

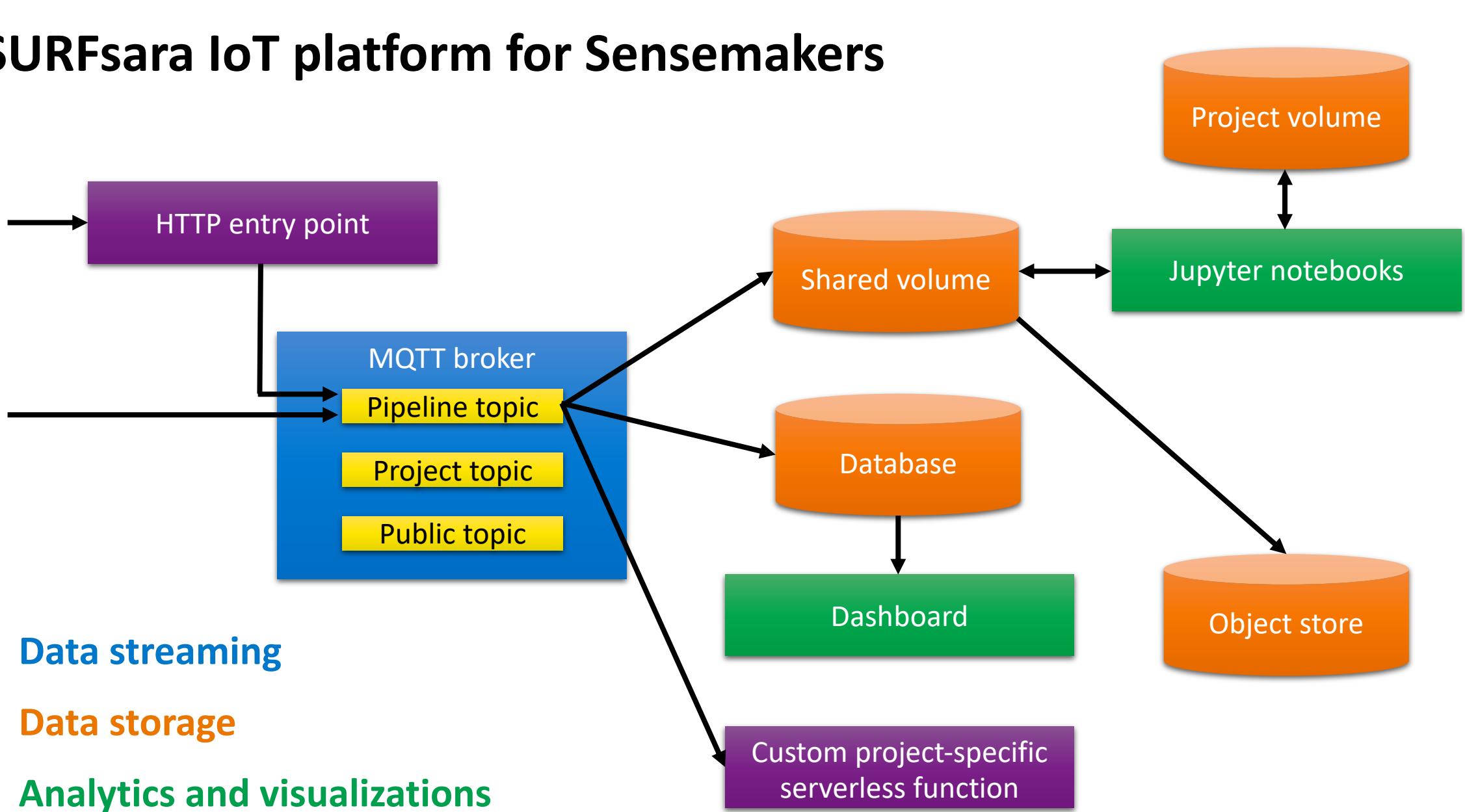
# SENSEMAKERS IOT PLATFORM

David Šálek  
08/10/2019



SURF SARA

# SURFsara IoT platform for Sensemakers



- **Data streaming**
- **Data storage**
- **Analytics and visualizations**
- **Event-driven actions**

# 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
- 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" },
>tag_fields": {"foo": "bar"}, "foo": "bar"}
```

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

# Automated data pipeline – Ingesting data

## ■ Publish to MQTT

- Username and password are required.

```
mosquitto_pub -t pipeline/test_project/test_device -m \  
'{"app_id": "test_project", "dev_id": "test_device", "payload_fields": {"temperature": 42}}' \  
-h mqtt.sensemakersams.org -p 9998 -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

- Automated pipeline

```
mosquitto_pub -t pipeline/test_project/test_device -m \  
'{"app_id": "test_project", "dev_id": "test_device", "payload_fields": {"temperature": 42}}' \  
-h mqtt.sensemakersams.org -p 9998 -u test_project -P 1234
```

- Project-specific topic

```
mosquitto_pub -t test_project/test_device -m 42 \  
-h mqtt.sensemakersams.org -p 9998 -u test_project -P 1234
```

- Public topic

```
mosquitto_pub -t public -m 42 \  
-h mqtt.sensemakersams.org -p 9998 -u public -P 1234
```

# Data storage

- **Shared volume**

- Every message is appended to a file specific to a device and a day of message arrival (not the measurement time).
- 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**.
- **dev\_id** and **tag\_fields** are used as tags.
- All numeric values are converted into floats.

