An introduction to the Data Services Hub

What is the DSH?

What is the DSH?

AWESOME

Why is the DSH awesome?

Why is the DSH awesome?

Data platform

Why is the DSH awesome?

Data platform

Data (as events)

Why is the DSH awesome?

Data platform
Data (as events)
Sharing

Why is the DSH awesome?

Data platform
Data (as events)
Sharing
Processing

Why is the DSH awesome?

Data platform
Data (as events)
Sharing
Processing
Scalable

Why is the DSH awesome?

Data platform
Data (as events)
Sharing
Processing
Scalable
Secure

Why is the DSH awesome?

Data platform
Data (as events)
Sharing
Processing
Scalable
Secure
Low-latency

What is the DSH?

What is the DSH?

A platform that does something with *streaming data*

Definition: platform

- A (software) platform is anything you can build (applications) on
- Provides reusable infrastructure
- Takes care of recurring and tedious tasks
- Should not hamper creativity

Definition: Streaming Data

...data that is generated continuously by thousands of data sources, which typically send in the data records simultaneously, and in small sizes (order of Kilobytes).

https://aws.amazon.com/streaming-data

Our definition: Streaming Data

A streaming data platform should also be able to continuously send selected data records to thousands of data sinks.

Data Streams

Data Streams

The DSH holds many different *data streams*

Data Stream

A sequence of digitally encoded signals used to represent information in transmission.

Federal Standard 1037C

Types of streaming data

Not all datastreams are created equal

Types of streaming data

Not all datastreams are created equal

Types of streaming data

Not all datastreams are created equal

```
$$ \begin{align}
    \text{One
   source, low
   volume} &|
   \text{many
  sources, high
   volume} \\\
   \text{Single
    sensor} &|
  \text{Stream
 processing} \\\
```

MQTT

- Messaging protocol
- ISO/IEC 20922 and, OASIS standard
- Lightweight messaging protocol
- Suitable for many simultaneous connections
- Widespread in the *Internet of Things*

Kafka

- Can handle huge volume of data
- Event-based
- Fast!

Kafka

- Can handle huge volume of data
- Event-based
- Fast!
- Messaging backbone for:
  - LinkedIn
  - Netflix
  - Yahoo
  - Twitter

MQTT vs Kafka

- MQTT
  - *usually* low volume *(default 10 msgs/sec)*
  - can have many sources/sinks (millions)
  - sources/sinks can reside outside of DSH
- Kafka
  - can have high volume (millions of

msgs/sec)

- *must* have few sources/sinks
- sources/sinks reside inside DSH

MQTT vs Kafka

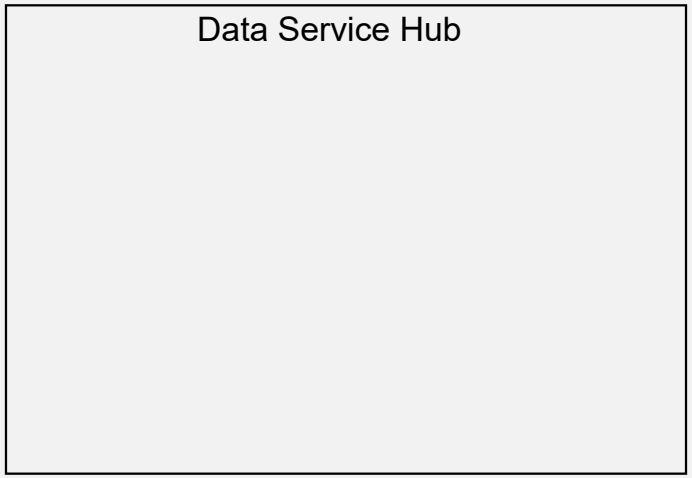
- MQTT
  - *usually* low volume *(default 10 msgs/sec)*
  - can have many sources/sinks (millions)
  - sources/sinks can reside outside of DSH
- Kafka
  - can have high volume (millions of

msgs/sec)

