OPPO 基于 Apache Flink 的实时数仓实践

Best practices for building real-time data warehouse in OPPO

演讲人：张俊
职位 ：OPPO 大数据平台研发负责人

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

FLINK FORWARD

Contents
目录

Contents
目录

FLINK FORWARD

关于 OPPO 移动互联网业务

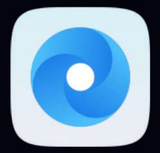
About OPPO mobile Internet business

ColorOS 月活跃用户超 3 亿，发布数十款应用

ColorOS has more than 300 million monthly active users and released tens of applications

主题商店

Theme Store



浏览器

Browser

游戏中心

Game Center

软件商店

App Store

云服务

Cloud Service

搜索

Search

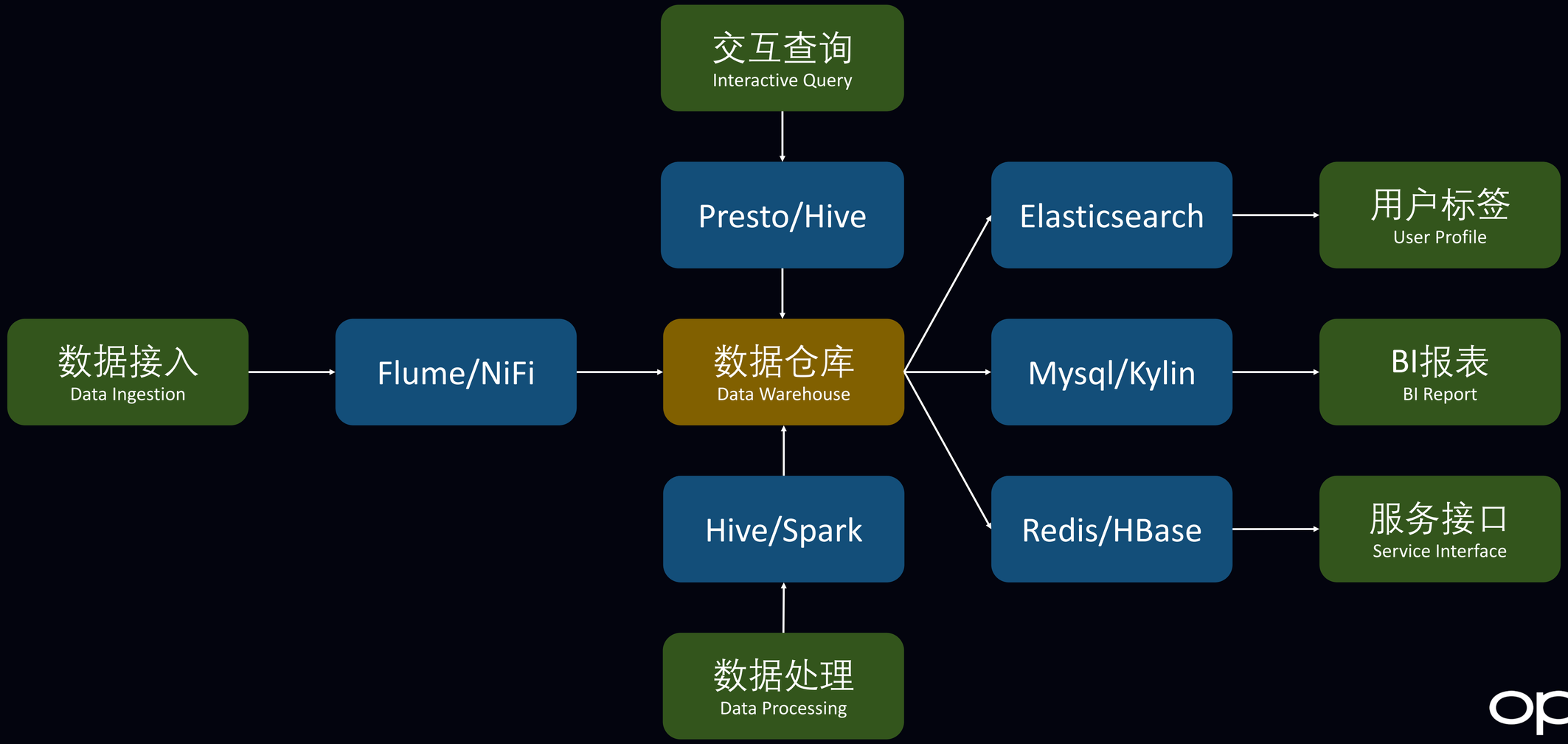
小游戏

Mini-game

oppo

以数仓为中心的数据架构

Data warehouse centric data architecture



数仓实时化的诉求

The requirements for the real-time data warehouse

小时/天级 → 分钟/秒级

hourly/daily → minutes/seconds

业务侧

Application aspect

- 实时报表：人群投放的曝光率/点击率

  Real-time report: impression/click-through rate

- 实时标签：用户当前所在的商圈

  Real-time profile: user's current location

- 实时接口：用户最近下载某APP的时间

  Real-time interface: the time when a user downloaded a given app

平台侧

Platform aspect

- 调度任务：集中在凌晨启动，集群压力大

  Scheduled task: mostly starts at mid-night which put high pressure on the cluster