# InfluxDB example queries

- Access InfluxDB from command line:

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

curl -XPOST \
"https://influxdb.sensemakersams.org/query?u=public&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"
```

# Using tags to select time series in InfluxDB

- Tags can be used to select particular time series from measurements.
- For example, human readable name for each device can be used as a tag.

```
SHOW TAG VALUES WITH KEY = "foo"  
SELECT * FROM /.*/ WHERE "foo"::tag =~ /bar/'
```

```
influx -host influxdb.sensemakersams.org -port 443 -ssl \  
-username public -password 1234 -database test_project \  
-execute 'SELECT * FROM /.*/ WHERE "foo"::tag =~ /bar/' -format '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
- web interface: <https://minio.sensemakersams.org>
- command-line interface:

```
mc config host add sensemakers https://minio.sensemakersams.org/ \
public 1234 --api s3v4
```

```
mc ls sensemakers
mc cp --recursive sensemakers/data/test_project .
```



# 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 **data**.
- SQL queries of the data in the object store are possible.

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

- Metadata give a summary of available sensor readings in the data file.

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

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

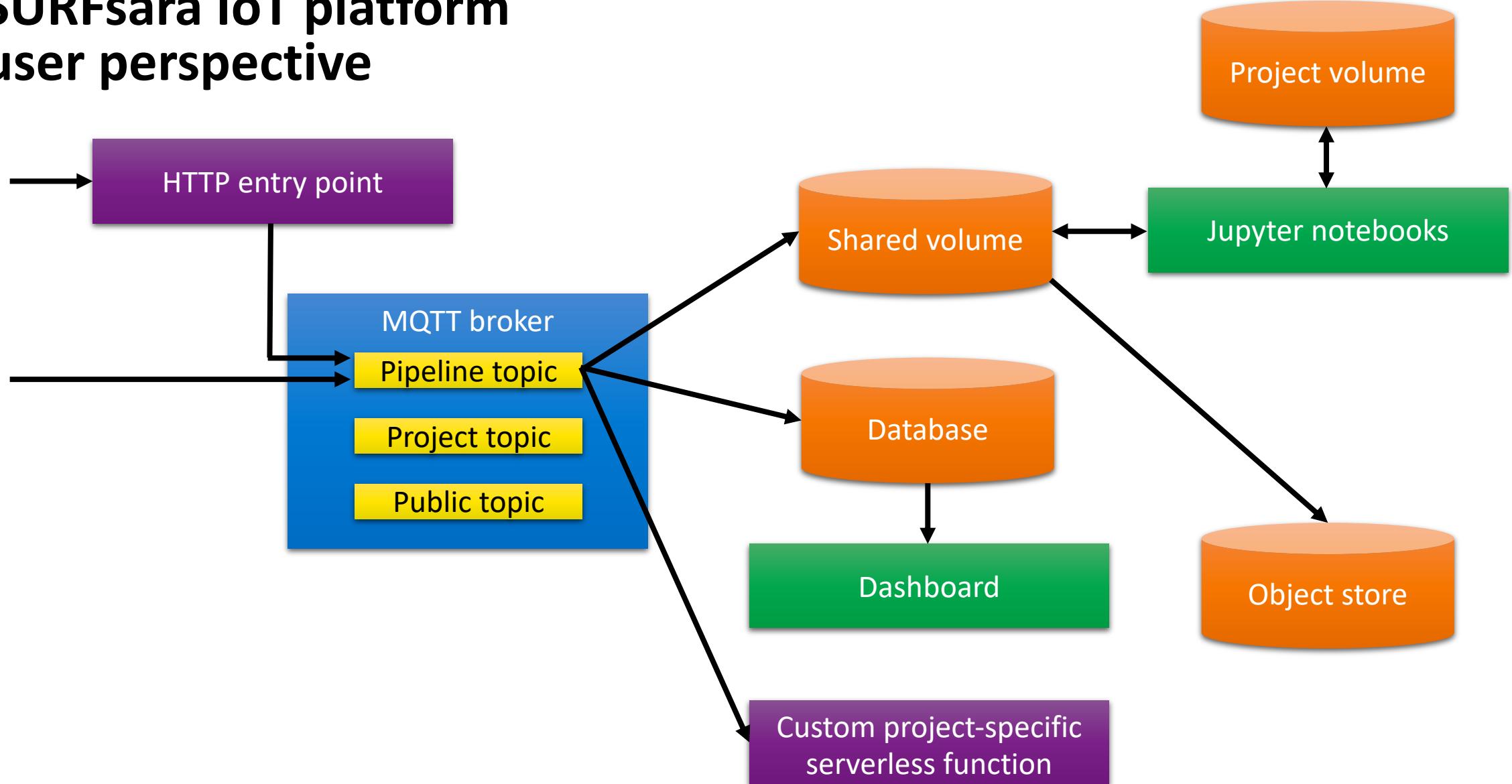
[https://openfaas.sensemakersams.org/async-function/<app\\_id>](https://openfaas.sensemakersams.org/async-function/<app_id>)

