APACHE FLINK

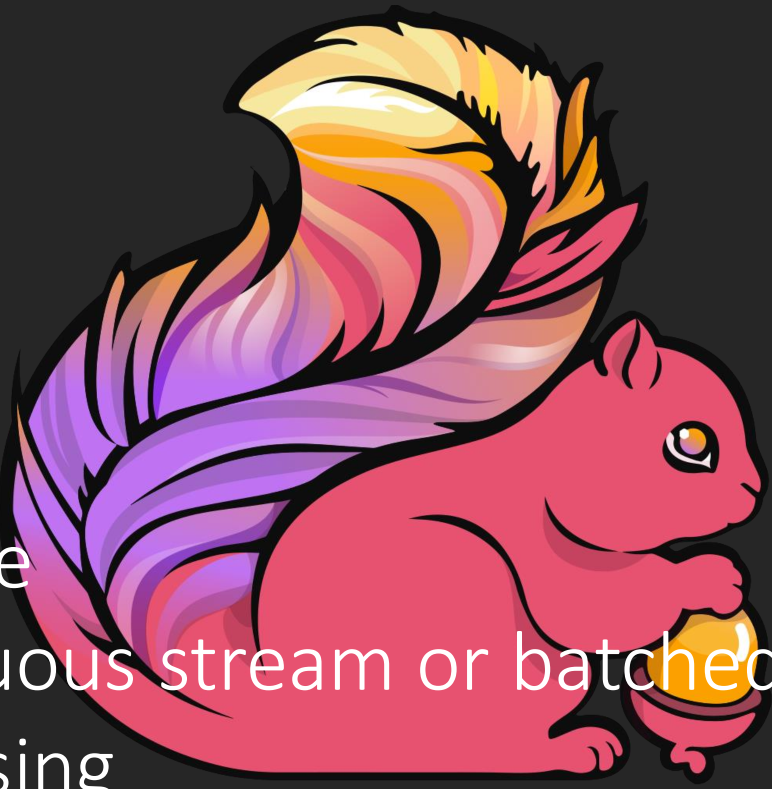
CONCEPTUAL ARCHITECTURE

Group Small

In Todays Presentation

- What is Apache Flink
- Flink Component Breakdown
- Flink Architecture
- Evolution
- Concurrency
- Use Cases
- Derivation Method
- Lessons Learned

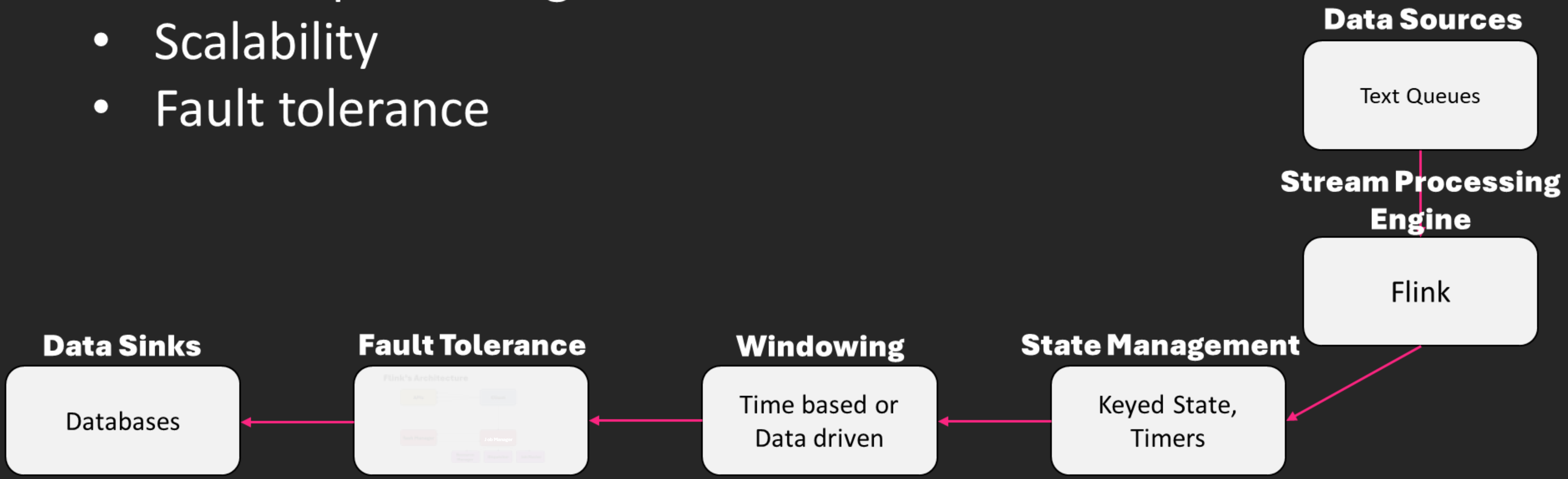
WHAT IS APACHE FLINK?



- Open Source
- Distributed Engine
- Processes continuous stream or batched data
- Real-time processing

Distributed Engine?

- Distributed Stream Processing Engine to be exact!
- Parallel processing
- Scalability
- Fault tolerance



