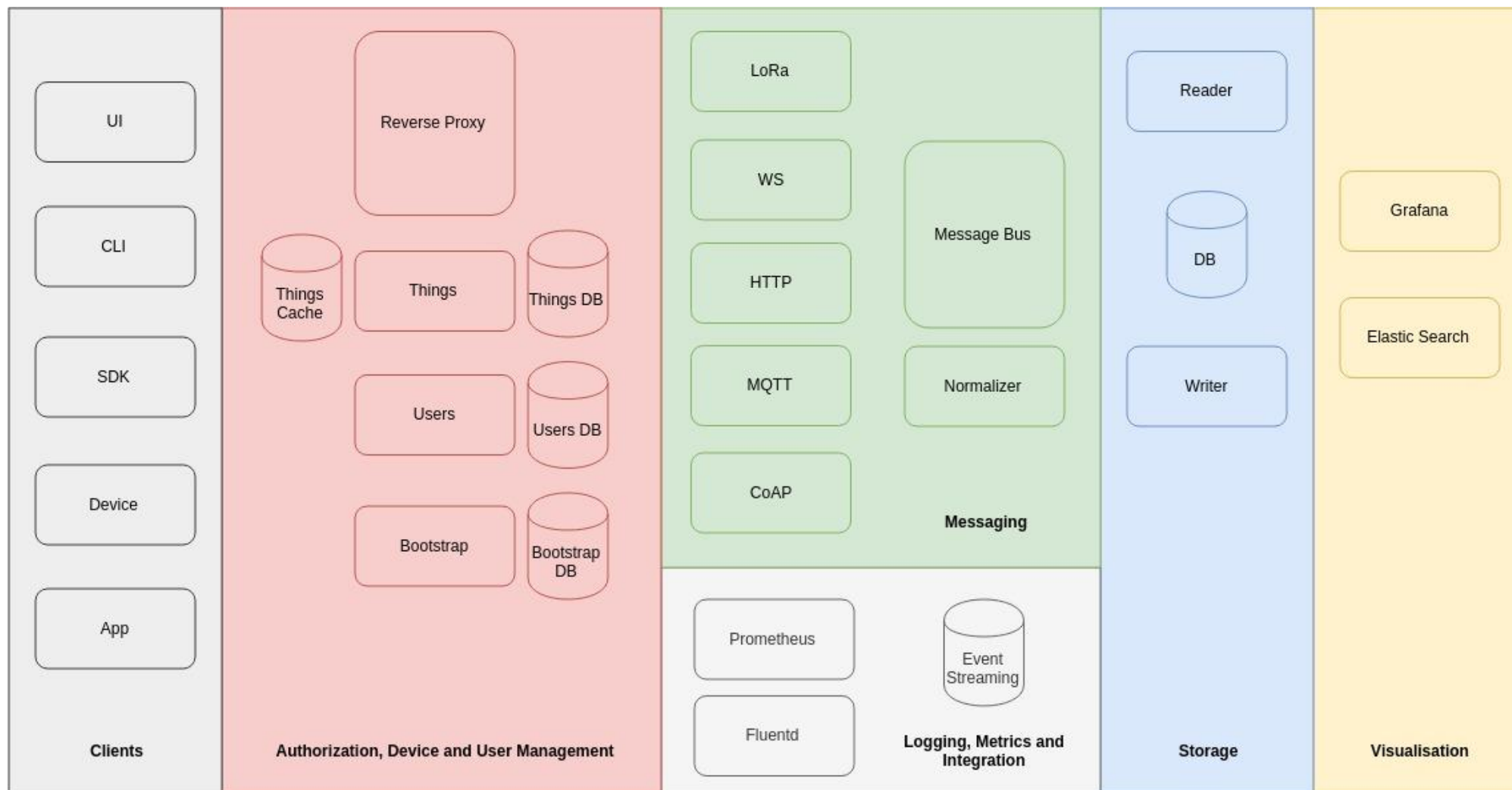
Mainflux - Overview

- Microservice Architecture
- Golang wherever possible - Go Kit
- NATS Message Bus
- NginX
 - TLS/DTLS Termination
 - Reverse Proxy for UI
- SQL database for structured data
- NoSQL database for Telemetry:
 - InfluxDB
 - MongoDB
 - Cassandra
- Mainflux Scales from PRi class devices to multi datacenter with Kubernetes and Cassandra DB
- Deployment:
 - Native
 - Docker containers (compose provided)
 - Kubernetes scripts



