Microservices architecture for collecting data related to electric consumption

Cloud Computing Technologies project

Federico Garegnani

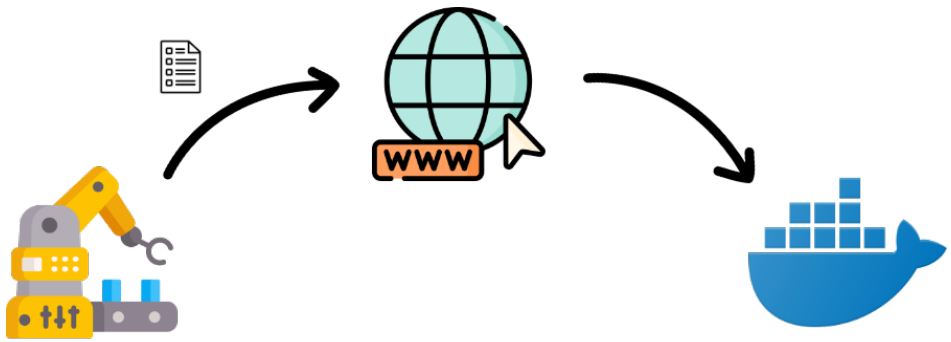
Università degli Studi di Milano

September 16, 2024

Table of contents

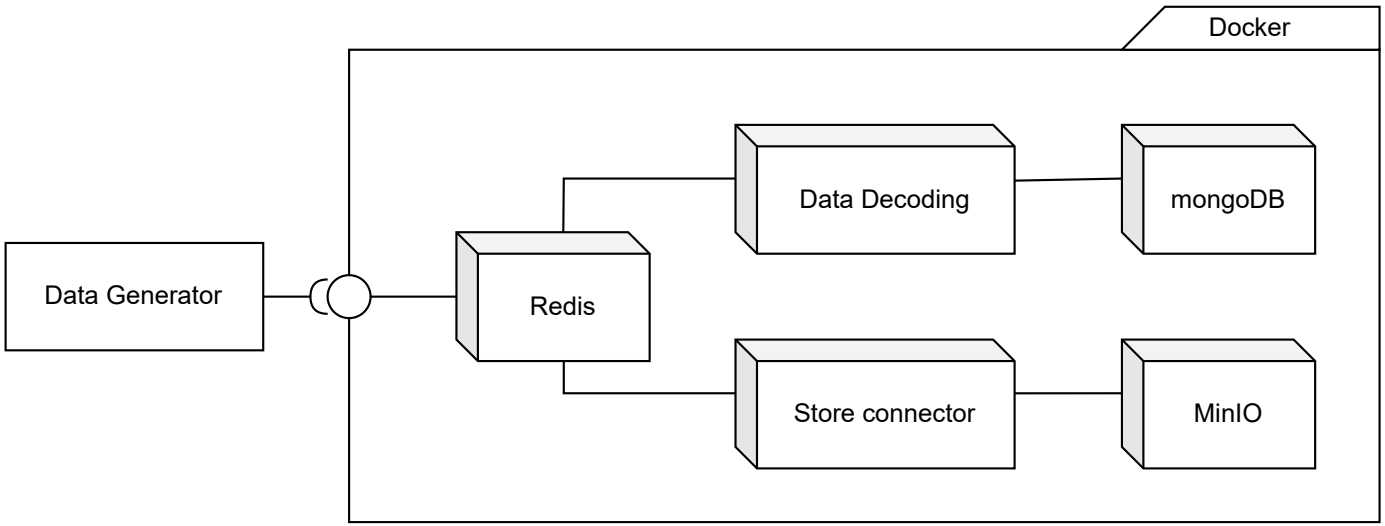
Application domain and working principle

- Industrial environment.
- An energy meter sends data regarding the electric consumption of a machinery through the internet.
- Data are received, analysed and collected by the cloud application.



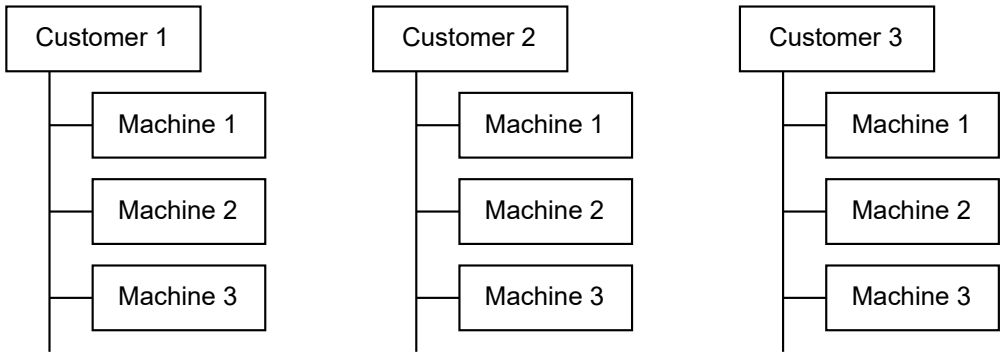
General structure of the system

- 5 microservices.
- Every microservice carry on a specific task.
- Only one entry point.