Data Sources

Text Queues

Stream Processing Engine

Flink

State Management

Keyed State, Timers

Windowing

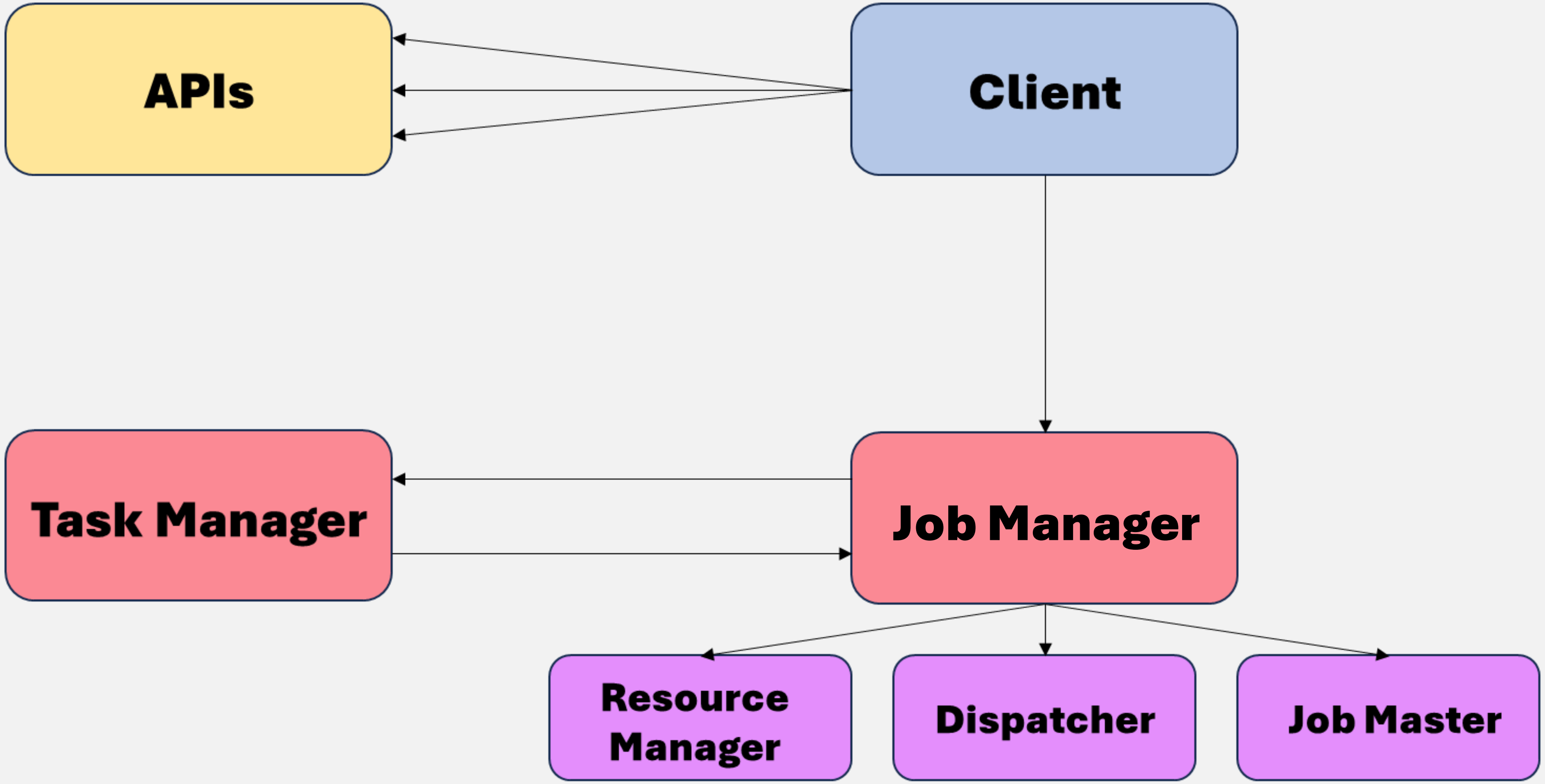
Time based or Data driven

Fault Tolerance

Data Sinks

Databases

Flink's Architecture

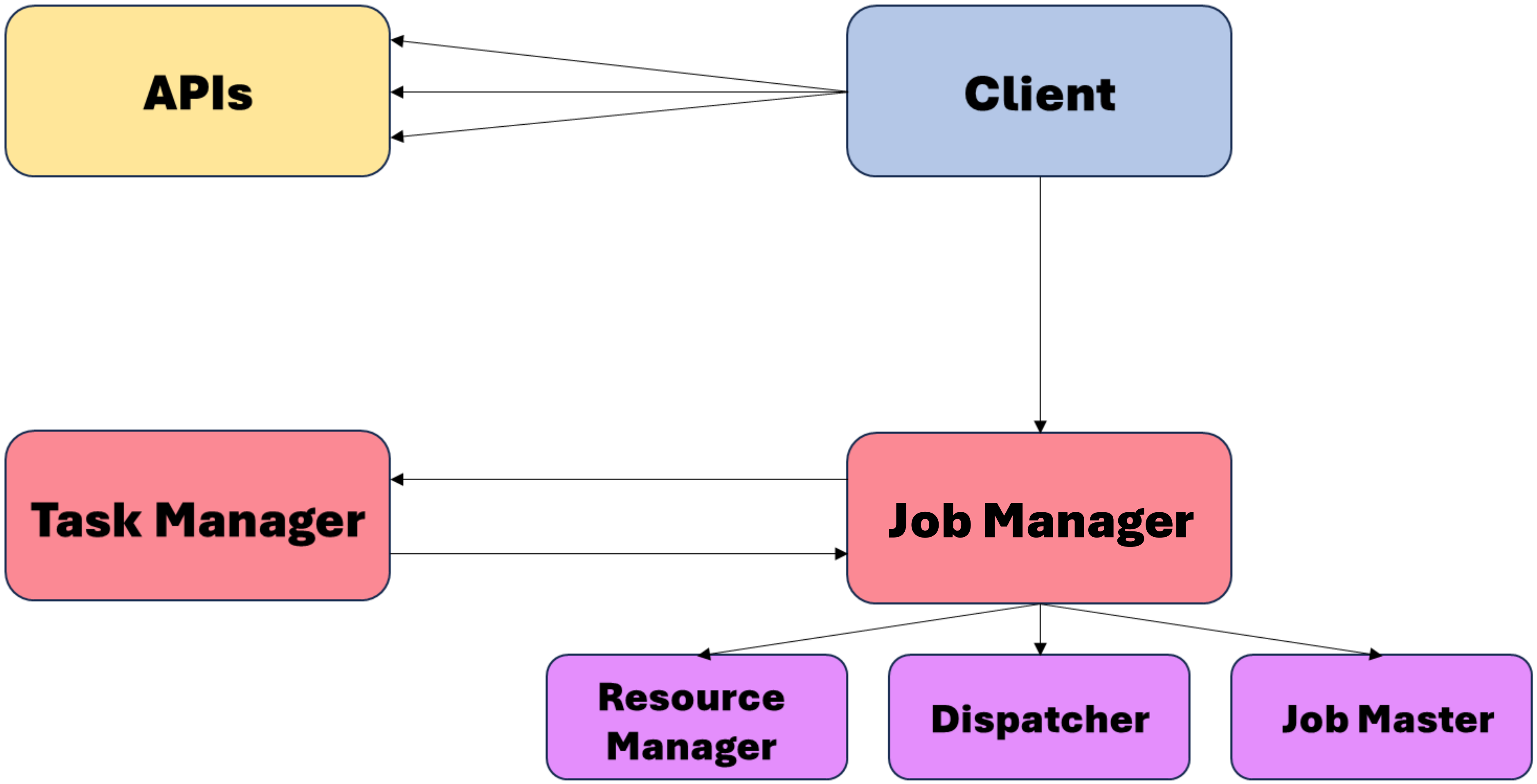


Job Manager

- Coordinating and managing Flink jobs
  - Job submissions
  - schedules tasks
  - Monitors execution

- Can have multiple Job Managers, with one of those managers being the leader.
  - Eliminates single point of failure.

Flink's Architecture



APIs

Client

Task Manager

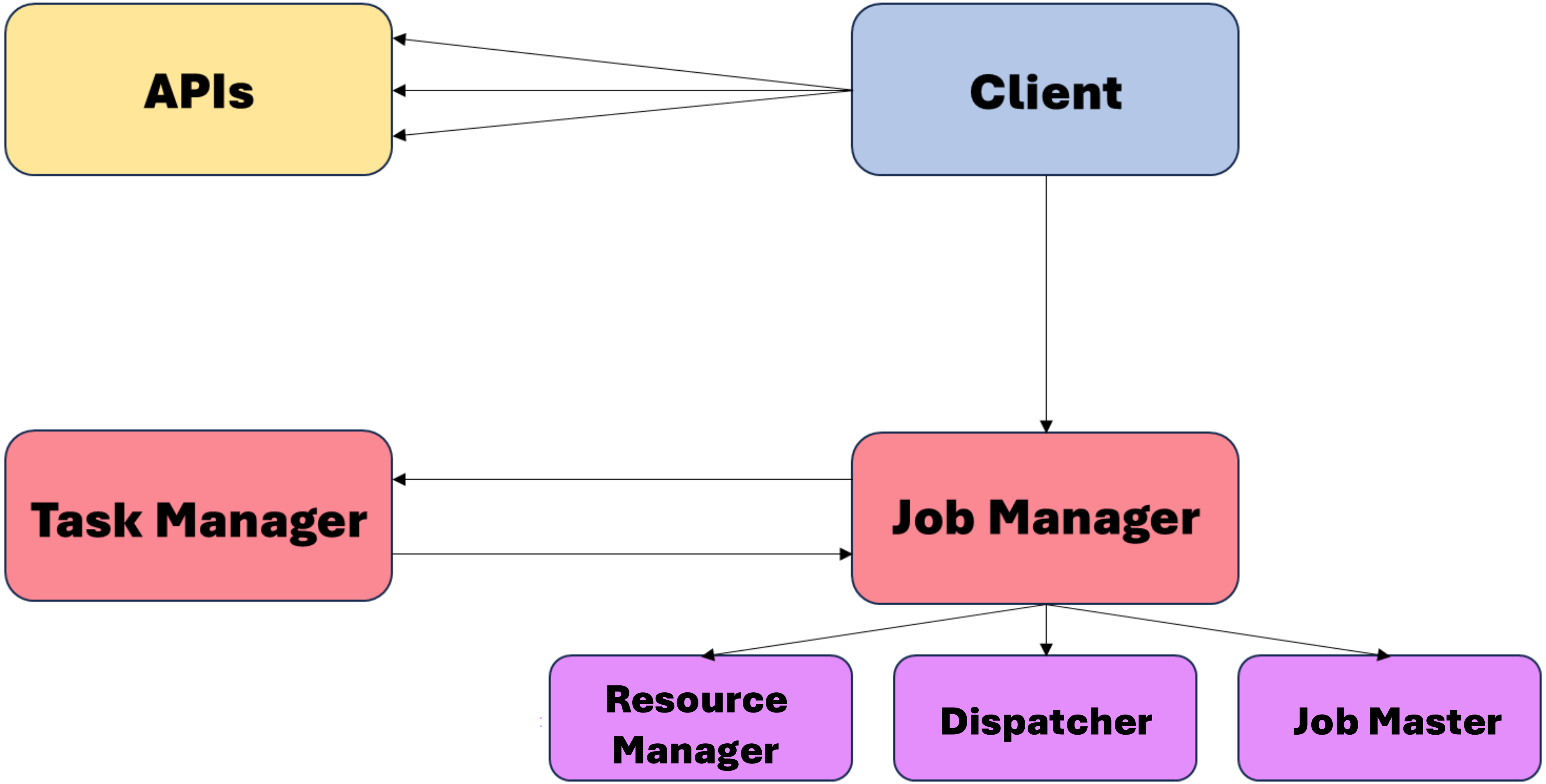
Job Manager

Resource Manager

Dispatcher

Job Master

Flink's Architecture



APIs

Client

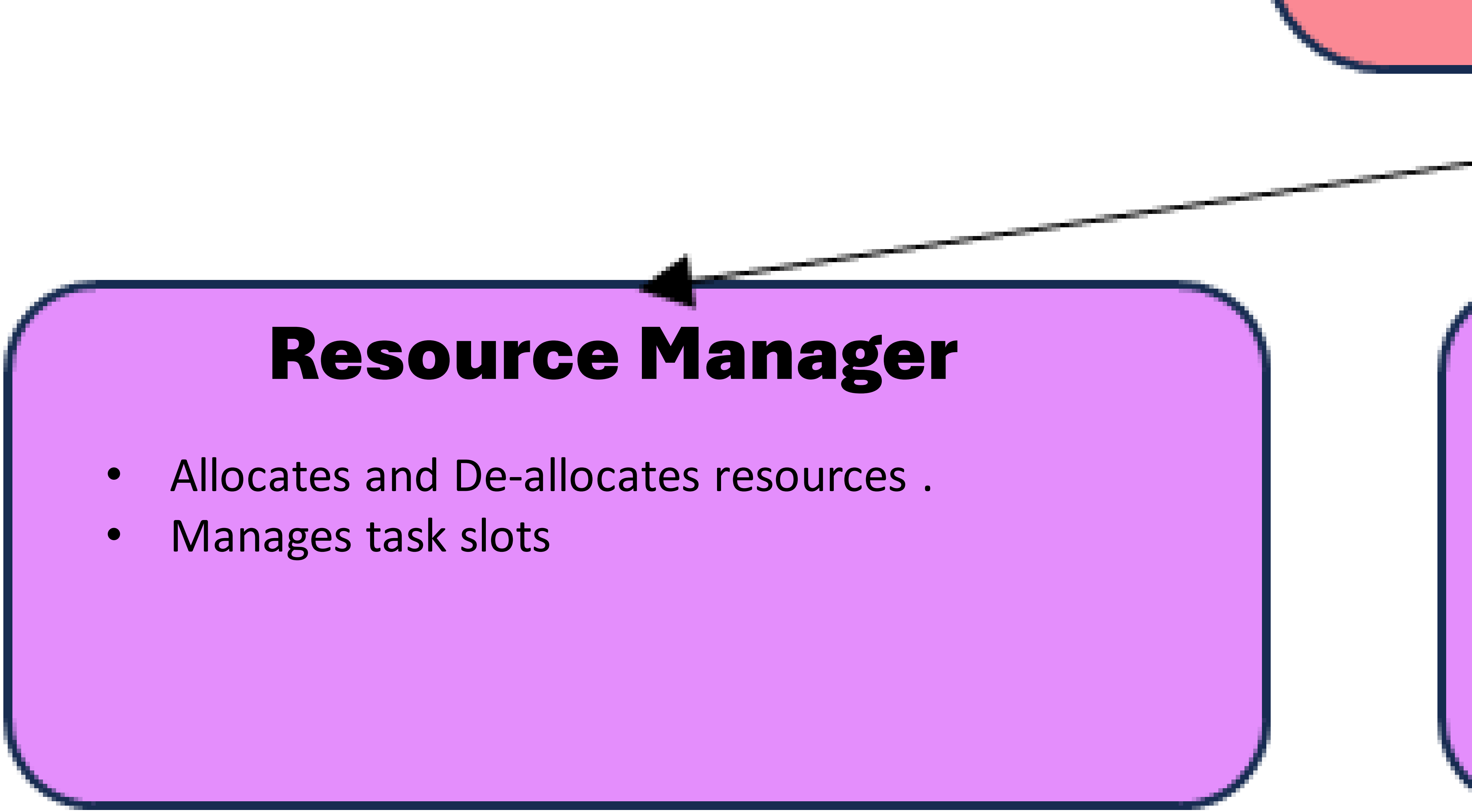
Task Manager

Job Manager

Resource Manager

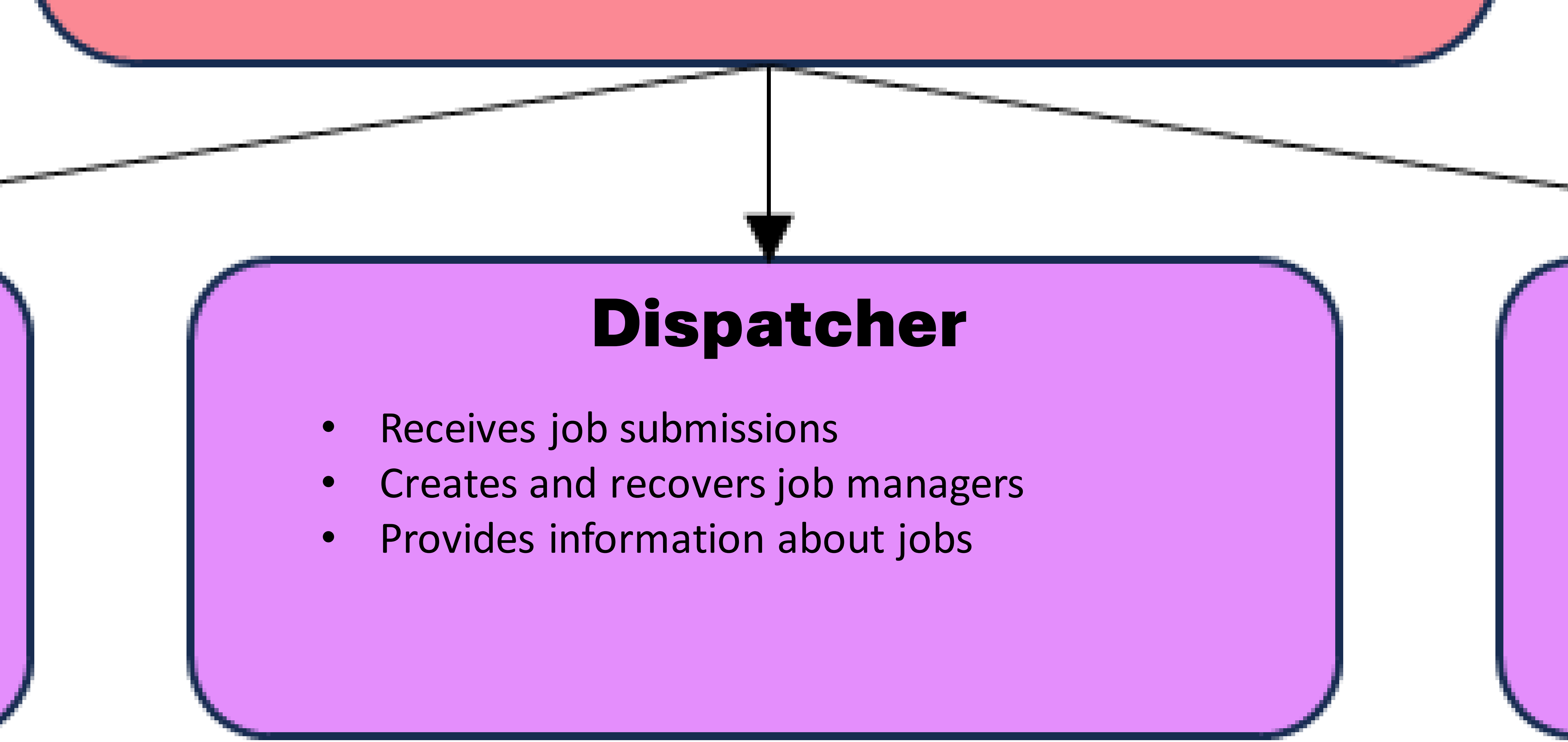
Dispatcher

Job Master



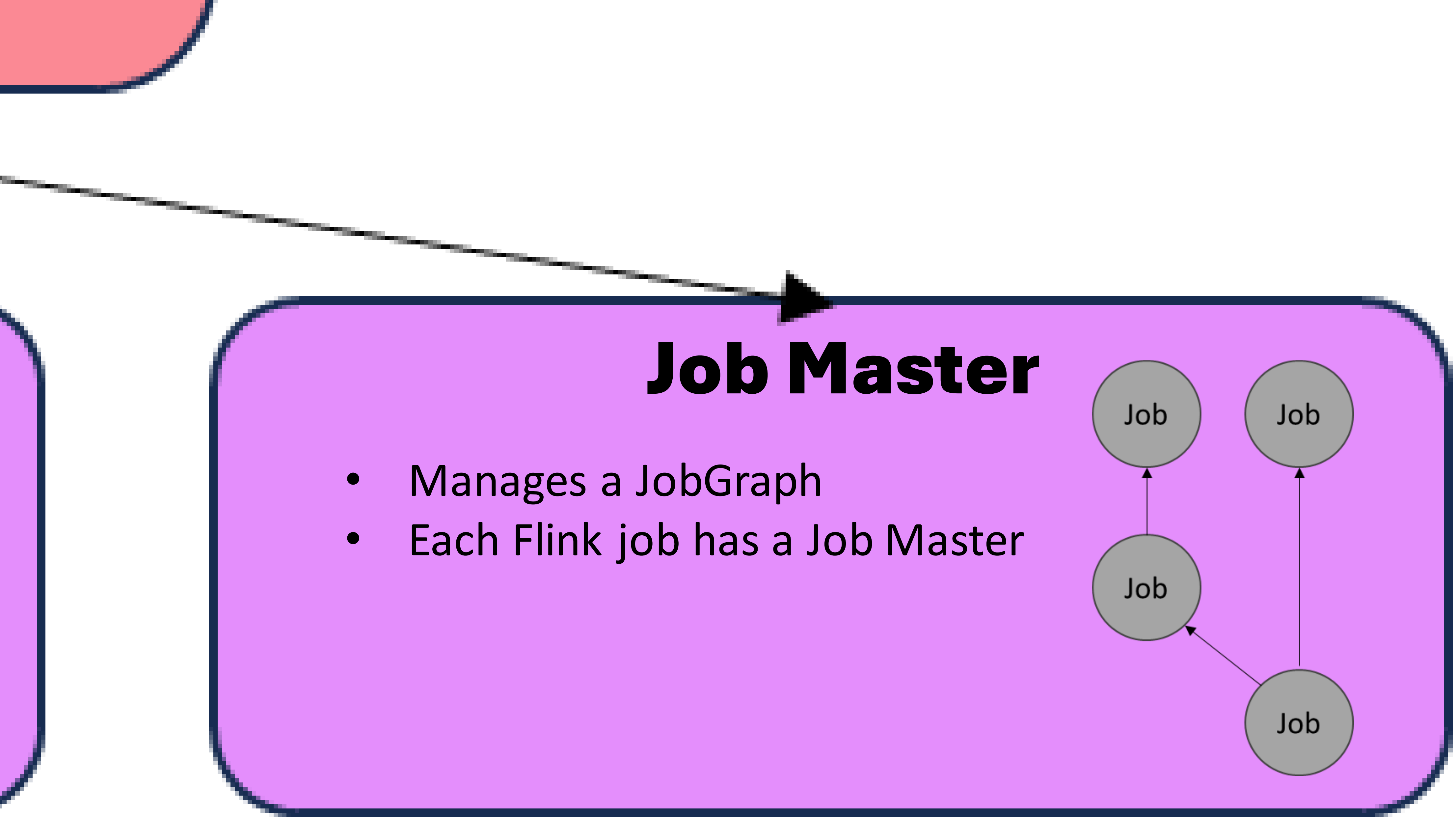
Resource Manager

- Allocates and De-allocates resources .
- Manages task slots



Dispatcher

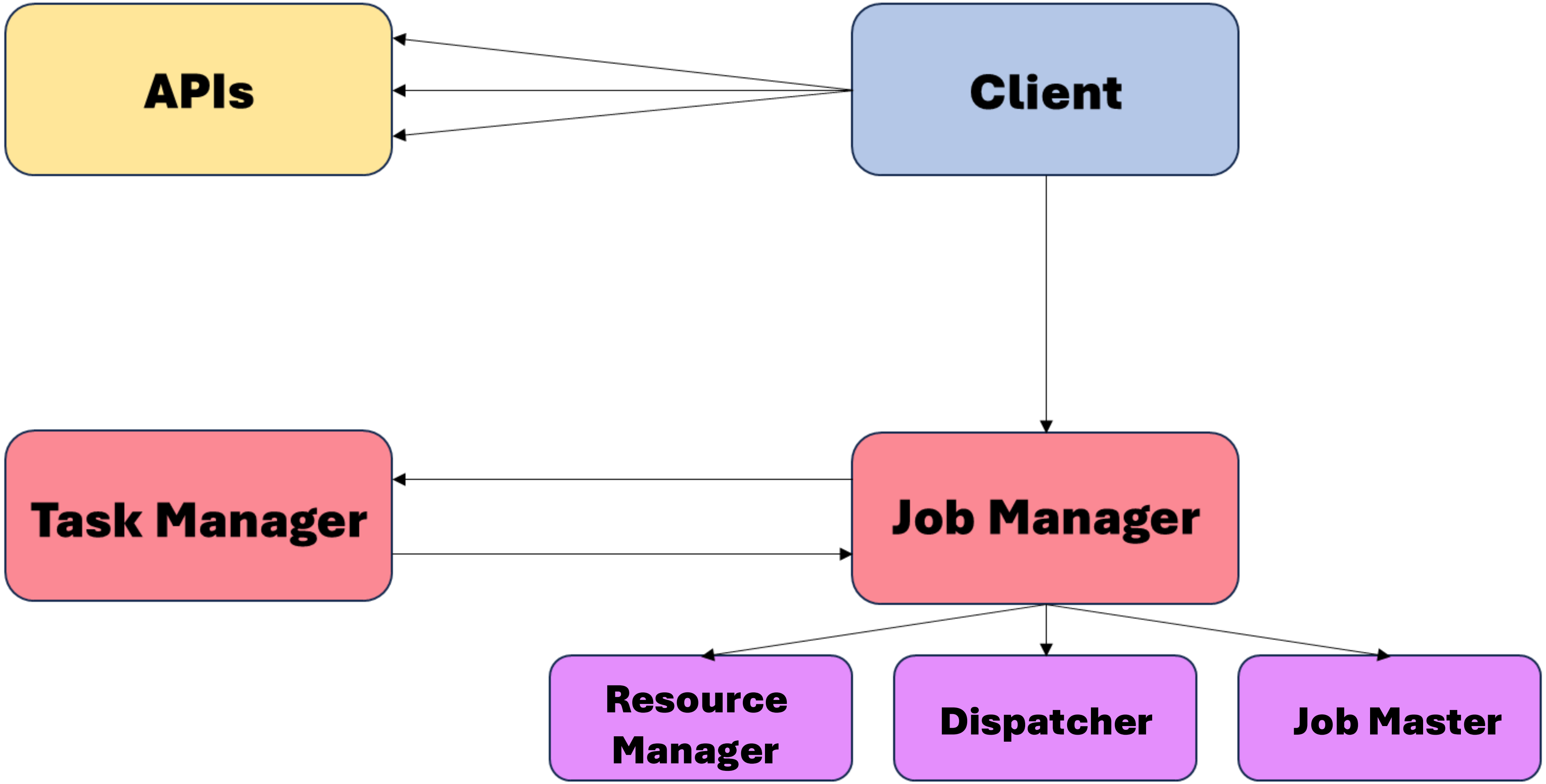
- Receives job submissions
- Creates and recovers job managers
- Provides information about jobs