- 用户标签：全量导入耗费数小时

  User profile: takes hours to import at batch

- 数据质量：很难及时发现数据异常

  Data quality: hard to catch data exceptions timely

FLINK FORWARD

实时数仓的现状

Current status of the real-time data warehouse

Flink 集群： 500+

Flink cluster size

Kafka 集群：200+

Kafka cluster size

实时库表： 500+

Real-time table count

实时作业： 300+

Real-time job count

日处理总数： 10 Trillion

Total data processed per day

日处理峰值： 300 Million/s

Peak data rate per second

oppo

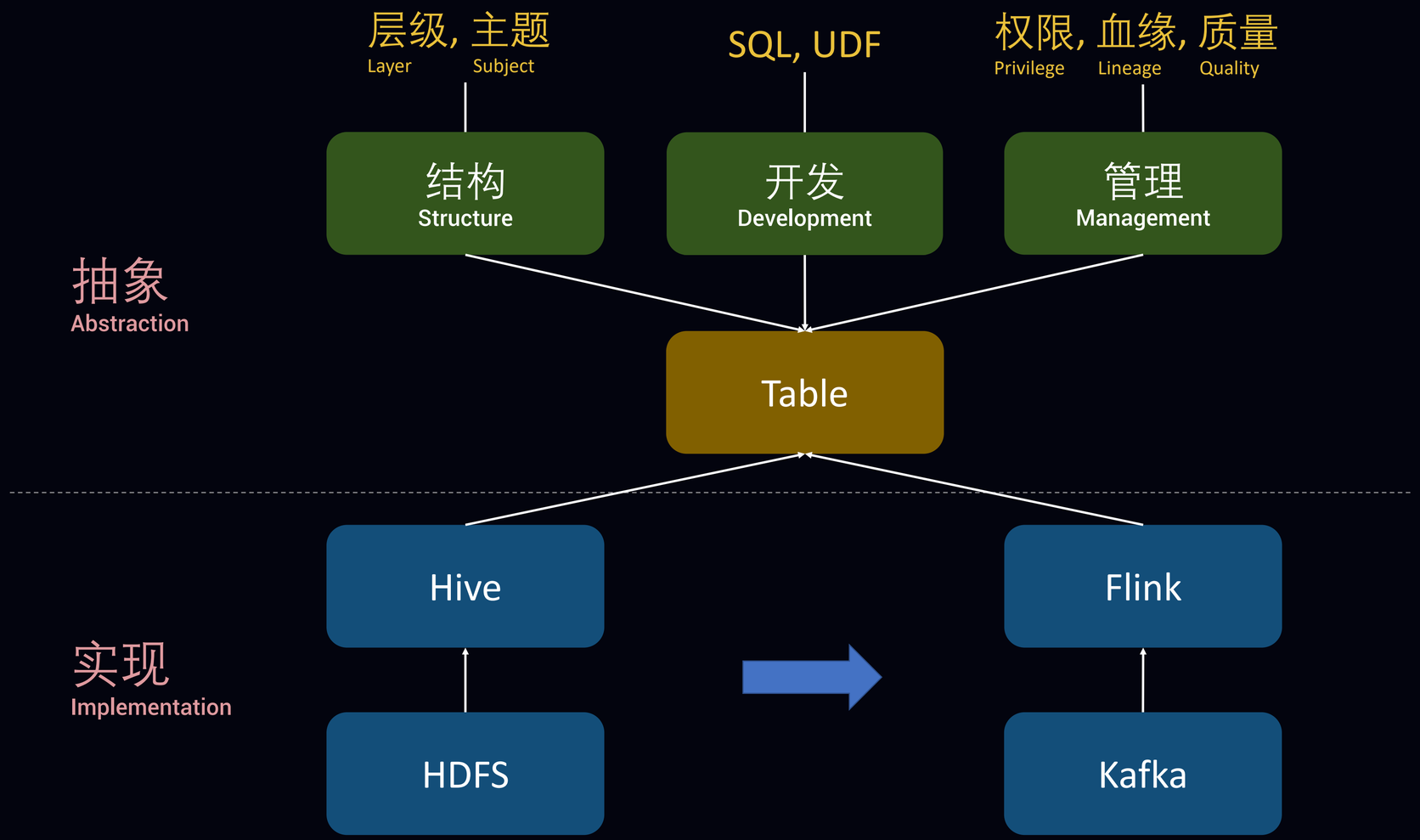
Contents
目录

实时数仓 vs 离线数仓

Real-time data warehouse vs Offline data warehouse

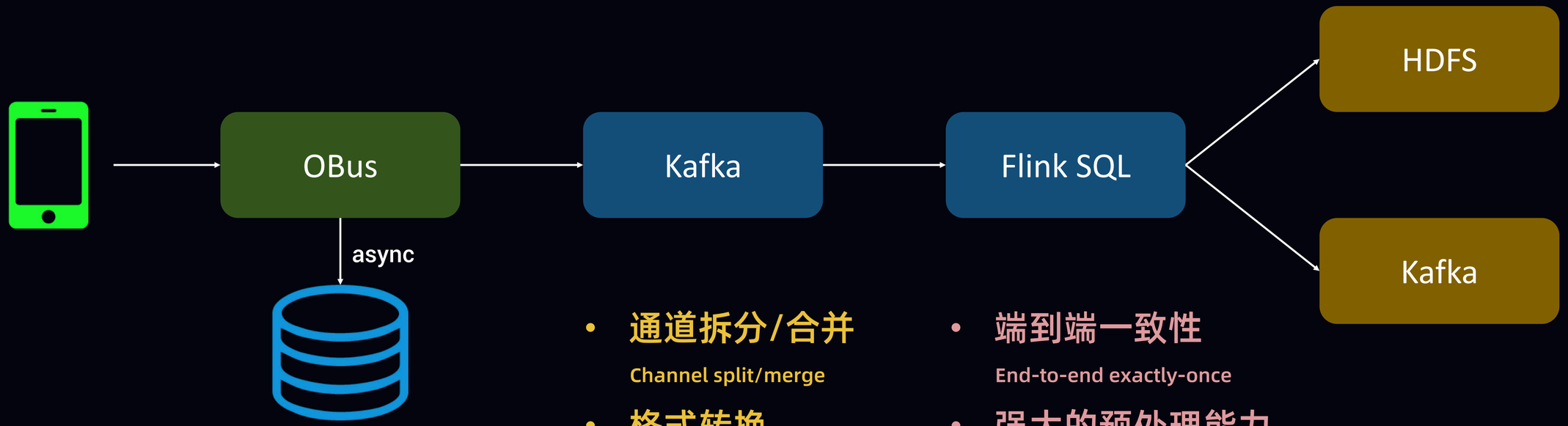
离线到实时数仓的平滑迁移

Smooth migration from offline to real-time data warehouse



离线实时一体化的接入链路

