What I am doing

Research and familiarize with stream processing techniques, mainly focus on apache storm

Advantages of storm

Fault-tolerant: when workers die, Apache Storm will automatically restart them. If a node dies, the worker will be restarted on another node.

Flexibility: Apache Storm provides flexibility by integrating into any programming language.

Scalability: Storm uses ZooKeeper to coordinate various configurations in the cluster so that the Storm cluster can be easily expanded

Data processing guarantee. A real-time system must ensure that all data is processed successfully.

Disadvantages of storm

Not suitable for smaller datasets： Apache Storm is a distributed system and not a good choice for small-scale applications.

Stateless: users need to manage their own state

Apache Spark – Structured Streaming

Advantages:

- unified programming model(untyped APIS, typed APIS,spark.sql)
- fault tolarance and consistency: ensure data processing consistency in the event of failures and recovery to the state before the failure.
- High-performance processing: use micro-batch to simulate stream process.

Disadvantages:

- not suitable for highly real-time scenarios: latency typically ranges from seconds to minutes.
- depend on Spark cluster: need more resource and cost.
- need time to be familiar with it.

Apache Beam

Advantages:

- can use Apache Spark, Apache Flink, Google Cloud Dataflow and so on in the framework Apache Beam.
- also has fault tolerance and data consistency.
- Scalability and flexibility: for different needs use different self-defined functions

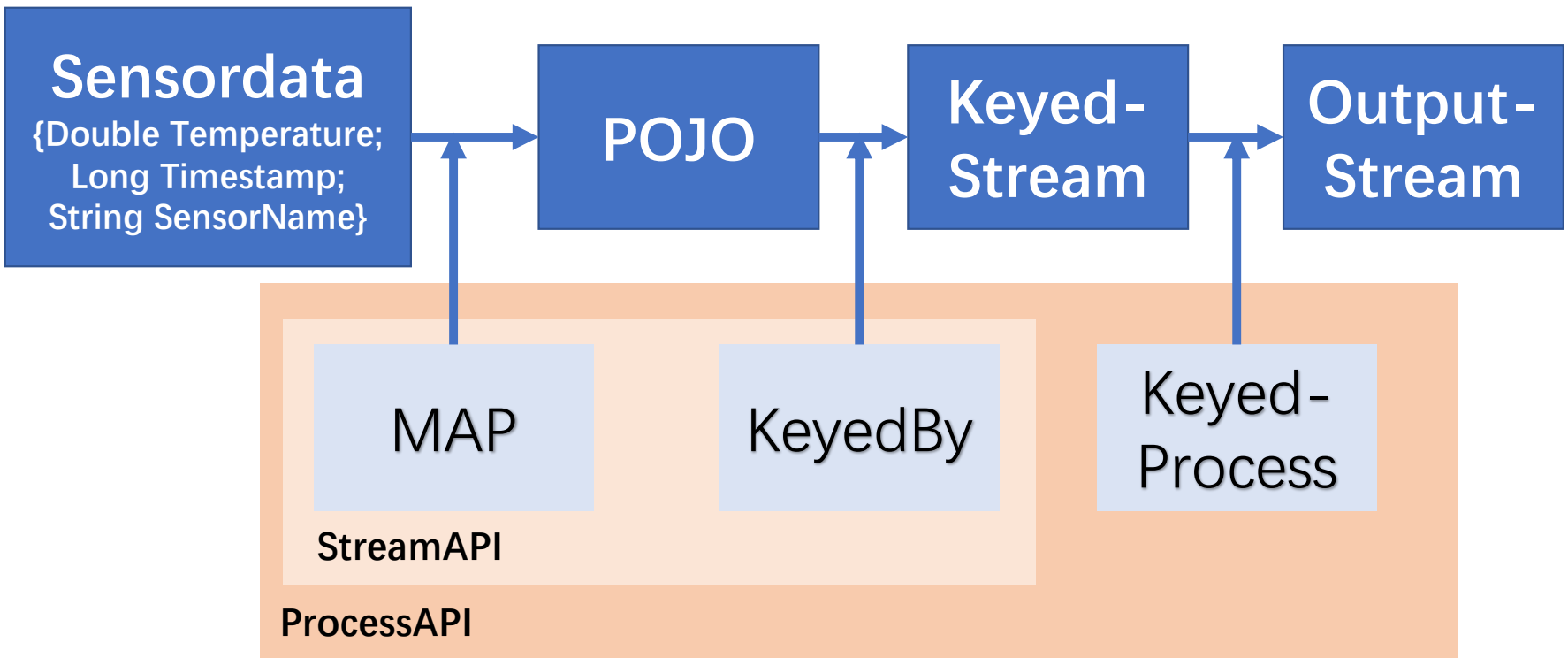
Disadvantages:

- need time to be familiar with it. Because it may occur problems, which caused by different execution engine.
- may occurs problems, which caused by different execution engine.
- configuration and environment may hard to handle.

Apache Flink

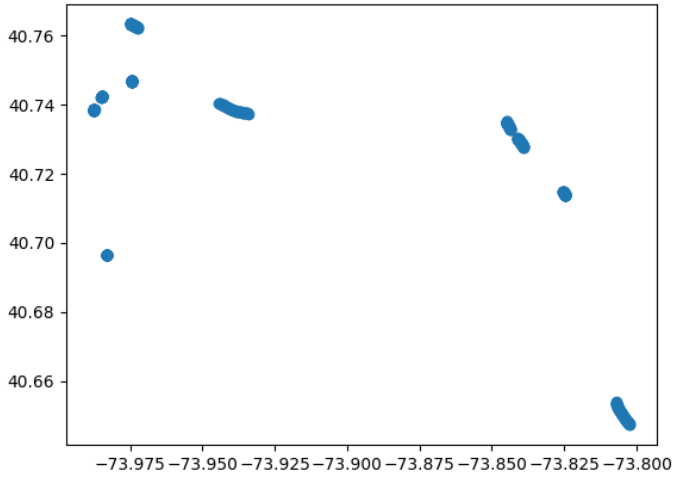
+ efficient
+ User Friendly
+ Easy to deploy and manage
+ Various data streaming tool, API
- Supports only limited types of streaming data sources

Implementation



Data source

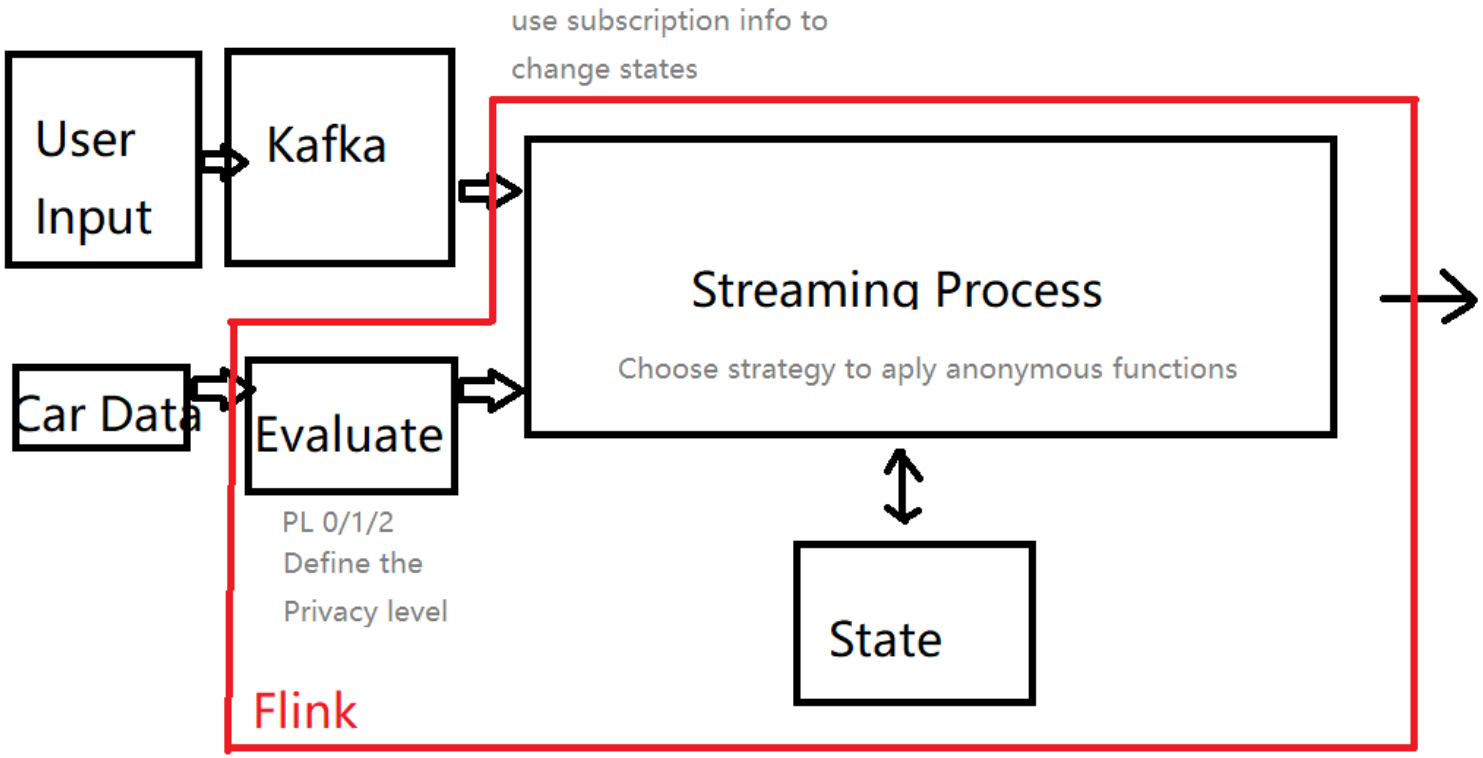
- BDD100K modified
- Content:
  - GPS
  - IMU
  - Timestamp
  - Camera Data
- Extract single trajectory



Scenarios

- Privacy near home.
  - Type: user defined
  - Parameters: PL if activated, distance threshold
  - Rules: Euclidean distance via GPS input.
  - Activity: PL change
- Car crash
  - Type: predefined
  - Parameters: accelerate threshold
  - Rules: sudden acceleration in counter direction of current speed
  - Activity: PL limit downgrade

Some Idea


use subscription info to
change states
User
Input
Kafka
Streaming Process
Choose strategy to aply anonymous functions
Car Data
Evaluate
PL 0/1/2
Define the
Privacy level
State
Flink

Next Phase

- Data source and Kafka public and subscribe dummy stream generator.
- Flink and Spark continue
- Scenario detail
- Implementation of BA