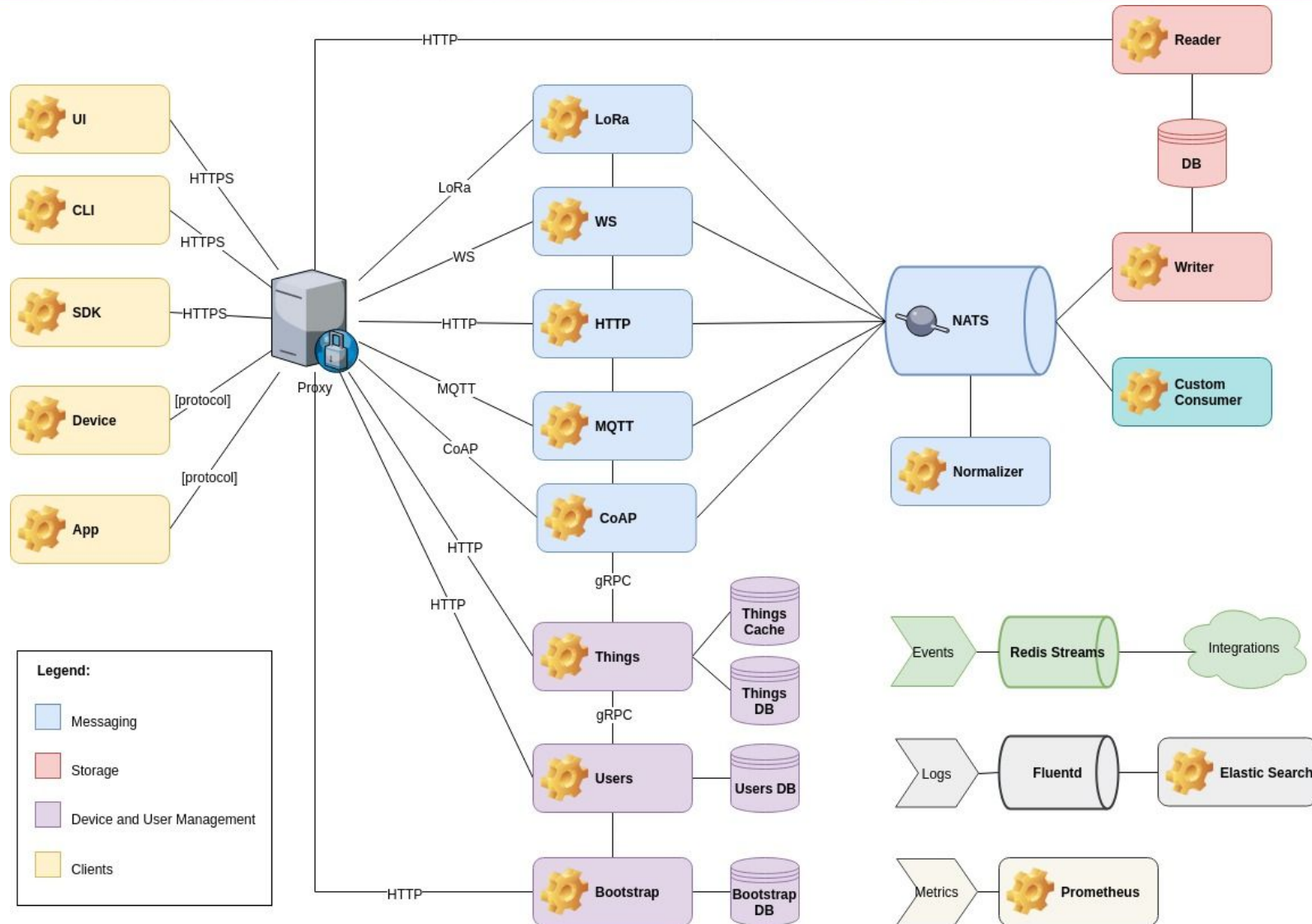
Mainflux - Architecture

S OPEN SOURCE SUMMIT
China 2019



Mainflux - Architecture

OPEN SOURCE SUMMIT
China 2019



Mainflux - Architecture

- **Users**

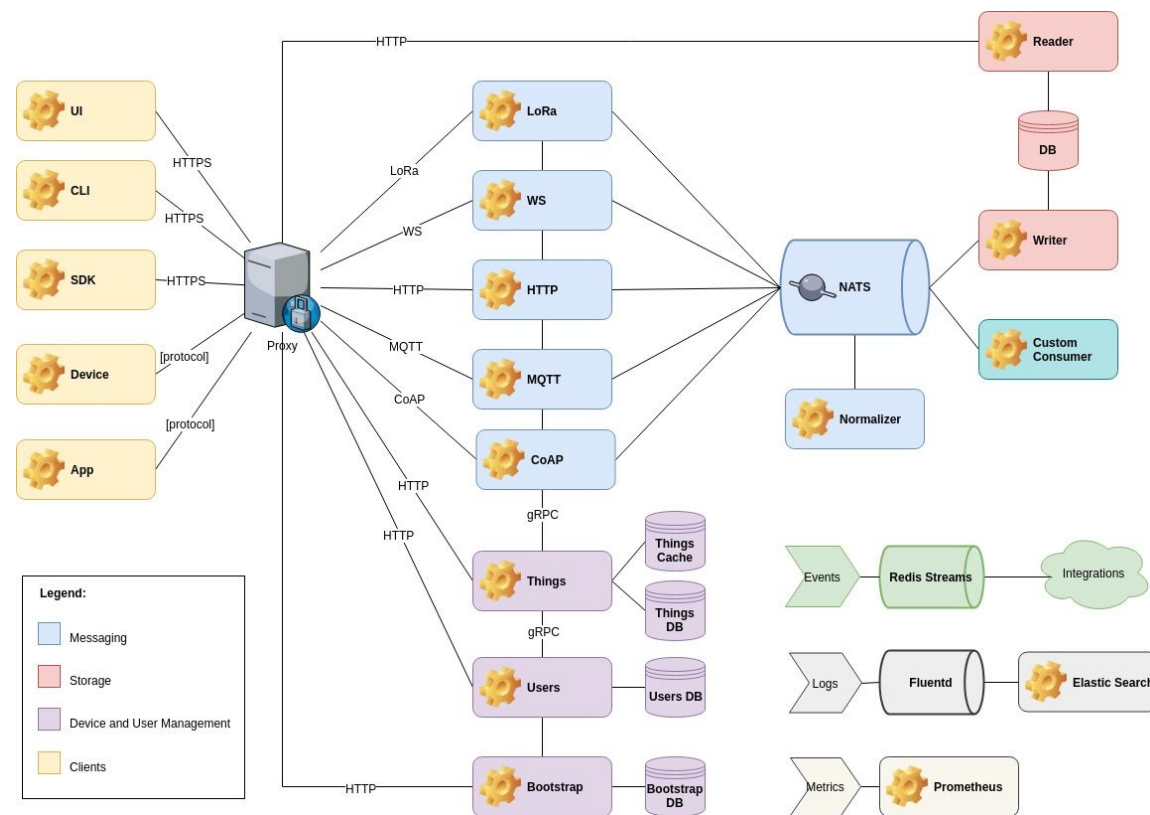
- User represents a human user of the system and is basic entity in the Mainflux IoT Platform. User is authenticated by email and password.
- Once authenticated user receives JWT to use for further actions.
- Each user is Admin within his domain

- **Things**

- Connected devices and applications are the same entity. We call them Things.
- Internal representation of every device is saved to database

- **Channels**

- Channel connects Things (devices and/or applications)
- Only Things connected to the same Channel can communicate with each other.