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

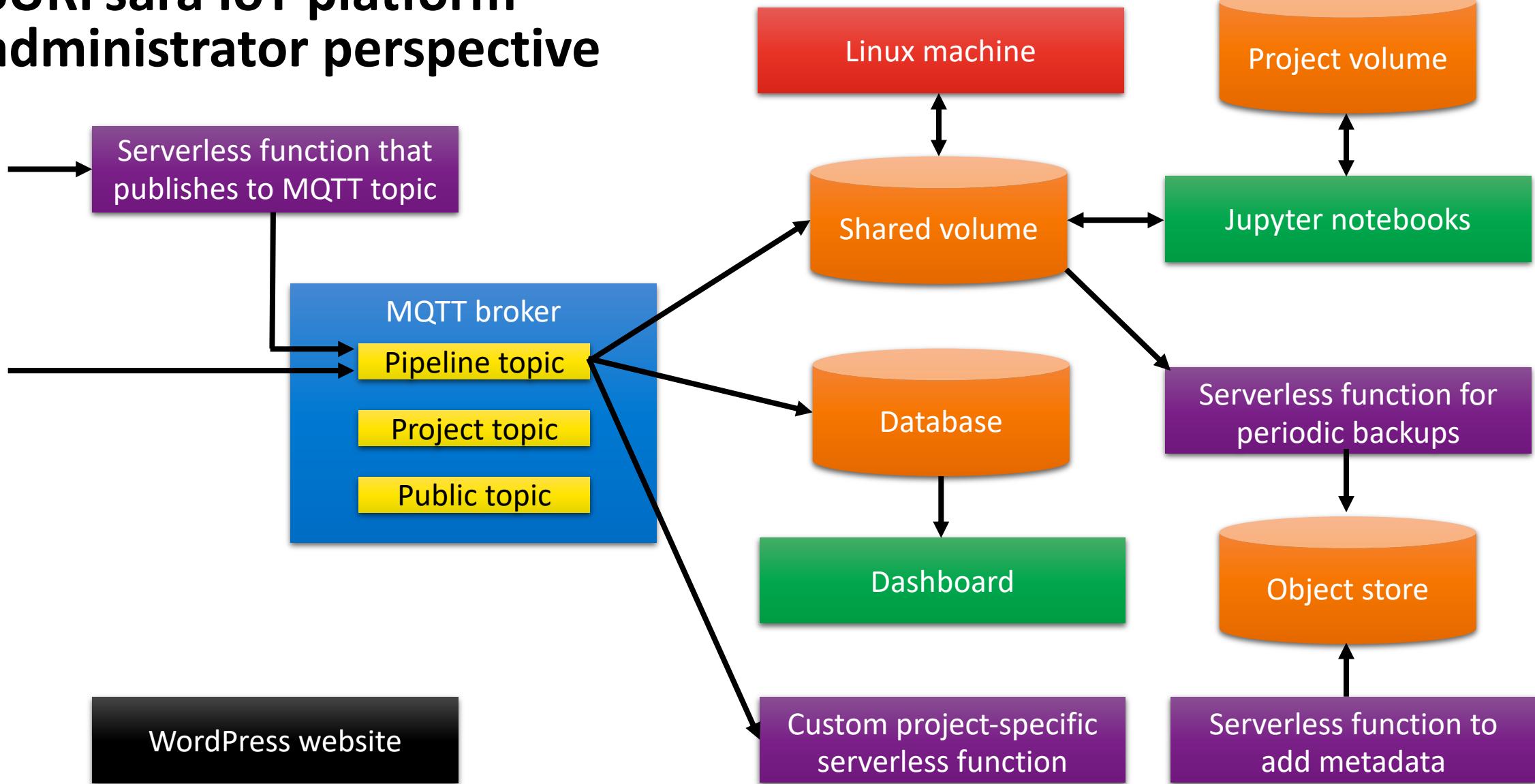
# Using the SURFsara IoT platform for Sensemakers

- The Sensemakers IoT platform is designed to host **multiple projects**.  
(credentials/resources are given on a project-basis)
- One project is dedicated to **experimentation** → available for everyone  
(ask the credentials)
- There is a dedicated **github repository** for:
  - platform overview and technical **documentation**
  - **example code**
  - dashboard backups
  - <https://github.com/sensemakersamsterdam/sensemakers-iot-platform>

# SURFsara IoT platform user perspective



# SURFsara 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**



# Useful links

- Website: <https://www.sensemakersams.org>
- MQTT broker: [mqtt.sensemakersams.org](https://mqtt.sensemakersams.org)
- Minio object store: <https://minio.sensemakersams.org>
- Grafana dashboards: <https://grafana.sensemakersams.org>
- Jupyter notebooks: <https://jupyter.sensemakersams.org>
- OpenFaaS serverless functions: <https://openfaas.sensemakersams.org>
- Github repository: <https://github.com/sensemakersamsterdam/sensemakers-iot-platform>