小米流式平台架构演进与实践

The Application and Architecture of the Streaming Computing Platform in Xiao Mi

夏军
小米流式平台负责人

FLINK FORWARD # ASIA
实时即未来 # Real-time Is The Future

FLINK FORWARD

Contents
目录

背景介绍

Background

01

Our vision

为小米各业务线提供流式数据的一体化/平台化解决方案

Building Integrated Streaming Platform for Businesses in Xiao MI

流式数据存储

Streaming Data Storage

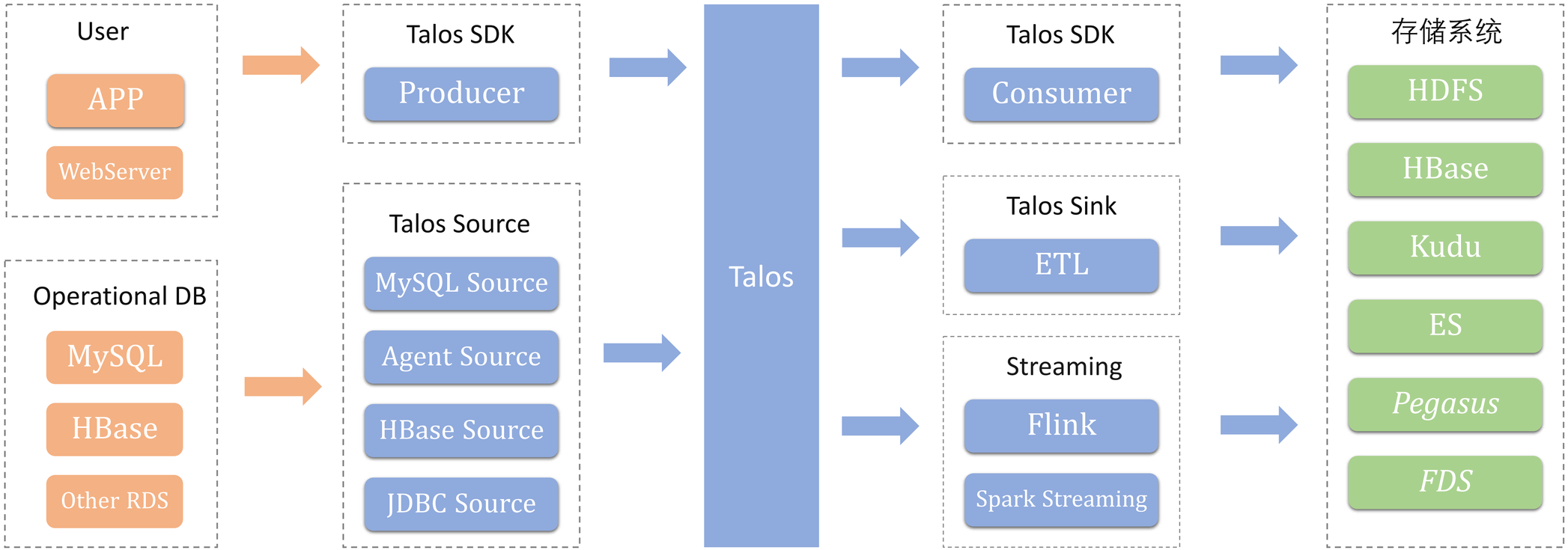
流式数据接入和转储

Streaming Data Access and Transfer

流式数据处理

Streaming Data Process

Architecture of Streaming Platform



Business Scale

小米商城 | 小米金融 | 小爱音箱 | 小米贷款 | 米家生活

MIUI | 小米运动 | 小米音乐 | 多看阅读 | 小米视频

1.2 Trillion
Messages/day

43 Million
Peak Messages/s

1.6 PB
Transfer Bytes/day

1.5 w+
Transfer Jobs

800 +
Streaming job

200 +
Flink job

7000亿
Flink Messages/day

1PB +
Flink Bytes/day

小米流式平台发展历史

The History of Streaming Platform in Xiao Mi

02

History

Streaming Platform 1.0

- Scribe
- Kafka
- Storm

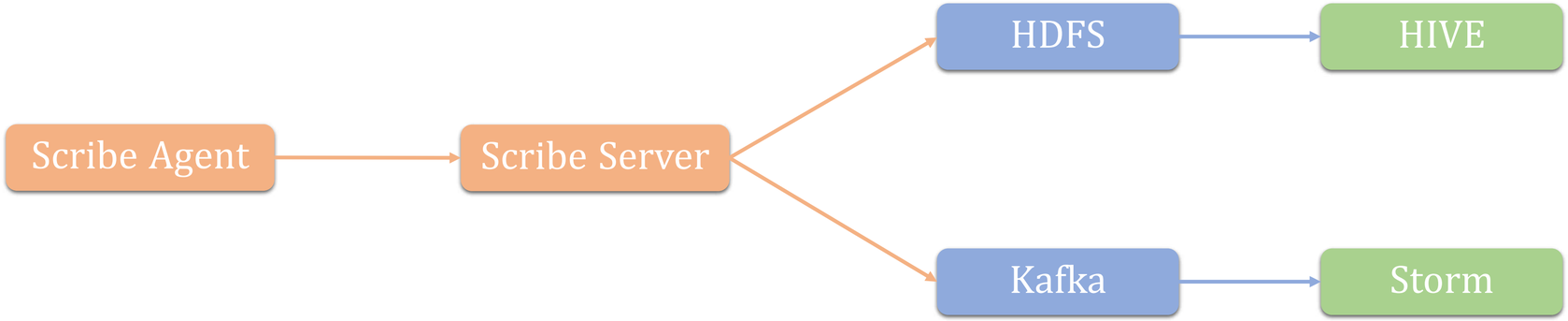
Streaming Platform 2.0

- Talos
- Talos Source
- Talos Sink
- Spark Streaming

Streaming Platform 3.0

