



Design Pattern of Hippo Service

Roger

What is this?

Borrow the concept from

Microservice

????????????????

單一大架構

Step 1: get data



Step 2: process data



Step 3: dump data



Step 4: get other data



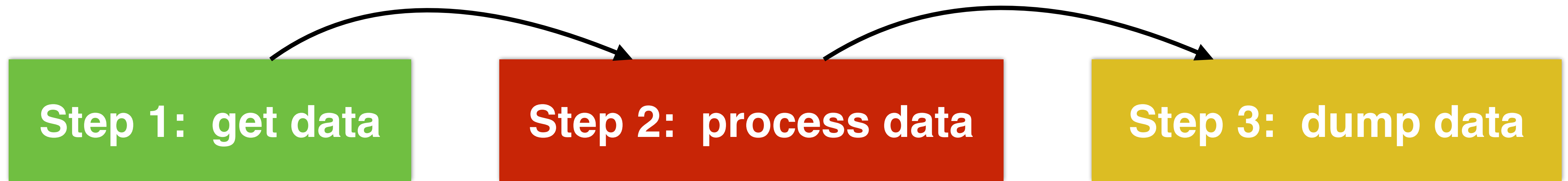
Step N:

Problem1: App太大包

Problem 2: 一步驟掛, 全掛

Problem 3: 每一步驟不能用不同技術

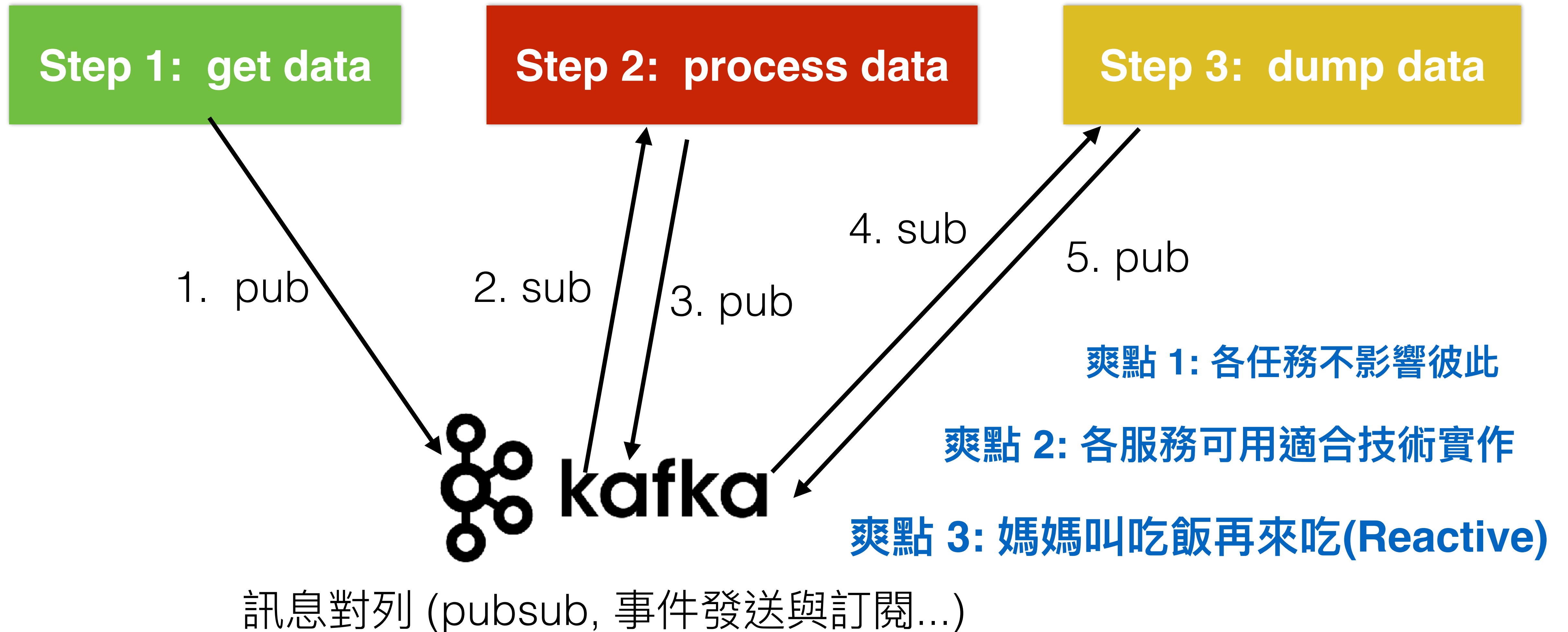
排程架構



Problem 1: 每一步驟該設幾點?

Problem 2: 一步驟掛, 還是接著跑!

微服務架構

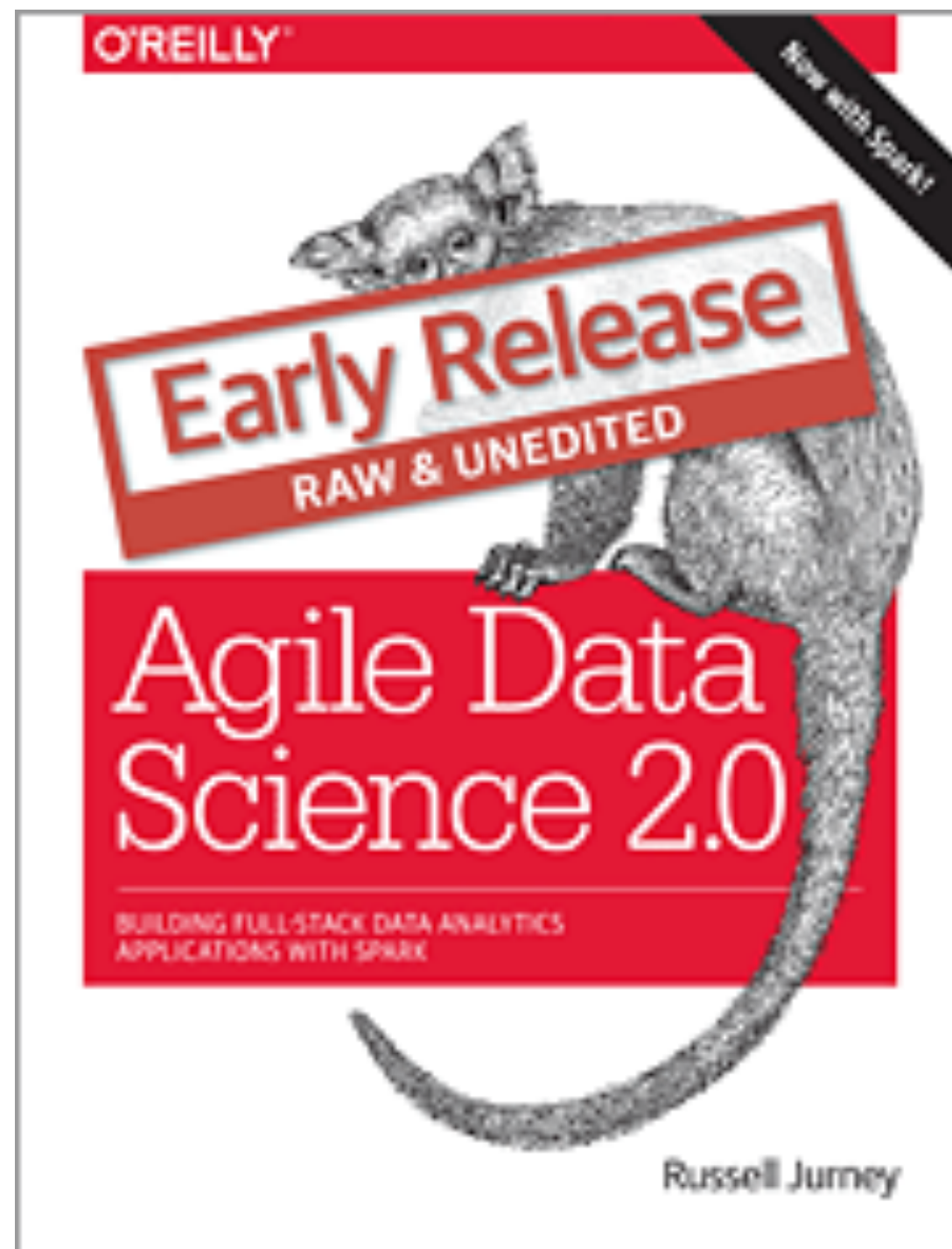


So we need our Data Microservice!

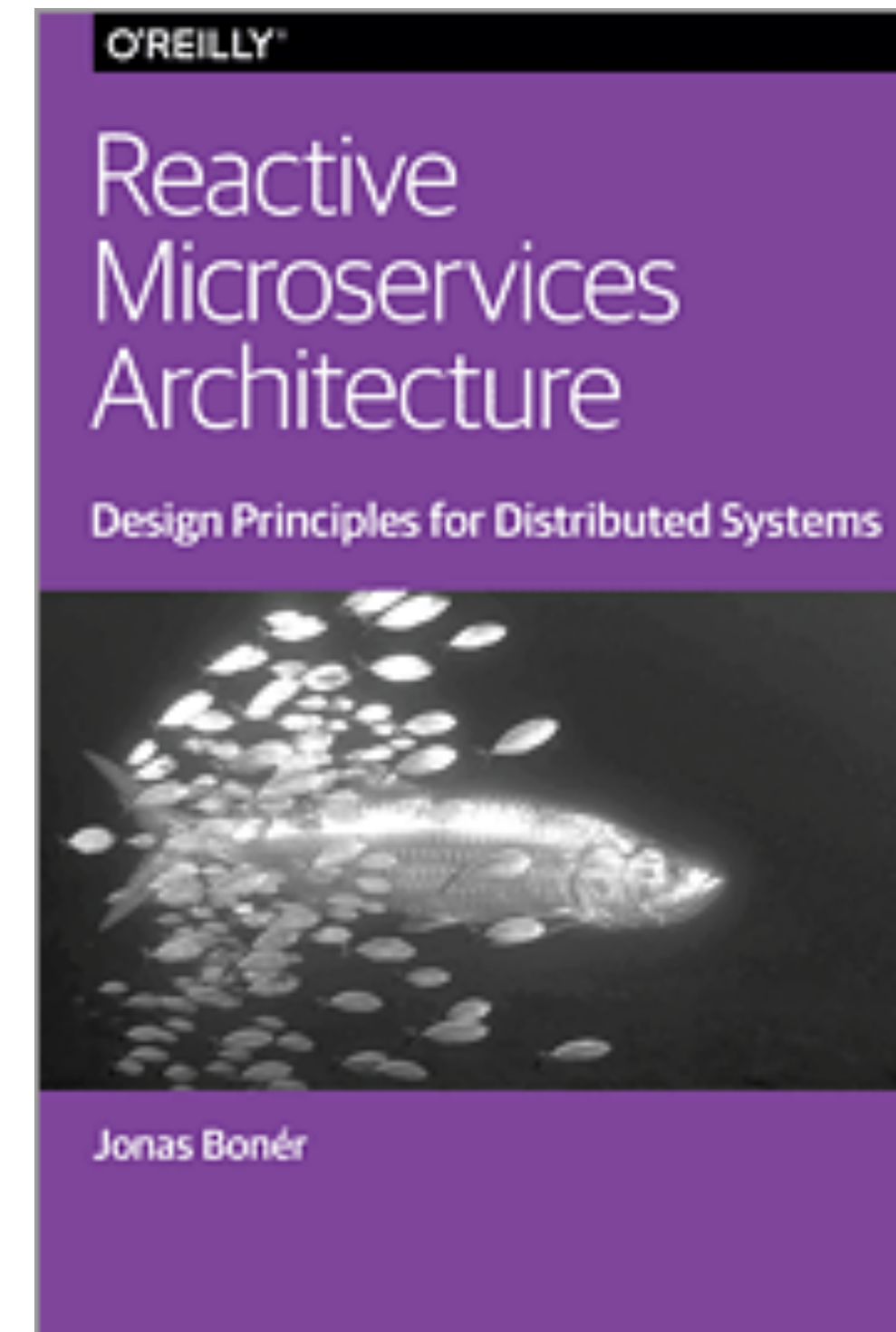


HIPPO

Reference



Agile Data Science 2.0
Building Full-Stack Data Analytics Applications
with Spark



Reactive Microservices Architecture
Design Principles for Distributed Systems

HIPPO types by purpose

ETL

- **Frontier**
- **Batch-ETL**
- **DB-ETL**
- **Stream-ETL**

Application

- **Query Service**
- **Web Service**

Science

- **ML Service**

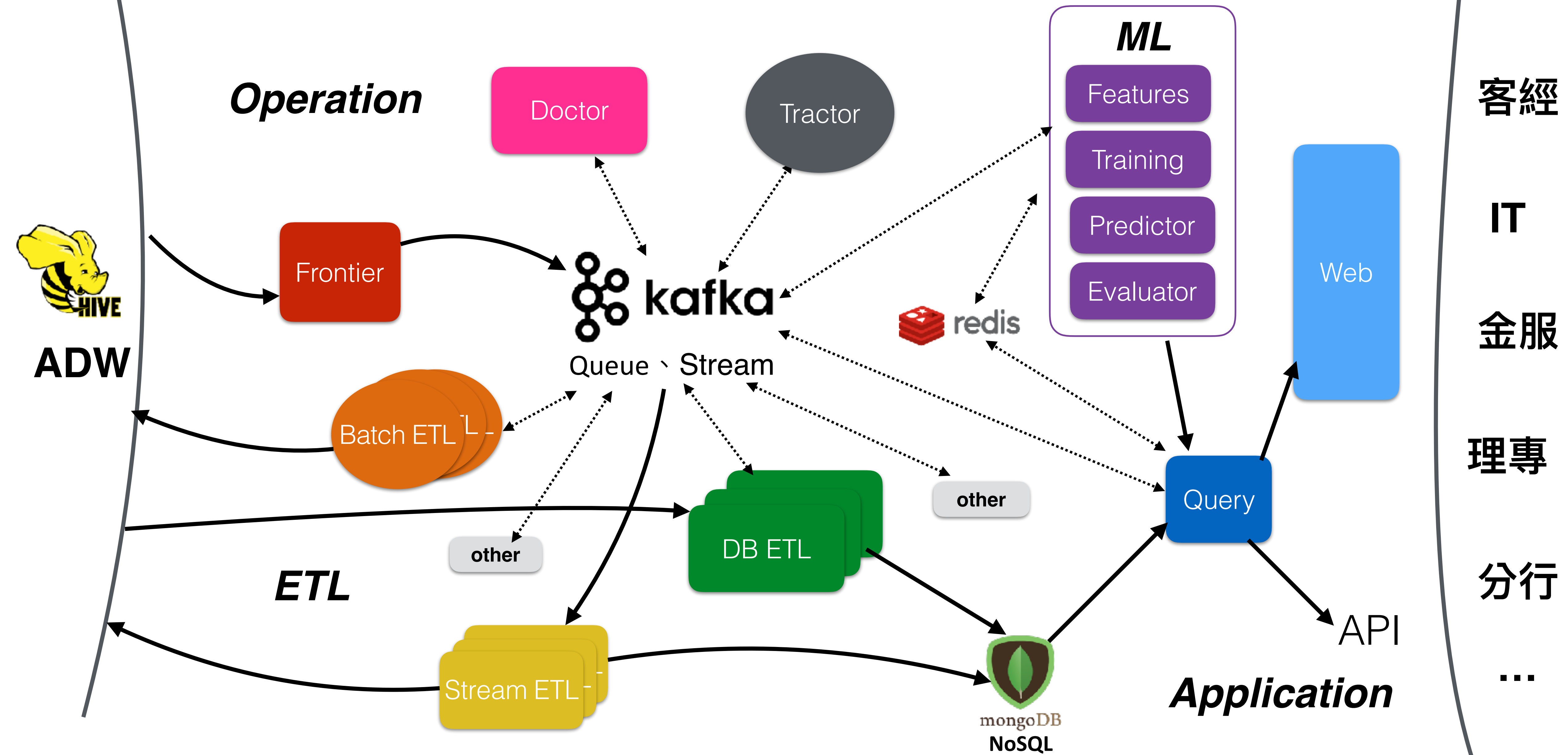
Operation

- **Doctor**
- **Tractor**

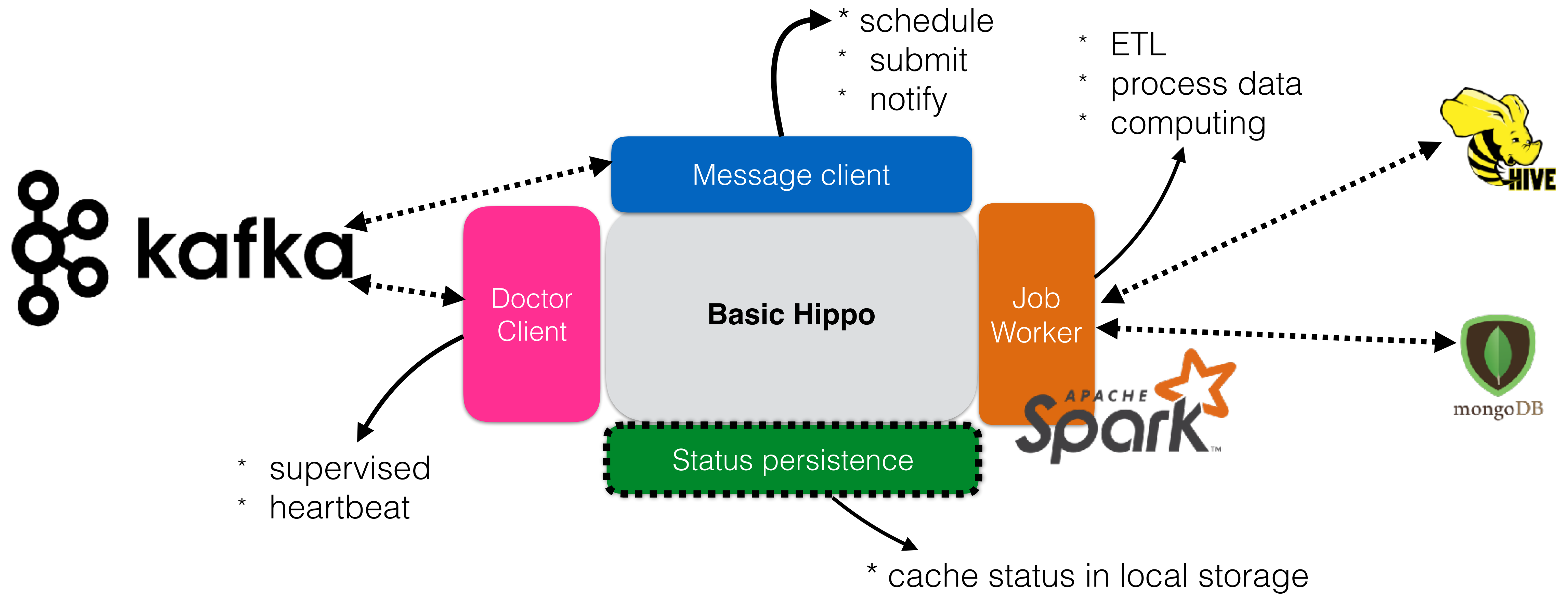
Other

general purpose
or
mixed purpose

Future HIPPO Views



Basic hippo structure

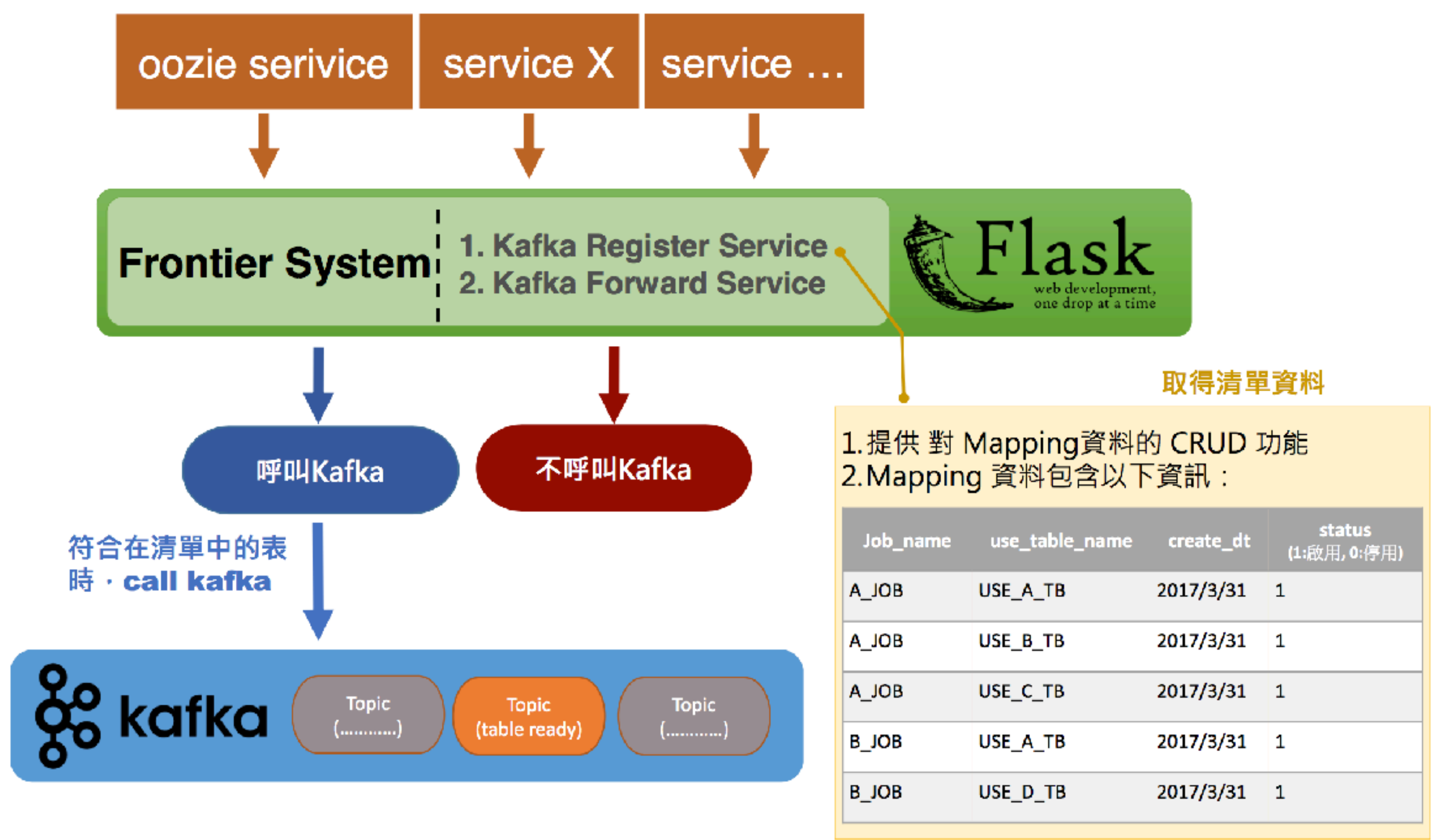


language

ETL

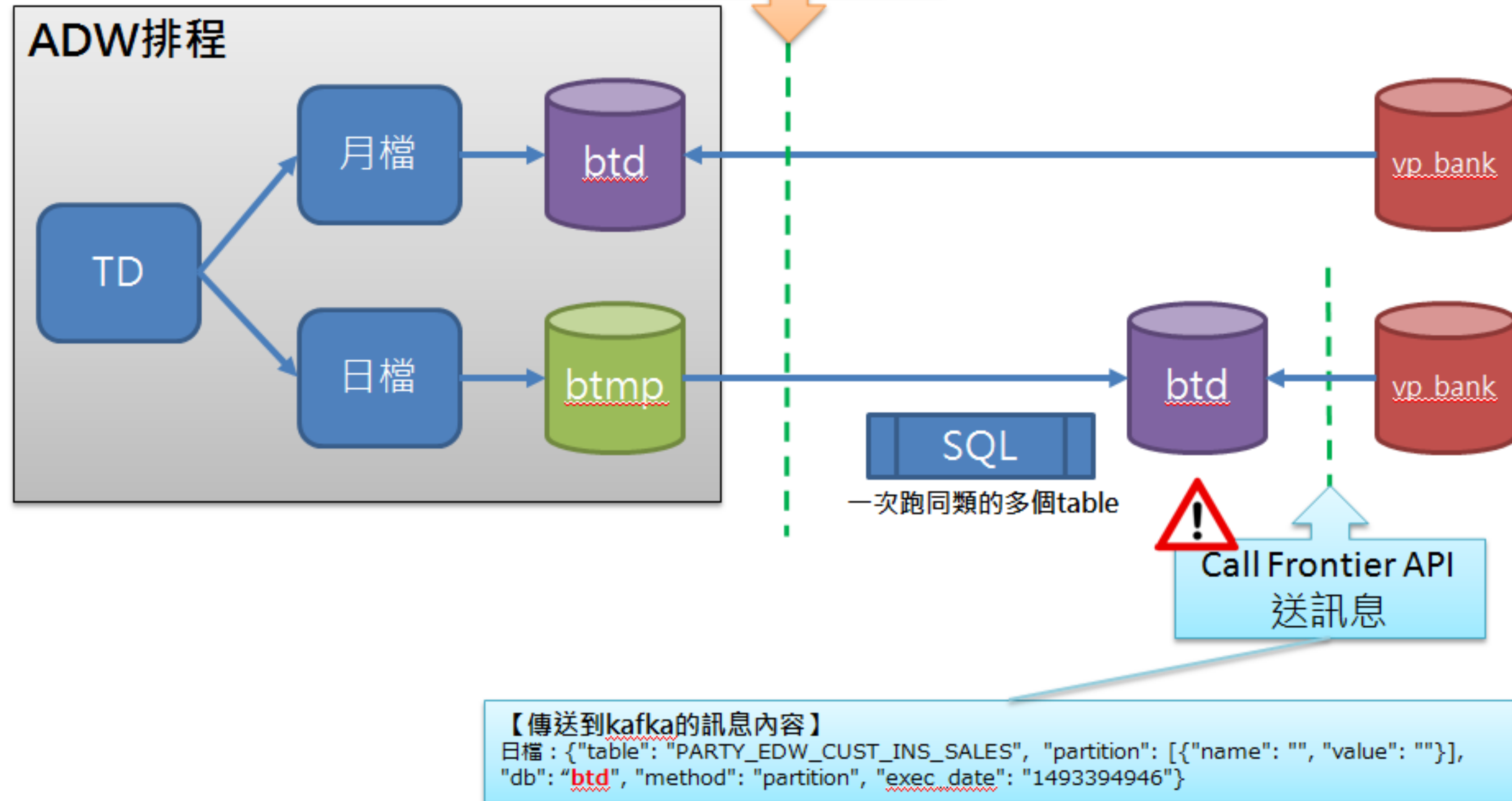
Frontier

架構圖



【傳送到kafka的訊息內容】

月檔: {"table": "PARTY_TRU_ACCUM_BAL_M", "partition": [{"name": "YYYYMM", "value": "201703"}],
"db": "btd", "method": "overwrite_partition", "exec_date": "1493394946"}
日檔: {"table": "PARTY_EDW_CUST_INS_SALES", "partition": [{"name": "", "value": ""}], "db":
"btmp", "method": "partition", "exec_date": "1493394946"}

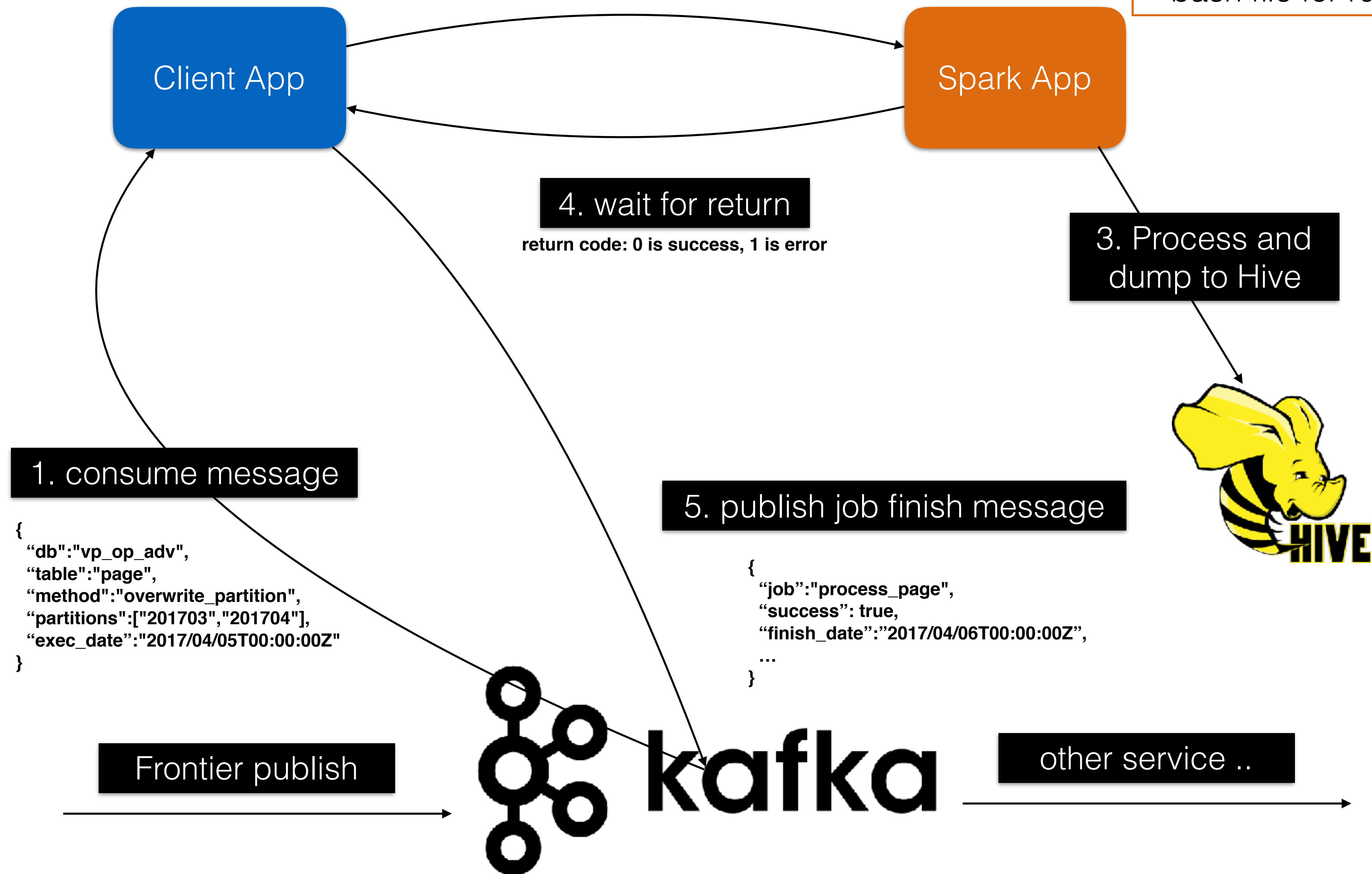


Batch-ETL

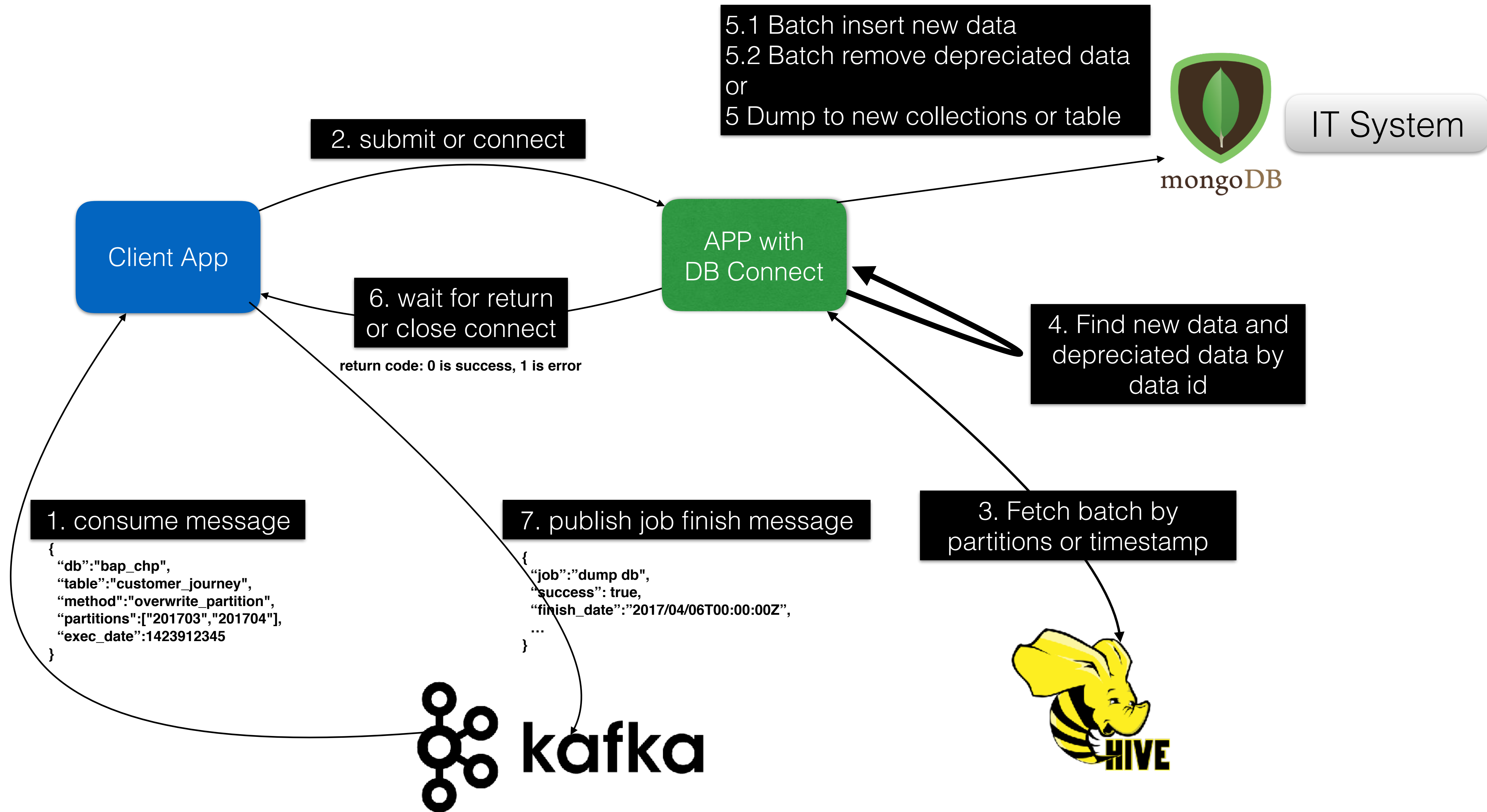
- * Kafka client
- * Process lib
- * Akka or Thread

2. spark-submit
service call "bash" or "command"

- * Yarn-client
- * Hive SQL
- * jar or py
- * bash file for running app



DB-ETL



Stream-ETL

Same source, but different consume purposes



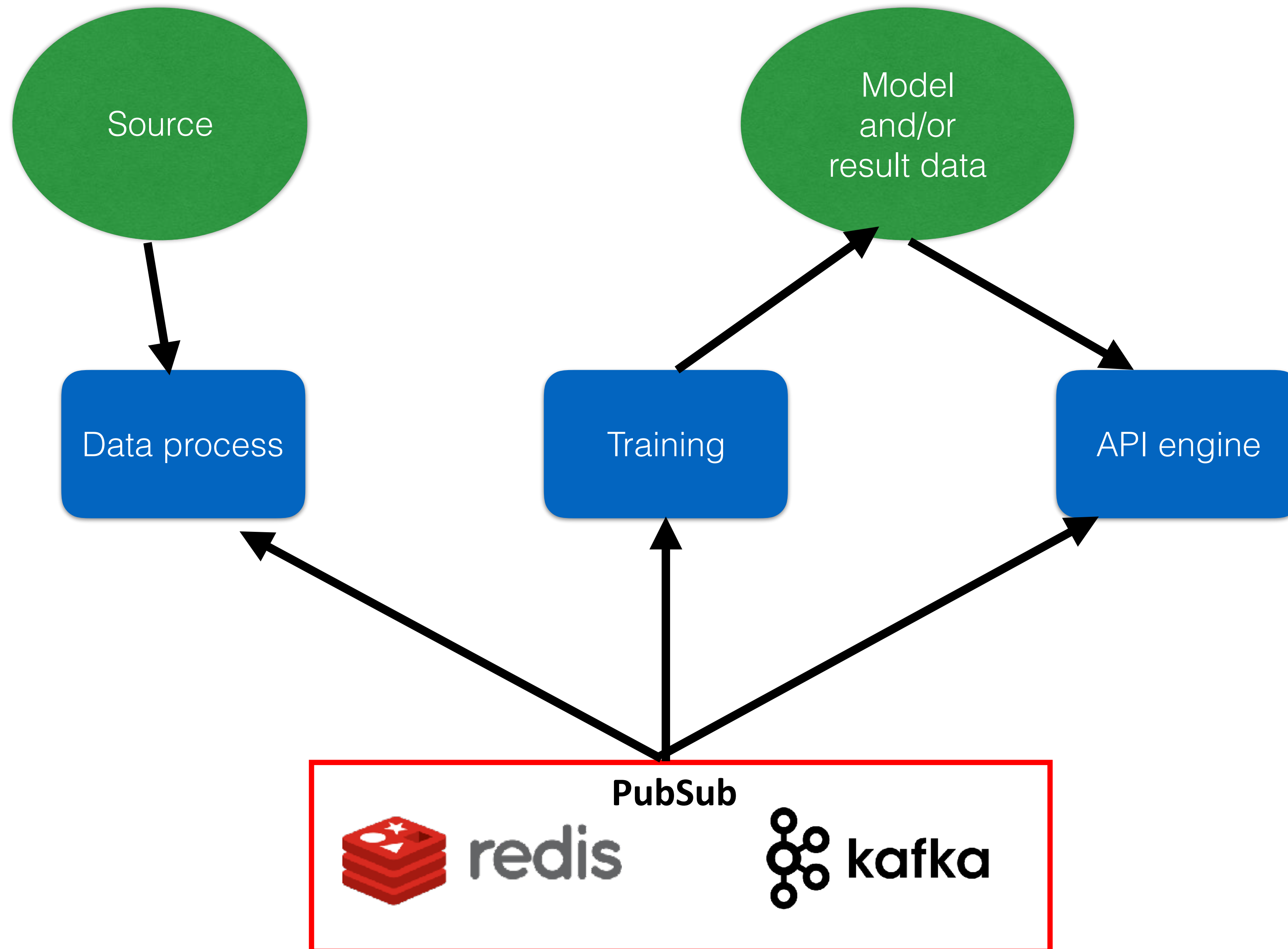
Neha Narkhede, 2016

ETL Is Dead, Long Live Streams: real-time streams w/ Apache Kafka

https://www.youtube.com/watch?v=I32hmY4diFY&list=PLVeYbWw30yOKWUZOrLqeVjWf_Vfvy3AGf

Science

ML Service

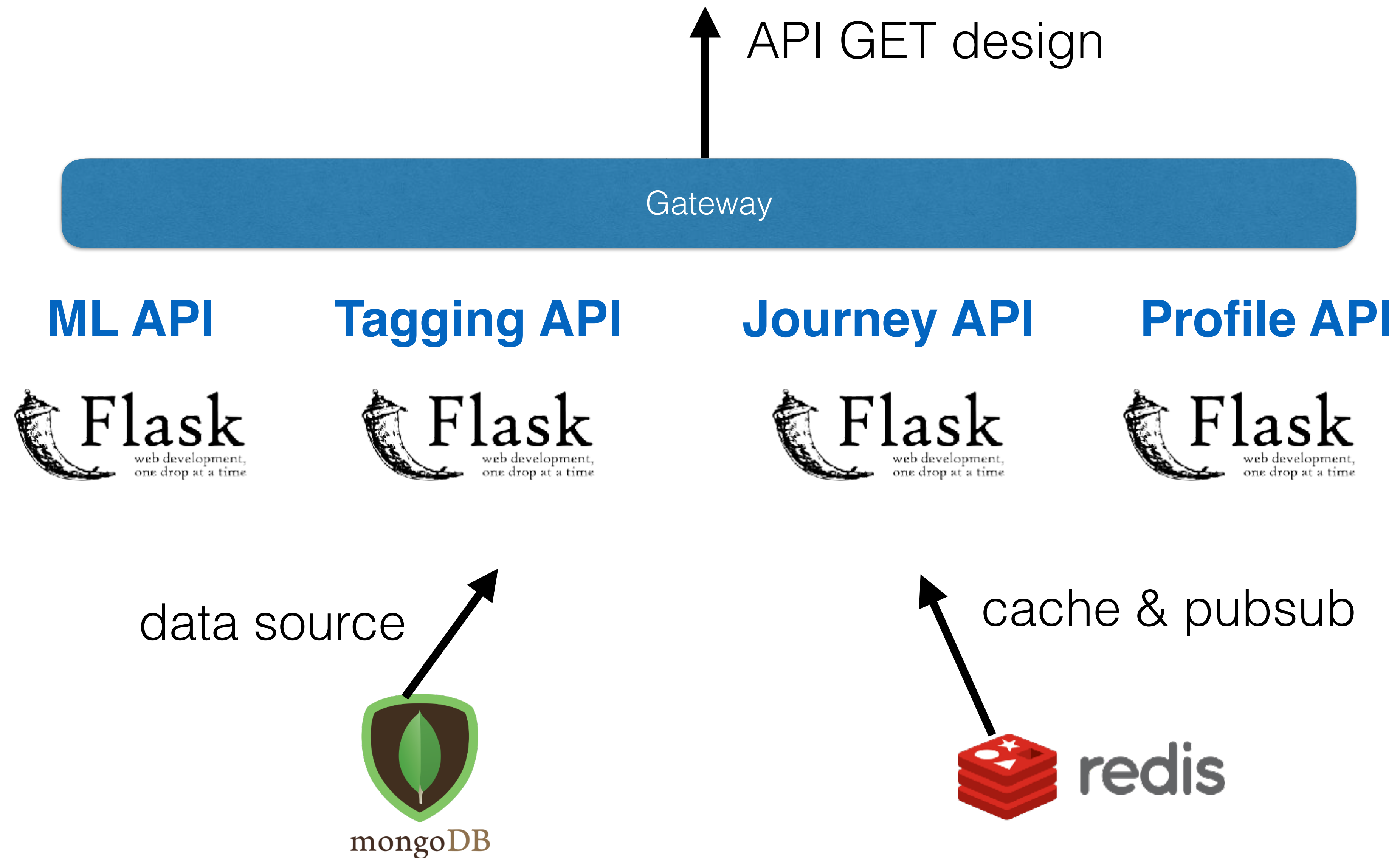


Repository

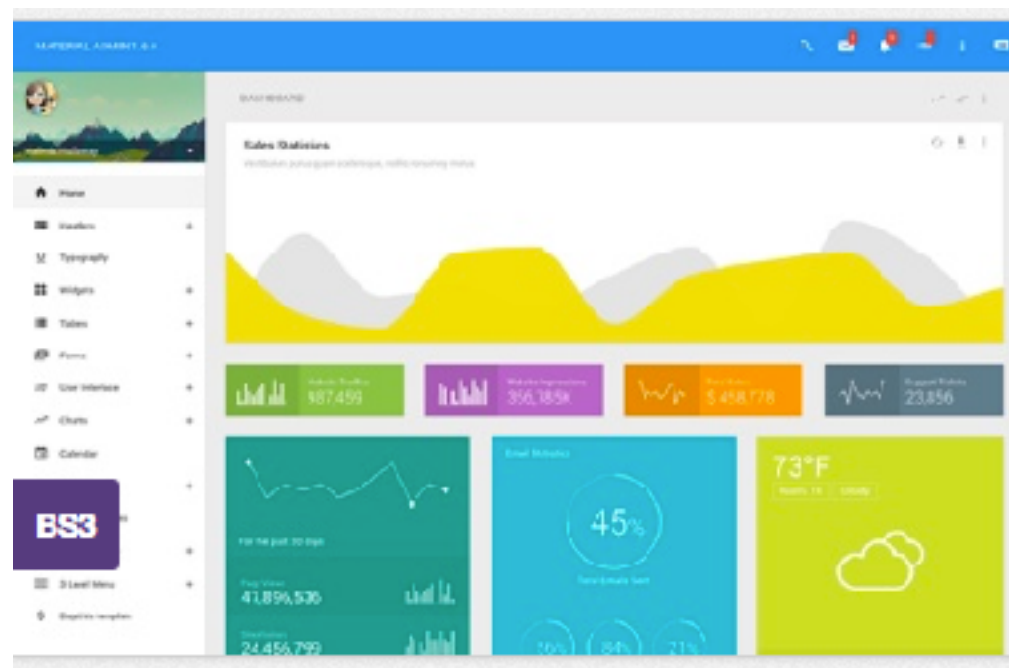
github.com/b967050008/MLServices

Application

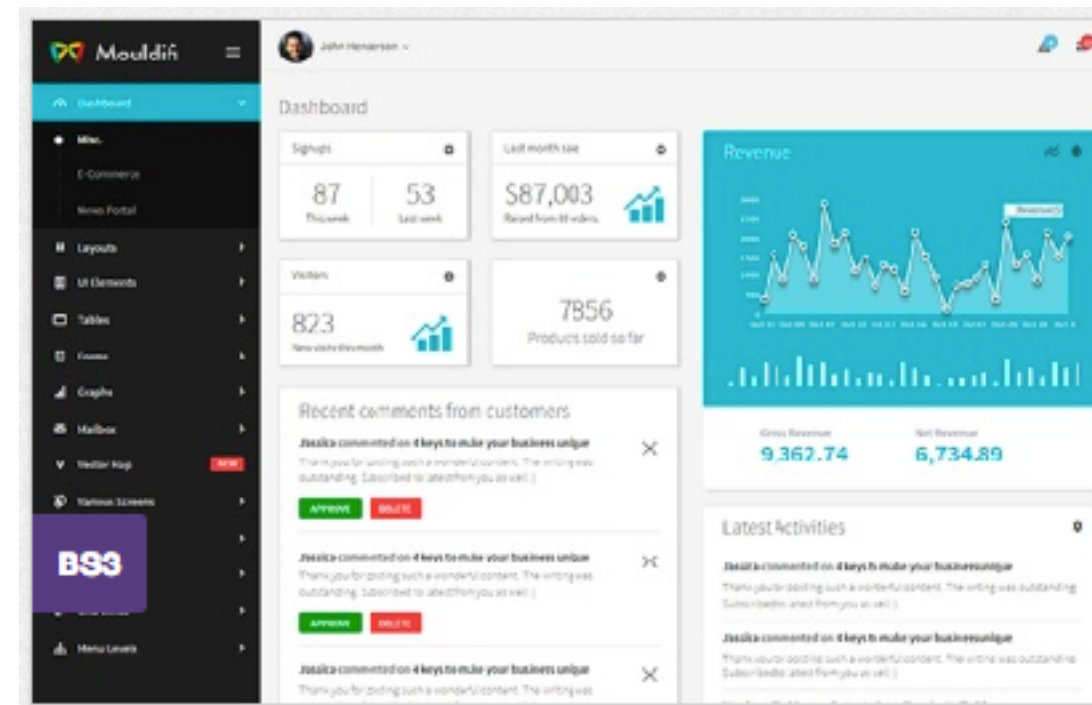
Query Service



Web Service



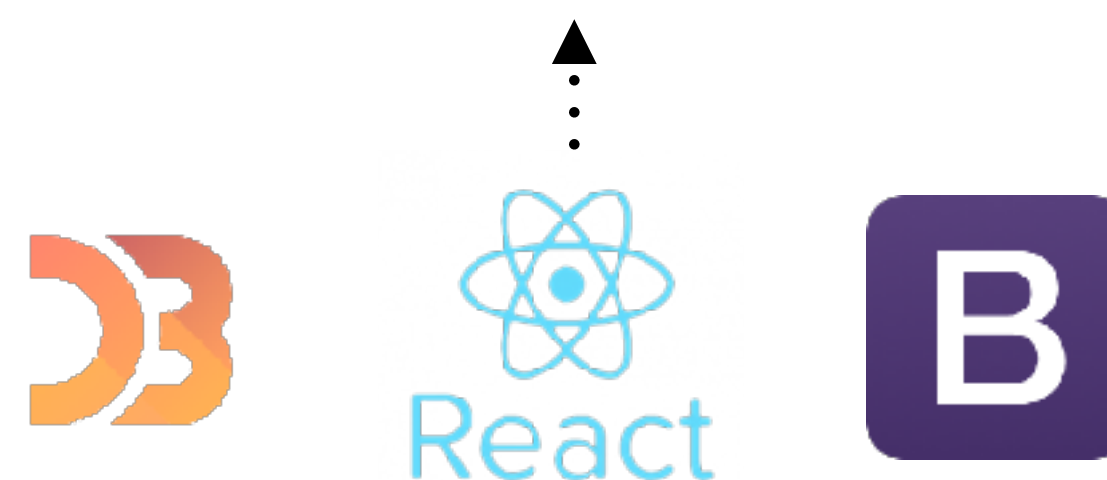
總體指標探索



個人化資訊檢視

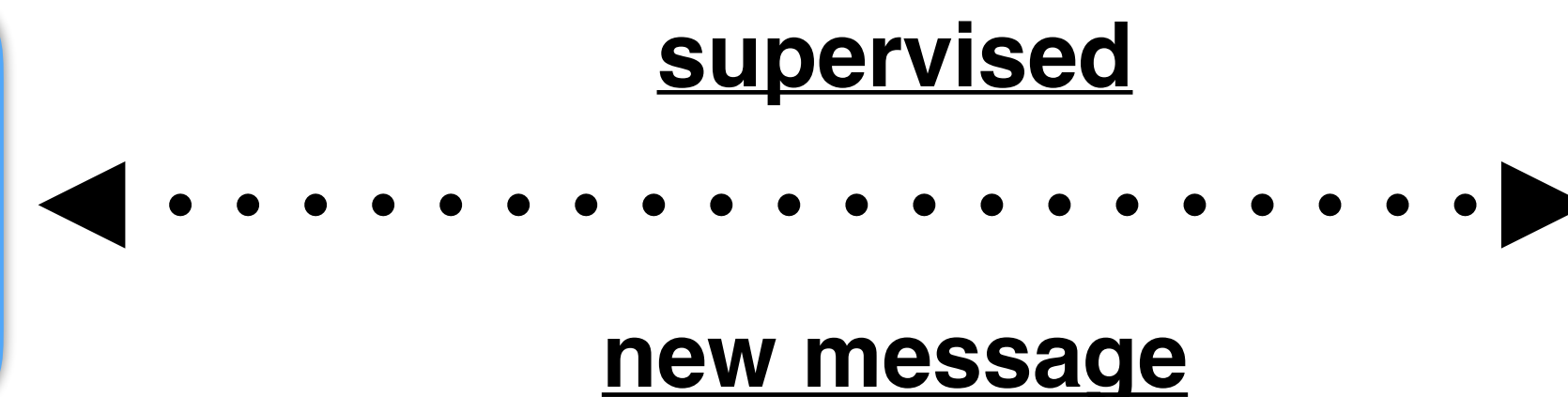


服務安全性及完整性監控



Frontend framework - Mitch Team

socket.io
data push

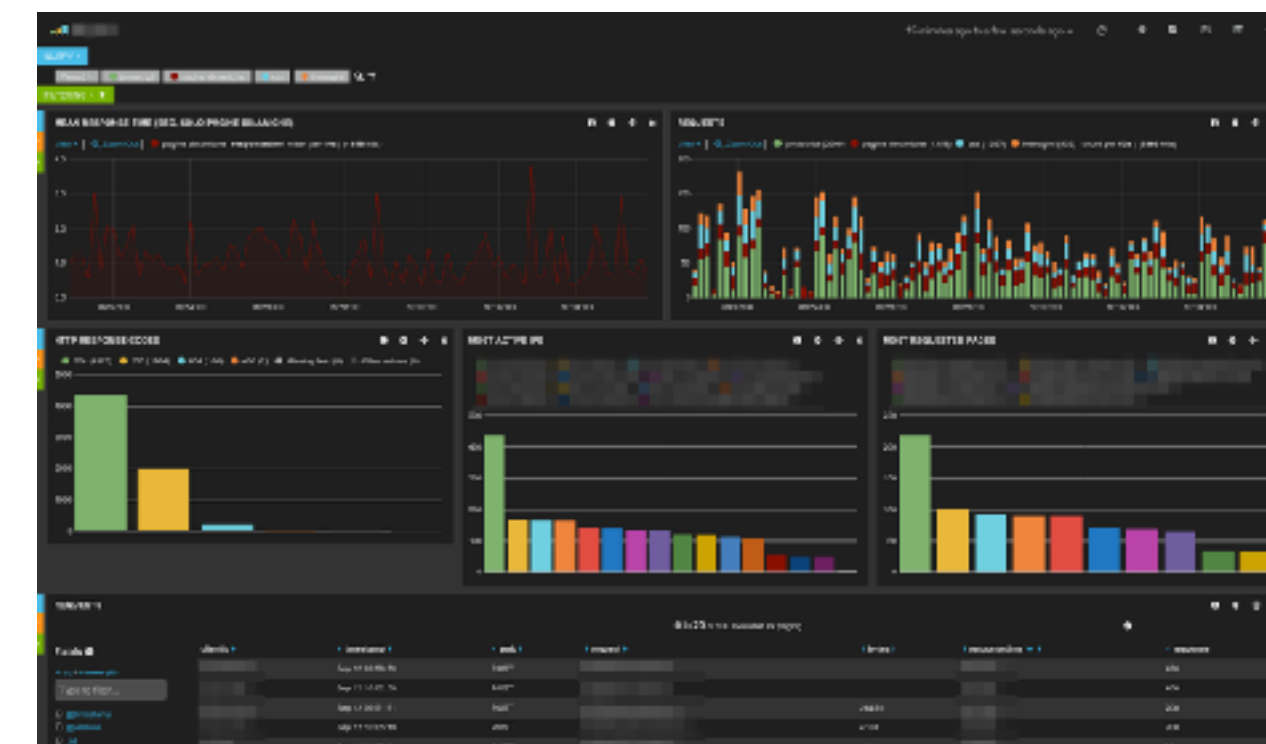


Backend service - HIPPO

Operation

Doctor

Scheduler
Status persistence
Supervise



Doctor
Server

Doctor Client

Doctor Client

Doctor Client

Doctor Client



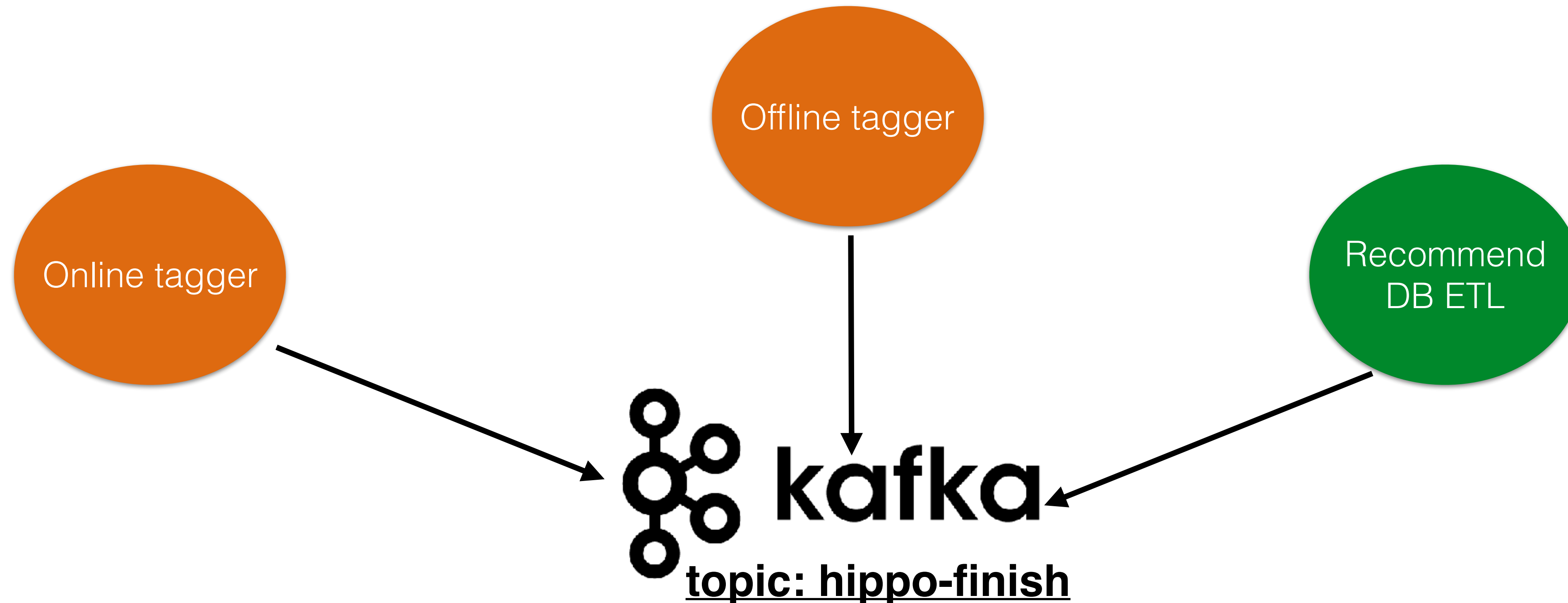
Heartbeat



kafka

topic: health-examination

Tractor



track hippo mission status

