IOT#BC-TRACE

Technologies for Big Data Management - A.Y. 2022-23

Project Description

The objective of the project is to define a prototype tool to store traces of IoT data on a public platform like Ethereum.
In order to guarantee scalability and performance, the tool must also consider an additional message broker component based on Apache Kafka https://kafka.apache.org/.
Students can use the IoT simulator:
https://github.com/massimocallisto/iot-simulator

Project demonstration should cover the following topics:

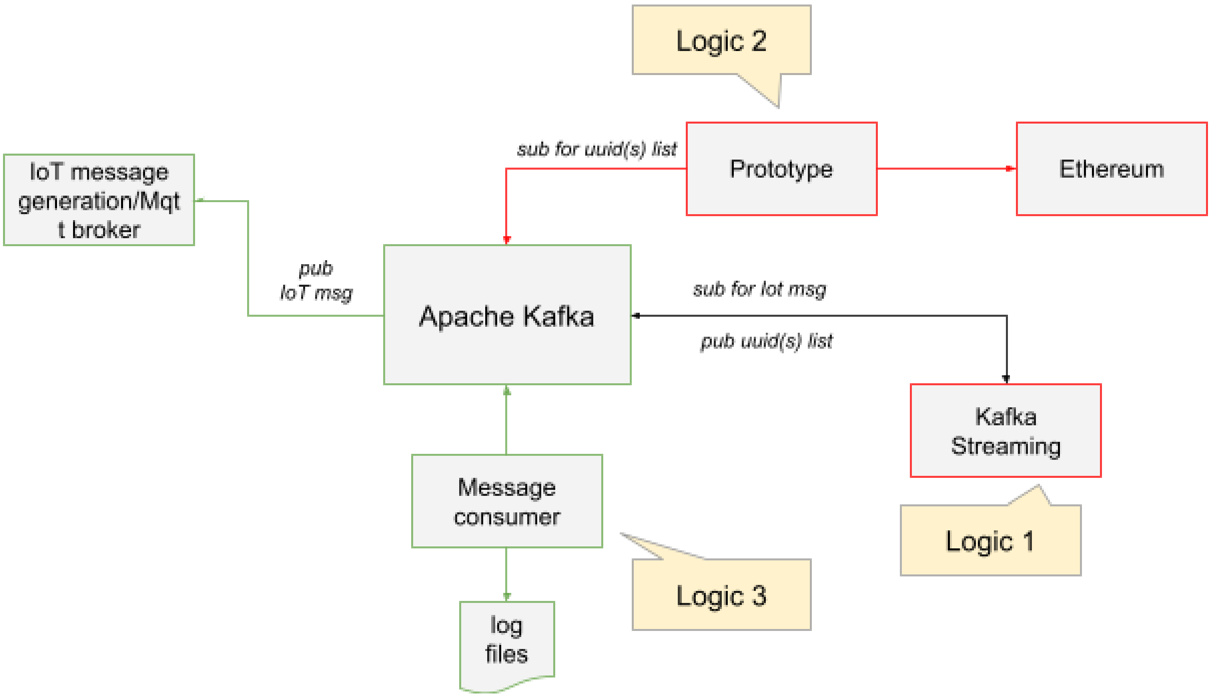
- Publish IoT messages from the simulator to a Kafka topic `MQTT`
- [Logic 1] Write a Kafka Stream[1] component that subscribes on the topic `MQTT` and:
    - Implements a window function[2] (eg. 5 minutes) to group all the messages that belong to the same time window
    - Produce a new message on topic `IOTRACE` with the following JSON format:

      ```
      {
        "tz" : "2020-04-06T10:50:39.137+02:00", // ISODATE of the kafka message
        "uuids" : ["UUID1","UUID2", ...]
      {
      ```

      `The UUID values are the ones inside the IoT messages. Those values are unique by definition among all the IoT messages`[3]
- [Logic 2] Write a prototype (e.g. in node.js) that subscribes to `IOTRACE` topic and persist the message on Ethereum
- [Logic 3] Configure a Console consumer that dumps the IoT message from `MQTT` topic to a file[4]. This file will be used to retrieve the original IoT messages from the UUIDs stored in Ethereum

The high level architecture can be defined as follows:

[1] https://kafka.apache.org/33/documentation/streams/
[2] https://developer.confluent.io/learn-kafka/kafka-streams/windowing/
[3] https://drive.google.com/file/d/1_LtJxNHLeSvt3EtRIYQpc8SpDWsYbDHk/view?usp=sharing
[4] https://www.edureka.co/community/9419/writing-the-kafka-consumer-output-to-a-file



Note that the green parts of the architecture will be discussed in the Lab course.

Expected results are:

- Data immutability and proof evidence over the data
- Performance analysis when changing the rate of IoT messages produced
- Implementation of the prototypes (e.g. node.js)