Intro to Flink

Using Stream Processing for Chase Prediction

Anup Sethuram.   08-April-2025

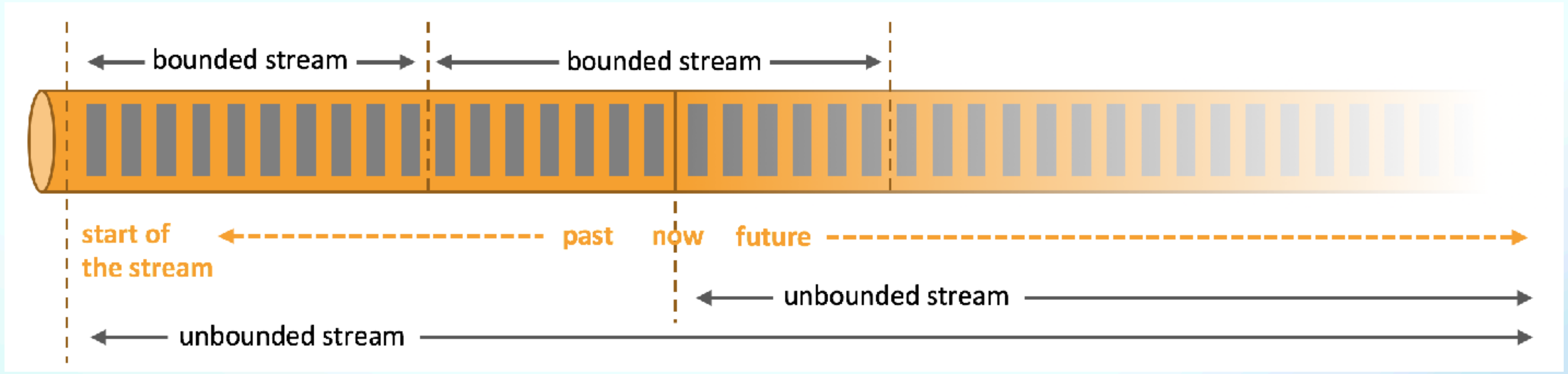
Agenda

>

- Overview
- HLA & Concepts
- Prototyping a Flink App
- Resources
- QnA

Data

Its nature



Monitoring

Fraud Detection

Realtime ETL

Alerting Systems

Realtime Dashboards

Clickstream Analysis

Use Cases

IoT

Predictive Maintenance

Navigation Apps

Realtime Recommendation-Engines

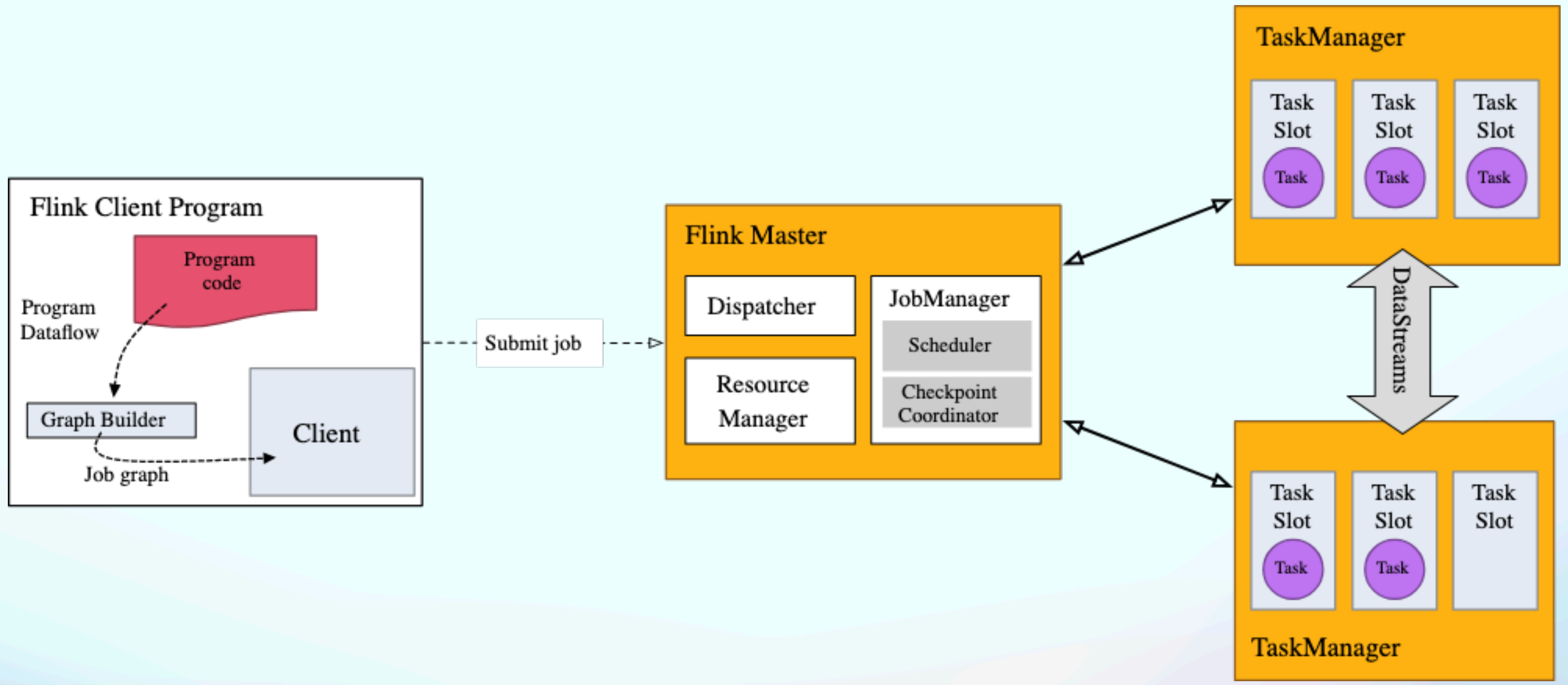
Gaming

Flink

Stateful Distributed Stream Processing Framework

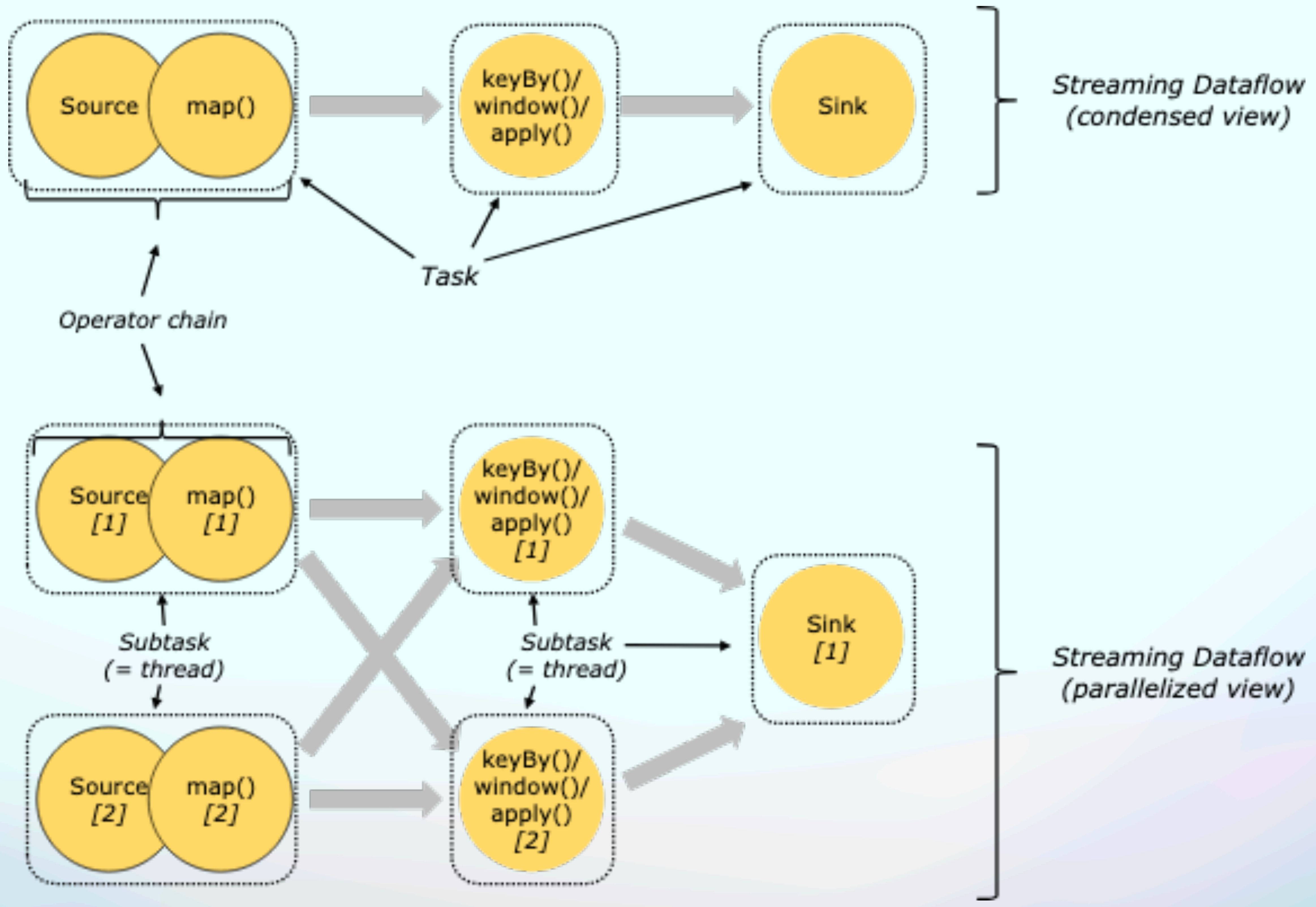
HLA

Execution



Dataflow

>

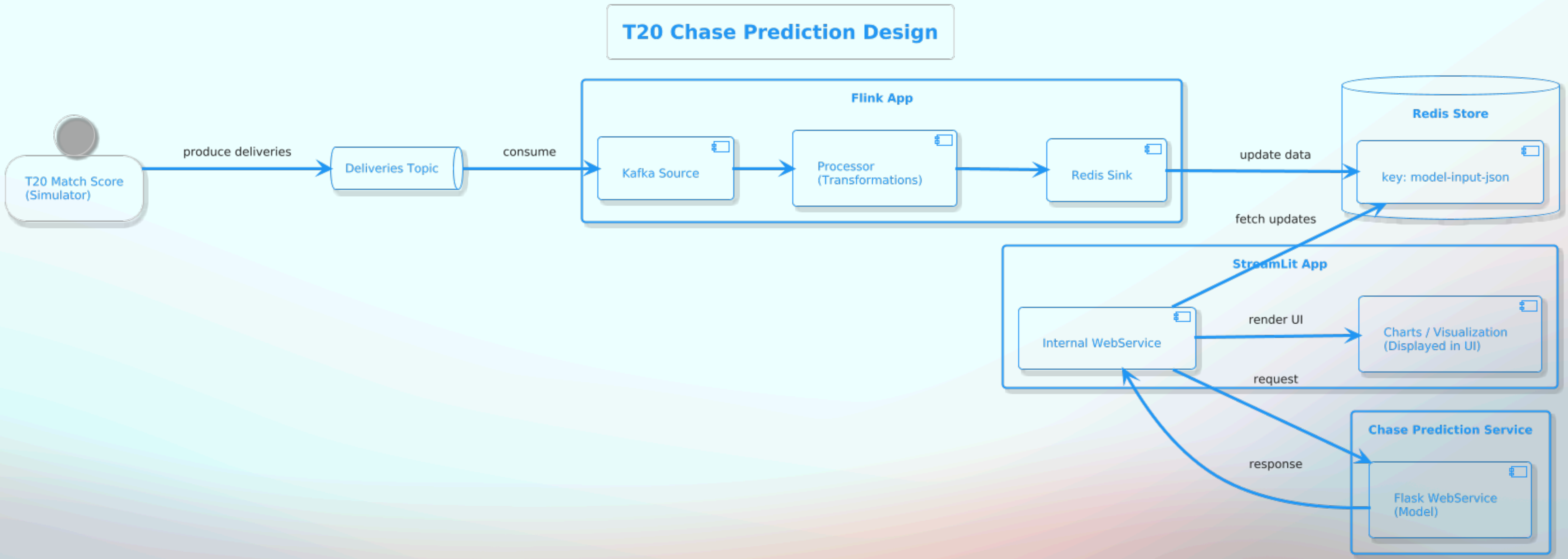

Source
map()
keyBy()/ window()/ apply()
Sink
Streaming Dataflow (condensed view)
Task
Operator chain
Source [1]
map() [1]
keyBy()/ window()/ apply() [1]
Sink [1]
Subtask (= thread)
Subtask (= thread)
Source [2]
map() [2]
keyBy()/ window()/ apply() [2]
Streaming Dataflow (parallelized view)

Concepts

>

- **Flink APIs** : SQL, Table API, **DataStream** API, Stateful Stream Processing
- **Operators**: Map, Filter, Window, Triggers, Connect, Broadcast ..
- **State** Persistence
- **Time**: Event / Process
- **Fault Tolerance**: Checkpoints & Savepoints

Flow



T20 Chase Prediction Design
T20 Match Score (Simulator)
produce deliveries
Deliveries Topic
consume
Flink App
Kafka Source
Processor (Transformations)
Redis Sink
update data
Redis Store
key: model-input-json
fetch updates
StreamLit App
Internal WebService
render UI
Charts / Visualization (Displayed in UI)
request
Chase Prediction Service
response
Flask WebService (Model)

Demo

T20 Cricket Chase Prediction

Best Practices

>

- **CI/CD Pipeline:** Automate testing, deployment, and rollbacks with a robust CI/CD pipeline.
- **Event Streaming:** Utilise platforms like Kafka or Kinesis for reliable, scalable data ingestion.
- **Checkpoints & Savepoints:** Ensure fault tolerance with regular checkpoints and manual savepoints.
- **Time Semantics:** Optimise processing by choosing event time for late data and process time for low latency.
- **Unified Pipeline:** Reuse the same pipeline for both real-time streaming and batch processing.
- **Monitoring & Alerts:** Implement comprehensive monitoring and proactive alerting using tools like Prometheus and Grafana.
- **Configurable Framework:** Build a flexible framework that externalises configurations and centralises common logic. Or make use of Managed Services.

Further Learning

Resources & References

Chase Prediction Repos: https://nupsea.github.io/

https://nightlies.apache.org/flink/flink-docs-release-2.0/

https://flink.apache.org/2025/03/24/apache-flink-2.0.0-a-new-era-of-real-time-data-processing/

?

Thank you!

eanups@yahoo.com