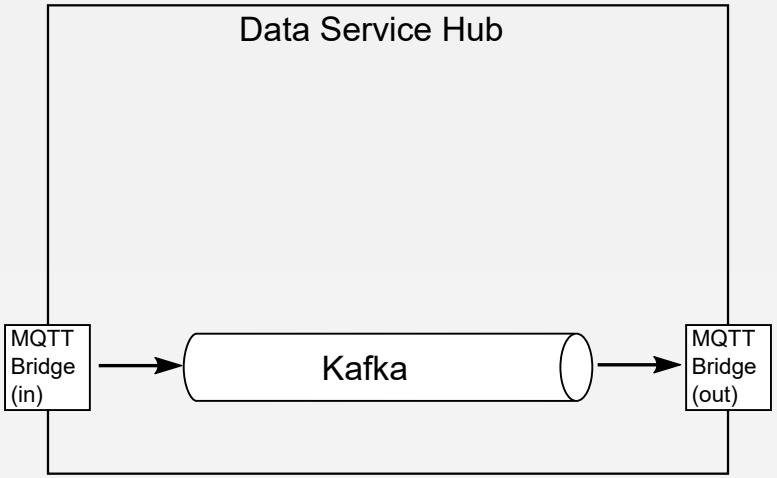
- *must* have few sources/sinks
- sources/sinks reside inside DSH

$$ \text{MQTT} \cdot \frac{sources}{sinks} \approx \text{Kafka} \cdot \frac{sources}{sinks} $$

