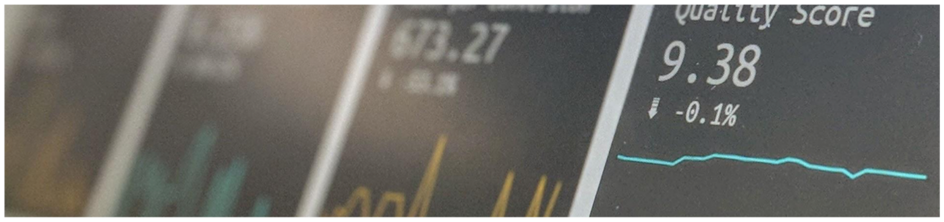
ELIoT Backbone Complete Feature List

Detailed overview of key functionalities of the IoT Backbone Platform.

IoT Backbone solution was built with maximum effort to provide customers with a cost-effective, scalable and highly reusable corporate IoT platform. Design was shaped in order to address high variety of business use cases that can be seamlessly integrated using standard interfaces.

Cost-Effectiveness

The key principle considered while building this platform was the fact that the product won't be used without the clear business case – the price of underlaying infrastructure simple matters. Therefore platform is adapting concept of shared or serverless infrastructure to significantly reduce costs while still allowing almost unlimited horizontal and vertical scaling. More you use, more you pay. That is fair!



Scalability

The design of the solution relies on an idempotent streaming layer that is highly scalable and provides customers with very low latencies between acquisition and serving. Both factors were repeatedly tested using load testing toolset that is bundled with the platform, including load simulator and performance dashboards.

Reduced Costs Factor

- Every component was chosen with respect to the **price** as a one of the **most important criteria.**
- **Serverless infrastructure** is a great mean how to make solution scalable and allows you to **pay** only for **what you** really **consume**.
- Fixed costs of components can be broken down and billed as a **shared costs.**
- Thanks to very thorough tagging approach, **costs** can be very easily **tracked** on **very granular level**.

Latency and Throughput

- **Idempotent** stream processing design enables **higher parallelism** and thus **higher throughput**
- All components are designed as **horizontal scalable**, temporal load peaks are **automatically handled** with **no latency impact**

ELIoT Backbone Complete Feature List

Detailed overview of key functionalities of the IoT Backbone Platform.

Security

Because the platform is built on top of the cloud service, it also inherits most of its guarantees (e.g. ISO certifications). All endpoints are restricted only to mutual service communication and secrets are automatically rotated according to customer standards. Identity management is ensured by Active Directory that jointly with central API management ensures proper security standards.

Microservice Architecture

The purpose of this platform is to be a backbone of the wider IOT solution. For this reason platform is designed to be modular and follow microservice architecture. This is valid not just for internal setup of components but also externally. Every component that requires configuration exposes its own API and automatically generated documentation using Open API standards via Swagger files. This enables clear separation between general ingestion and use case specific layer.

Platform Operations

Platform is designed as fault-tolerant, that means that in case of outage of any component, no message is lost, in case of anomality, these messages are stored in poison queues and thus accessible for platform operators. In addition to that every component is connected to central logging system and equipped with set of alerting rules that in cases of failure automatically sends pre-defined notifications.

Security Measures

- Identity management is ensured by **Active Directory**
- **Access** to the platform **is restricted** and exposed through the **single end-point** (API Management Service).
- **Key rotation mechanism** is put in place and ensure proper secret management.
- Platform **shares** most of the **security guarantees as** underlying **cloud service**

Microservice Architecture

- Composition of **loosely coupled** services/components
- Every **component** that requires authentication **exposes API**
- **Documentation** via **Open API format** (using Swagger files)

Platform Operations

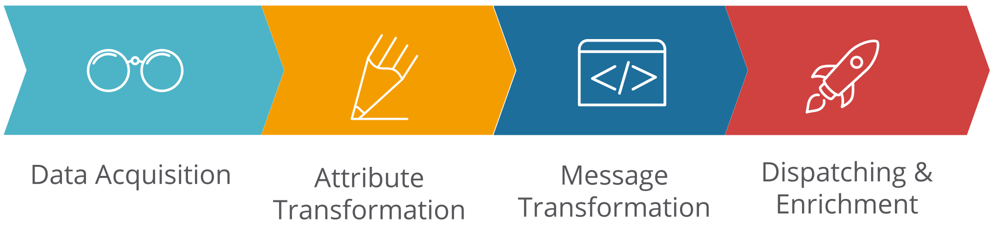
- **Fault-tolerant architecture** (no message is lost any time)
- **Logging is centralized**
- **Alert mechanism in place**

ELIoT Backbone Complete Feature List

Detailed overview of key functionalities of the IoT Backbone Platform.