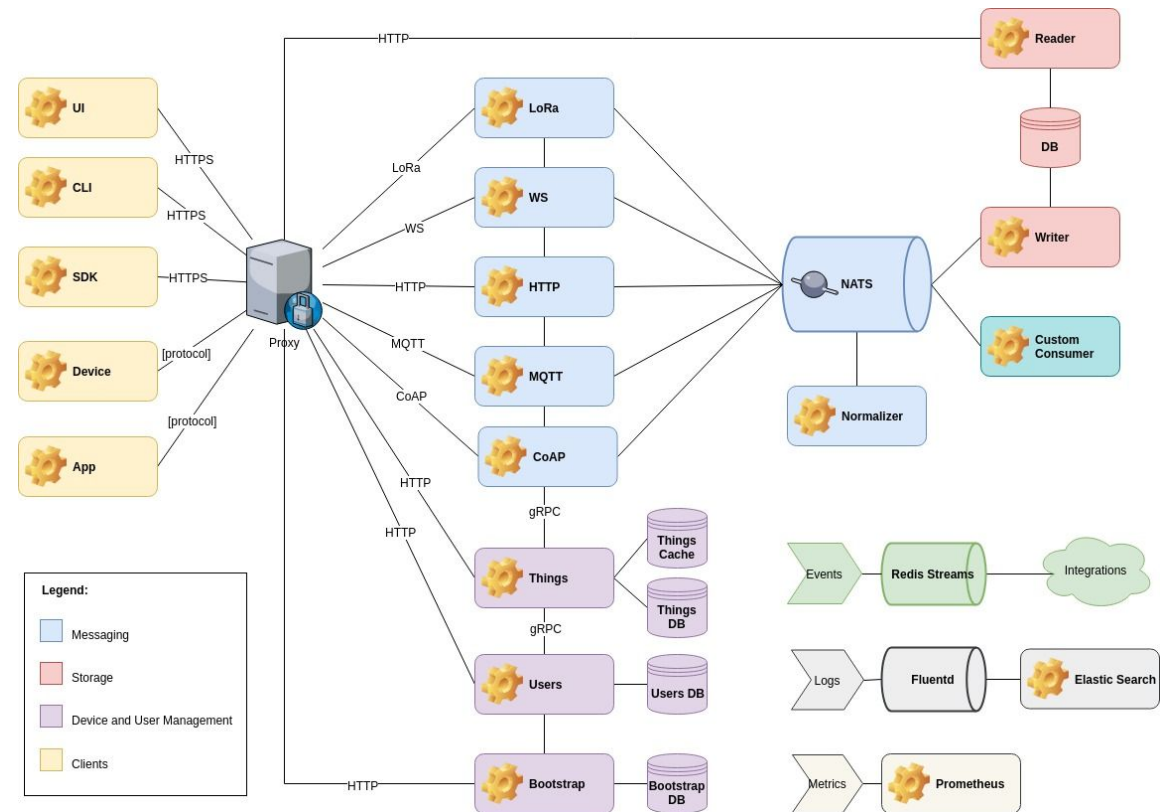
Mainflux - Security

Authentication with Mainflux keys

- Mainflux key is a secret key that's generated at the Thing creation

Mutual TLS Authentication with X.509 Certificates

- Client-to-server authentication using client-side X.509
- This is called two-way or mutual authentication
- Mainflux supports mTLS over HTTP, WS, and MQTT protocols (no CoAP at the moment)
- Thing key will be used to create x.509 certificate
- HTTPS – Authorization header does not have to be present
- MQTTS – Password filed in CONNECT message must match the key from the certificate
- WSS – Authorization header or authorization query parameter must match cert key