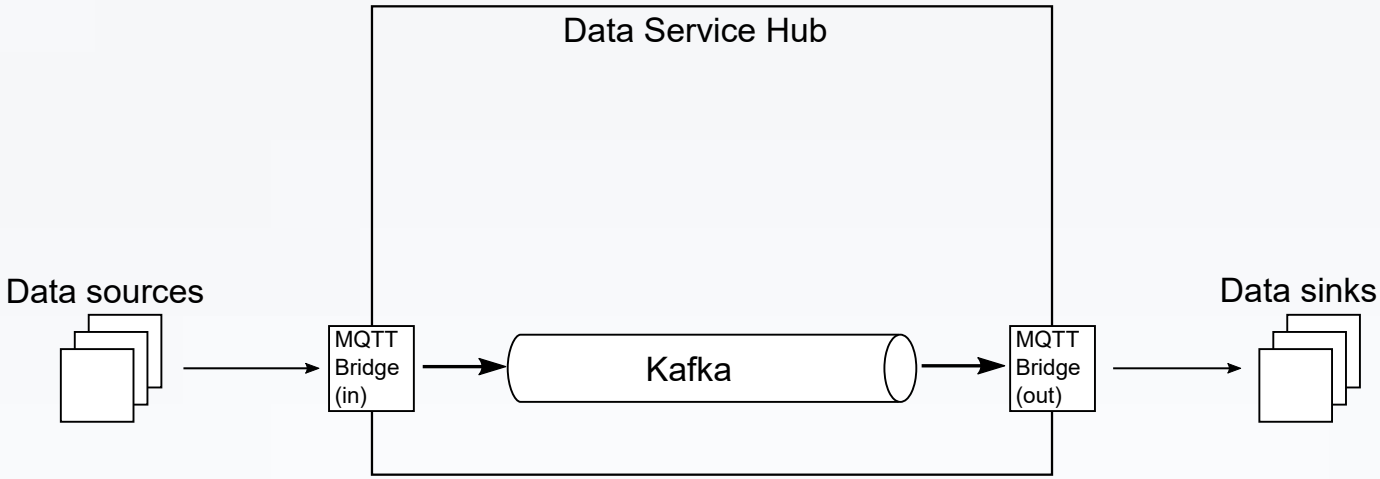
Overview



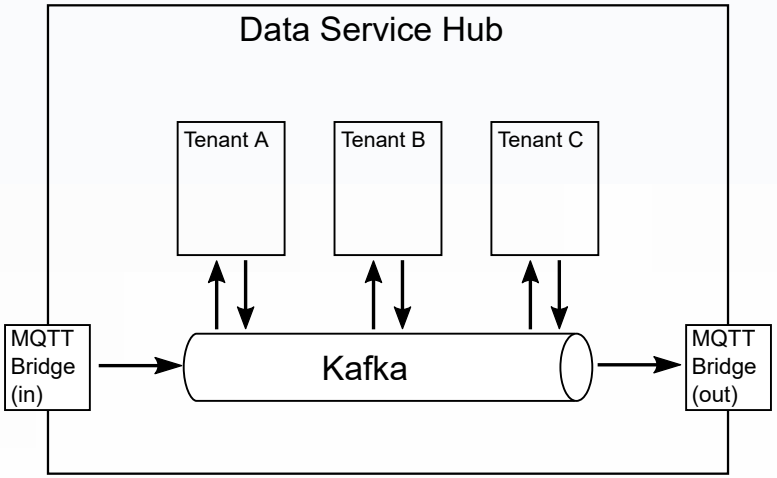
Overview



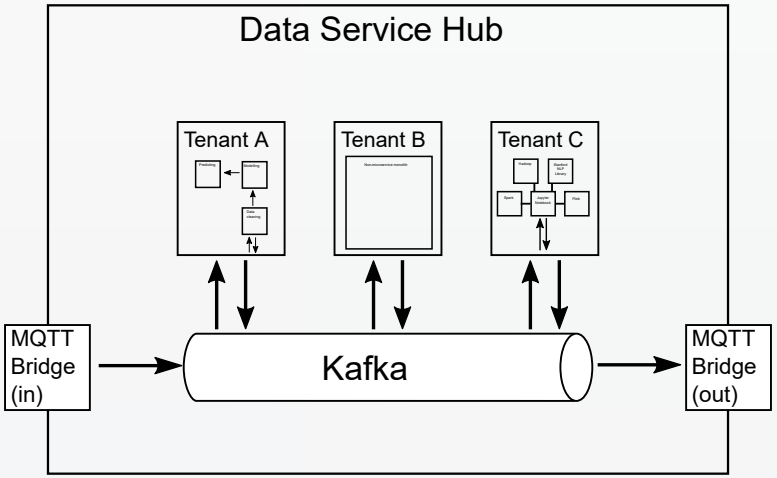
Overview


Data Service Hub
Data sources
MQTT
Bridge
(in)
Kafka
MQTT
Bridge
(out)
Data sinks

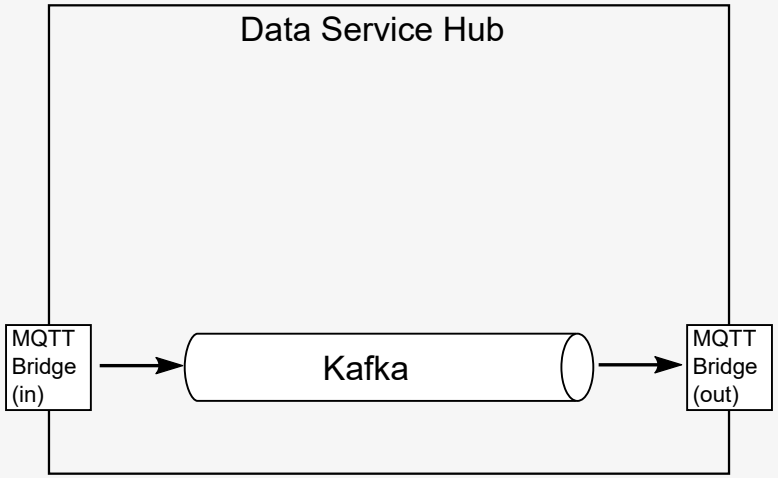
Overview


Data Service Hub
Tenant A
Tenant B
Tenant C
MQTT
Bridge
(in)
Kafka
MQTT
Bridge
(out)

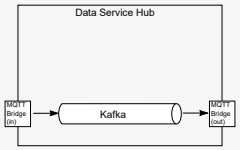
Overview


Data Service Hub
Tenant A
Tenant B
Tenant C
MQTT
Bridge
(in)
Kafka
MQTT
Bridge
(out)

MQTT

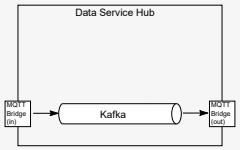


MQTT bridge



- Protocol adapter
  - MQTT interface with Kafka

MQTT bridge



- Protocol adapter
  - MQTT interface with Kafka
- Like MQTT: allows wildcard subscriptions:

```
/platform/stream/topic/#
```

External data sources

External data sources

- are not always MQTT ...