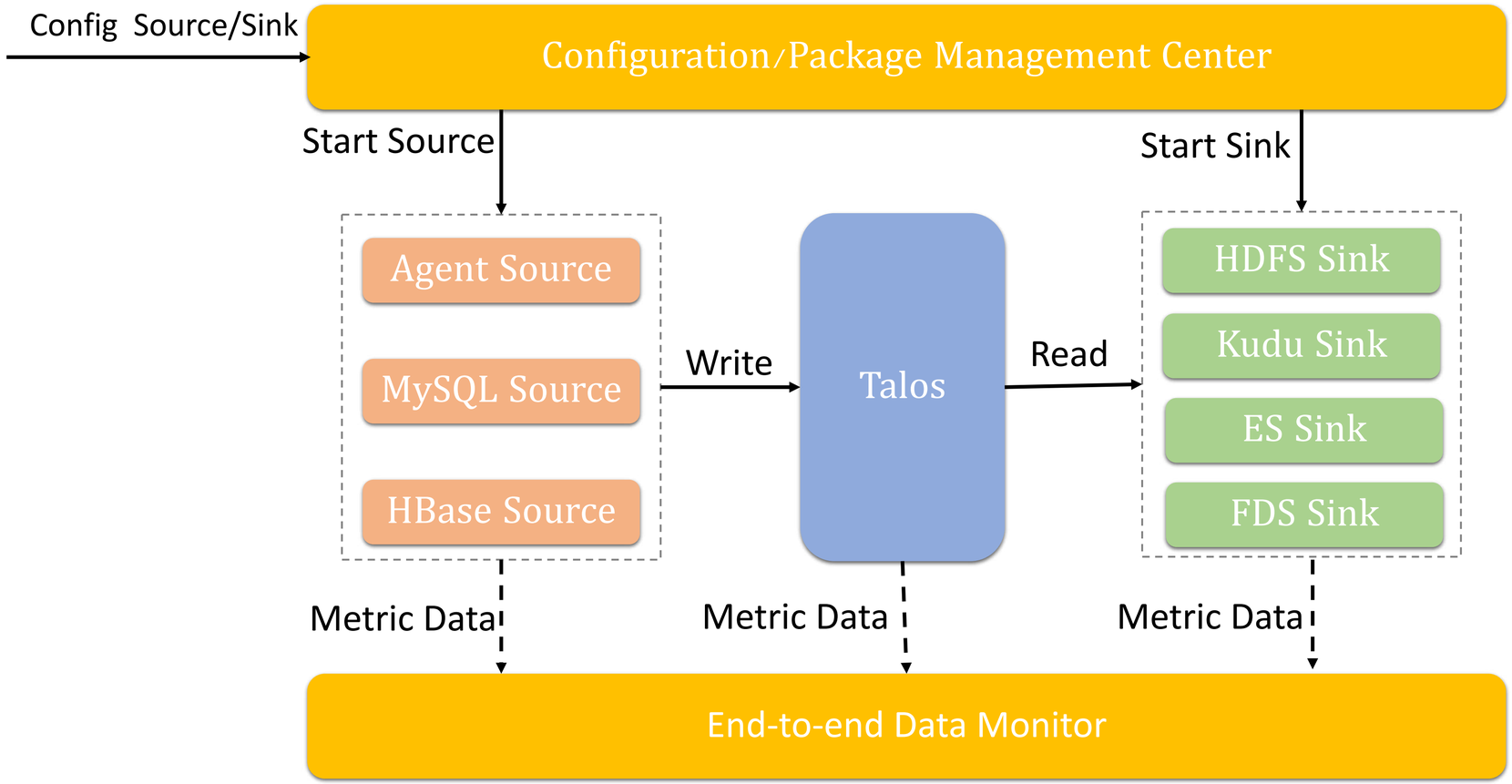
- Talos
- Talos Source
- Talos Sink
- Spark Streaming
- Talos Schema
- Flink
- Stream SQL

Streaming Platform 1.0



- 配置和包管理机制缺乏，维护成本较高
  Lack of configuration and package management mechanisms, high maintenance costs.
- Push模式架构，异常情况无法有效缓存数据，同时HDFS/Kafka 数据相互影响
  Push Mode architecture, abnormal conditions can not effectively cache data, while HDFS/Kafka data interacts.
- 全链路数据黑盒，缺乏监控和数据检验机制
  No metrics in the full pipeline, lack of monitoring and data verification mechanism.

Streaming Platform 2.0



Improvements

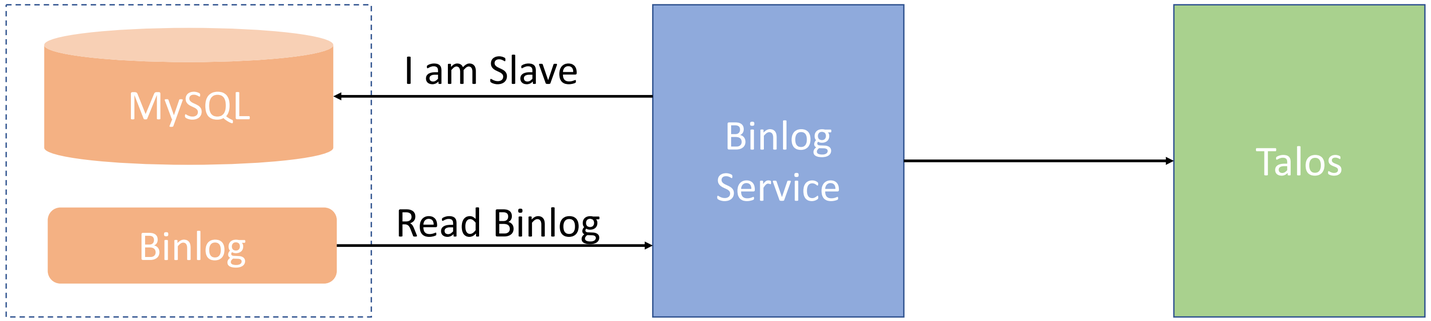
- Multi Source& Multi Sink：将系统集成复杂度由O（M*N）降低为O（M+N）
  Multi Source& Multi Sink：Reduce system integration complexity form O(M*N) to O(M+N).

- 引入Configuration 和 Package中心化管理机制，彻底解决升级，修改，上线等一系列问题
  Introduce Configuration and Package management mechanism to solve problems such as upgrade, modification and online.

- 端到端数据监控机制，实现全链路数据监控，量化全链路数据质量
  End-to-end data monitoring mechanism to achieve full pipeline alert and quantify full pipeline data quality.

