Kappa Architecture
Our Experience

Who am I

- CDO ASPgems
- Former President of Hispalinux (Spanish LUG)
- Author “La Pastilla Roja” first spanish book about Free Software.

Menu

- A little context about Kappa Architecture
- What's Kappa Architecture
- What is not Kappa Architecture
- How we implement it
- Real use cases with KA

A little context

- July 2, 2014 Jay Kreps coined the term Kappa Architecture in an article for O'reilly Radar

Questioning the Lambda Architecture

The Lambda Architecture has its merits, but alternatives are worth exploring.

by Jay Kreps | @jaykreps | +Jay Kreps | Comments: 19 | July 2, 2014

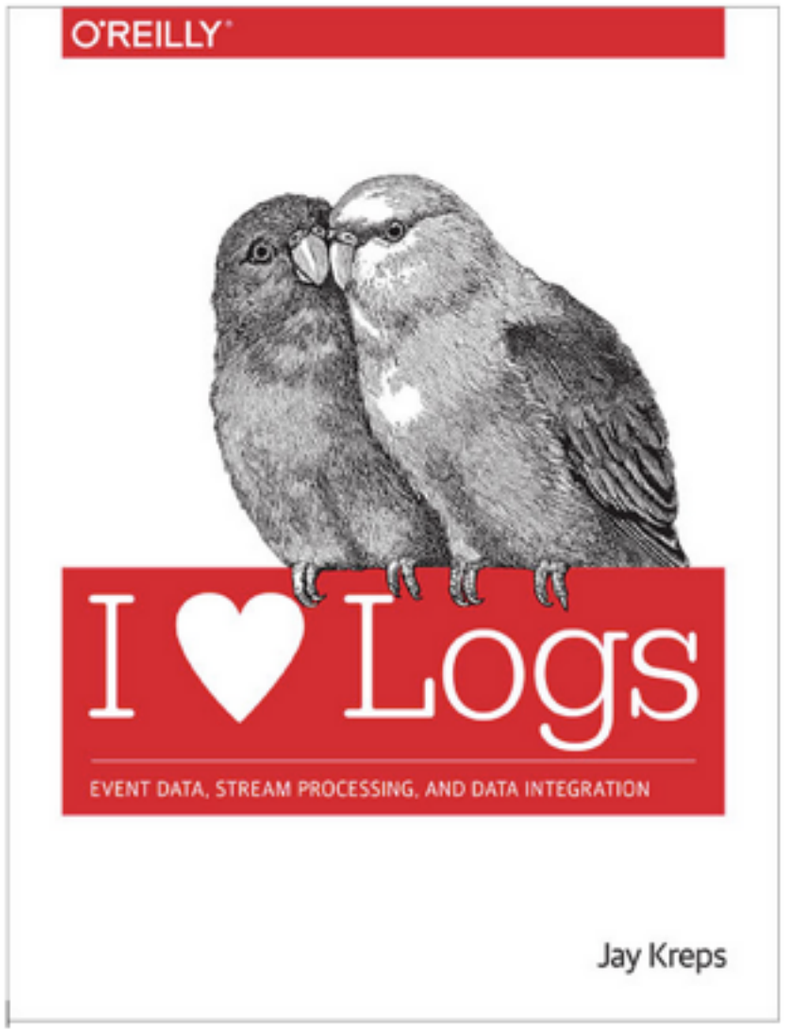
Maybe we could call this the Kappa Architecture, though it may be too simple of an idea to merit a Greek letter.

Who is Jay Kreps

- Jay has been involved in lots of projects:
  - Author of the essay:
    - The Log: What every software engineer should know about real-time data's unifying abstraction **(12/16/2013)**
    - https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying

Jay Kreps

- Author of the book: I ♥ Logs

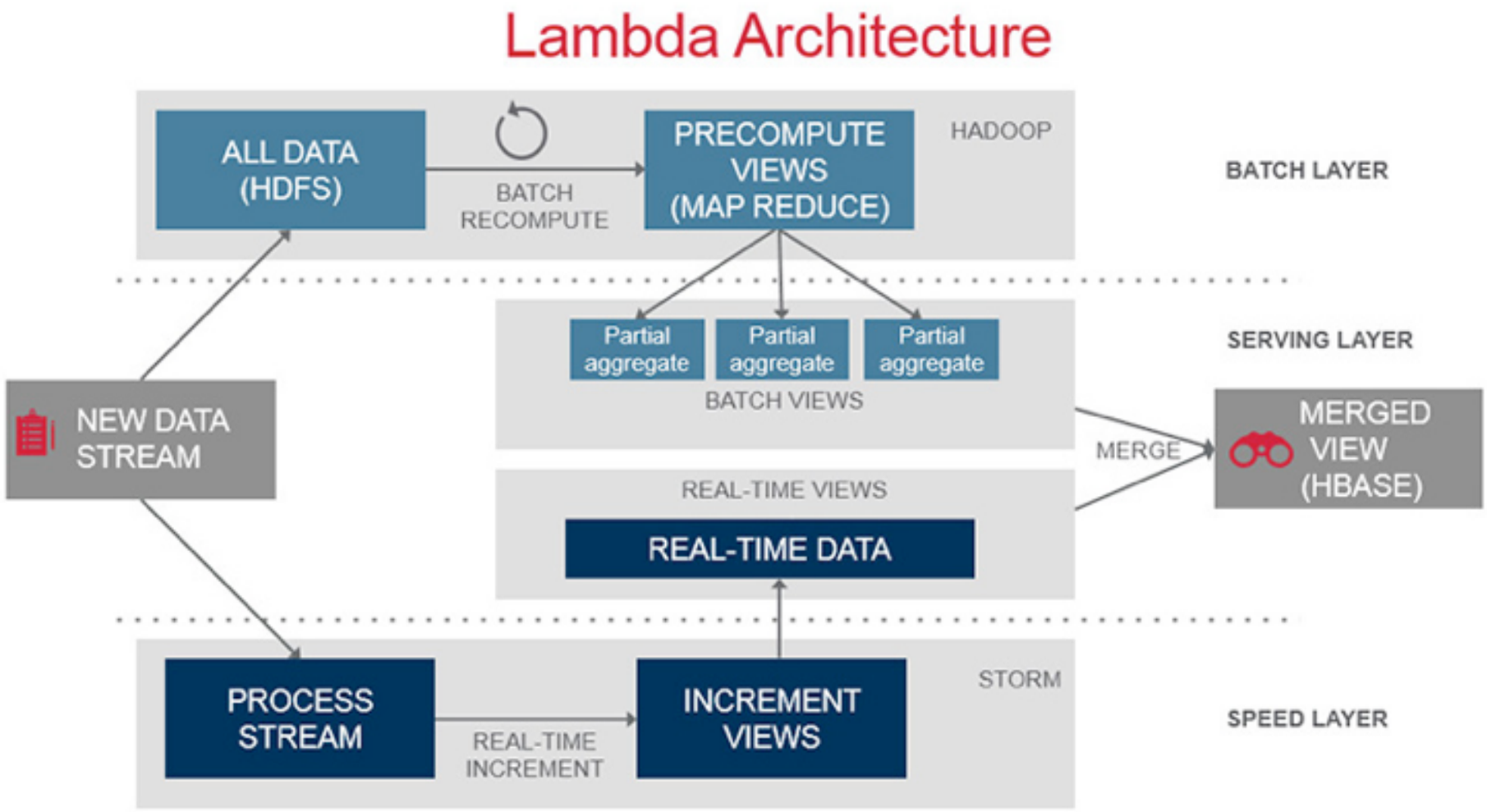


Jay Kreps

- Involved with projects as:
  - Apache Kafka
  - Apache Samza
  - Voldemort
  - Azkaban
  - Ex-Linkedin
  - Now co-founder and CEO of Confluent

Lambda Architecture

- Look something like this:



https://www.mapr.com/developercentral/lambda-architecture

Lambda Architecture

- **Batch layer** that provides the following functionality
  - managing the master dataset, an immutable, append-only set of raw data.
  - pre-computing arbitrary query functions, called batch views.

https://www.mapr.com/developercentral/lambda-architecture

Lambda Architecture

- **Serving layer**
  - This layer indexes the batch views so that they can be queried in ad hoc with low latency.
- **Speed layer**
  - This layer accommodates all requests that are subject to low latency requirements. Using fast and incremental algorithms, the speed layer deals with recent data only.

Lambda Architecture

- **batch layer datasets** can be in a distributed filesystem, while MapReduce can be used to **create batch views** that can be fed to the serving layer.
- The **serving layer** can be implemented using NoSQL technologies such as HBase,Apache Druid, etc.
- **Querying** can be implemented by technologies such as Apache Drill or Impala
- **Speed layer** can be realized with data streaming technologies such as Apache Storm or Spark Streaming

https://www.mapr.com/developercentral/lambda-architecture

Pros of Lambda Architecture

- Retain the input data unchanged.
  - Think about modeling data transformations, series of data states from the original input.
- Lambda architecture take in account the problem of reprocessing data.
  - this happens all the time, **the code will change**, and you will need to reprocess all the information. Lots of reasons and you will need to live with this.