External data sources

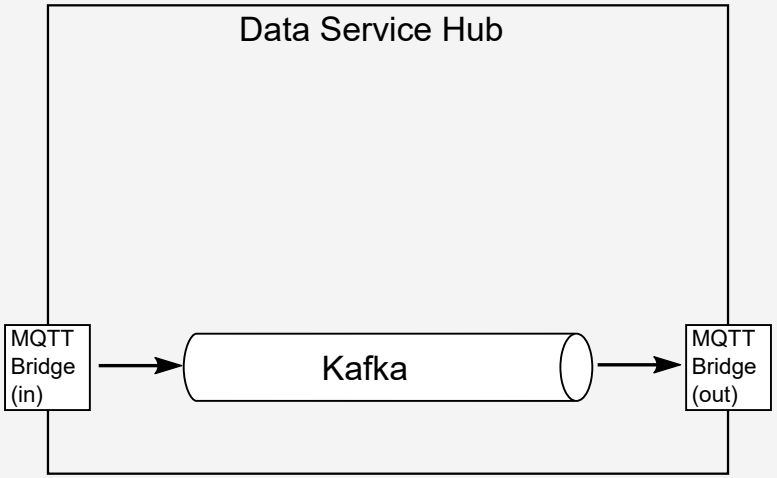
- are not always MQTT ...
- ... and will therefore require custom adapters

External data sources

- are not always MQTT ...
- ... and will therefore require custom adapters

We allow tenants to write their own adapters

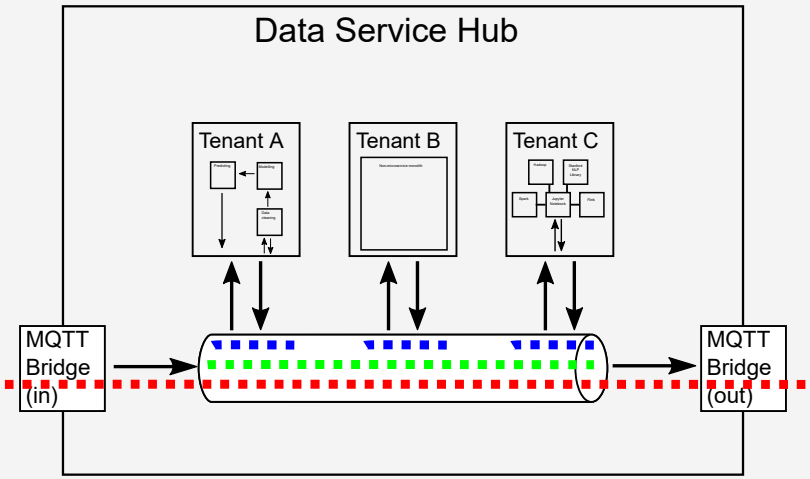
Kafka



Kafka

Three Kafka stream-types

- *stream.* topic
- *internal.* topic
- *scratch.* topic


Data Service Hub
Tenant A
Tenant B
Tenant C
MQTT
Bridge
(in)
MQTT
Bridge
(out)

Many data streams

- Streams need organizing
- DSH topics $ \approx $ Kafka topics
- Need to control access to topics

Stream Processing

Stream Processing

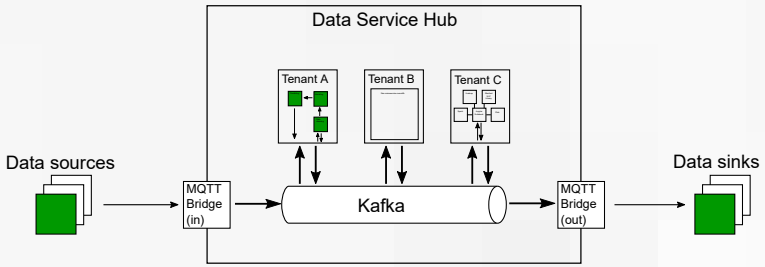
The DSH is a platform that does *stream processing*

Stream Processing

... is the processing of data in motion, or in other words, computing on data directly as it is produced or received.

https://data-artisans.com/what-is-stream-processing

Where to process



- At the source?
- On the DSH?
- At the sink?