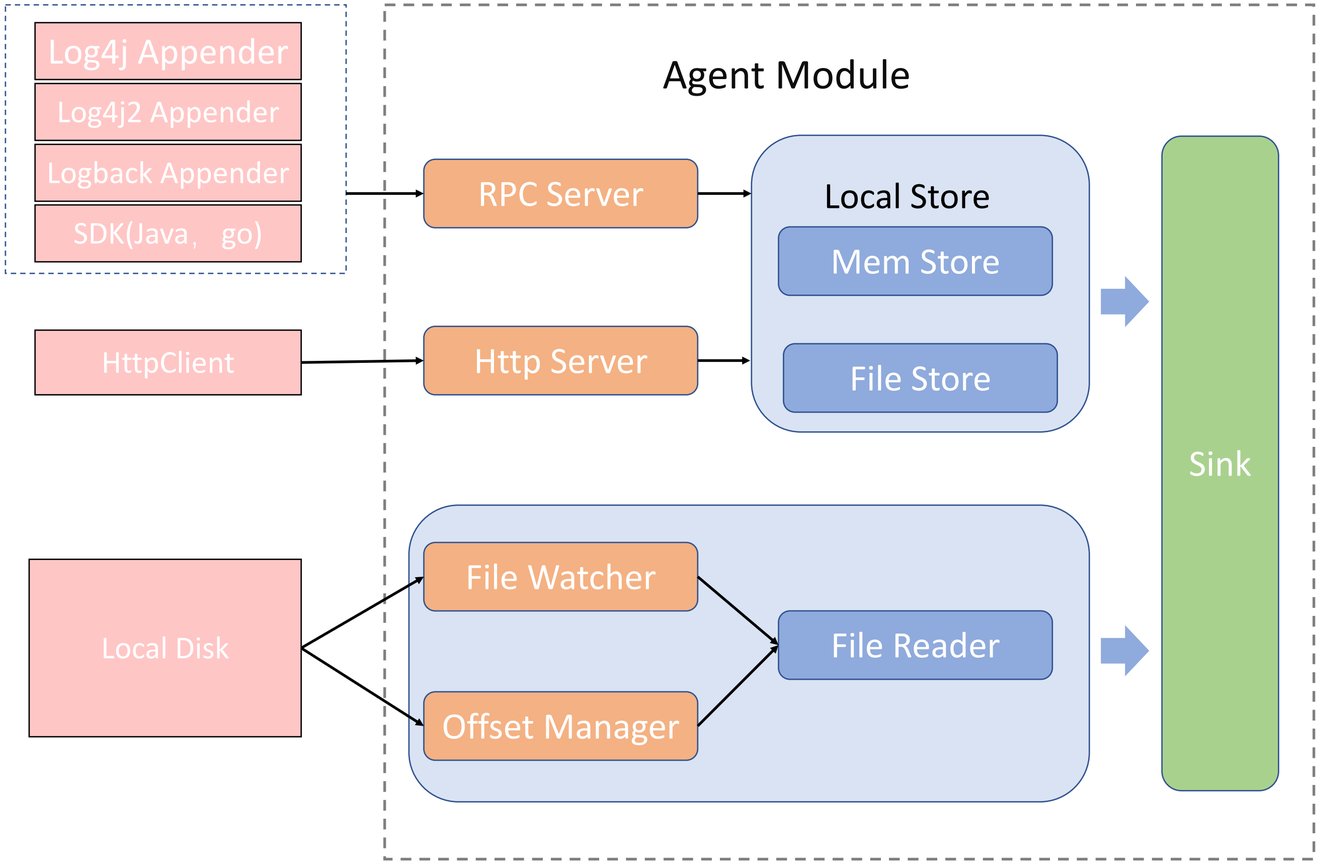
- 产品化解决方案，避免重复建设，解决业务运维问题
  Product solutions to avoid redundant construction and solve business operation and maintenance problems.

MySQL Source



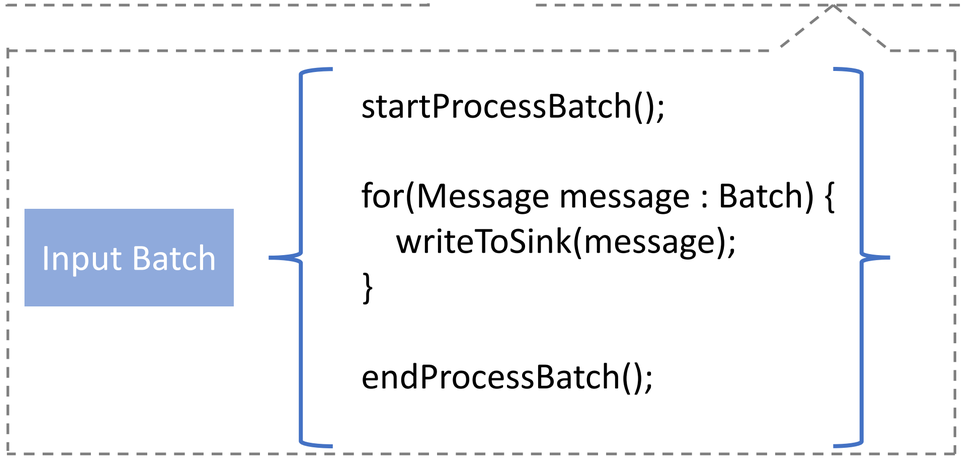
- **Binlog Service伪装成MySQL Slave，向MySQL 发送Dump binlog请求**
  Binlog Service masquerades as MySQL slave, sending Dump binlog requests to MySQL.
- **MySQL 收到Dump 请求，开始推动Binlog给Binlog Service**
  MySQL receives the Dump request and starts pushing binlog to the Binlog Service.
- **Binlog Service将binlog 以严格有序的形式转储到Talos**
  Binlog Service dumps the binlog in a strictly ordered form to Talos.