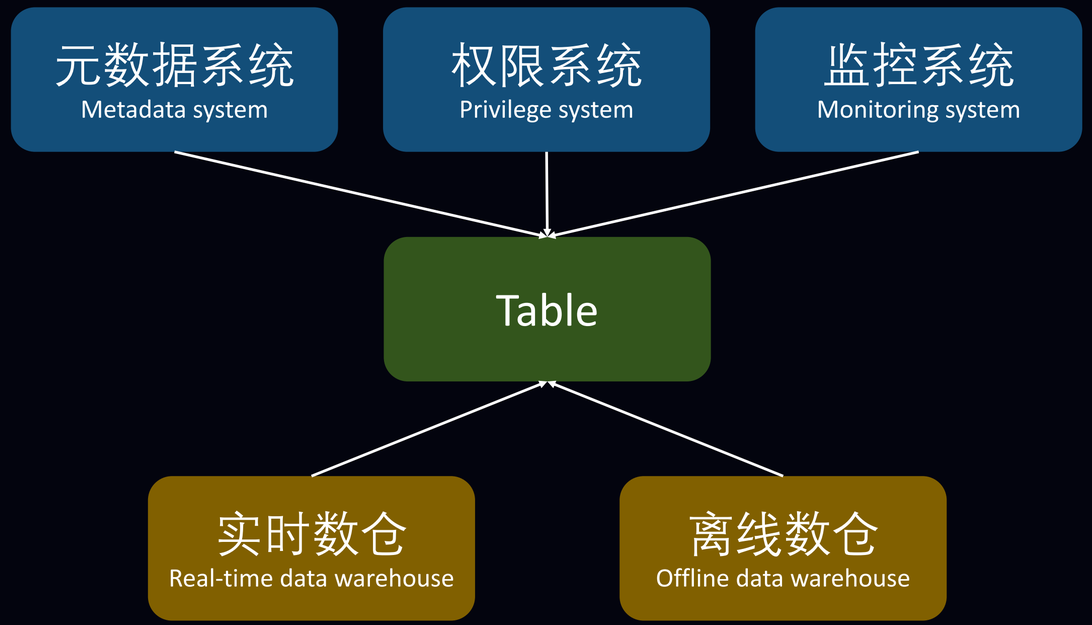
Unified ingestion pipeline for both offline and real-time data warehouse



- 通道拆分/合并
  Channel split/merge
- 格式转换
  Format conversion
- 数据监控
  Data monitoring
- 端到端一致性
  End-to-end exactly-once
- 强大的预处理能力
  Strong preprocessing power
- 一套代码实现
  Single codebase

离线实时一体化的管理流程

Unified management process for both offline and real-time data warehouse



- 表格式定义

  Table schema definition
- 表血缘追踪

  Table lineage tracing
- 表权限申请

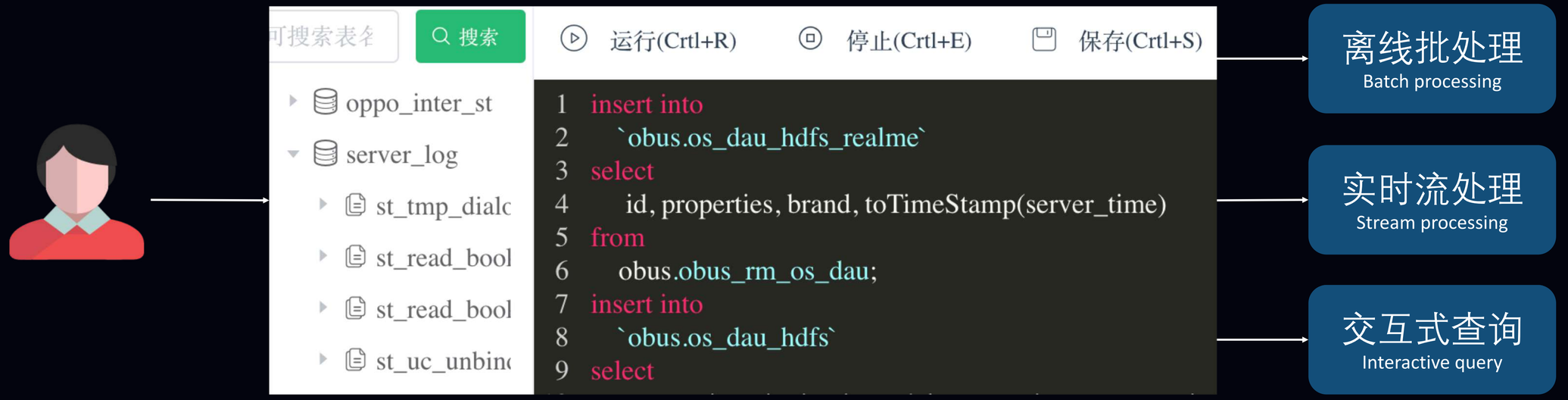
  Table privilege application
- 表监控配置

  Table monitoring configuration

离线实时一体化的开发环境

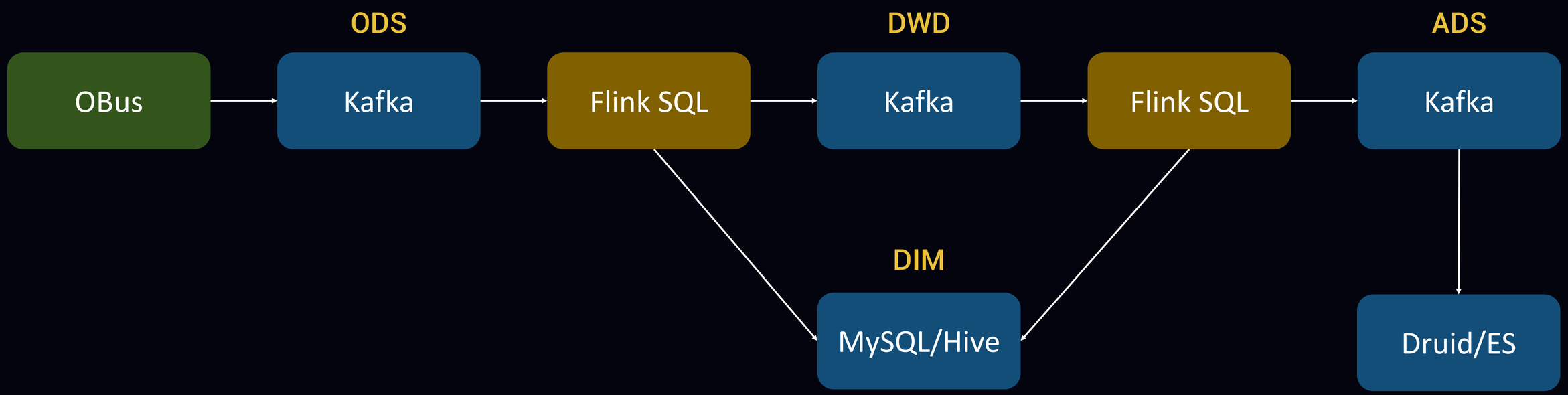
Unified development environment for both offline and real-time data warehouse