Cons of Lambda Architecture

- Maintain the code that need to produce the same result from two complex distributed system is painful.
  - Very different code for MapReduce and Storm/ Apache Spark
- Not only is about different code, is also about debugging and interaction with other products like (hive, Oozie, Cascading, etc)
- At the end is a problem about different and diverging programming paradigms.

So what is Kappa Architecture

- The proposal of Jay Kreps is so simple:
  - Use kafka (or other system) that will let you retain the full log of the data you need to reprocess.
  - When you want to do the reprocessing, start a second instance of your stream processing job that starts processing from the beginning of the retained data, but direct this output data to a new output table.

So what is Kappa Architecture

- part II
  - When the second job has caught up, switch the application to read from the new table.
  - Stop the old version of the job, and delete the old output table.

So what is Kappa Architecture

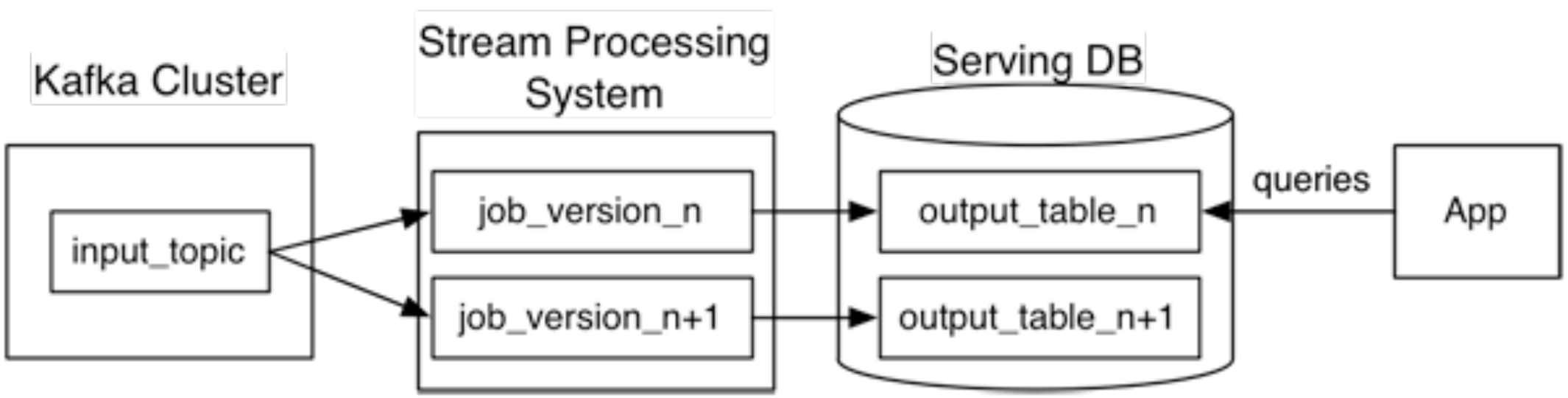
- part II
  - When the second job has caught up, switch the application to read from the new table.
  - Stop the old version of the job, and delete the old output table.

So what is Kappa Architecture

- part II
  - When the second job has caught up, switch the application to read from the new table.
  - Stop the old version of the job, and delete the old output table.

So what is Kappa Architecture

- This architecture looks something like this:



So what is Kappa Architecture

- The first benefit is that only you need to reprocessing only when you change the code.
- You can check if the new version is working ok and if not reverse to the old output table.
- You can mirror a Kafka topic to HDFS so you are not limited to the Kafka retention configuration.
- You have only a code to maintain with an unique framework.

So what is Kappa Architecture

- The real advantage is not about efficiency at all (You will need extra temporarily storage when reprocessing for example) is **allowing your team** to develop, test, debug and operate their systems on top of a **single processing framework**.