Agent Source



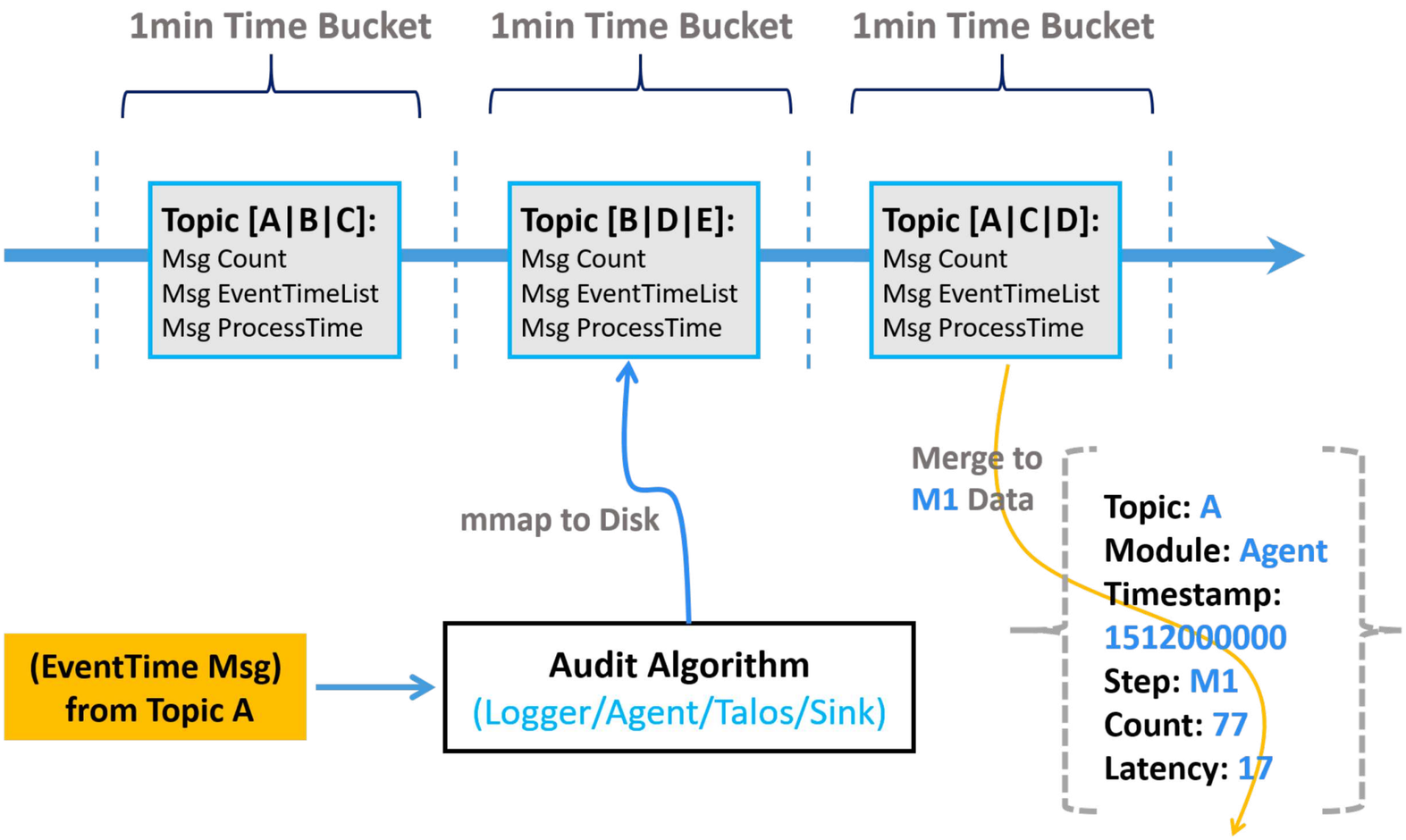
- **全场景覆盖：Rpc，Http，File**
  Full scene coverage：rpc，http and file
- **内存+文件双缓存**
  Double cache of memory and disk
- **动态监控文件，自动记录offset**
  Dynamically monitor files and record offsets
- **K8S环境深度整合**
  Deep integration of k8s environment

Talos Sink



- **抽象公共逻辑，不同Sink只需要实现Write逻辑**
  Abstract public logic, different sink only need to implement write logic
- **不同的Sink独立为不同的Job，避免相互影响**
  Different sinks are independent jobs, avoiding mutual influence
- **依赖Topic流量进行动态资源调度，资源使用最优化**
  Dynamic resource schedule based on topic traffic for resource optimization.

End-to-end Data Monitor



Problems

- Talos数据缺乏Schema管理
  Lack of schema management.

- Talos Sink模块不支持定制化需求，例如实现业务特定ETL操作
  Talos Sink do not support custom requirements, such as implementing business-specific ETL operations.

- Spark Streaming自身问题：不支持Event Time，端到端Exactly Once语义
  Talos Sink do not support custom requirements, such as implementing business-specific ETL operations.

