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

Definition: Streaming Data Platform

Definition: Streaming Data Platform

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

A better definition: Streaming Data

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

–according to us

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

One source, low volume | many sources, high volume

Types of streaming data

Not all datastreams are created equal

One source, low volume | many sources, high volume

Single sensor | Stream processing

Types of streaming data

Not all datastreams are created equal

One source, low volume | many sources, high volume

Single sensor | Stream processing

MQTT | Kafka

Streaming data on DSH

Focus on two types of streams:

- MQTT (manneke pis)
- Kafka (waterval)

MQTT

- Lightweight messaging protocol
- Suitable for many simultaneous connections
- Widespread use in *Internet of Things*

Kafka

- Highly scalable in volume of data
- Messaging backbone for LinkedIn, Netflix, Yahoo, Twiter, Goldman Sachs

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

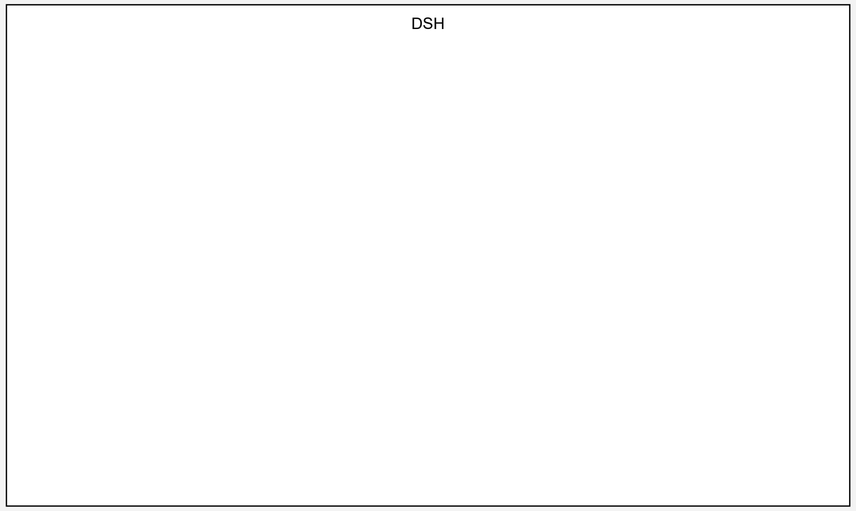
MQTT vs Kafka

- 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

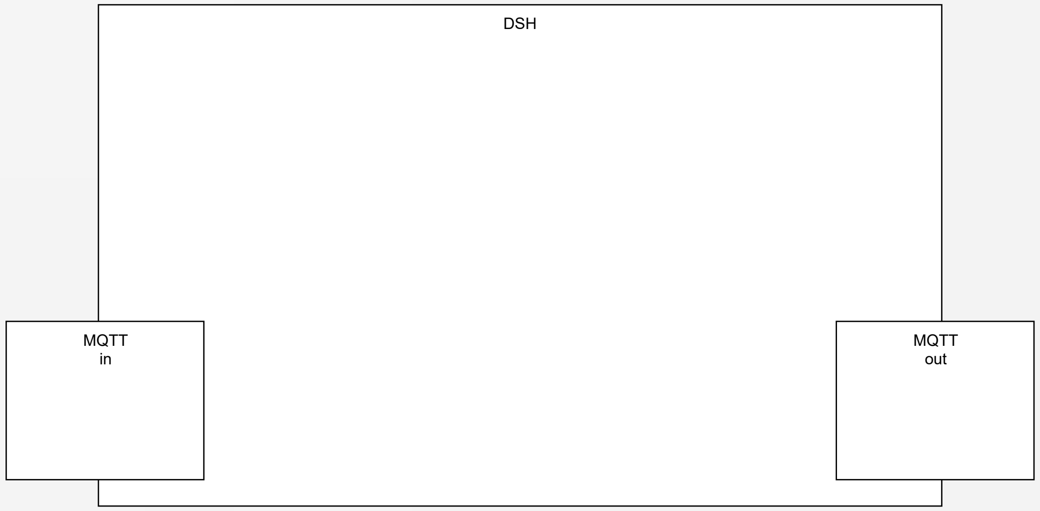
$$ \text{MQTT} \cdot \frac{sources}{sinks} \approx \text{Kafka} \cdot \frac{sources}{sinks} $$ $$