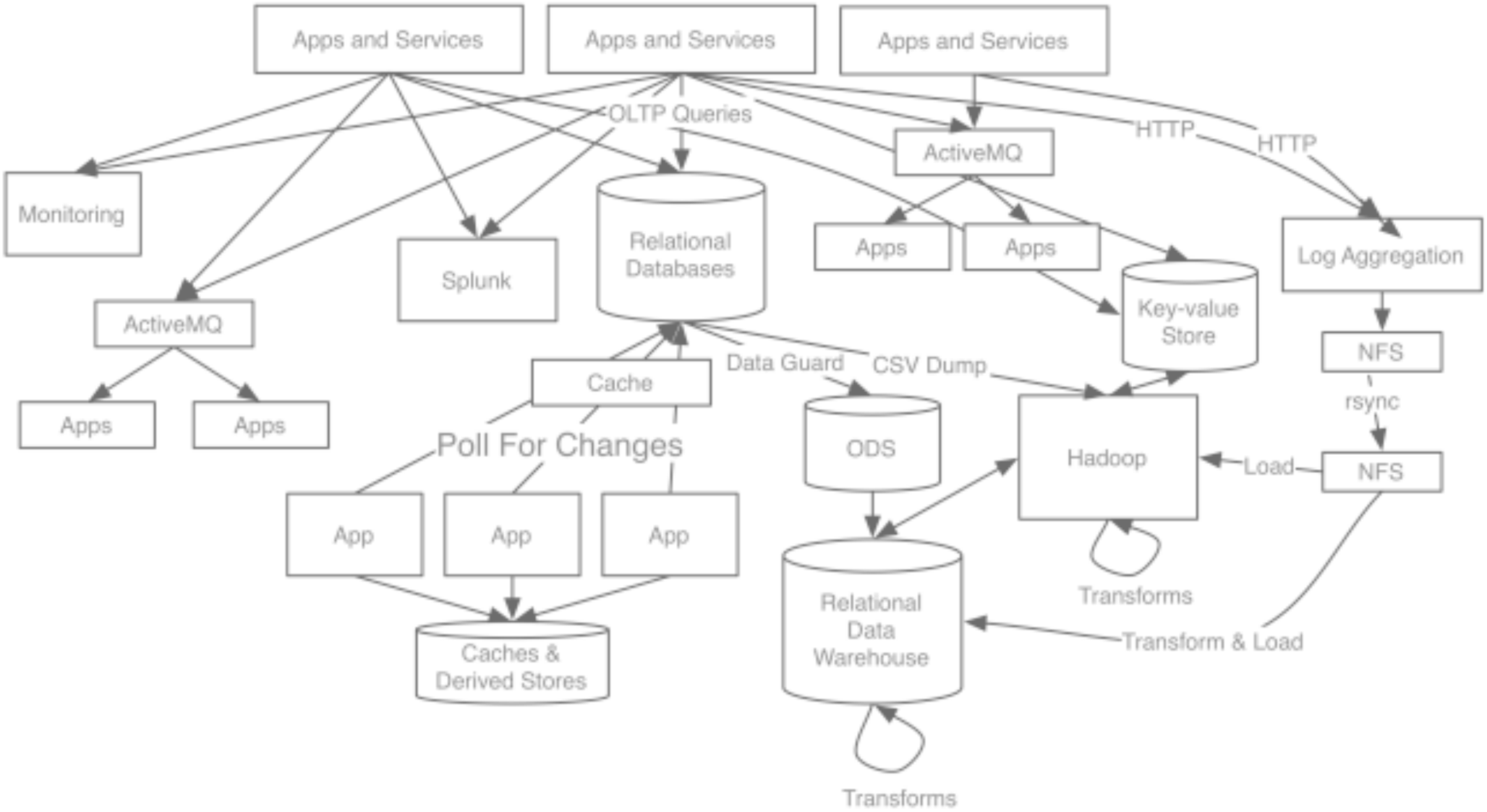
What is not Kappa Architecture

- Is not a silver bullet to solve every problem at Big Data.
- Is not a list of prescriptions of technologies. You can implement with your favorite frameworks.
- Is not a rigid set of rules. But helps to maintain the complex projects simple.

How we use Kappa Architecture

- We start working with projects with a complex structure like Linkedin looks at early stage.
- That's very usual.