基于Flink的实时数仓

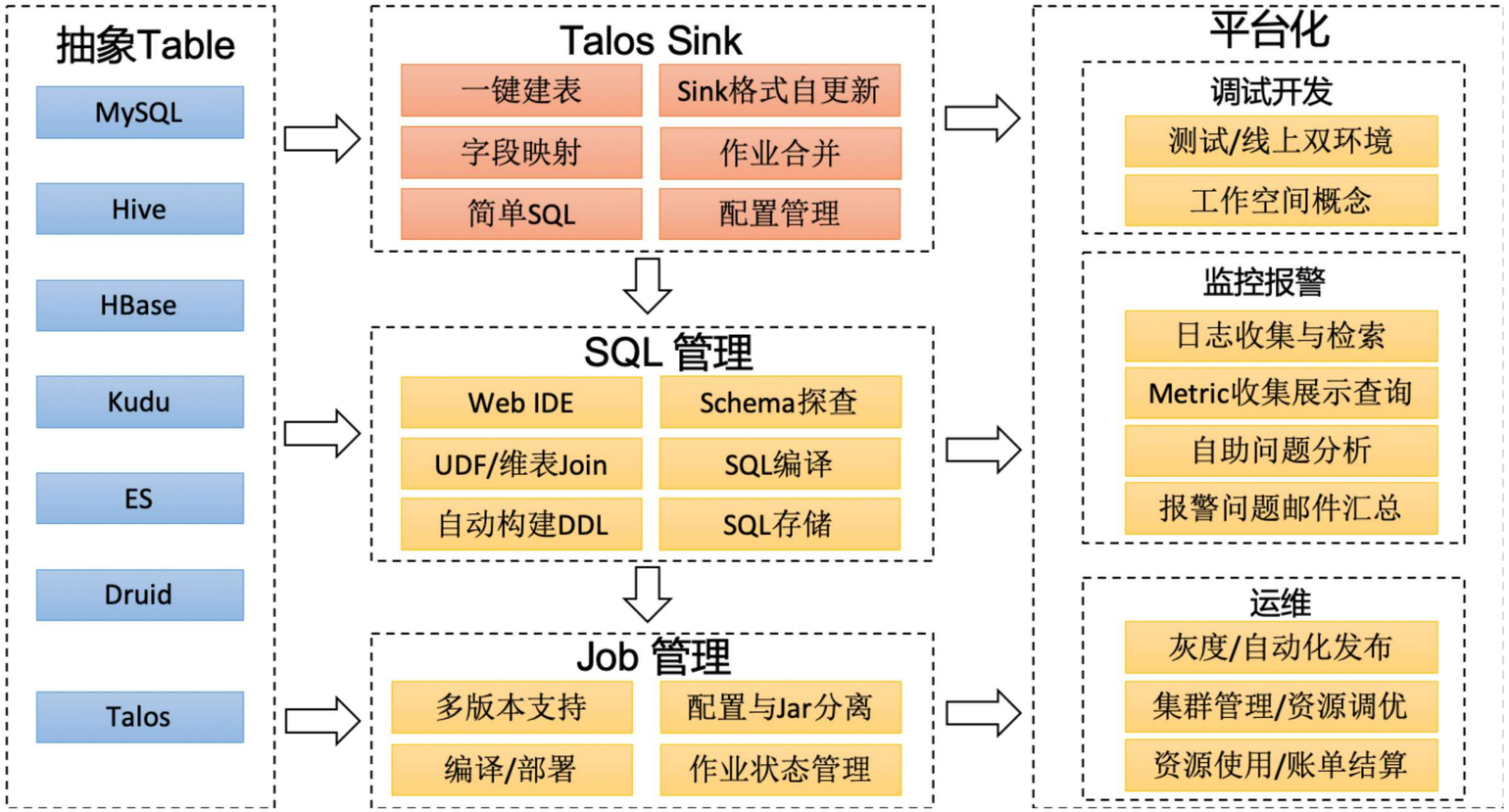
Real-time Data Warehouse Based on Flink

03

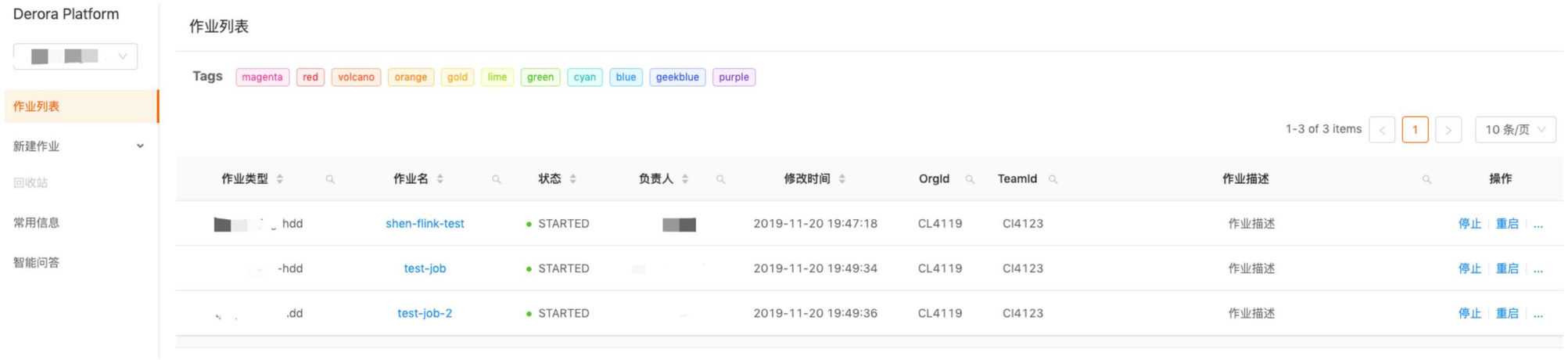
Design Philosophy

- 全链路Schema支持，实现数据校验，字段变更，兼容性检查
  Add Schema support for the whole pipeline, support data validation, field changes, compatibility checks.
- 全面推进Flink在小米的落地，大力推进Streaming SQL
  Fully promote Flink in Xiao MI, vigorously promote Streaming SQL.
- Streaming产品化，实现Streaming Job 和 Streaming SQL 的平台化管理
  Filly implement stream productization and realize platform management of Streaming Job and Streaming SQL.
- 基于Flink SQL 改造Talos Sink，支持业务逻辑定制化
  Transform Talos Sink based on Flink SQL, support business logic customization.

Architecture



Job Management



- **Job全生命周期管理，Job 权限管理，Job 标签管理等**
  Job lifecycle management, Job acl management, job tag management, etc.