- 用户体验一致
  Consistent user experience
- 开发流程统一
  Unified development process



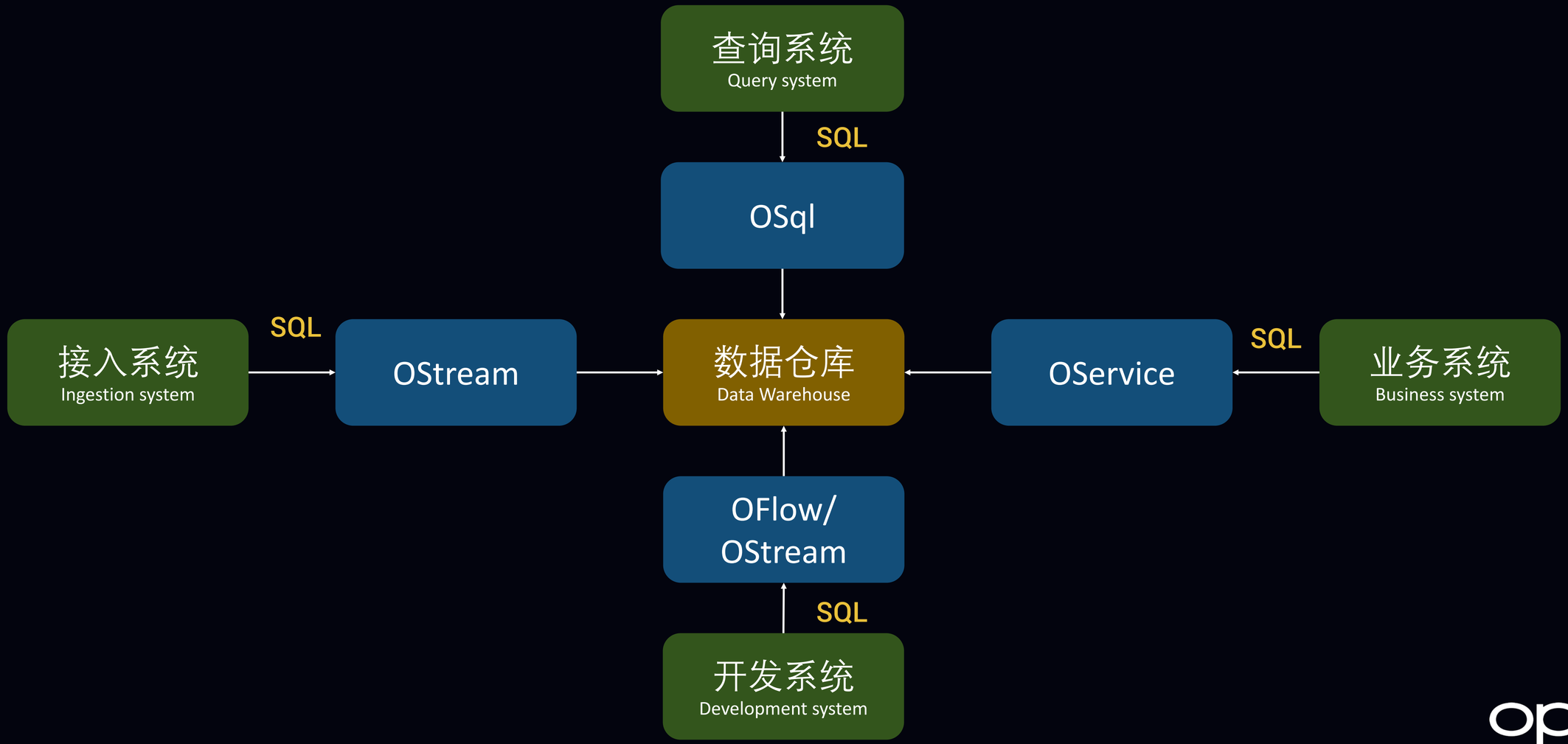
实时数仓的层级划分

The layers of the real-time data warehouse



SQL 一统天下的数据架构

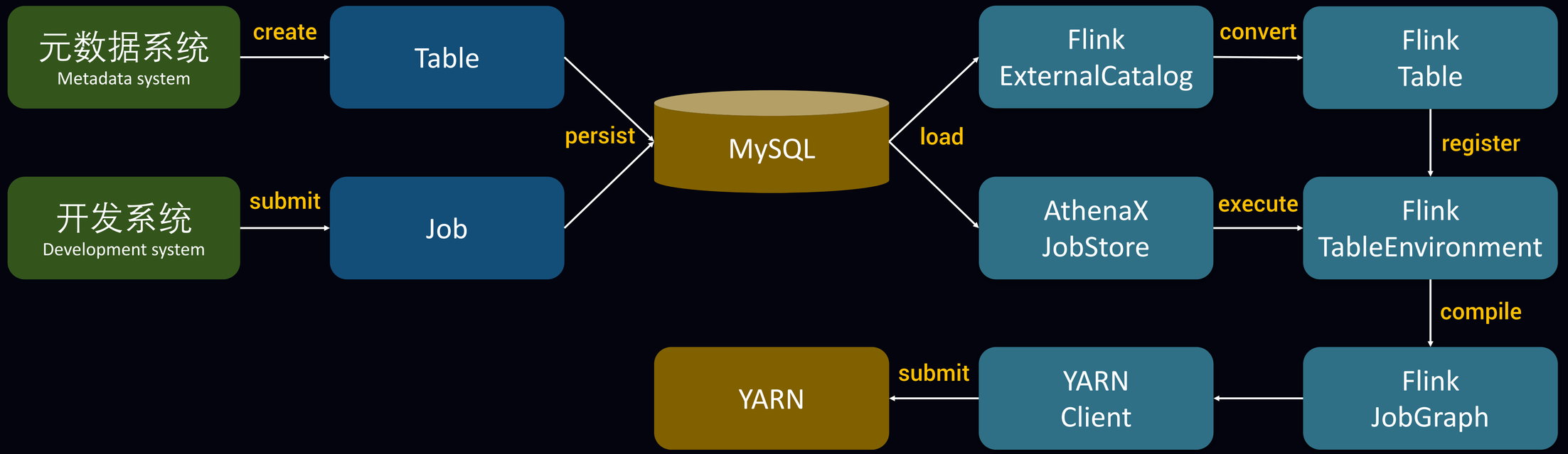
One SQL to rule them all



Contents
目录

SQL 开发与元数据管理的实现

The implementation of SQL development and metadata management

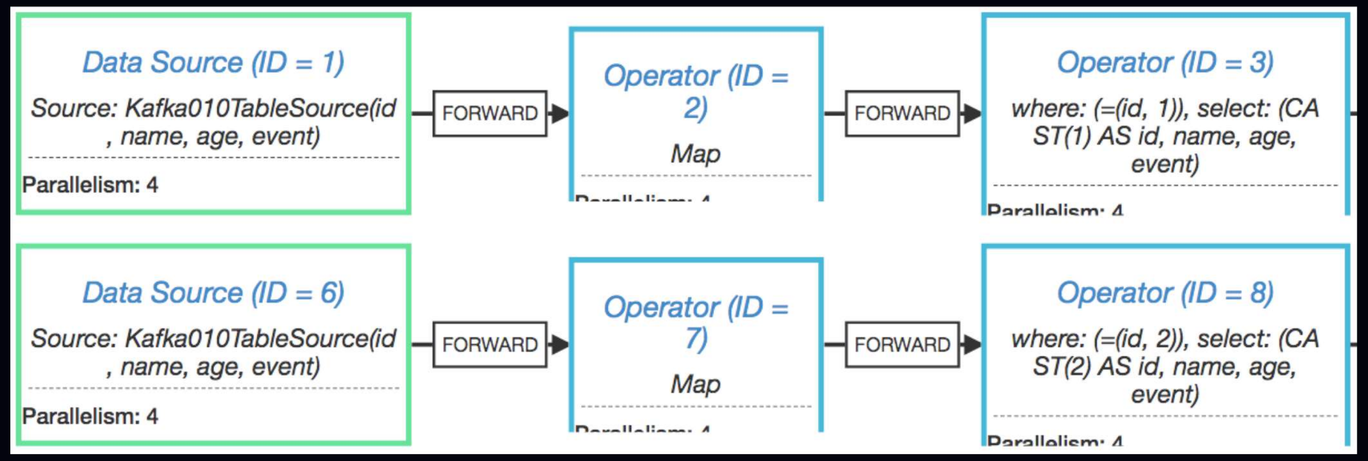


冗余消费 Kafka Topic 问题的优化

The optimization for duplicate consumption of kafka topic

```