Mainflux - DevOps

- **Events**

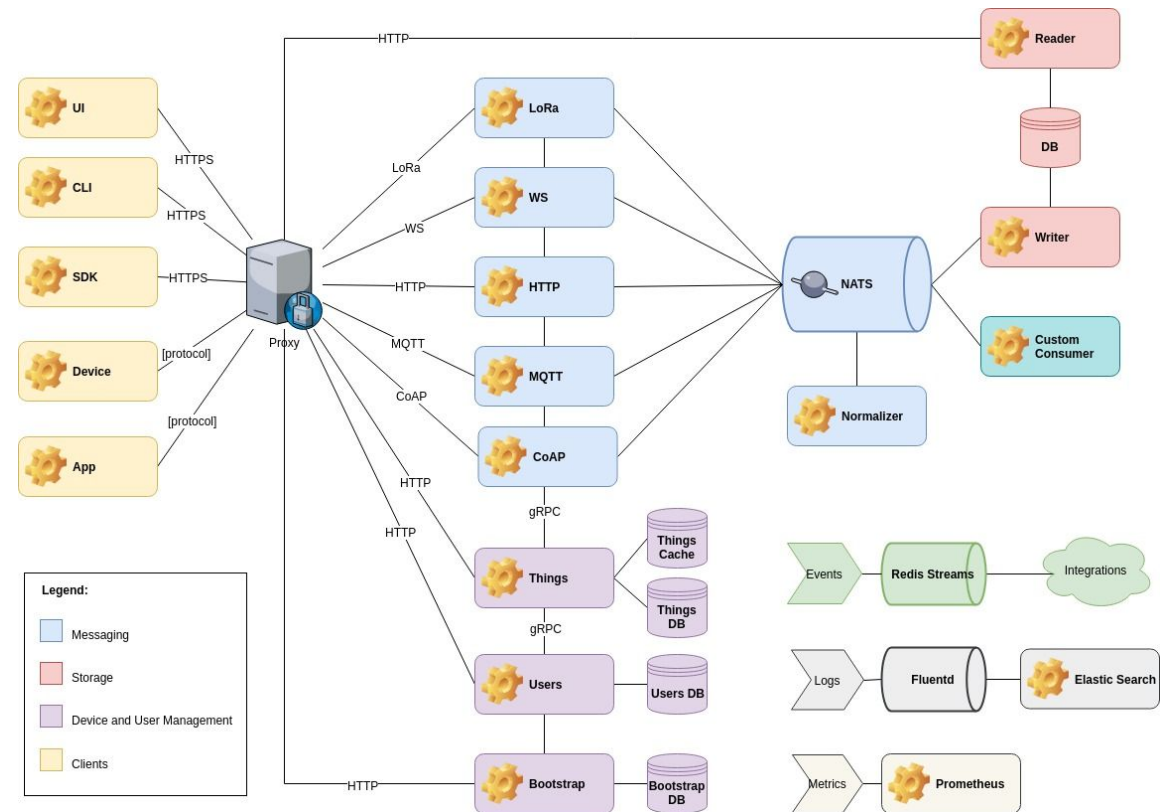
- Events are published to Redis Stream. This is useful for integration with 3rd party systems

- **Logs**

- Fluentd
- Elastic Search
- Kibana

- **Metrics**

- Prometheus
- Grafana





Edge



An open source, vendor neutral project (and ecosystem)