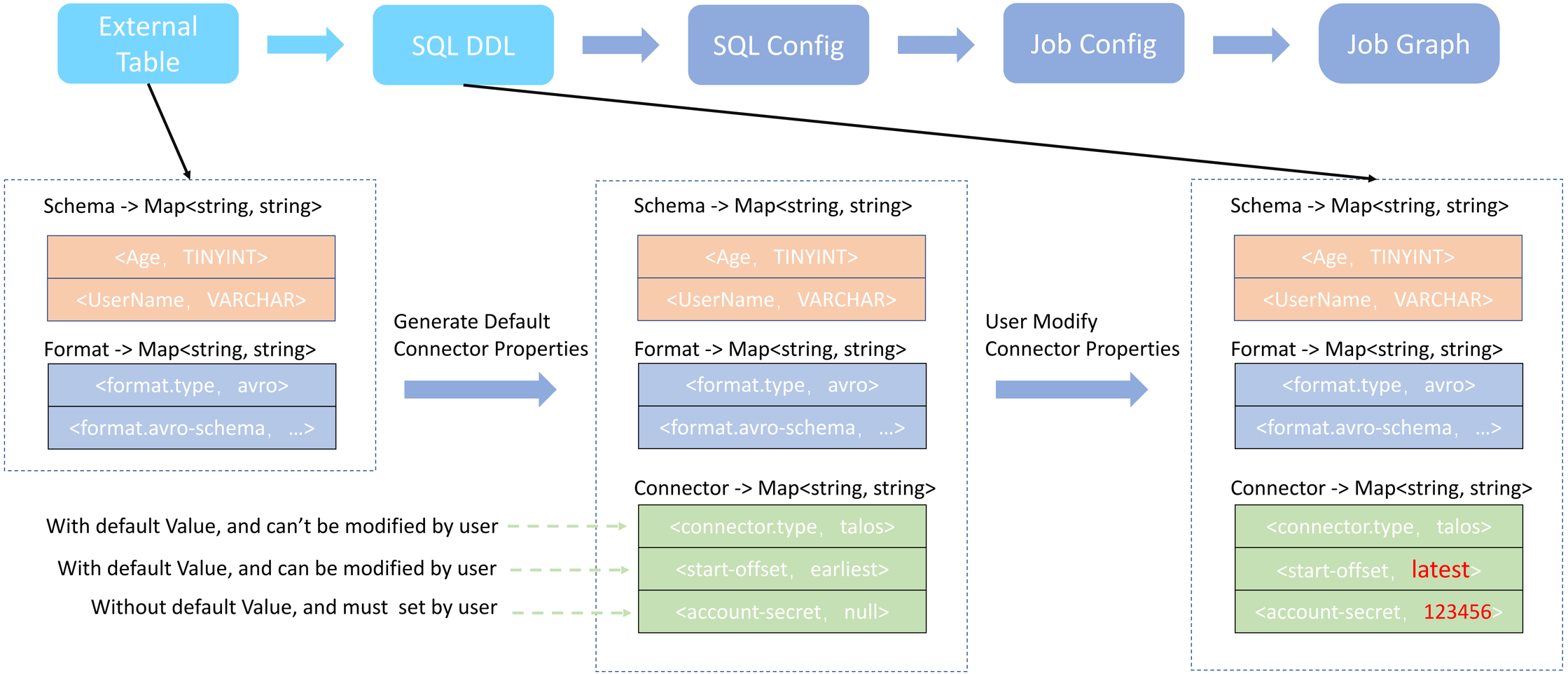
- **Job运行历史展示，方便追溯**
  Display job running history for trace.

- **Job状态与延迟监控，失败作业自动拉起**
  Monitor job status and processing delay, automatically restart the failed job.

SQL Management

- 将外部表描述为SQL DDL，主要包含Table Schema，Table Format，Connector Properties
  Describe the external table as SQL DDL, including Table Schema, Table Format, Connector properties.
- Source DDL + Sink DDL + SQL DML = SQL Config
  Source DDL + Sink DDL + SQL DML = SQL Config
- 将SQL Config转换成 Job Config，即转换为Stream Job的表现形式
  Convert SQL Config to Job Config, which is converted to the representation of Stream Job.
- 将Job Config转换为JobGraph，用于提交Flink Job
  Convert Job Config to Job Graph for submit flink job.

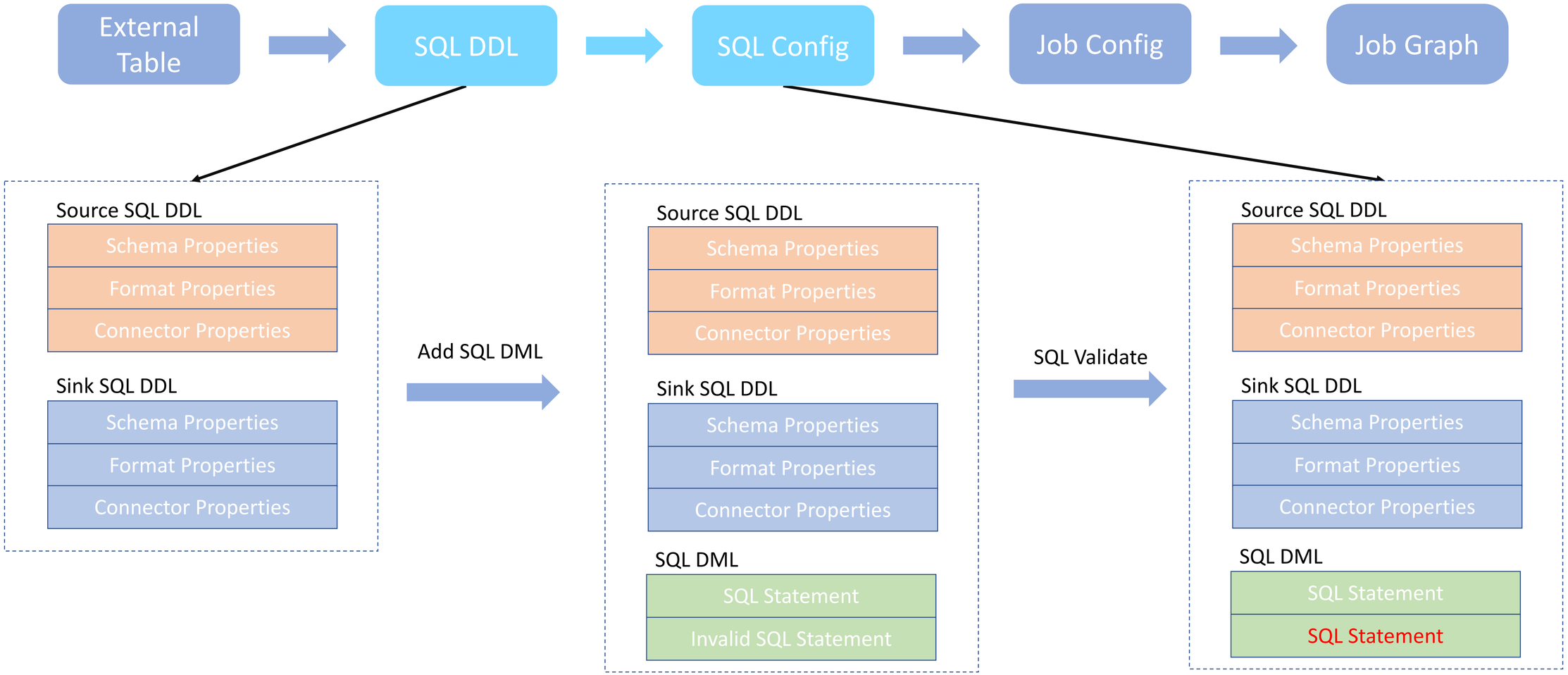
SQL Management



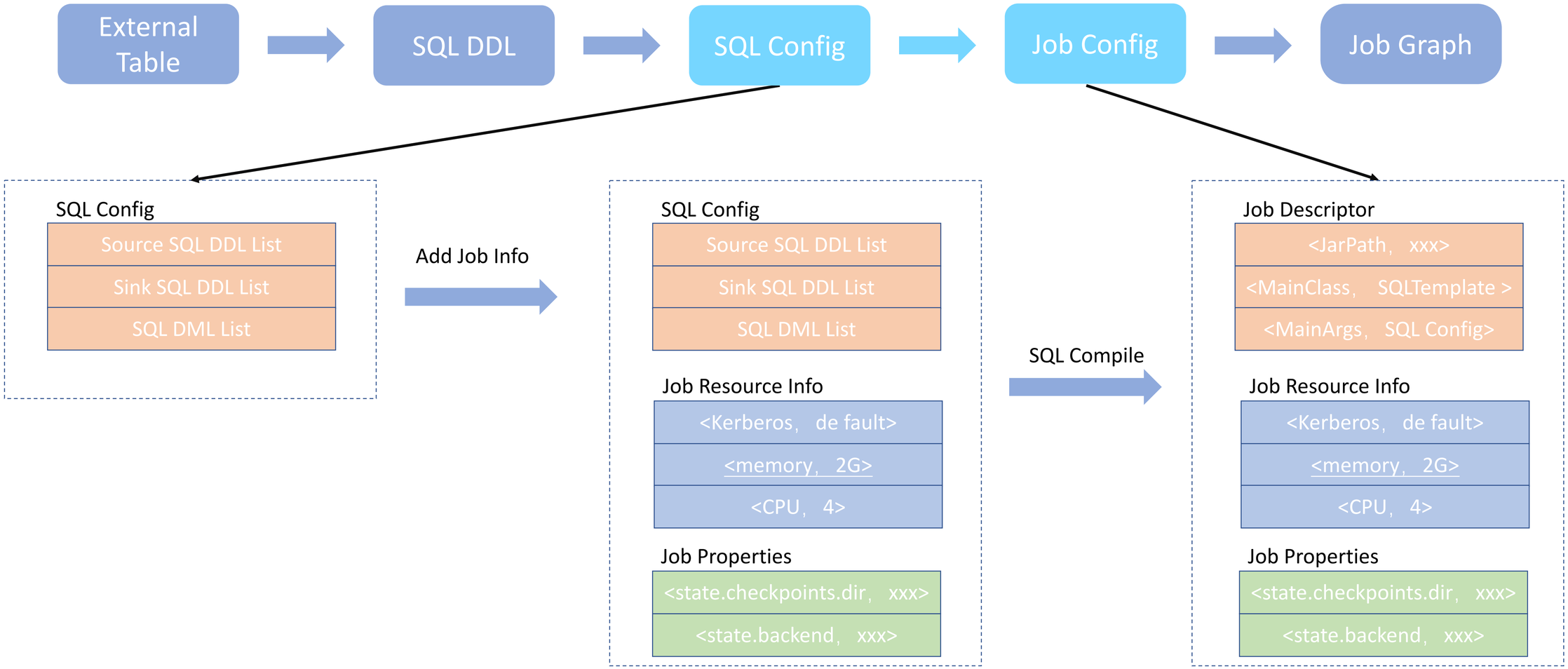
SQL Management：External Table Fetcher

- 自动获取TableSchema和TableFormat，并且去除了注册Flink Table的逻辑
  Automatically fetch Table Schema and Table Format, and remove the logic of register Flink Table.
- 获取Schema时，将外部表字段类型自动转换为Flink Table字段类型
  Automatically convert external table field types to Flink Table Field types when fetch Table Schema.
- 将Connector Properties 分成三类，参数带默认值，只有必须项要求用户填写
  Divide the Connector Properties into three categories with default parameters, and only the require user fill some specified items.
- 所有参数均采用Map<string, string>的形式表达，非常便于后续转换为TableDescriptor
  All parameters are expressed in the form of Map<string, string>, which is very ease to convert to Table Descriptor.