INSERT INTO `dw.sdk_log_cdo` SELECT * FROM dw.sdk_log
    WHERE system_id = '2' OR system_id = '1000';

INSERT INTO `dw.sdk_log_browser_client` SELECT * FROM dw.sdk_log
    WHERE system_id = '2007' AND event_info['eventTag'] = '10001';
```



单个作业多 SQL 读相同表

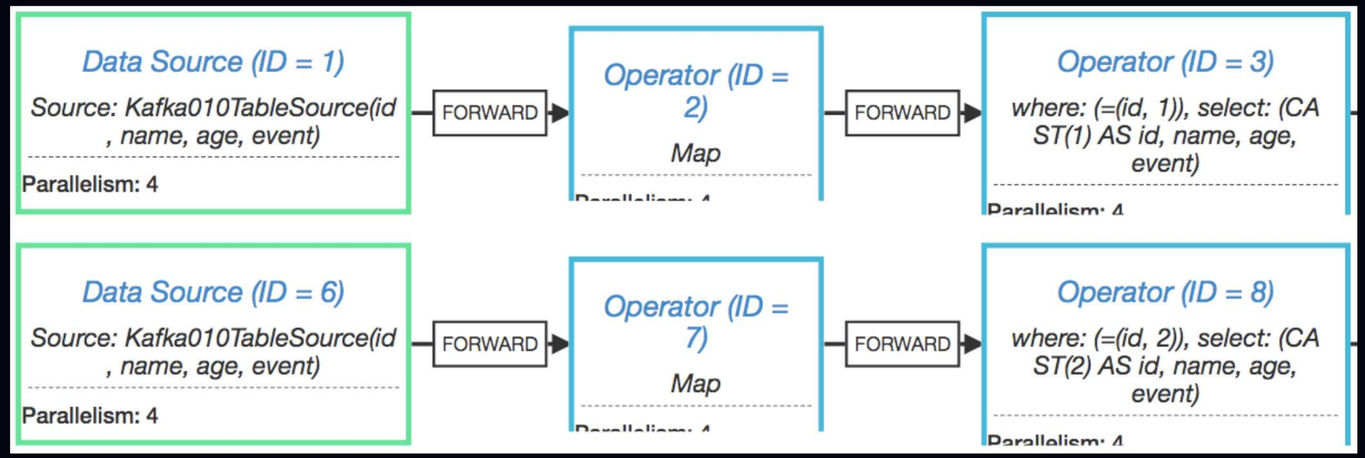
Multiple SQLs in single job with same source table

多次消费同一个 kafka topic

Consume the same kafka topic multiple times

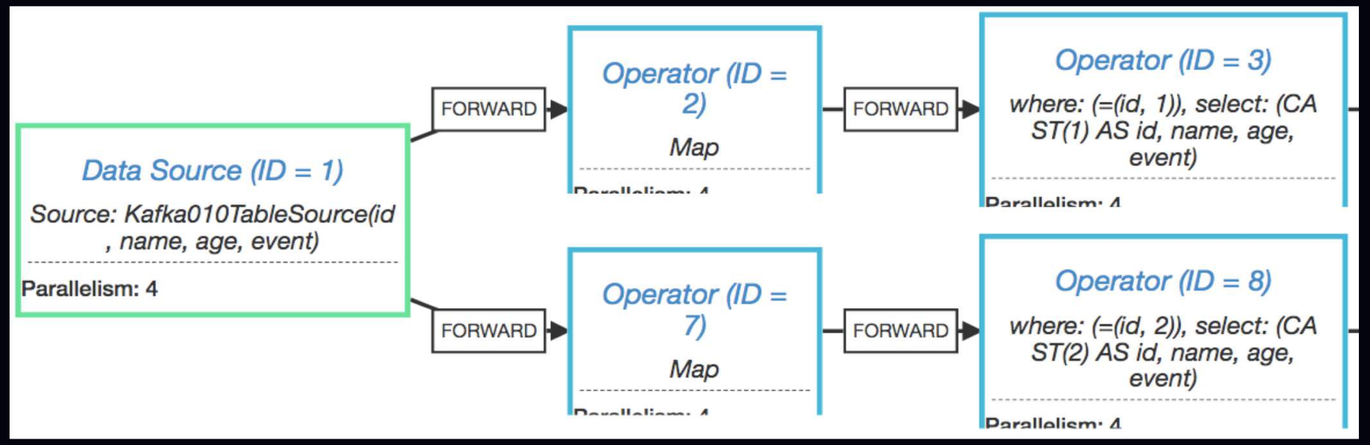
冗余消费 Kafka Topic 问题的优化 (Cont'd)

The optimization for duplicate consumption of kafka topic



改写 StreamGraph，合并冗余 DataSource

Rewrite StreamGraph and merge duplicate DataSource

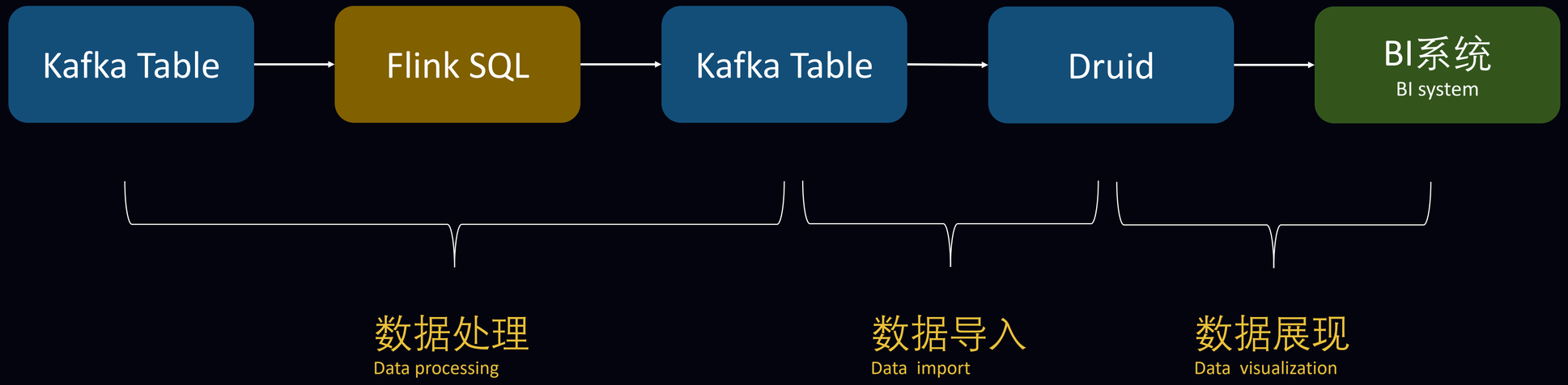


单作业只消费一次 kafka topic

Consume the kafka topic once in a single job

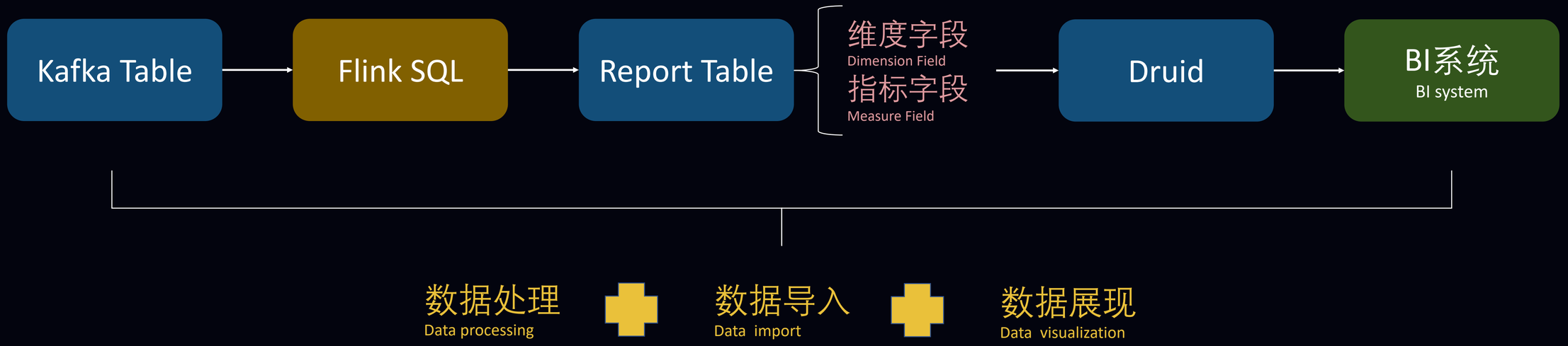
实时数据链路的自动化

Workflow automation of the real-time data pipeline



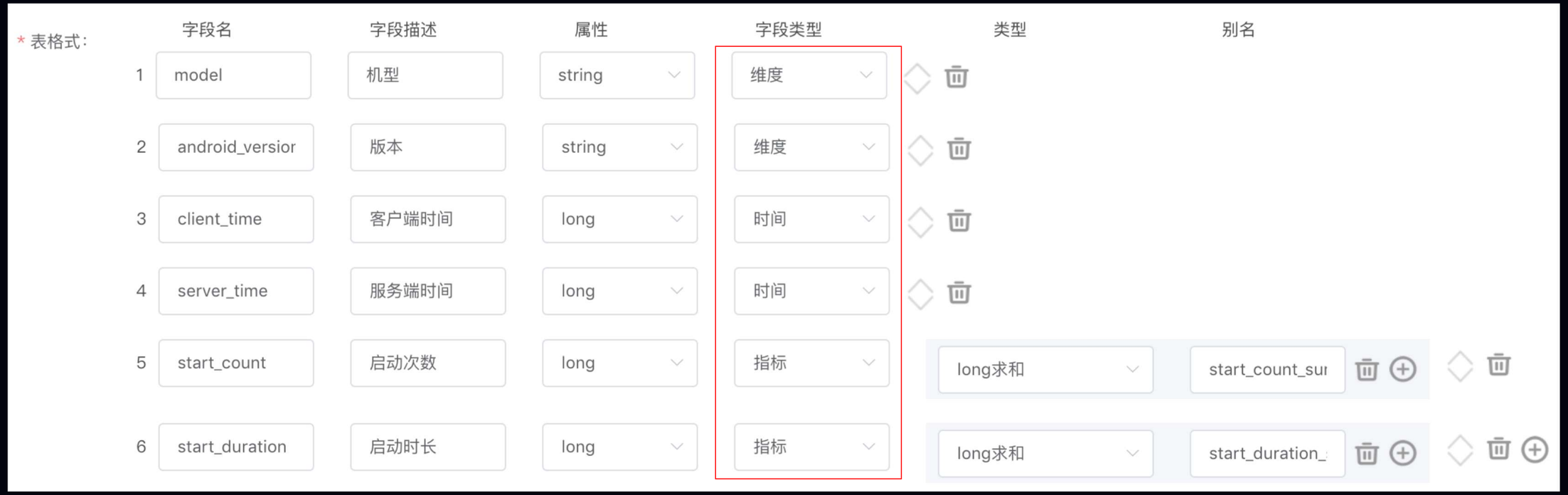
实时数据链路的自动化 (Cont'd)