A **microservice**, loosely coupled software framework for IoT edge computing

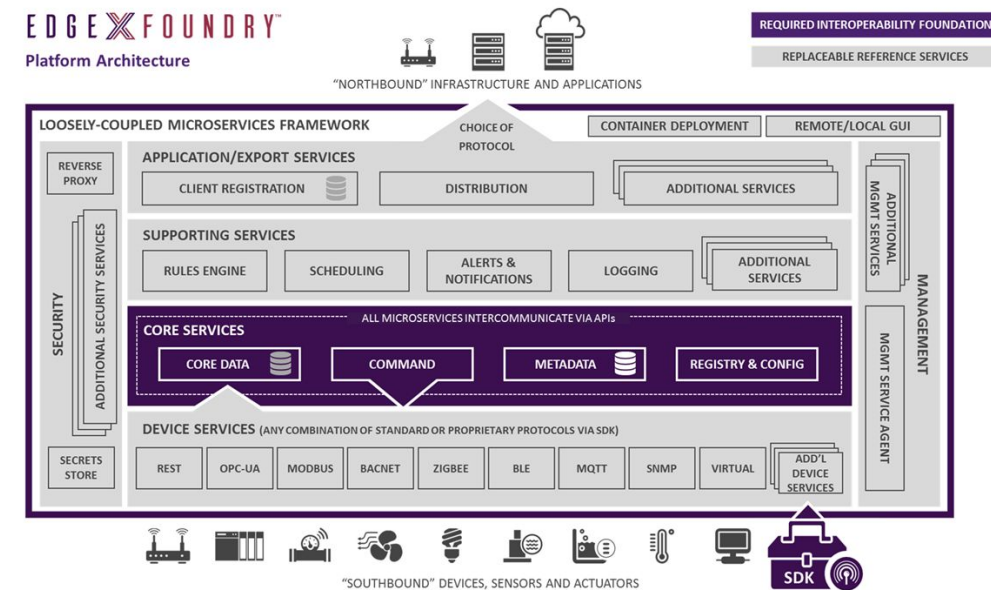
Hardware and OS agnostic

Linux Foundation, Apache 2 licensed 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

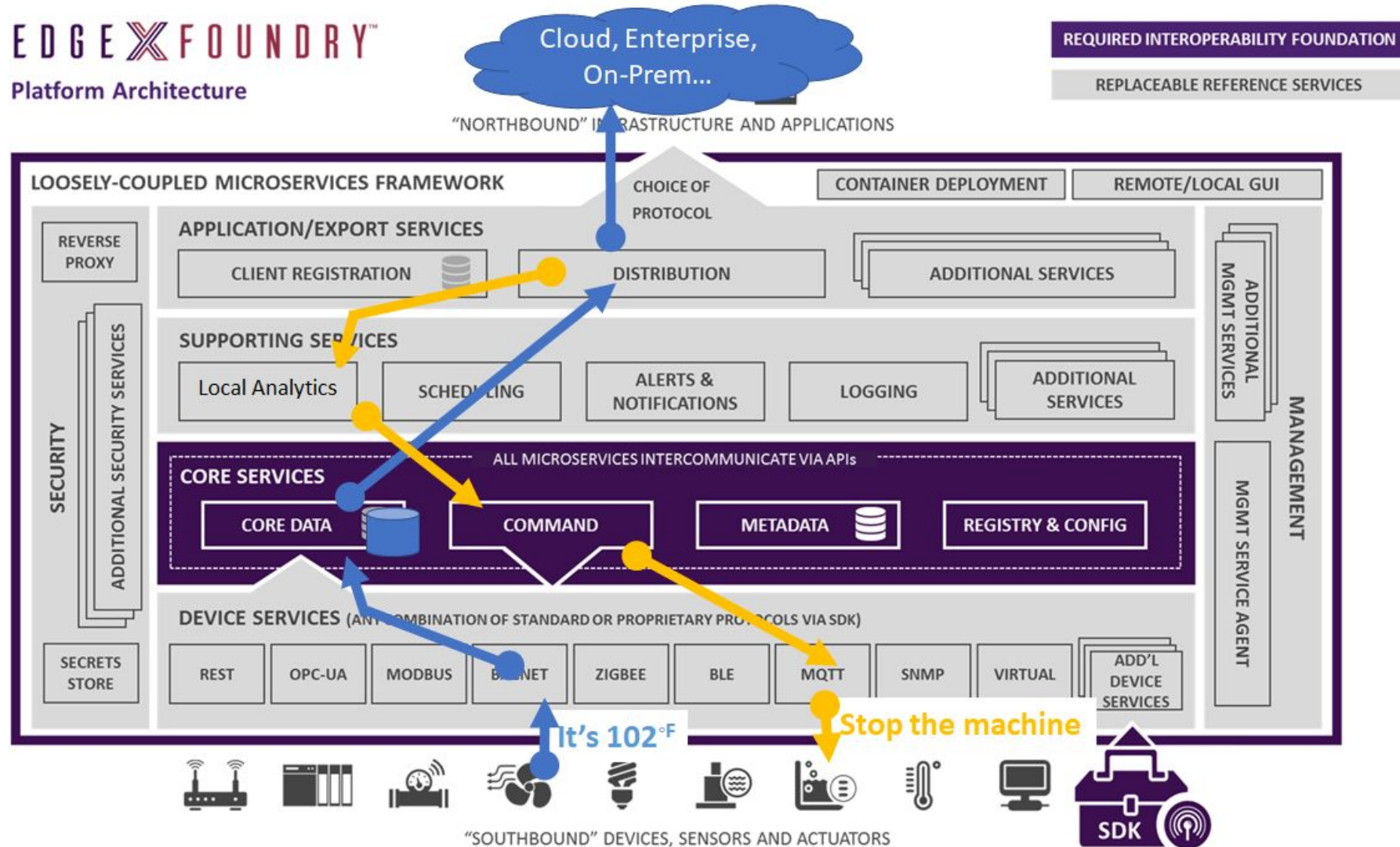
- A collection of a 10+ microservices
- Written in multiple languages (Java, Go, C, ... we are polyglot believers!!)
- EdgeX data flow:
 - Sensor data is collected by a **Device Service** from a thing
 - Data is passed to the **Core Services** for local persistence
 - Data is then passed to **Export 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