Modular and Highly Configurable Ingestion Pipeline

Ingestion pipeline as a core part of the platform Is designed as highly configurable automatic ingestion framework. For this reason it consists of four microservices in a series each of them handling different type of operation.



Acquisition of Telemetry and Device Data

Platform implements automatic based ingestion mechanism for both telemetry and device metadata information using push and pull scenarios. Platform has already predefined templates for different type of 3rd party device providers as SigFox etc. Whole mechanism is configuration based so onboarding of new group of devices is just a matter of one API call.

Attribute Level Transformation

The role of this component is to apply arbitrary transformation on the one or multiple values according to the provided configuration. Outputs of this step are than stored as a derived attribute next the original information. The example can be a decoder of SigFox messages which takes the binary (encoded) input and outputs one or multiple decoded values.

Message Level Transformation

This component handles message transformation into canonical format, .i.e. vendor agnostic common format. Transformation is conducted based on the provided transformation schema. Attributes that are not directly contained in transformation schema are not lost but kept as a part of the message.

Dispatching & Enrichment

Last component in a pipeline deals with the optional enrichment of the data, based on the information provided by the customer and the destination dispatching. According to the configuration messages are distributed to different tiers of storage as cold, warm or hot that are further being consumed by use cases.

Ingestion Process

- **Modular** pipeline with high degree of flexibility thanks to in-built **configuration-based processing**.

- **Acquisition** of telemetry and device data **regardless on the service/device provider**
- **Supports** both **push** and **pull** data **integration** scenarios
- Platform provides **a set of predefined** vendor-specific data acquisition **templates**.

- Platform enables users to configure **transformations** that makes messages unified and following **canonical format.**
- Transformation is done in the way that **no information is lost.**

- Transformed can be further routed and dispatched to different type of storage engines.
- Routing is **configuration based**
- Supported are **COLD**, **WARM** and **HOT storage** engines or paths.

ELIoT Backbone Complete Feature List

Detailed overview of key functionalities of the IoT Backbone Platform.

Data Serving

Platform has an ability to dispatch messages to dedicated storage engines. On top of these engines the platform exposes APIs that enables users to paginate over persisted records. Alternative could can be purely the path that archives the events for purely analytics purposes. In this case data are accessible in cold storage engines. Users can access only records they are authorized to.

How to access the data?

- **Data** can be **dispatched based** on the use case **requirements** to different type of storage engines
- **Access** is restricted to **authorized** users **only**
- Platform provides an **API** how **to** retrieve most **recent data**

Alerting and Notification

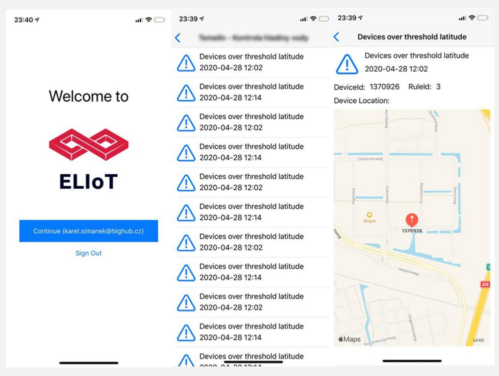
Alerting service a key component of the platform that allows users to setup a rules that are evaluated for every new message. All alerts are subsequently temporarily persisted and accessible through the API. In case of real-time serving demand these alerts can be sent to a hot path for further processing. Special case is the notification service that enables users to promptly react to some fresh events and send a notification as email or push notification.

How to be notified?

- Alerting **rules** can be **evaluated** for **every incoming message**
- **Alerts** are **accessible** through the **API** or can be **emitted** to **HOT path**.

White-label Mobile and Web App for Clients

In addition to previous functionalities, platform is shipped with pre-fabricated (white-label) apps for client applications. Easily customizable iOS application written in native Swift for real-time notification delivery and Java Script based front-end as a baseline for arbitrary client application. The main advantage of this addon is its eventual usage as a template for mutual communication with the platform (including authentication).


23:40
Welcome to
ELIoT
Continue (karel.simanek@bighub.cz)
Sign Out
23:39
[redacted]
23:39
Devices over threshold latitude

Devices over threshold latitude 2020-04-28 12:02
Devices over threshold latitude 2020-04-28 12:14
Devices over threshold latitude 2020-04-28 12:02
Devices over threshold latitude 2020-04-28 12:14
Devices over threshold latitude 2020-04-28 12:02
Devices over threshold latitude 2020-04-28 12:14
Devices over threshold latitude 2020-04-28 12:02
Devices over threshold latitude 2020-04-28 12:14
Devices over threshold latitude 2020-04-28 12:02
Devices over threshold latitude 2020-04-28 12:14
Devices over threshold latitude

Devices over threshold latitude 2020-04-28 12:02
DeviceId: 1370926 RuleId: 3
Device Location: