

# Spark Reference Architecture

## IoT platforms

Jan Macháček

## Abstract—TODO

## I. INTRODUCTION

TODO

## II. REFERENCE IMPLEMENTATION

This year appears to be the perfect time to start building internet-of-things platforms: the availability of sensors has never been greater, while their cost has never been lower. Together with wide availability of smart hubs—appliances that provide screen-less access to the nearby devices—and our prediction that these devices will provide local platform for the IoT devices, there will be growing need to build larger cloud-based applications that ingest the data, contextual information, and commands from the applications on these smart hubs. Such cloud-based application needs to be able to handle significant volume of incoming data in a reliable and responsive manner. The responsiveness and reliability expectations for the applications for the smart hubs far exceed those of ordinary web or mobile applications: the users are even less likely to tolerate delays or failures. The Lightbend stack, with help from Kafka, Apache Spark and Mesos are the perfect choices to implement the cloud-based portion of such a system. The application you see implements the crucial elements of an IoT platform: the ingestion service receives the data from the hubs and sensors. It then reconstructs the messages, checks their integrity and authenticity, decodes the payloads, and passes the data on to Kafka to deliver the payloads to the subscribed microservices. The back-pressure reporting is used to ensure that the infrastructure can flex up or down to deal with the load. Kafka (with Zookeeper) allows us to reason about the order of messages, the delivery guarantees; these then serve as constraints on the implementation of the microservices. Using this architecture, it is possible to clearly separate the ingestion / storage microservices from the streaming and batch analytics services. ?? illustrates the main components and the sizes of the data flows; it leaves out the infrastructural concerns.

### III. INFRASTRUCTURE

To run this application, the infrastructure needs to be able to make the most of the available computing resources (CPU, GPU, RAM, storage, etc.) and to provide sufficient service discovery functionality; it must also provide sufficient monitoring and log management. The monitoring components need to be able to trigger scale up or down as necessary, the triggers in this type of architecture are typically back-pressure indicators. The main infrastructure components are shown in ??.

Fig. 1. Architecture

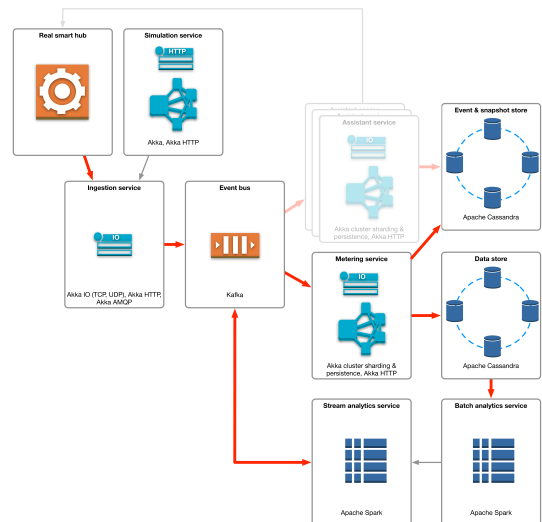
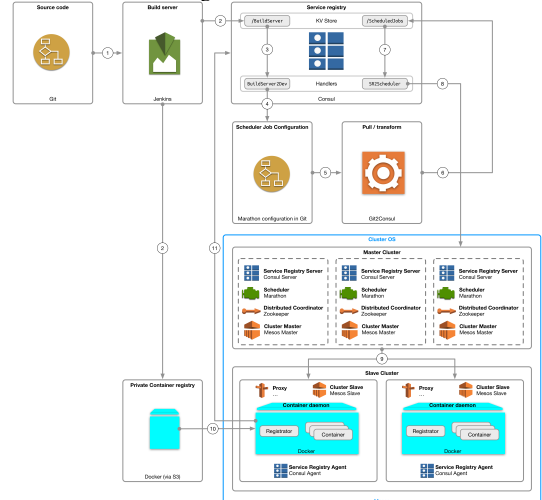


Fig. 2. Infrastructure



#### IV. SUMMARY

## REFERENCES