\frac{sources*{mqtt}}{sinks*{mqtt}}} \approx \text{Kafka} \cdot \frac{sources}{sinks} $$

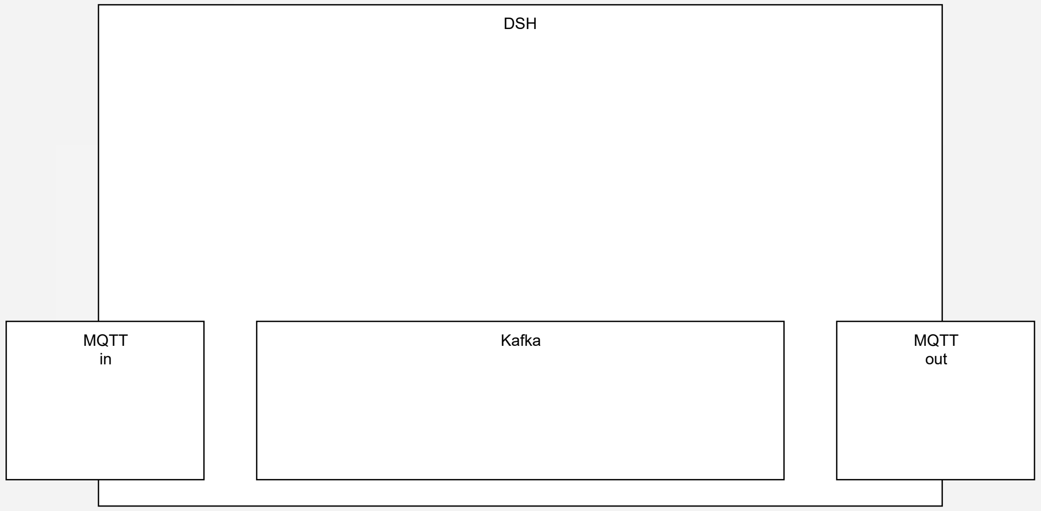
Overview



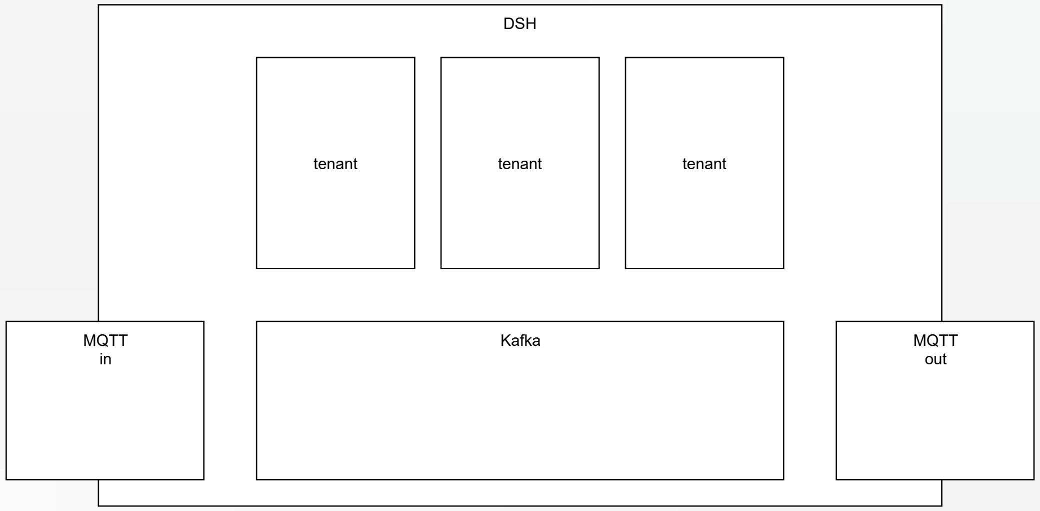
Overview



Overview



Overview



Bridge

- MQTT protocol adapter
  - acts *as if* it is MQTT broker
  - actually interfaces with Kafka
- like MQTT
  - allows wildcard subscriptions:

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

Bridge

$$\begin{align} \text{MQTT topic prefix} &= \text{Kafka cluster name} \\\
\text{MQTT topic infix} &= \text{Kafka topic name} \\\
\text{keys in Kafka} &= \text{MQTT topic suffix} \\\
\end{align}$$

Bridge

$$\begin{align} \text{MQTT topic prefix} &= \text{Kafka cluster name} \\\
\text{MQTT topic infix} &= \text{Kafka topic name} \\\
\text{keys in Kafka} &= \text{MQTT topic suffix} \\\
\end{align}$$

```
MQTT(topic="/tt/cam/id", data="...")
```

$=$

```
Kafka(cluster="tt", topic="stream.cam.*", key="id", data="...")
```

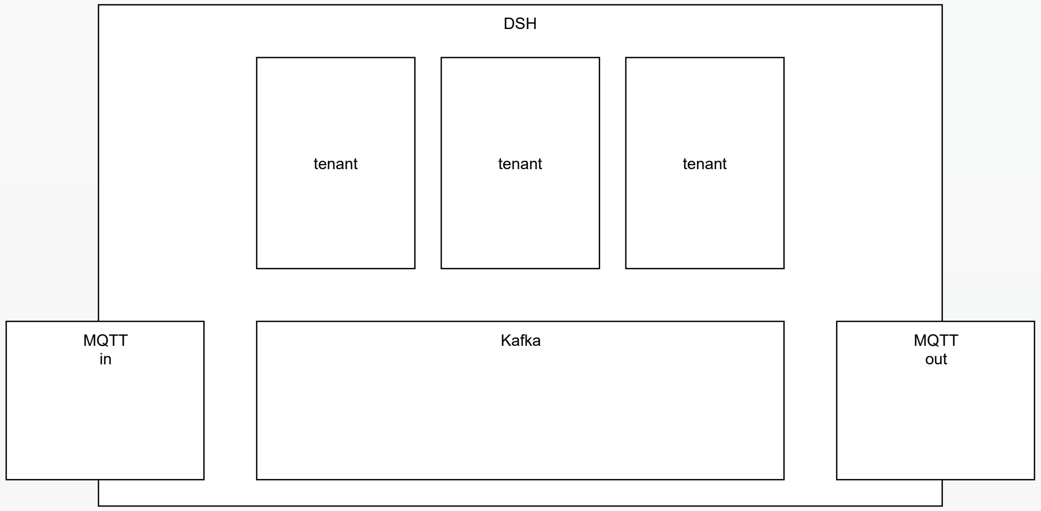
Rarely updated data sources

- Latest value store indexing service
- tracks keys in a stream
- distributed in-memory key-value store