Many ways to process the data

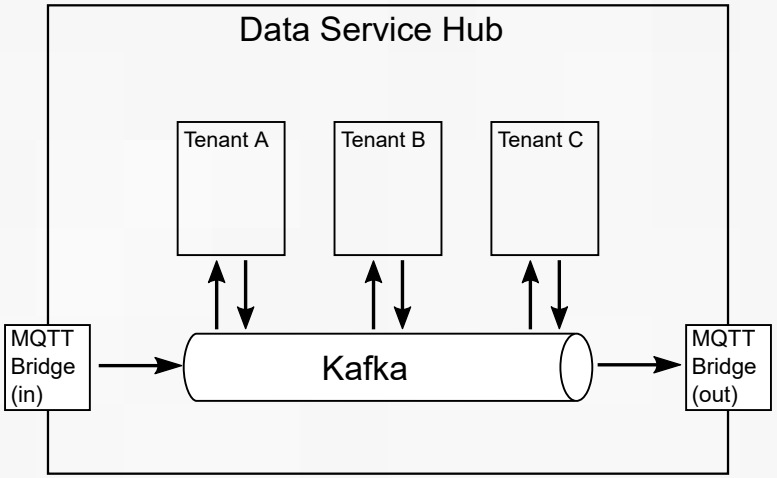
- Many frameworks for stream processing
- No framework fits all use-cases
- DSH does not dictate a framework

No *One framework to rule them all*, but the DSH to *bind them*.

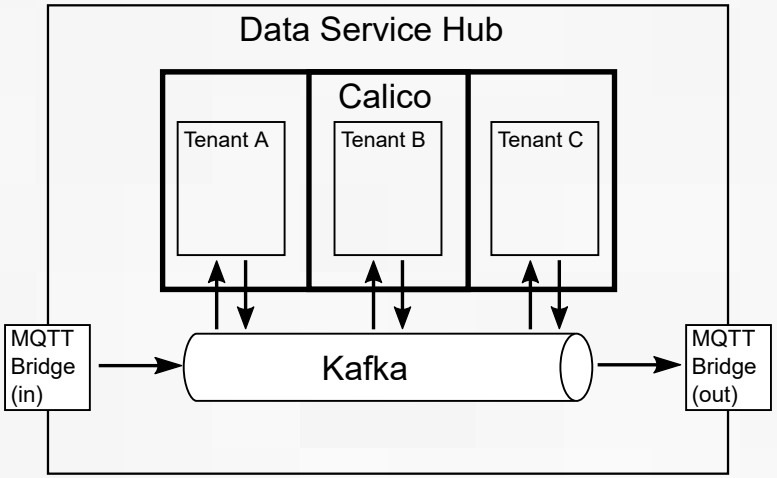
Security nightmare

- Need to allow other people on your platform for proximity
- And they can use whatever software they want on the platform