EdgeX - How it Works

OPEN SOURCE SUMMIT
China 2019

EDGE X FOUNDRY™
Platform Architecture





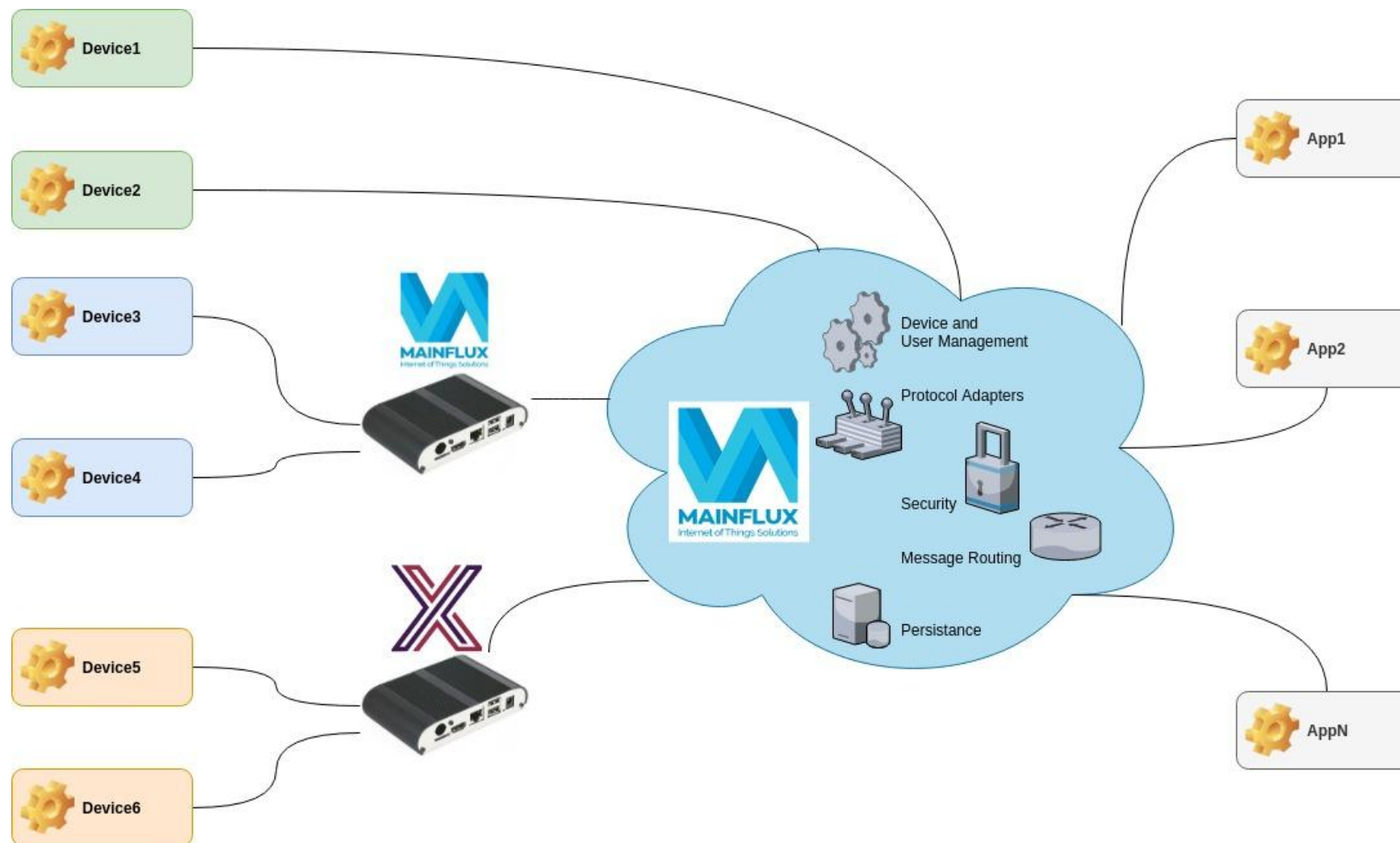
Integration



Edge-Cloud

OPEN SOURCE SUMMIT

China 2019



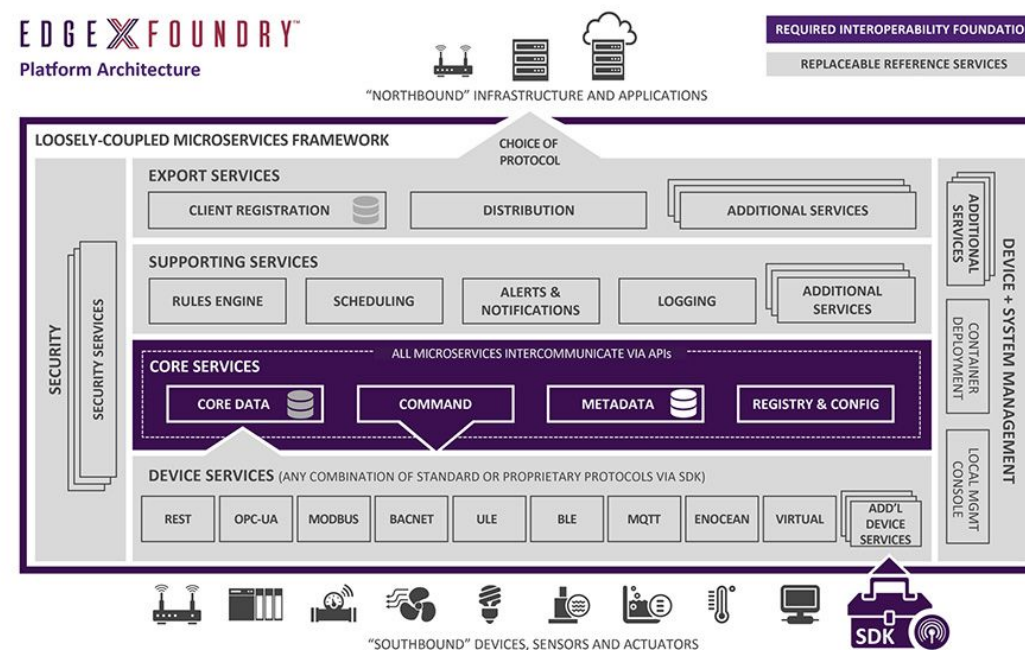
Mainflux on the Edge

- Unified IoT Platform - same code in the cloud and on the gateway
- Strong security - X.509 certificates and device tokens
- Small footprint - Docker images ~5MB
- Edged - IoT agent on the gateway
- Elm-based UI for administration



EdgeX on the Edge

- Plethora of Device Services
- Linux Foundation LF Edge project
- MFX-1 distribution maintained by Mainflux Labs company
- Edged by Mainflux - IoT agent on the gateway
- SMA - remotely managed
- UI by VMware
- **Edgeflux** - full-blown cloud UI for managing EdgeX gateways - COMING SOON!





Thank You!