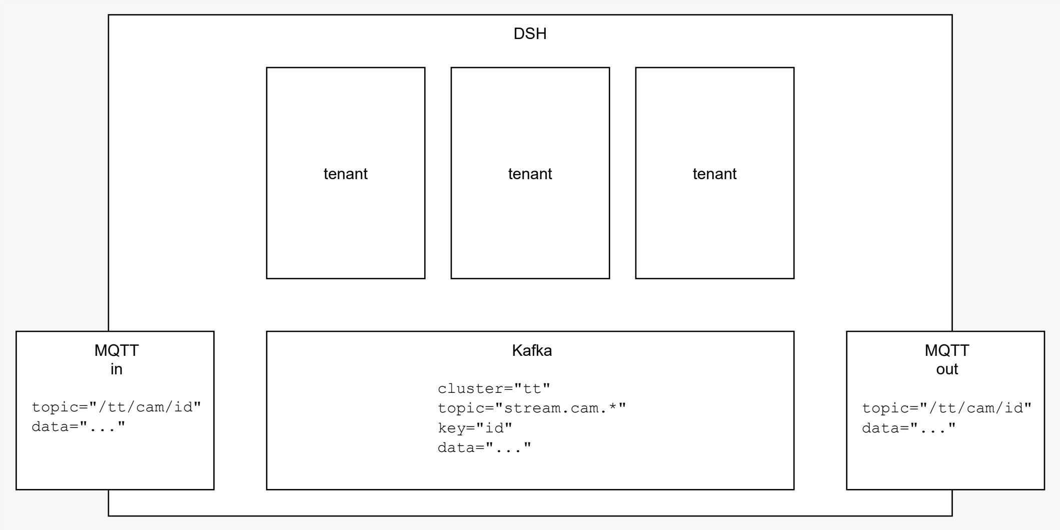
External data sources

- are not always MQTT
- do not always stream
- will require custom adapters
  - allow tenants to write their own

Overview



Overview


DSH
tenant
tenant
tenant
MQTT
in
topic="/tt/cam/id"
data="..."
Kafka
cluster="tt"
topic="stream.cam.*"
key="id"
data="..."
MQTT
out
topic="/tt/cam/id"
data="..."

Wrap-up

- MQTT for low volume, many sources/sinks
- Kafka for high volume, few sources/sinks
- bridge (protocol adaptor) to tie them together
- custom data source adapters for external data
- latest value store for instant syncing with rarely changing data source

Stream Processing Platform

Stream Processing Platform

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 edge where possible and necessary
- Close to the data (on the DSH) if you need a lot of data from multiple sources

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*.

Wrap-up

- DSH can process streams:
  - but is not always the right place to do it
  - and does not dictate how to process them

Security nightmare

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

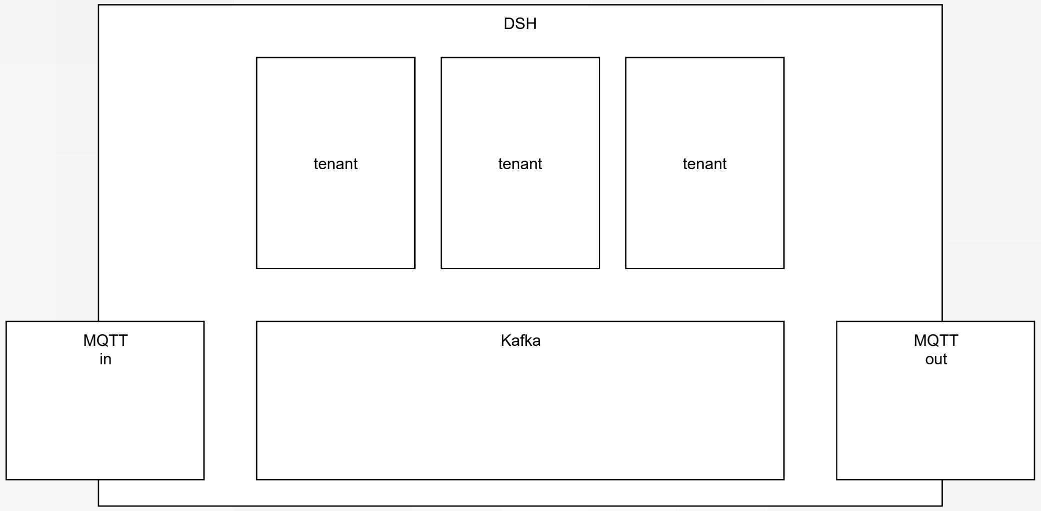
DC/OS

- Started with DC/OS as base platform
- Supported by most stream processing frameworks
- Tenants run docker containers on top