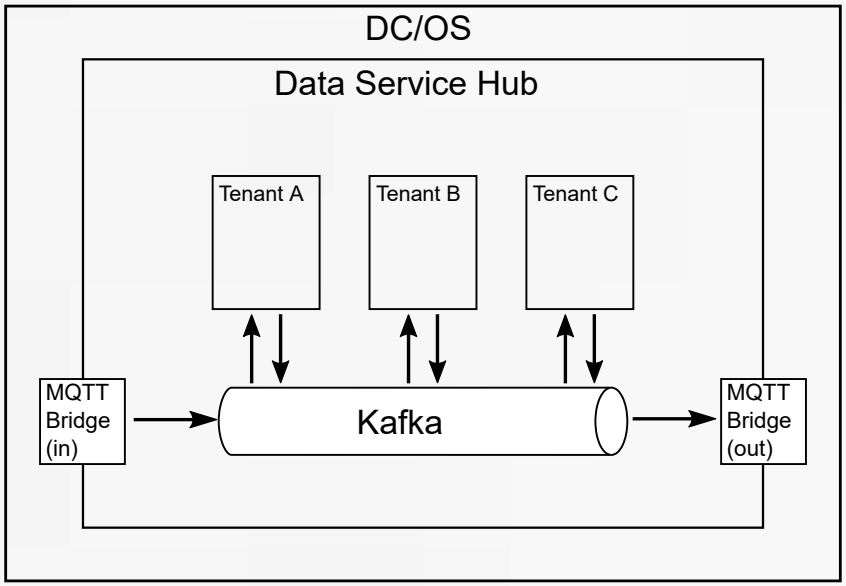
Base DSH

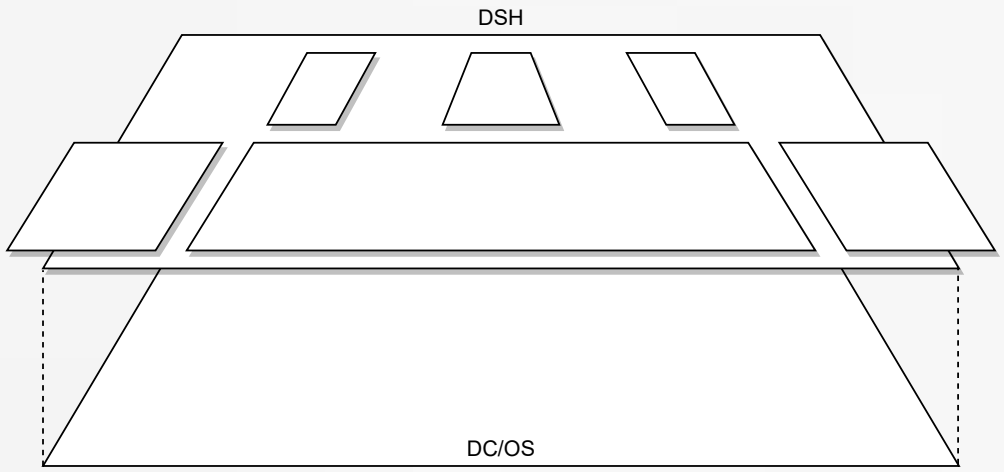

Data Service Hub
Tenant A
Tenant B
Tenant C
MQTT Bridge (in)
Kafka
MQTT Bridge (out)

Calico


Data Service Hub
Calico
Tenant A
Tenant B
Tenant C
MQTT
Bridge
(in)
Kafka
MQTT
Bridge
(out)

DC/OS


DC/OS
Data Service Hub
Tenant A
Tenant B
Tenant C
MQTT Bridge (in)
Kafka
MQTT Bridge (out)


DSH
DC/OS

Securing

- Custom container manager
  - for ease of use
  - to *force* correct use of Docker
- Custom resource manager
- Calico to ensure network isolation

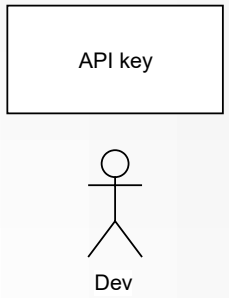
Wrap-up

- DC/OS as base
- Docker + extra security
- Tenant network isolation

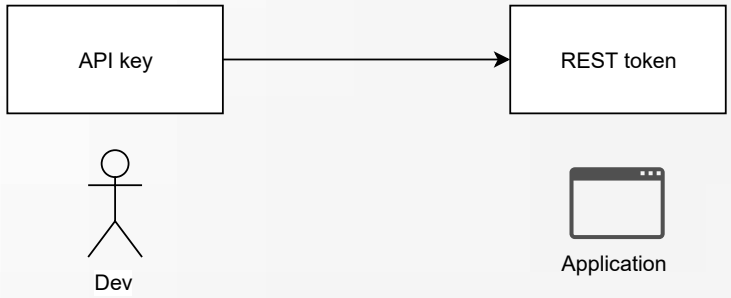
Authentication Nightmare

- Certificates for tenant (container) authentication towards Kafka
- API key to authenticate tenants that want to let devices/things/users connect to the platform
- REST token for authentication of MQTT token requests
- Tokens for MQTT authentication of devices/things/users