Workflow automation of the real-time data pipeline



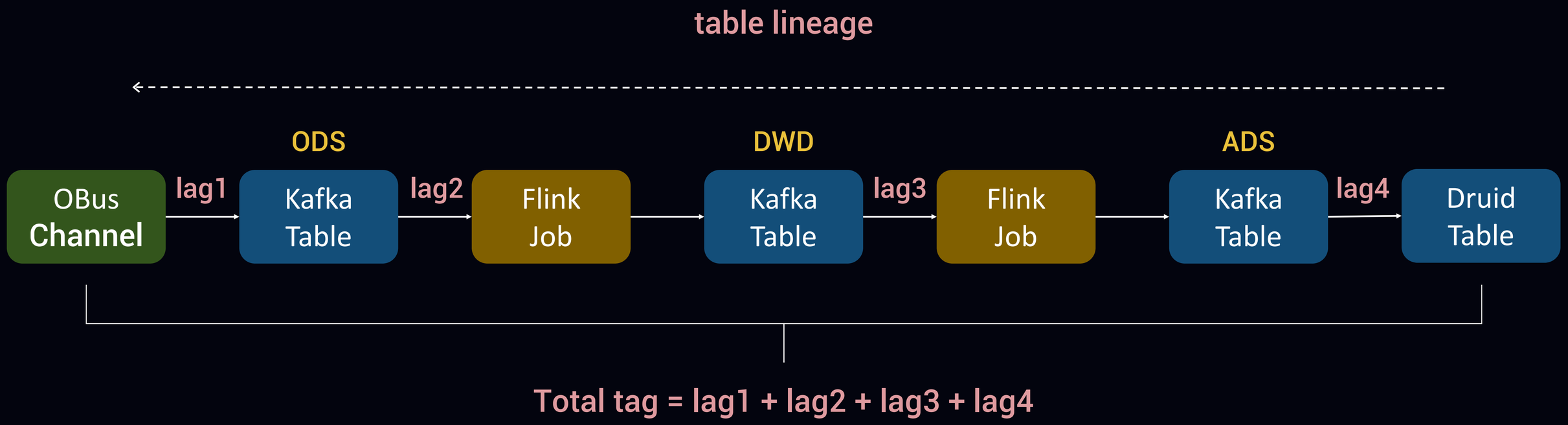
实时数据链路的自动化 (Cont'd)

