

# SENSEMAKERS IOT PLATFORM

David Šálek  
22/05/2019



SURF SARA

# Website

- **WordPress**
- <https://www.sensemakersams.org>

# MQTT

- MQTT stands for Message Queuing Telemetry Transport
- extremely simple and lightweight messaging protocol designed for constrained devices and low-bandwidth, high-latency or unreliable networks
- became a standard for the Internet of Things
- **Mosquitto** is an open-source MQTT broker and serves as a backbone of the Sensemakers IoT platform. <https://mosquitto.org/>
- publish/subscribe to topics
- TLS encryption, user authentication, access control lists



# Automated data pipeline

- Messages sent to the **pipeline/<app\_id>/<dev\_id>** topic are automatically:
  - stored in InfluxDB database
  - appended in JSON format to files in a shared volume
  - sent to a serverless function (if in place) to enable event-driven actions

[https://openfaas.sensemakersams.org/async-function/<app\\_id>](https://openfaas.sensemakersams.org/async-function/<app_id>)

# Automated data pipeline – Message format

- Messages sent to the **pipeline/<app\_id>/<dev\_id>** topic have to comply with this JSON format.
- Time (in milliseconds) is optional.  
In case it is not provided, the server time upon message arrival will be used.
- Additional fields are allowed.
- Examples:

```
{"app_id": "test_project", "dev_id": "test_device",  
 "payload_fields": {"temperature": 42},  
 "time": 1557244616000}
```

```
{"app_id": "test_project", "dev_id": "test_device",  
 "payload_fields": {"temperature": 42, "foo": "bar"},  
 "foo": "bar"}
```

```
{  
  "type": "object",  
  "properties": {  
    "app_id": {  
      "type": "string"  
    },  
    "dev_id": {  
      "type": "string"  
    },  
    "payload_fields": {  
      "type": "object"  
    },  
    "time": {  
      "type": "integer"  
    }  
  "required": [  
    "app_id",  
    "dev_id",  
    "payload_fields"  
  ]  
}
```

# Automated data pipeline – Ingesting data

## ■ Publish to MQTT

- Username and password are required.
- port 9999 with encryption, port 9998 without encryption

```
mosquitto_pub -t pipeline/test_project/test_device -m \  
'{"app_id": "test_project", "dev_id": "test_device", "payload_fields": {"temperature": 42}}' \  
--cafile ca.crt \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9999 \  
-u test_project -P 1234
```

## ■ HTTP endpoint

- Password needs to be passed in a header, TLS encryption is in place.
- Works well with The Things Network HTTP integration.

```
curl -XPOST https://openfaas.sensemakersams.org/function/faas-mqtt --data \  
'{"app_id": "test_project", "dev_id": "test_device", "payload_fields": {"temperature": 42}}' \  
-H "X-Api-Key:1234"
```

# Publishing to MQTT topics with encryption

- Automated pipeline

```
mosquitto_pub -t pipeline/test_project/test_device -m \  
'{"app_id": "test_project", "dev_id": "test_device", "payload_fields": {"temperature": 42}}' \  
--cafile ca.crt \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9999 \  
-u test_project -P 1234
```

- Project-specific topic

```
mosquitto_pub -t test_project/test_device -m 42 --cafile ca.crt \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9999 -u test_project -P 1234
```

- Public topic

```
mosquitto_pub -t public -m 42 --cafile ca.crt \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9999 -u public -P 1234
```

# Publishing to MQTT topics without encryption

- Automated pipeline

```
mosquitto_pub -t pipeline/test_project/test_device -m \  
'{"app_id": "test_project", "dev_id": "test_device", "payload_fields": {"temperature": 42}}' \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9998 \  
-u test_project -P 1234
```

- Project-specific topic

```
mosquitto_pub -t test_project/test_device -m 42 \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9998 -u test_project -P 1234
```

- Public topic

```
mosquitto_pub -t public -m 42 \  
-h lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9998 -u public -P 1234
```

# Data storage

- **Shared volume**

- Every message is appended to a file specific to a device and a calendar day.
- The filename and path is defined in the following way:  
**app\_id/dev\_id-YYYY-mm-dd.json**

- **InfluxDB** <https://docs.influxdata.com/influxdb/>



- InfluxDB is an open-source time series database.
- Every message is written as a point to InfluxDB measurement **dev\_id** in database **add\_id**.
- Each point consists of the fields from **payload\_fields**.
- All numeric values are converted into floats.