Job Master

- Manages a JobGraph
- Each Flink job has a Job Master

Flink’s Architecture



APIs

Client

Task Manager

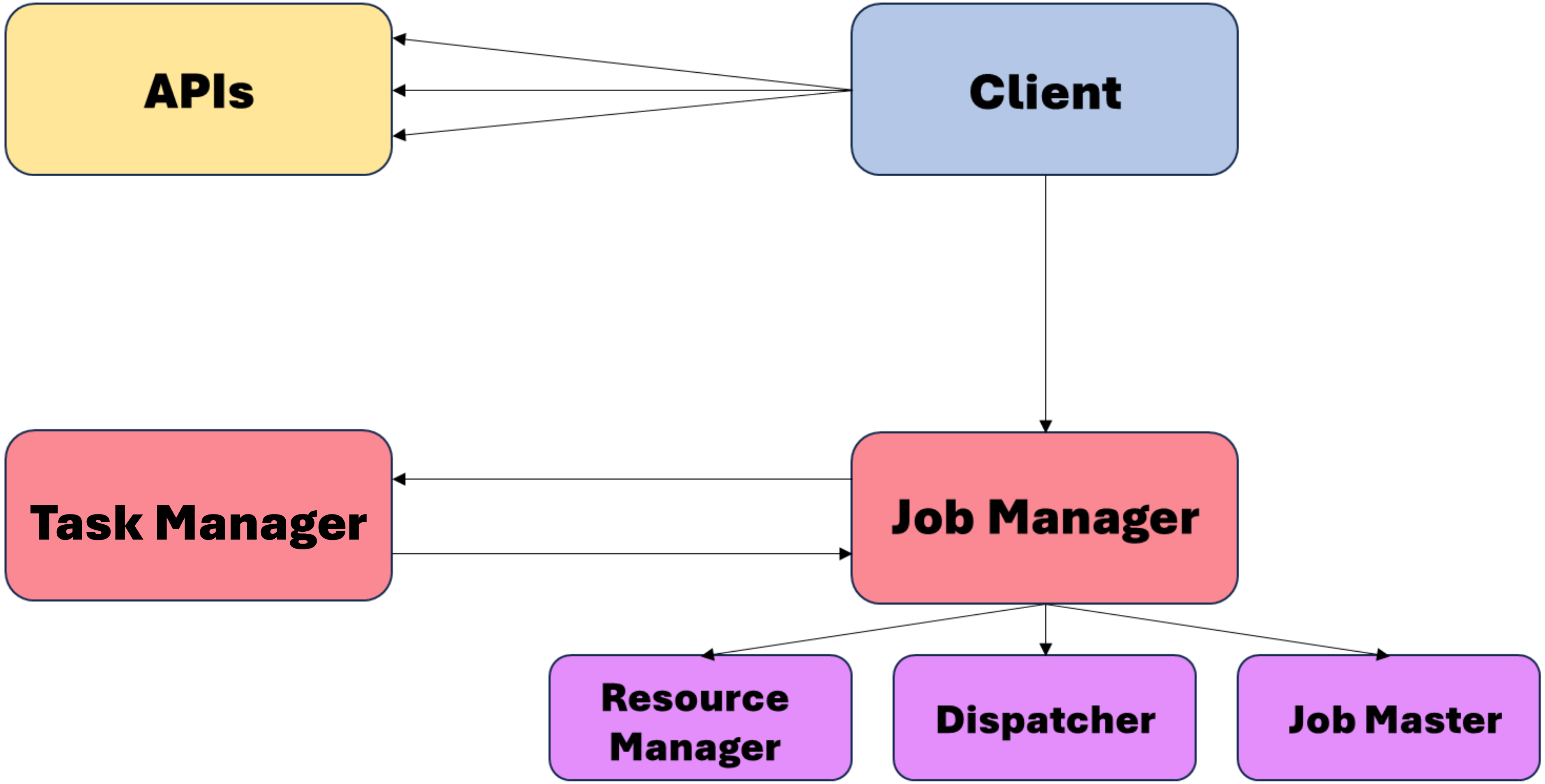
Job Manager

Resource Manager

Dispatcher

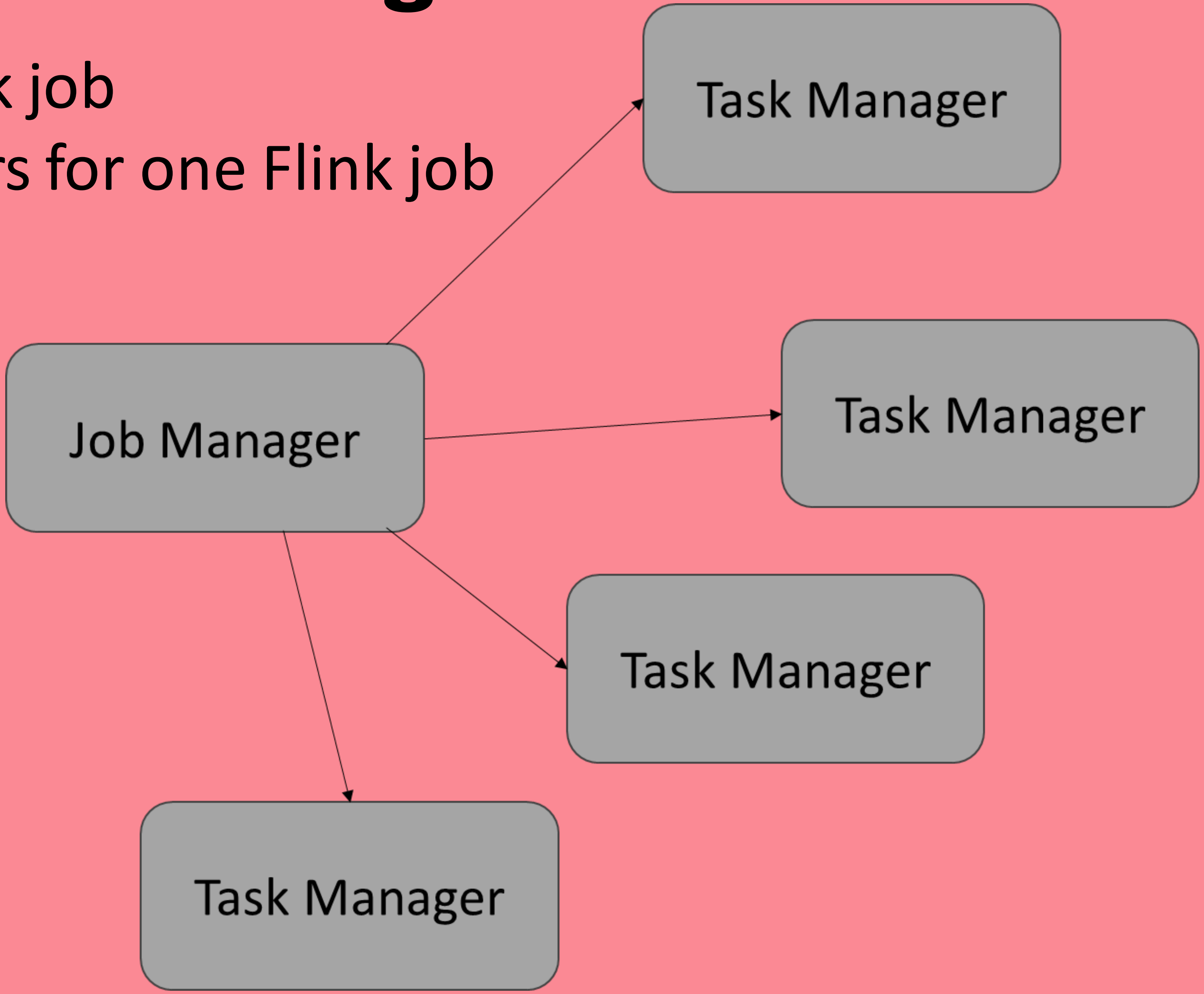
Job Master

Flink’s Architecture

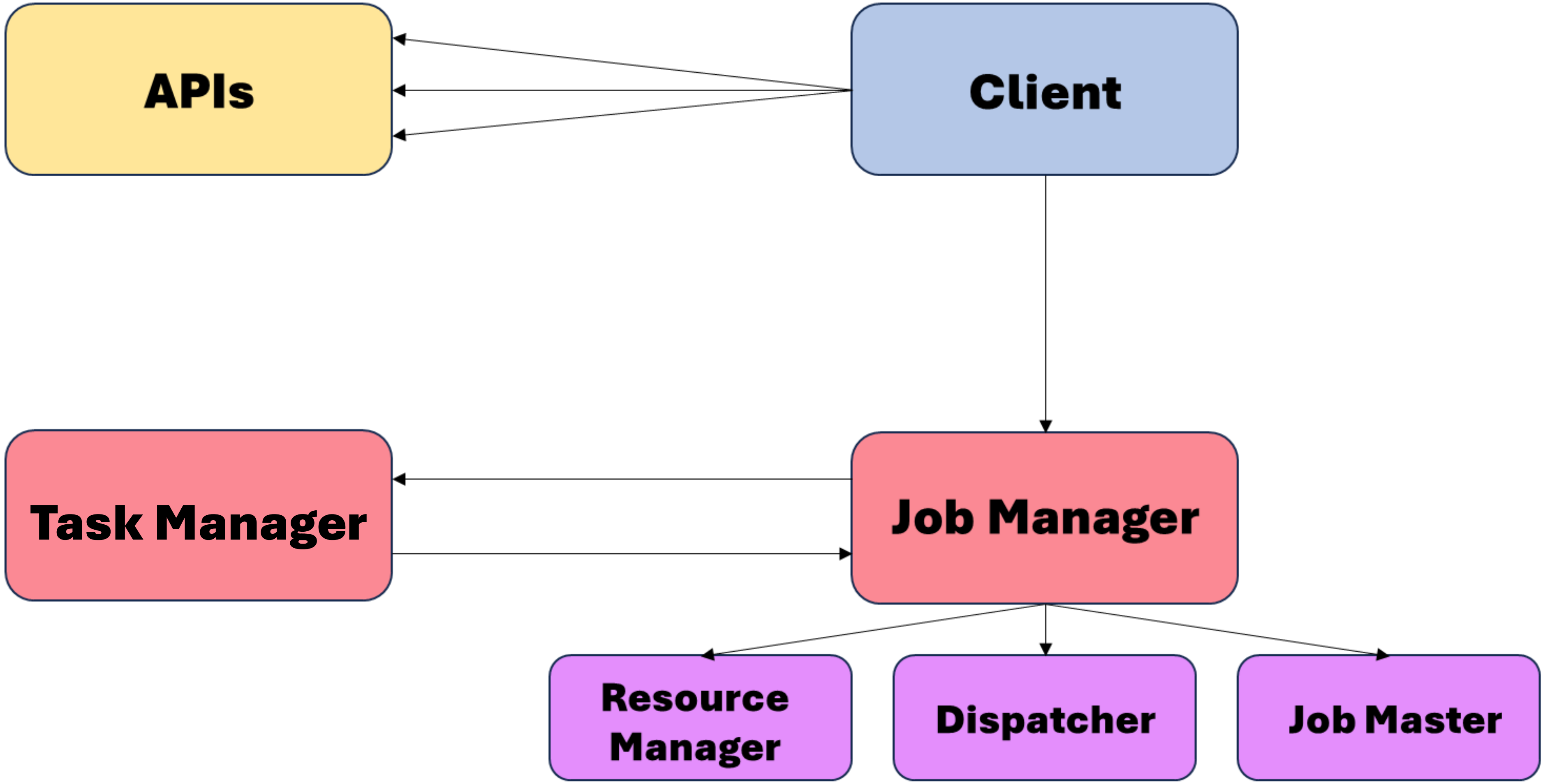


Task Manager

- Execute tasks in a Flink job
- Multiple task managers for one Flink job
- Handle data exchange
- Fault Tolerance