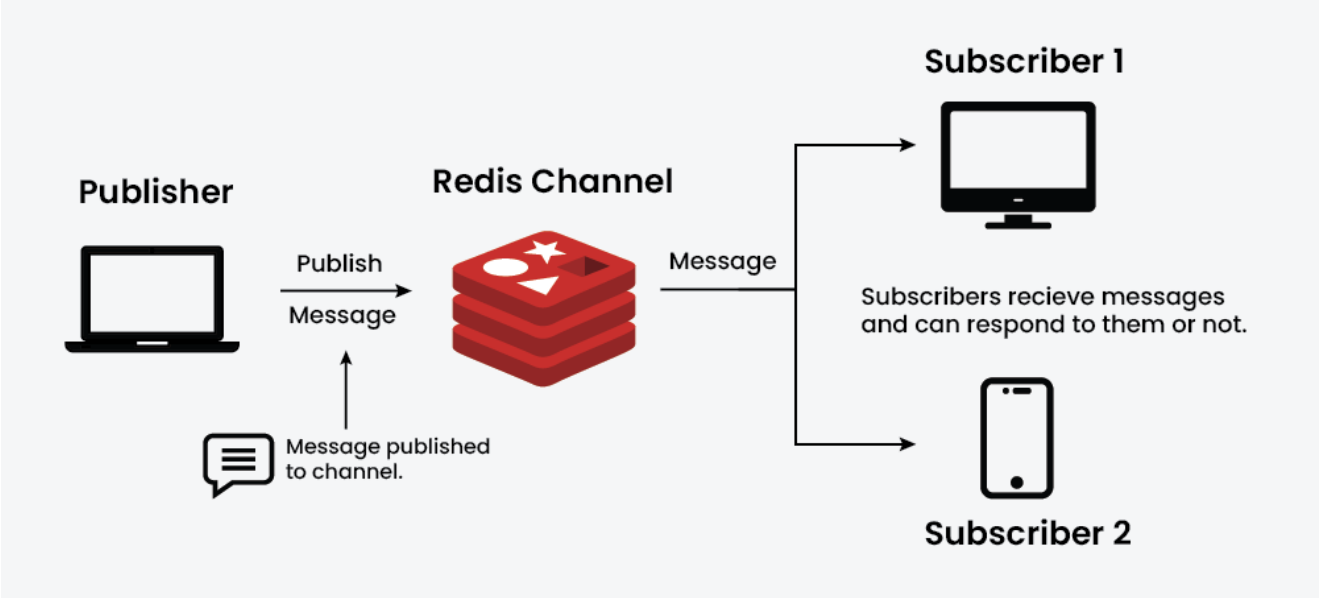
Data Generator

- It simulates data produced by each power meter.
- It publish data on a specific Redis channel.
  E.g. Data from `machine1` of `customer1` are published on `data.customer1.machine1`



Redis

- It is a key-value database that offers the functionalities of a message broker.
- It receives data from all the meters and forward them to two microservices: *DataDecoding* and *Storeroom*.
- It is the entry point of the entire system.



Data decoding

- It subscribes to Redis channel '`data.*`'.
- It receives a hexadecimal string that is split to obtain data, hour and numeric value of the measure.
  E.g. `'2024-09-07T19:53:19.561339'` $\rightarrow$ `'1268b41553b7'`
  Numeric value 68 `DEC` $\rightarrow$ 44 `HEX`.
  Then the two strings are concatenated: `'1268b41553b744'`.
- It prepares a JSON document and writes it into the database.

```
{
  "customer": "customer1",
  "machine": "machine1",
  "date": "2024-07-18T18:05:05Z",
  "EE": 32
}
```

Figure: Example of document

MongoDB Database

- It stores all measurements.
- It provides all data if a user wants to download or visualize them.

Store connector

- It subscribes to Redis channel '`data.*`'.
- It encapsulates raw data (hexadecimal strings) into text files.
- A unique filename is set to every file, given following this scheme `f"{self.machine_name}_{current_time}_{unique_id}"`, where `unique_id = uuid.uuid4()`.
- Files are sent for permanent storage.

MinIO Object Store

- It is an object store platform, organized in buckets and fully compatible with Amazon S3.
- It takes care of the permanent storage of raw data.
- One bucket for each customer.