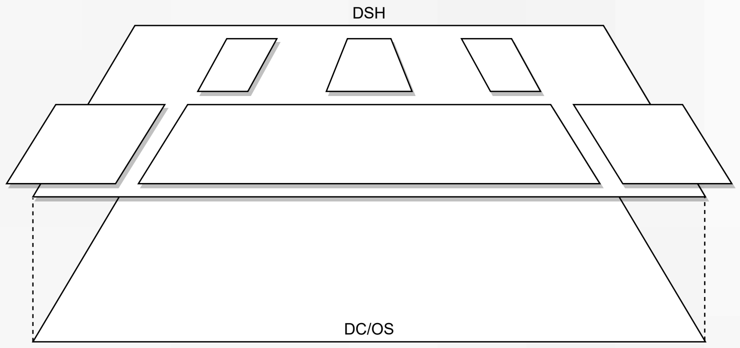
Securing

- Custom container manager to force correct use of Docker
- Custom resource manager to control resource requests
- Calico for network isolation

DC/OS



DC/OS



Wrap-up

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

Data Stream Platform

Data Stream Platform

a platform that holds many different *data streams*

Data Stream

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

Federal Standard 1037C

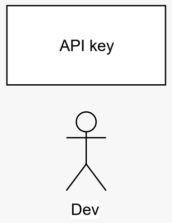
Many data streams

- Streams need organizing
- DSH topics $ \approx $ Kafka topics
- Need to control access to topics
  - Manage at topic level using custom tooling
  - Based on Access Control Lists (ACLs)

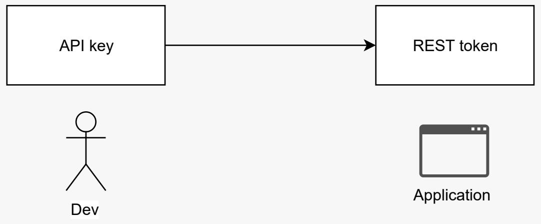
Authenticate

- 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

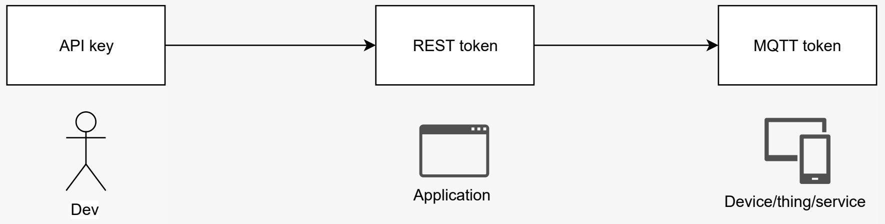
Authentication relations



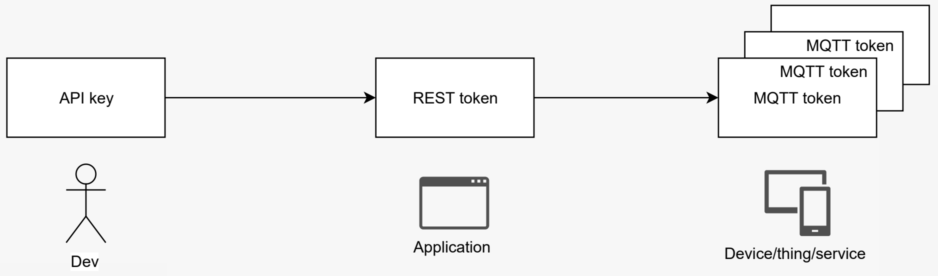
Authentication relations



Authentication relations



Authentication relations



Device management

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

Access control

- Fine-grained on MQTT
  - read `/tt/topic/fixed/tenant/+/#`
  - write `/tt/topic/other/tenant/`
- Coarse-grained on Kafka
  - read/write on topic-level

Kafka

Three Kafka stream-types

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

Wrap-up

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