Aalto University
School of Science

CS-E4640 Big Data Platforms

Taste of Big Data Platforms

Rohit Raj (rohit.raj@aalto.fi)
Master student of SECCLO

Purpose and Content

- **My experiences about the design of a real world big-data platform and how it changed my perception.**

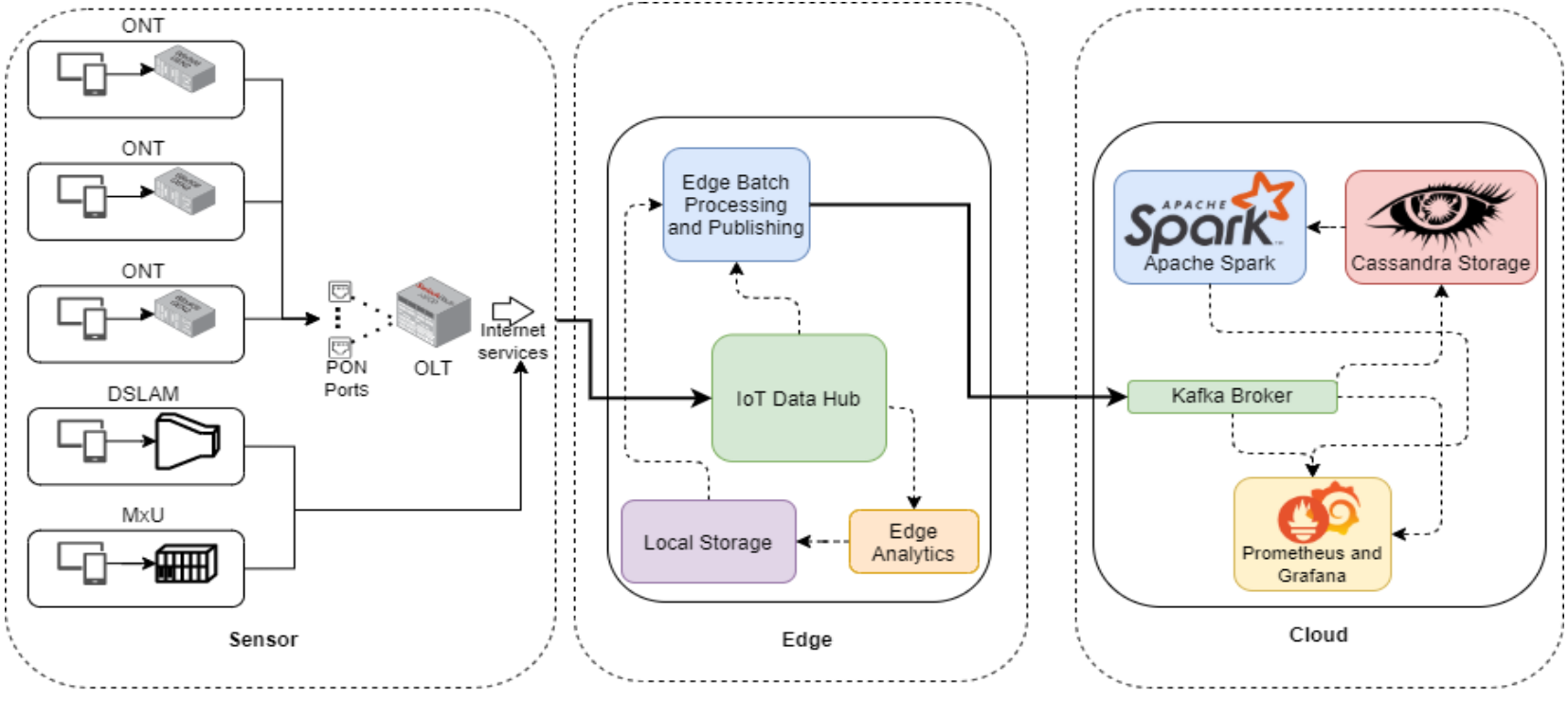
- About this presentation:
  - Goals with which I started
  - My first design choices
  - Changes in the design and perception

Goal of GPON Big Data platform

- My generic goals were:
  - Do stuff at a large scale: data ingestion, management and analytics

- My goals for the GPON monitoring infrastructure were:
  - Reliably monitoring a large-scale GPON infrastructure
  - Build a resilient and elastic big data infrastructure for network operations
  - With fast analytics (results) provided to GPON maintainers

A Real-Life Example: GPON Monitoring Infrastructure



https://version.aalto.fi/gitlab/bigdataplatforms/cs-e4640/-/tree/master/tutorials/netopanalytics

Initial design aims

- **Everyone has a different role in developing and operating a BDP**
  - Developer, System Engineer, Data Scientist, Manager etc
- **My focus while creating the design:**
  - Major aims as a developer
    - Focus on faster ingestion
    - Fault can occur at a variable rate so be prepared for that
    - Providing faster aggregation queries: let the maintainers know ASAP
    - Use most popular products like MongoDB and Mosquitto MQTT broker
  - Minor aim as a system engineer:
    - *Deploy and scale the monitoring infrastructure*

Initial Design

- **So, I created an initial design for the monitoring infrastructure**
  - It was similar to the figure on slide 4 except for the database
- I initially went with MongoDB as database choice
  - Easy to use NoSQL document store DB with huge community support on internet
  - "Looked good enough" for this use-case

BUT ...

Design changes while development

- **I did some design changes while development**

- **I changed the choice of database from:**
  - MongoDB -> Cassandra
    - *Columnar oriented DBs are faster for analytical queries*
    - *Availability & Partition Tolerance (AP) thorough Consistent Hashing is important for high availability**
    - *Support for eventual consistency*

*** Read more at: https://www.ibm.com/cloud/learn/cap-theorem**

Design changes while deployment

- **One of the design changes while 1st test deployment**

- **I changed the choice of edge sub-system broker**
  - Mosquitto -> VerneMQ
    - *Easier to scale horizontally*
    - *Inbuild support for JMX and Prometheus exports*

Changing perceptions about BDP

- **I felt it's difficult predict everything in advance**
  - I had to re-evaluate choices as perceptions changed
  - Example: going from MongoDB to Cassandra
- **I had to develop the mindset to select based on use-case**
  - *Example: MongoDB would have been better if we were doing long-term warehousing due to better consistency support*
- **I feel it is we should build the ecosystem around the platform**
  - *Thinking about the systems' integration rather than the technology*
  - *Ex: Better monitoring through just a change in broker flavour*

Changing Perceptions – II

- **For me as a developer, the new technologies sounded daunting**
  - *For example, during my first experience on Kafka, the official documentation was unnecessarily complex*
- **I felt that databases can be hard depending on use case**
  - *Getting Cassandra up and running was difficult*
- **However, I think that Apache Spark turned out much easier to use**
  - *I just had to develop an idea about RDD and dataframe APIs*
  - *Spark maybe difficult for other products or use-cases but not this one* ☺

Sum up

- **Different roles may have different approach**
  - *Define you own role (developer and system person in my case)*
- **Ever evolving platform**
  - *We might need to change our approach with time (broker change)*
  - *Might be tradeoffs (MongoDB vs Cassandra)*
- **A big data platform is more than bunch of technologies**
  - *Rather a complete ecosystem*
  - *And it is important to have a very deep understanding of the solution*
- **First impression/taste will be vastly different and will change**
  - *My perception changed from technology to system*
  - *I felt that learning technology is easier but more important is design mindset*
    - For example: Like when to use MongoDB and when to use Cassandra

Thank You

Questions?