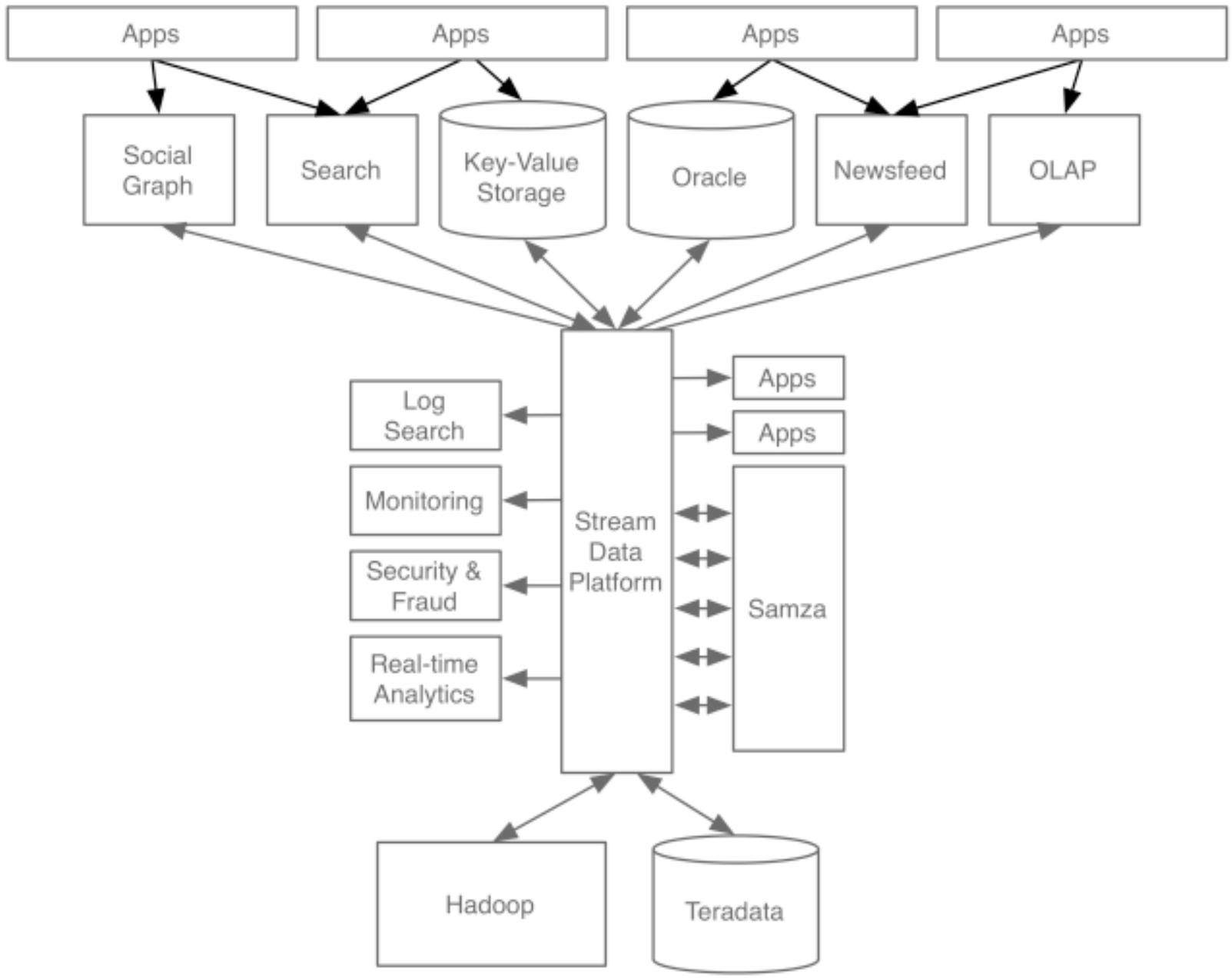
How we use Kappa Architecture



How we use Kappa Architecture

- We try to refactoring the data flows to fix in a Kappa Architecture.

How we use Kappa Architecture



How we use Kappa Architecture

- We use Kafka as Stream Data Platform
- Instead of Samza we feel more comfortable with Spark Streaming.
- At ASPGems we choose Apache Spark as our Analytics Engine and not only for Spark Streaming.

How we use Kappa Architecture

- At the end, Kappa Architecture is design pattern for us.
- We use/clone this pattern in almost our projects.
- We have projects of every size, volume of data or speed needing and fix with the Kappa Architecture.

Use Cases

Telefónica - MSS

- We use KA to calculate near real time KPIs, SLAs related with the managed security system.
- We simplify the data flow of the input data.
- Kafka in the streaming data platform.
- As MPP we use CassandraDB.

IOT - OBD II

- One of our clients install On Board Devices in the cars of its customers.
- We implement an API to got all the information in real time and inject the information in Kafka.
- The business rules are implemented in a CEP running into Apache Spark Streaming.
- As MPP we use Elastic Search.

Questions

Thank you

Juantomás García

juantomas@aspgems.com

@juantomas