# InfluxDB example queries

- Access InfluxDB from command line:

```
influx -host influxdb.sensemakersams.org -port 443 -ssl \
        -username test_project -password 1234
```

- Example queries:

```
SHOW DATABASES
USE test_project
SHOW MEASUREMENTS
SHOW SERIES

precision rfc3339
SELECT * FROM test_device
SELECT * FROM test_device WHERE time > now() - 1h
```

# Download data from InfluxDB

- Download data in the csv format:

```
influx -host influxdb.sensemakersams.org -port 443 -ssl \
--username test_project --password 1234 --database test_project \
--execute 'SELECT * FROM test_device' --format 'csv'

curl -XPOST \
"https://influxdb.sensemakersams.org/query?u=test_project&p=1234" \
--data-urlencode "db=test_project" \
--data-urlencode "q=SELECT * FROM test_device" | jq -r \
"(.results[0].series[0].columns), (.results[0].series[0].values[]) | @csv"
```

# Data storage

## ■ Object store

- Minio is an open-source object store compatible with Amazon S3  
<https://min.io/>
- used for periodic backups
- can be used for uploading larger files, e.g. images
- <https://minio.sensemakersams.org>
- command-line interface:

```
mc config host add sensemakers https://minio.sensemakersams.org/ \
sensemakers 1234
```

```
mc ls sensemakers
mc cp --recursive sensemakers/backup .
```



# Periodic backups and SQL queries in the object store

- All data files from the shared volume from the previous day are automatically copied to the object store every night.
- Every project gets its own folder in the bucket called **backup**.
- SQL queries of the data in the object store are possible.

```
mc sql sensemakers/backup/test_project/ \
--query "SELECT time,payload_fields FROM S3Object WHERE dev_id='test_device'"
```

# Metadata for the files in the object store

- Metadata for every data file in the object store is written to the bucket called **metadata**.

```
mc sql --recursive --query "SELECT * FROM S3Object" sensemakers/metadata/
```

```
{  
  "filename": "test_project/test_device-2019-05-12.json",  
  "date": "2019-05-21",  
  "app_id": "test_project",  
  "dev_id": "test_device",  
  "keys": [  
    "conductivity",  
    "light",  
    "moisture",  
    "temperature"  
  ],  
  "num_events": 3  
}
```

# Public access and data sharing

- Minio supports multiple users and setting policies at the bucket/folder/object level.
- <https://docs.min.io/docs/minio-multi-user-quickstart-guide.html>
- Public access to the files in a particular project folder can be defined by a policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::backup/test_project/*"  
            ],  
            "Sid": ""  
        },  
        {  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:s3:::backup"  
            ],  
            "Condition": {  
                "StringLike": {  
                    "s3:prefix": [  
                        "test_project/*",  
                        "test_project"  
                    ]  
                }  
            },  
            "Sid": ""  
        }  
    ]  
}
```

# Public access and data sharing

- Add the policy and define a new user with that policy.

```
mc admin policy add sensemakers public policy.json  
mc admin user add sensemakers public public1234 public
```

- The public user can access the shared data.

```
mc config host add sensemakers-public https://minio.sensemakersams.org/ \  
public public1234  
  
mc ls sensemakers-public/backup/test_project  
mc cp --recursive sensemakers-public/backup/test_project .  
  
mc sql sensemakers-public/backup/test_project/ \  
--query "SELECT time,payload_fields FROM S3Object WHERE dev_id='test_device'"
```

# Dashboards

- **Grafana** is available for visualizing data from InfluxDB and alerting.
- <http://docs.grafana.org/>
- <https://grafana.sensemakersams.org>



# Serverless functions

- OpenFaaS framework is available to deploy serverless functions.

- <https://www.openfaas.com/>



OPEN FAAS

- <https://openfaas.sensemakersams.org>

- Functions can be deployed by a platform administrator using **faas-cli**.
- The HTTP endpoint for ingesting data to the platform is an OpenFaaS function that publishes the received message over MQTT to the pipeline topic for the corresponding project.

# The Things Network HTTP integration

- In case **app\_id** in TTN does not correspond to the one in the Sensemakers IoT platform, it can be overwritten in a URL query.
- In case the MQTT user is not identical to **app\_id**, it can be specified in URL query variable **mqtt\_user**.
- The MQTT password is passed as a custom **X-Api-Key** header.

<b>URL</b> The URL of the endpoint	<input type="text" value="https://openfaas.sensemakersams.org/function/faas-mqtt?app_id=test_project"/>
<b>Method</b> The HTTP method to use	<input type="text" value="POST"/>
<b>Authorization</b> The value of the Authorization header	<input type="text"/>
<b>Custom Header Name</b> An optional custom HTTP header that you would like to add to the request	<input type="text" value="X-Api-Key"/>
<b>Custom Header Value</b> The value of the custom Header	<input type="text" value="1234"/>

# Jupyter notebooks

- Jupyter notebooks are available for data analytics purposes.
- <https://jupyter.org/hub>
- <https://jupyter.sensemakersams.org>
- Every project gets its own Jupyter server with private storage space and access to the shared storage under /home/shared



# Linux machine

- Linux machine running **Ubuntu** is available.
- Shared volume is mounted under **/data**
- SSH access