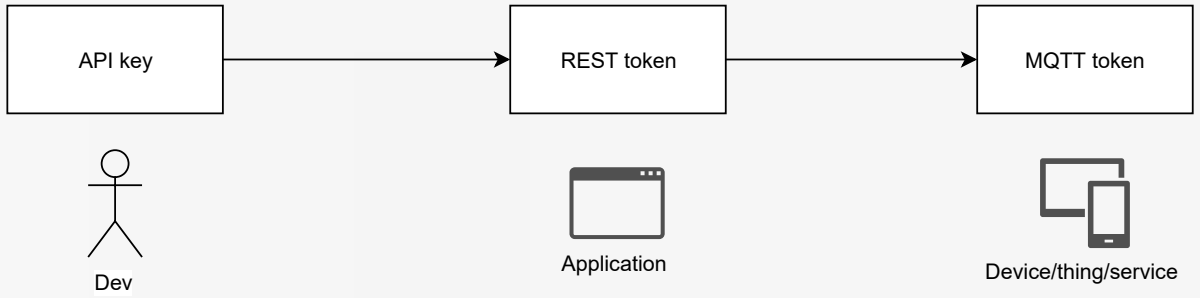
Authentication relations



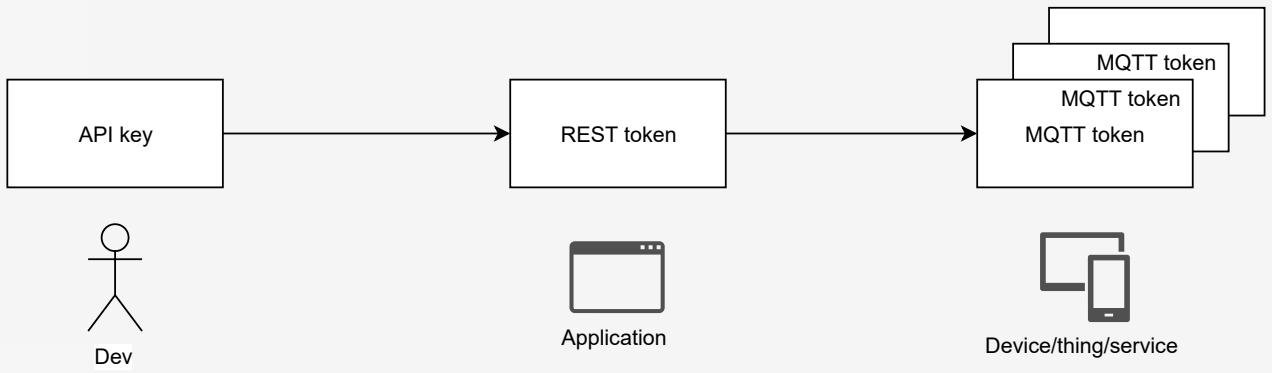
Authentication relations



Authentication relations



Authentication relations



Device management

- DSH does not manage devices

Device management

- DSH does not manage devices
- Up to the tenant to implement
- Provides the necessary building blocks

Access control

- Fine-grained on MQTT
  - Access Control Lists (ACLs)
  - read `/tt/topic/fixed/tenant/+/#`
  - write `/tt/topic/other/tenant/`
- Coarse-grained on Kafka
  - read/write on topic-level
  - implemented using custom tooling

Wrap-up

- API keys, REST token & MQTT tokens
- Kafka certificates
- ACLs on all streams/topics
- Kafka topics scheme

Practical part; MQTT

MQTT