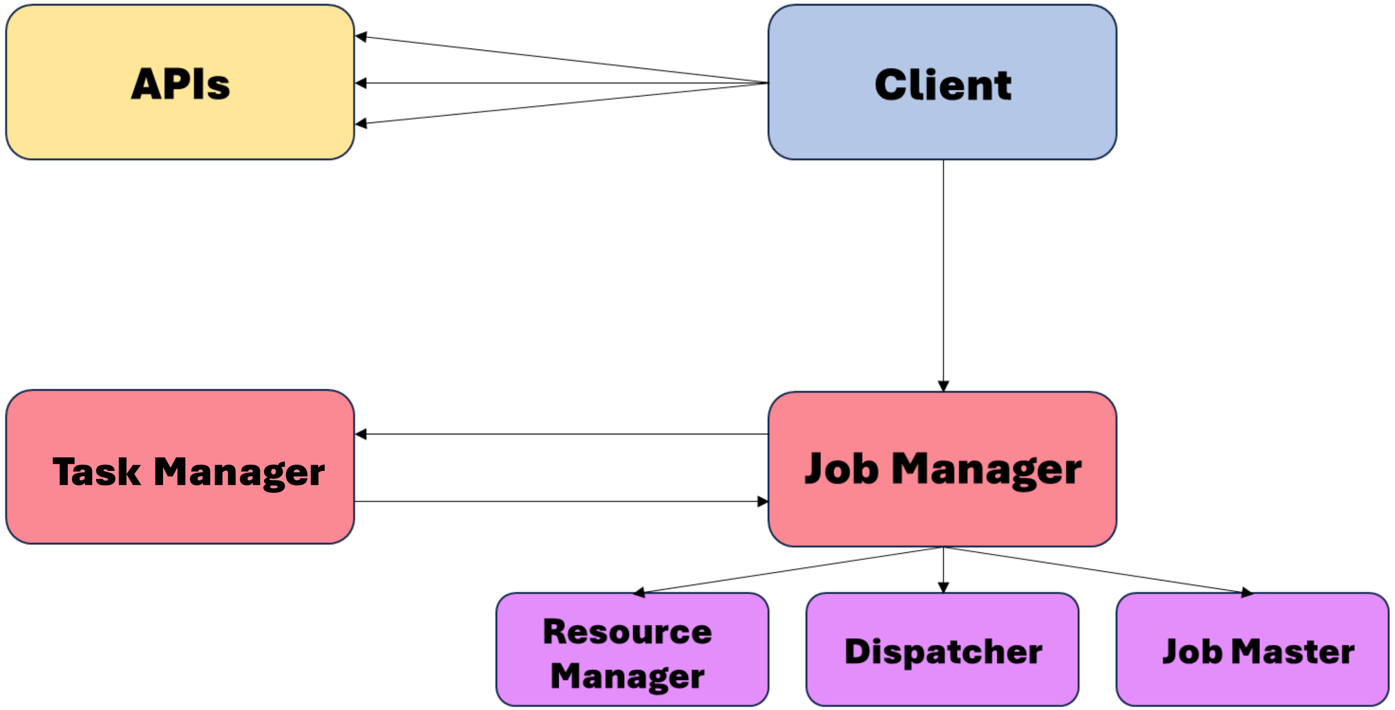
Flink’s Architecture



Architectural Styles

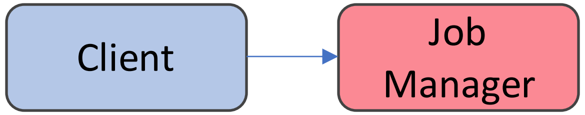
- Client – Server
- Pipe and Filter

Flink's Architecture



Design patterns Used

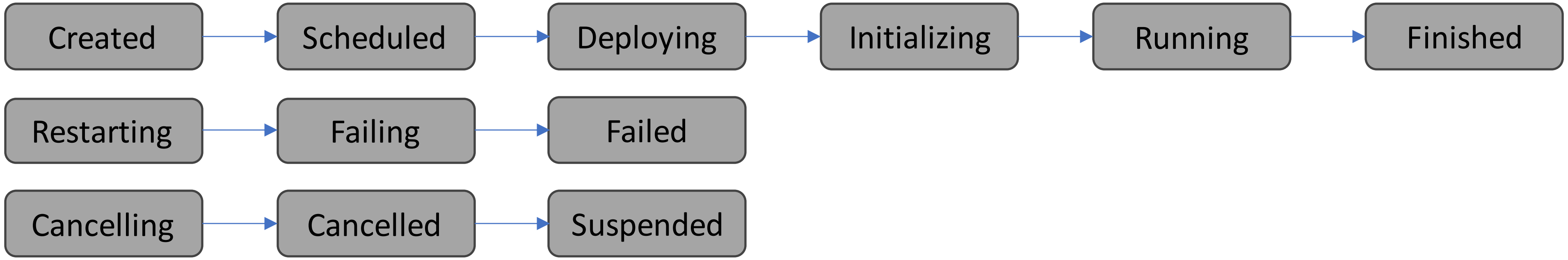
- Façade



Client

Job Manager

- State Design: ExecutionGraph



Created

Scheduled

Deploying

Initializing

Running

Finished

Restarting

Failing

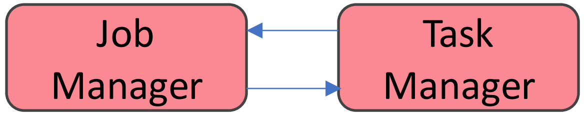
Failed

Cancelling

Cancelled

Suspended

- Master – Slave:



Job Manager

Task Manager

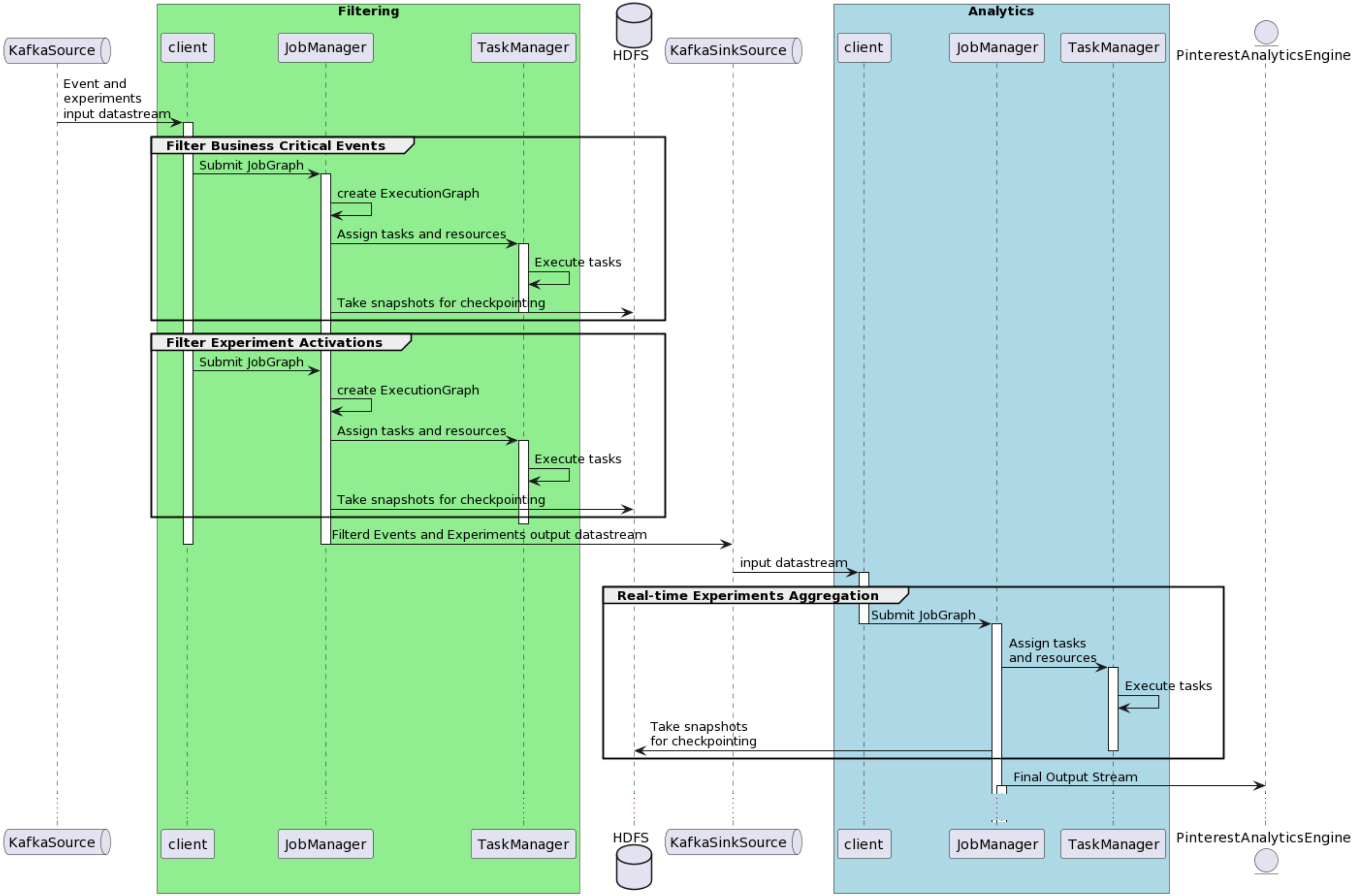
Concurrency

- Parallelism
- Data Partitioning
- Stateful Processing
- Checkpointing

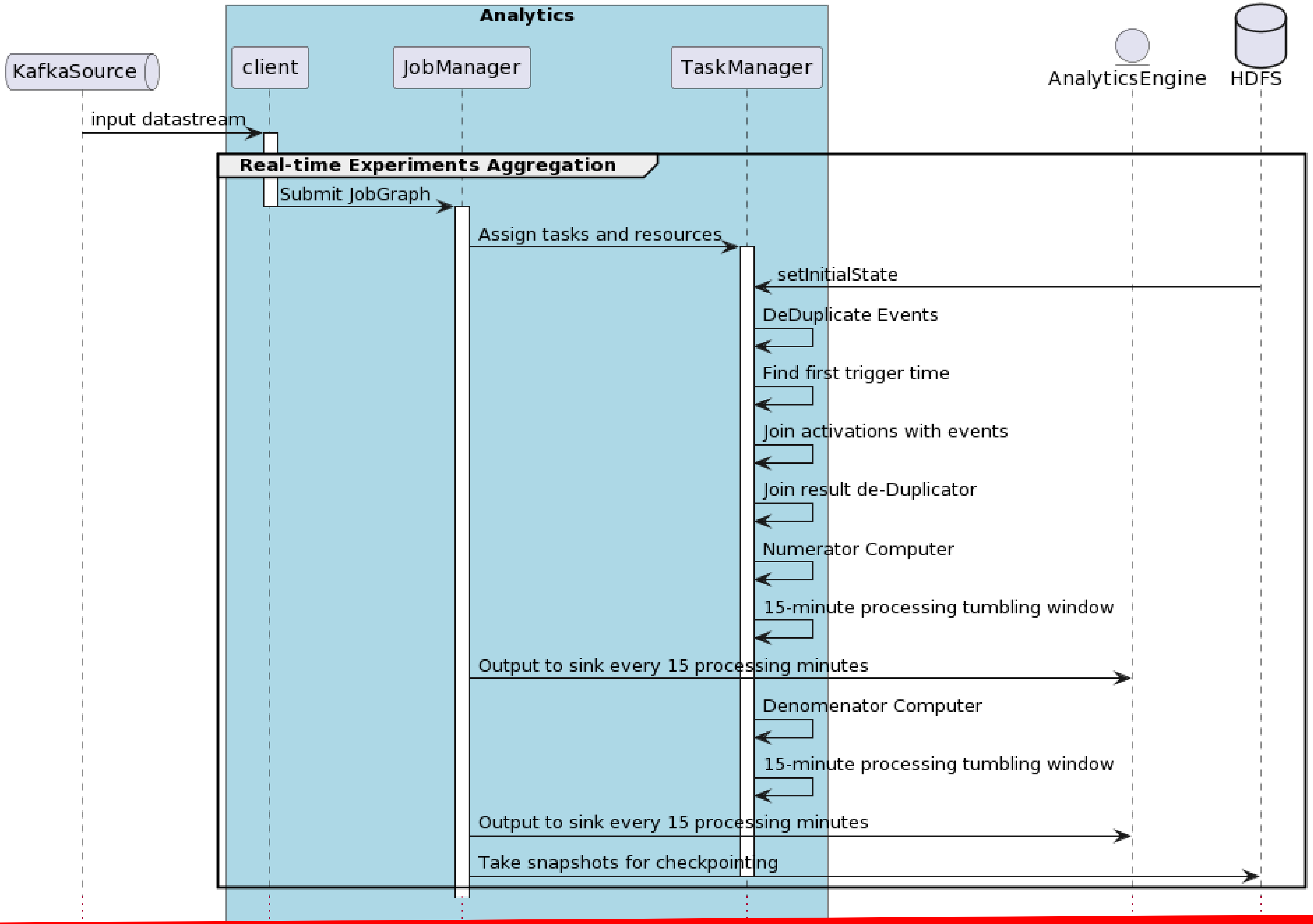
Pinterest Real-Time Experiment Analysis Use Case

- Track progress
- Evaluate performance
- Goals are to catch:
  - Experiments performing poorly
  - Experiments performing well

Sequence Diagram



Sequence Diagram


Analytics
KafkaSource
client
JobManager
TaskManager
AnalyticsEngine
HDFS
input datastream
Real-time Experiments Aggregation
Submit JobGraph
Assign tasks and resources
setInitialState
DeDuplicate Events
Find first trigger time
Join activations with events
Join result de-Duplicator
Numerator Computer
15-minute processing tumbling window
Output to sink every 15 processing minutes
Denomenator Computer
15-minute processing tumbling window
Output to sink every 15 processing minutes
Take snapshots for checkpointing

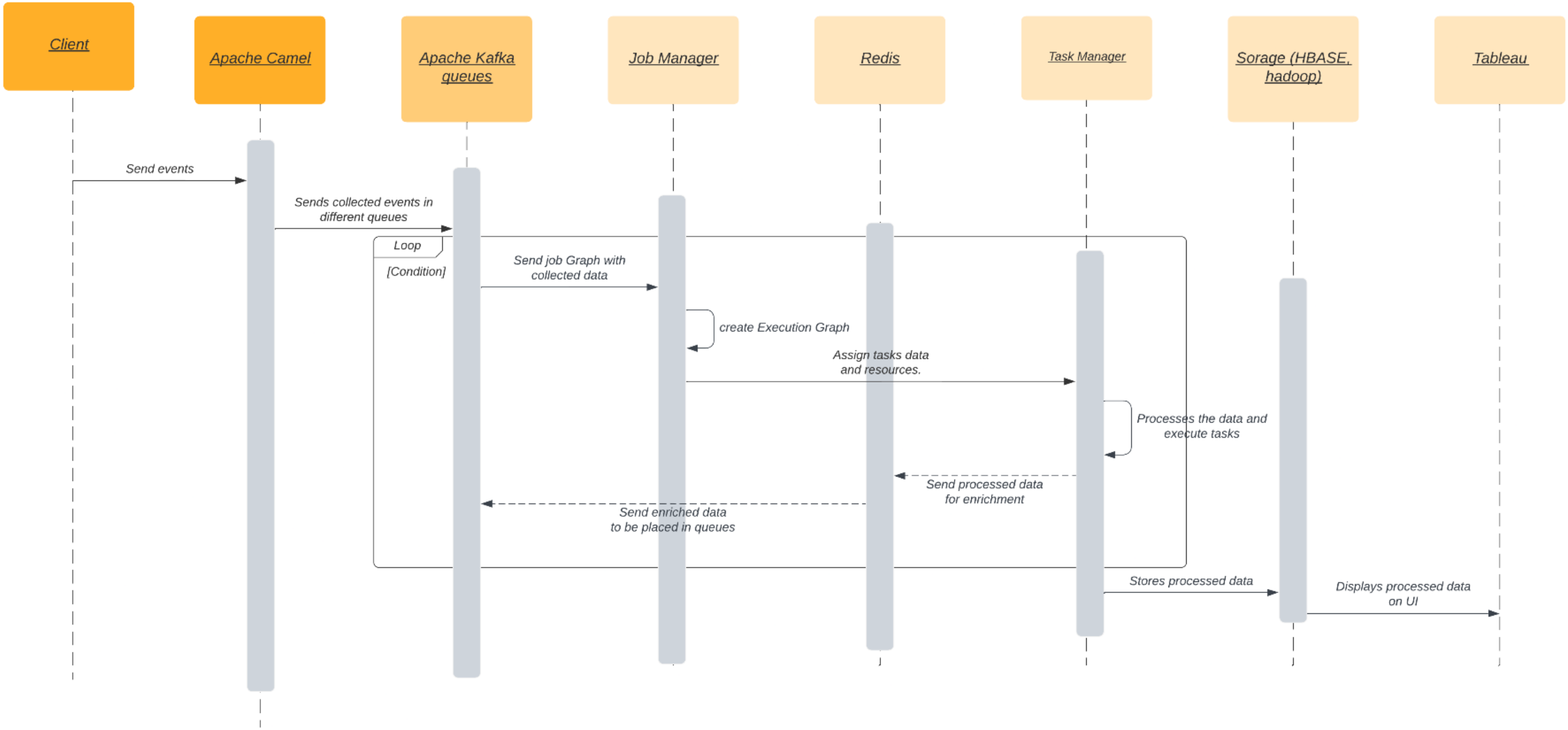
Bouygues Telecom Real Time Monitoring and Analysis

- Identify problems that customers are facing when calling or using their service
- Collect data to analyze the entire traffic
- Detect what is happening on the network in real time

bouygues
TELECOM

Bouygues telecom sequence diagram

Corneille Bobda | October 14, 2023


Client
Apache Camel
Apache Kafka queues
Job Manager
Redis
Task Manager
Sorage (HBASE, hadoop)
Tableau
Send events
Sends collected events in different queues
Loop
[Condition]
Send job Graph with collected data
create Execution Graph
Assign tasks data and resources.
Processes the data and execute tasks
Send processed data for enrichment
Send enriched data to be placed in queues
Stores processed data
Displays processed data on UI

Derivation Method

- Apache Flink Documentation
- Apache Website

Lessons Learned

- Usefulness and Value

- What we wish we did differently
  - Learn as a group before taking roles

QUESTIONS?

THANK YOU