```
ssh root@lb.cluster-meetup-demo.aws.surfsaralabs.nl -p 9997
```

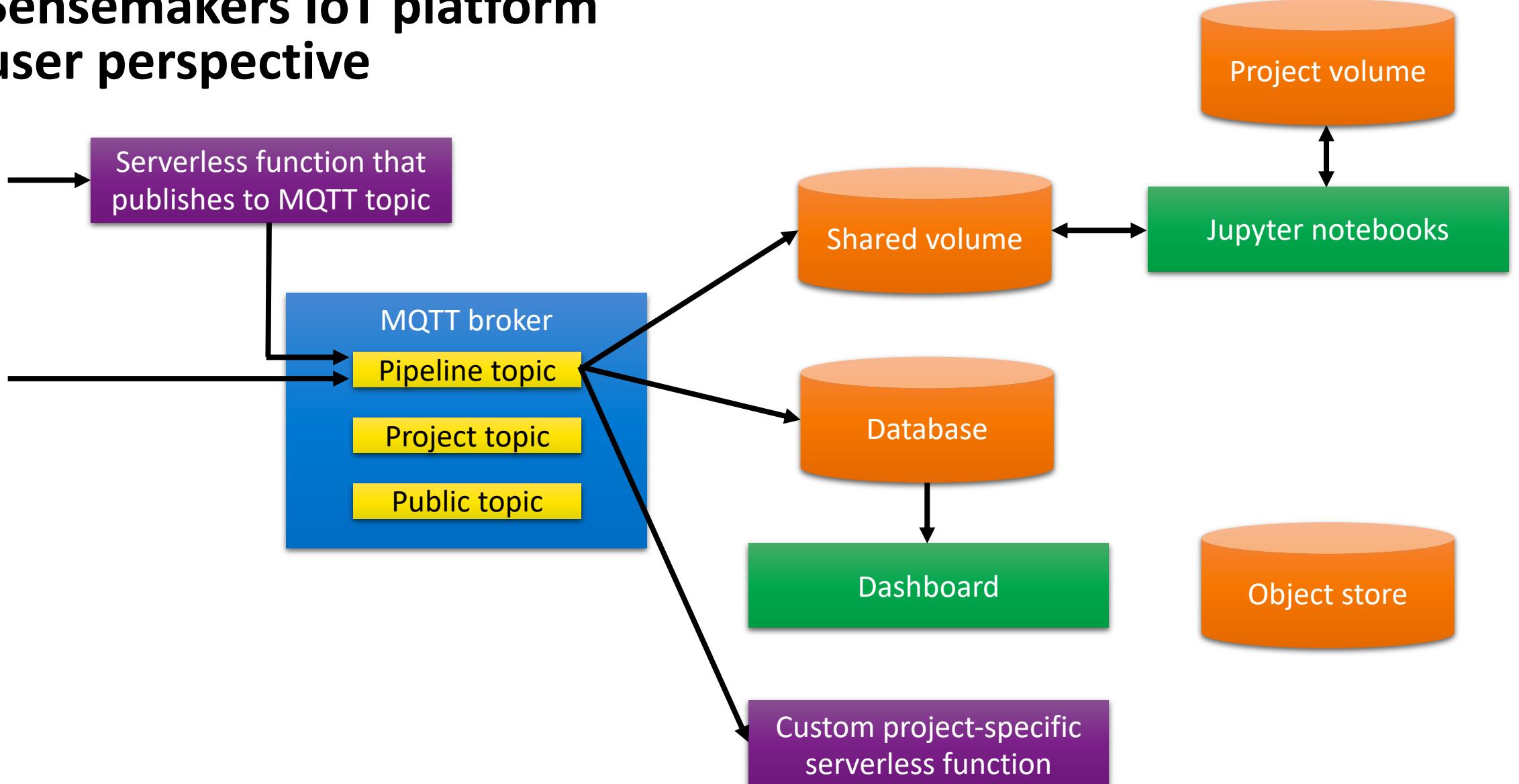
# Event-driven actions/talking back to devices

- Event-driven actions (i.e. executing code triggered by an incoming message) can be implemented in the following ways:
  - Run code that subscribes to an MQTT topic on your own machine.
  - Run code that subscribes to an MQTT topic on the Linux machine in the Sensemakers IoT platform.
  - Deploy a serverless function.

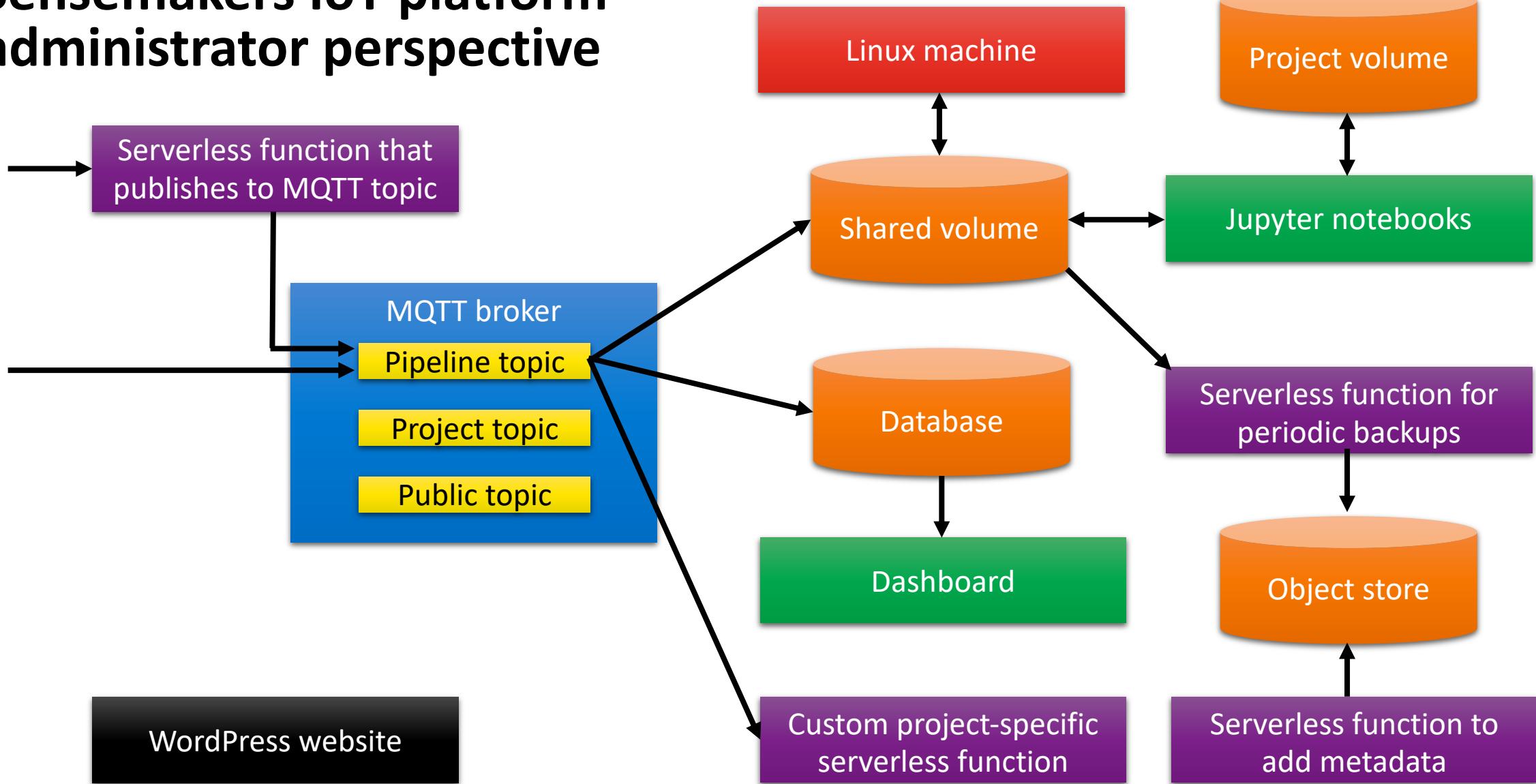
[https://openfaas.sensemakersams.org/async-function/<app\\_id>](https://openfaas.sensemakersams.org/async-function/<app_id>)

- The MQTT broker can be used to send messages back to devices. (provided the devices are capable of subscribing to MQTT topics)

# Sensemakers IoT platform user perspective



# Sensemakers IoT platform administrator perspective



# SENSEMAKERS IOT PLATFORM

 David Šálek

 E-mail: [david.salek@surfsara.nl](mailto:david.salek@surfsara.nl)

 <https://www.linkedin.com/in/davidsalek/>

Driving innovation together

 SURF SARA

**Driving innovation together**