Workflow automation of the real-time data pipeline



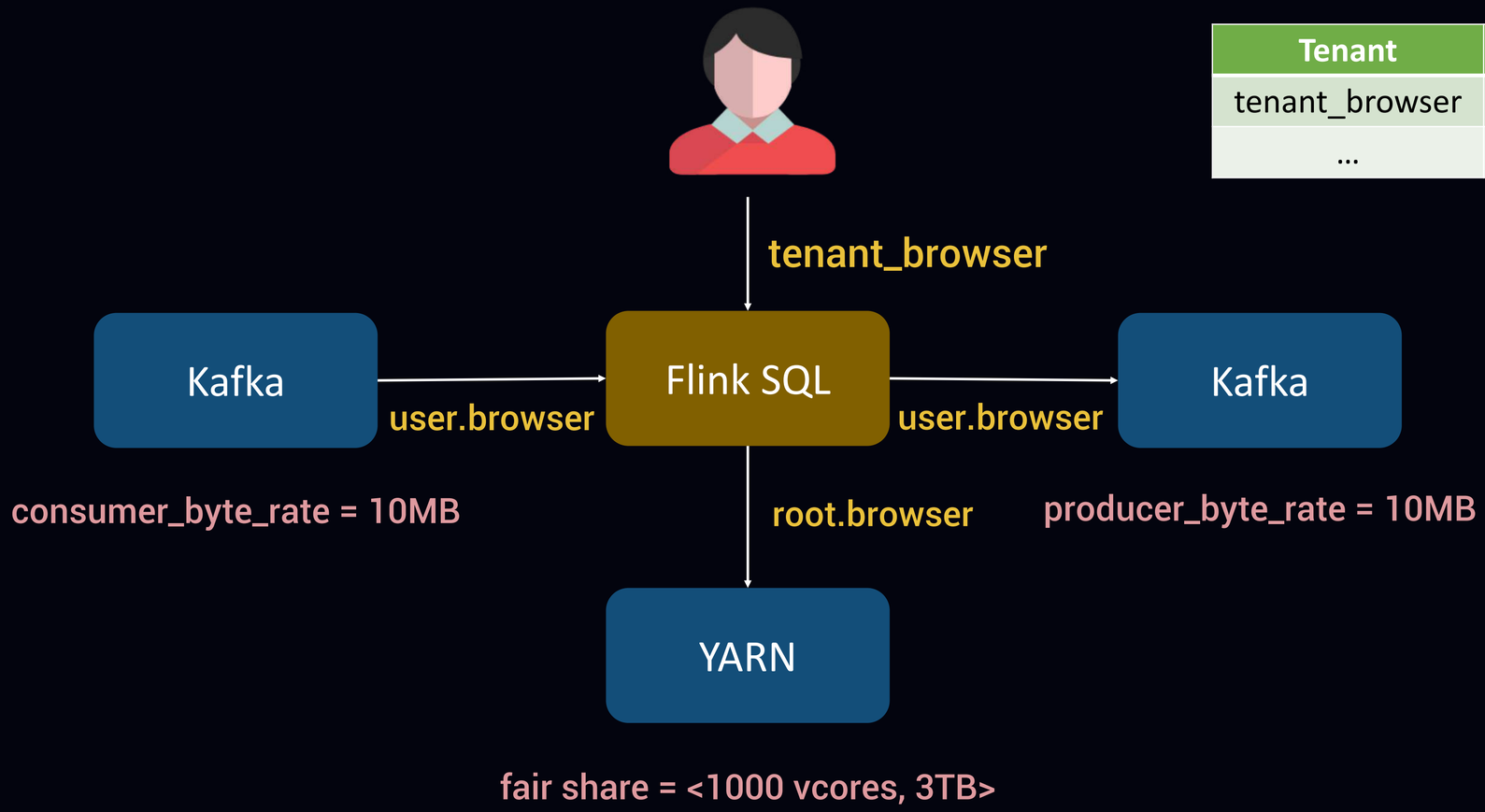
实时数据链路的延迟监控

Latency monitoring of the real-time data pipeline



实时数据链路的多租户管理

Multi-tenancy management of the real-time data pipeline



Tenant	Business group	Kafka User	YARN Queue
tenant_browser	browser	user.browser	root.browser
...	...	...	...

- 业务隔离
  Business isolation
- 成本核算
  Cost accounting
- Kafka认证与配额
  Kafka authentication and quota
- YARN队列调度
  YARN queue scheduling

Contents
目录

FLINK FORWARD

更便捷的 SQL 开发

Towards easier SQL development

表达能力
Expressiveness

连接类型
Connector

开发模板
Template

开发规范
Specification

oppo

更细粒度的资源调度

Towards finer-grained resource scheduling

不同的SQL，计算密集度差别很大

The computational intensity gap between different SQLs may be huge

CPU核数	CPU使用率	内存使用率	负载	流量in	流量out	最近监控时间	操作
64	99.99%	41.64%	188.7	548.29 Mbps	74.11 Mbps	2019-11-09 14:28:51	详情
64	99.99%	50.02%	204.6	422.68 Mbps	188.5 Mbps	2019-11-09 14:41:42	详情
64	99.99%	31.93%	158.0	298.6 Mbps	167.14 Mbps	2019-11-09 14:26:13	详情
64	99.99%	48.89%	332.3	933.56 Mbps	223.03 Mbps	2019-11-09 14:26:13	详情
64	99.98%	27.35%	222.1	98.19 Mbps	42.28 Mbps	2019-11-09 14:20:50	详情
64	99.98%	37.06%	140.7	129.63 Mbps	27.85 Mbps	2019-11-09 14:33:46	详情

64	37.89%	34.96%	18.62	1.21 Gbps	510.03 Mbps	2019-11-09 14:26:19	详情
64	37.32%	36.76%	20.53	1.19 Gbps	870.04 Mbps	2019-11-09 14:26:54	详情
64	36.47%	41.20%	17.75	1.48 Gbps	431.1 Mbps	2019-11-09 14:26:14	详情
64	35.65%	30.98%	19.68	956.8 Mbps	597.06 Mbps	2019-11-09 14:28:51	详情
64	35.59%	22.40%	20.76	216.95 Mbps	63.51 Mbps	2019-11-09 14:41:37	详情
64	34.98%	24.18%	20.66	1.34 Gbps	77.08 Mbps	2019-11-09 14:20:50	详情

FLINK FORWARD

自动化的参数配置

Towards automatic parameter configuration

Operator
Parallelism

State
Backend

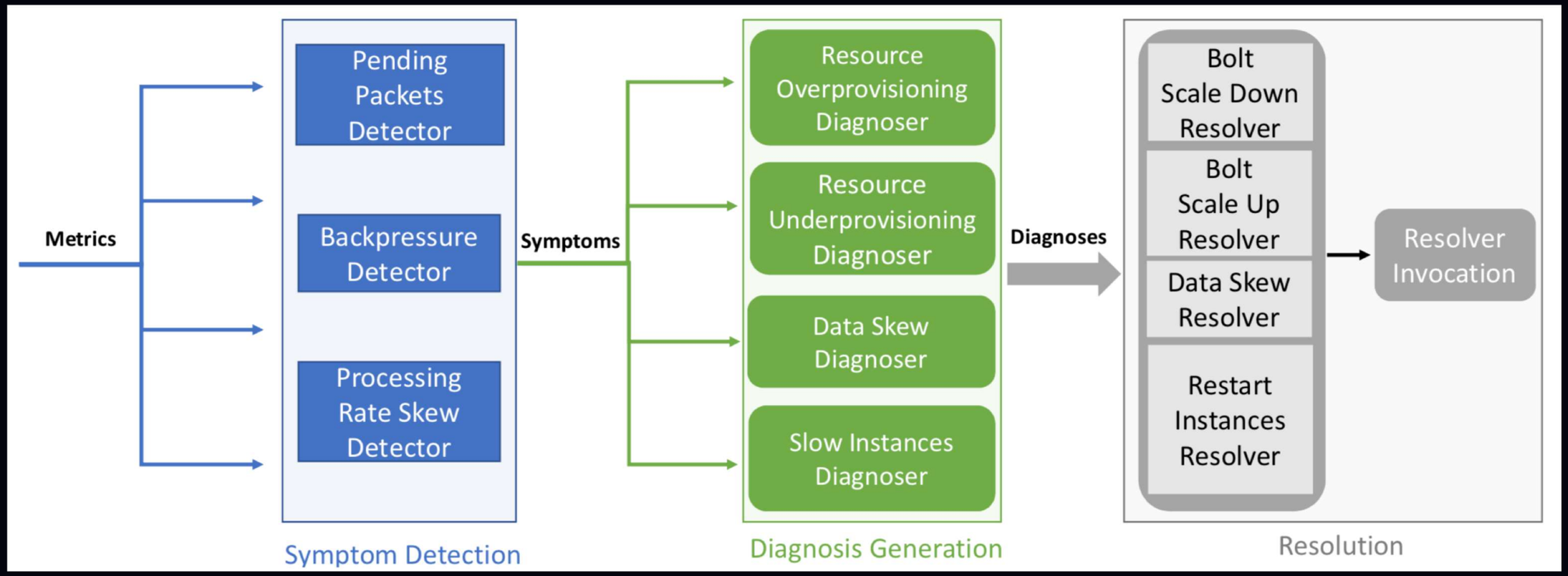
Watermark
Interval

oppo

自动化的伸缩调优

Towards automatic scaling and tuning

Dhalion: Self-Regulating Stream Processing in Heron




THANKS