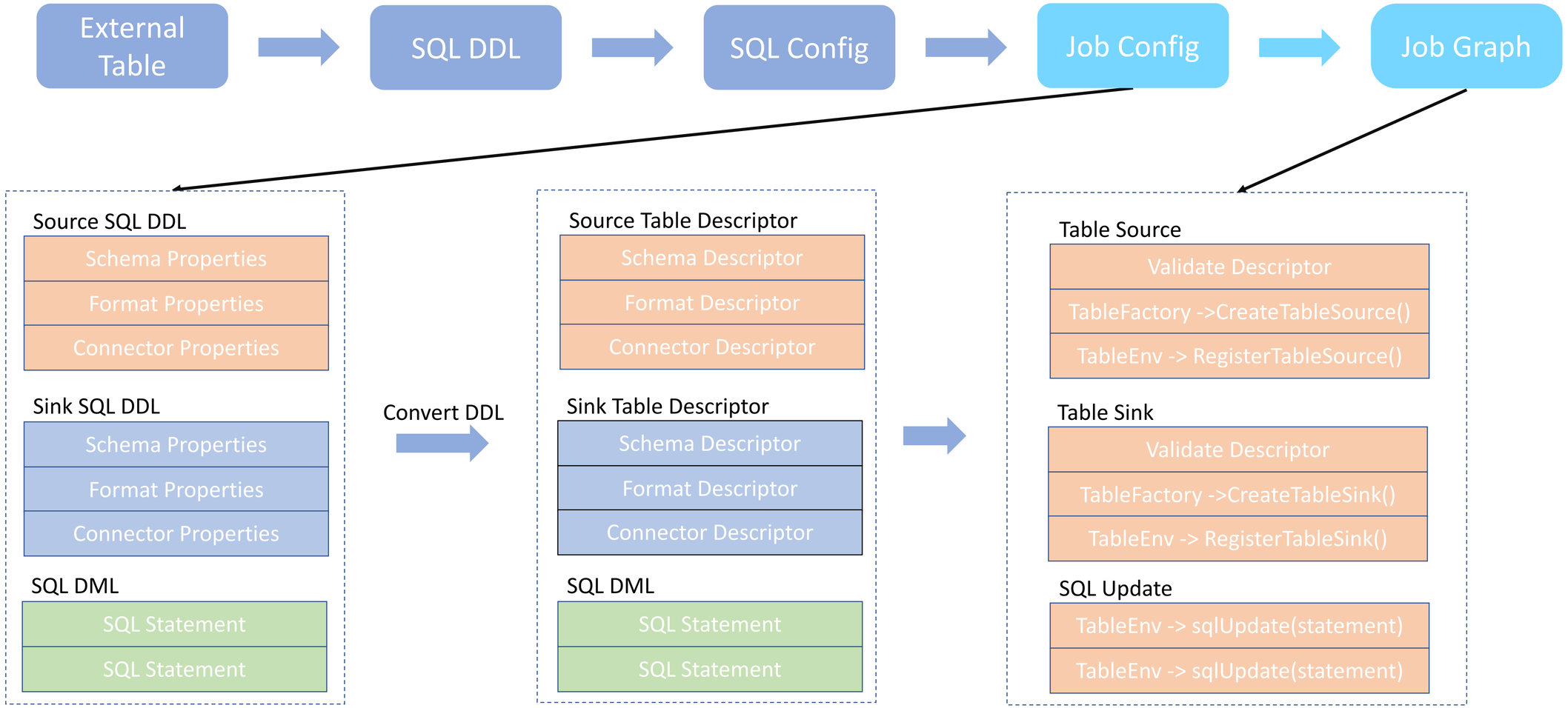
SQL Management



SQL Management



SQL Management



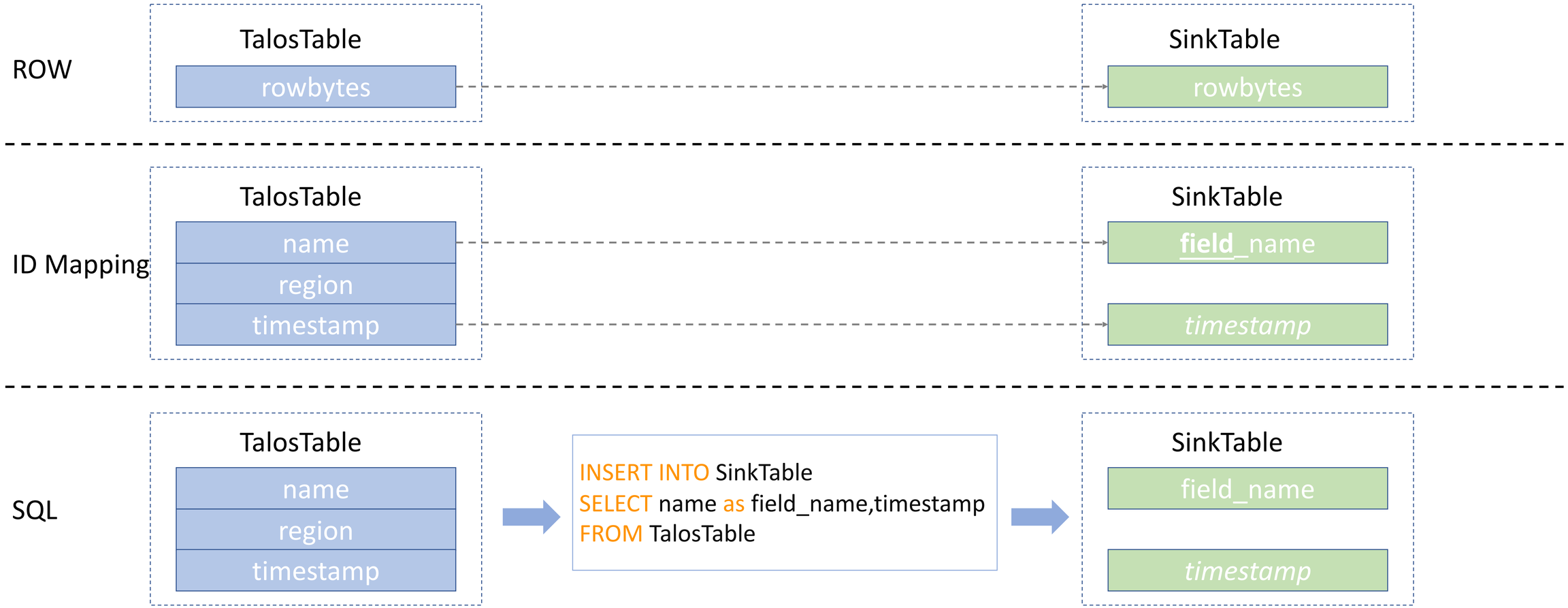
SQL Management：Template Job


```
// register Table Source
for (SqlDdl sqlDdl: sqlConfig.getSourceSqlDdl()) {
   TableSource tableSource = TableFactoryUtil.findAndCreateTableSource(new TableDescriptor(sqlDdl));
   tbEnv.registerTableSource(sqlDdl.getTable().tableName, tableSource);
}

// register Table Sink
for (SqlDdl sqlDdl: sqlConfig.getSinkSqlDdl()) {
  TableSink tableSink = TableFactoryUtil.findAndCreateTableSink(new TableDescriptor(sqlDdl));
   tbEnv.registerTableSink(sqlDdl.getTable().tableName, tableSink);
}

// exec sql statement
for (String sqlLine : sqlConfig.getSqlDml()) {
   tbEnv.sqlUpdate(sqlLine);
}
```

Talos Sink


ROW
TalosTable
rowbytes
SinkTable
rowbytes
ID Mapping
TalosTable
name
region
timestamp
SinkTable
field_name
timestamp
SQL
TalosTable
name
region
timestamp
INSERT INTO SinkTable
SELECT name as field_name,timestamp
FROM TalosTable
SinkTable
field_name
timestamp

未来规划

Future Plans

04

Future Plans

- **Streaming Job推进和平台化建设**
  Streaming Job promotion and platform construction.
- **统一离线数仓和实时数仓**
  Unify offline data warehouse and real-time data warehouse.
- **数据血缘分析与展示**
  Construct and display data pedigree.
- **Flink 社区参与**
  Participate in the Flink community.


THANKS