

# 大数据训练营 — 模块七

## 数据仓库、ETL 和数据开发体系： DW、ETL、Data Platform

极客时间

金澜涛





# 目录

数据开发体系：

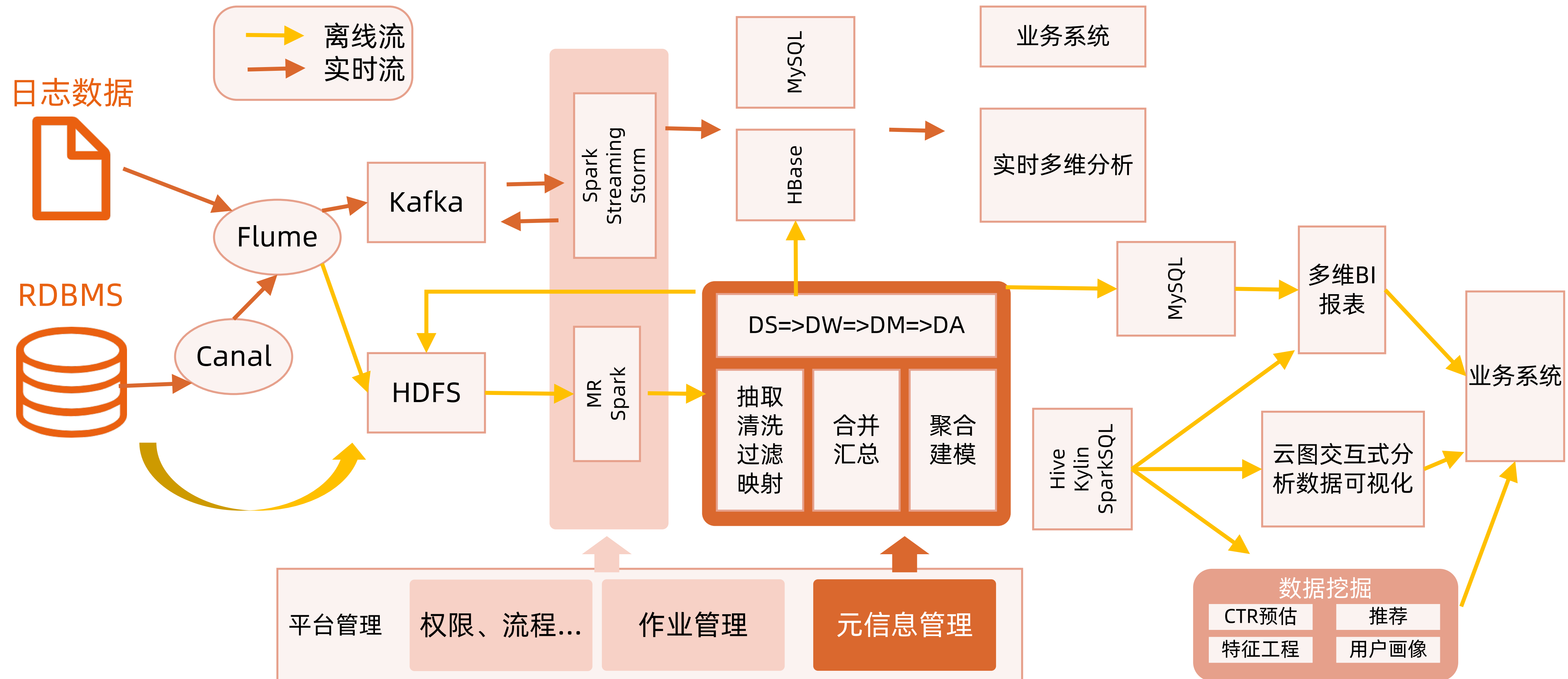
1. 数据平台整体架构
2. ETL 系统与技术
3. 日志收集和相关开源框架
4. 数据库传输和相关开源框架
5. 调度和监控系统
6. 元数据系统
7. 数据质量系统

# 1. 数据平台整体架构

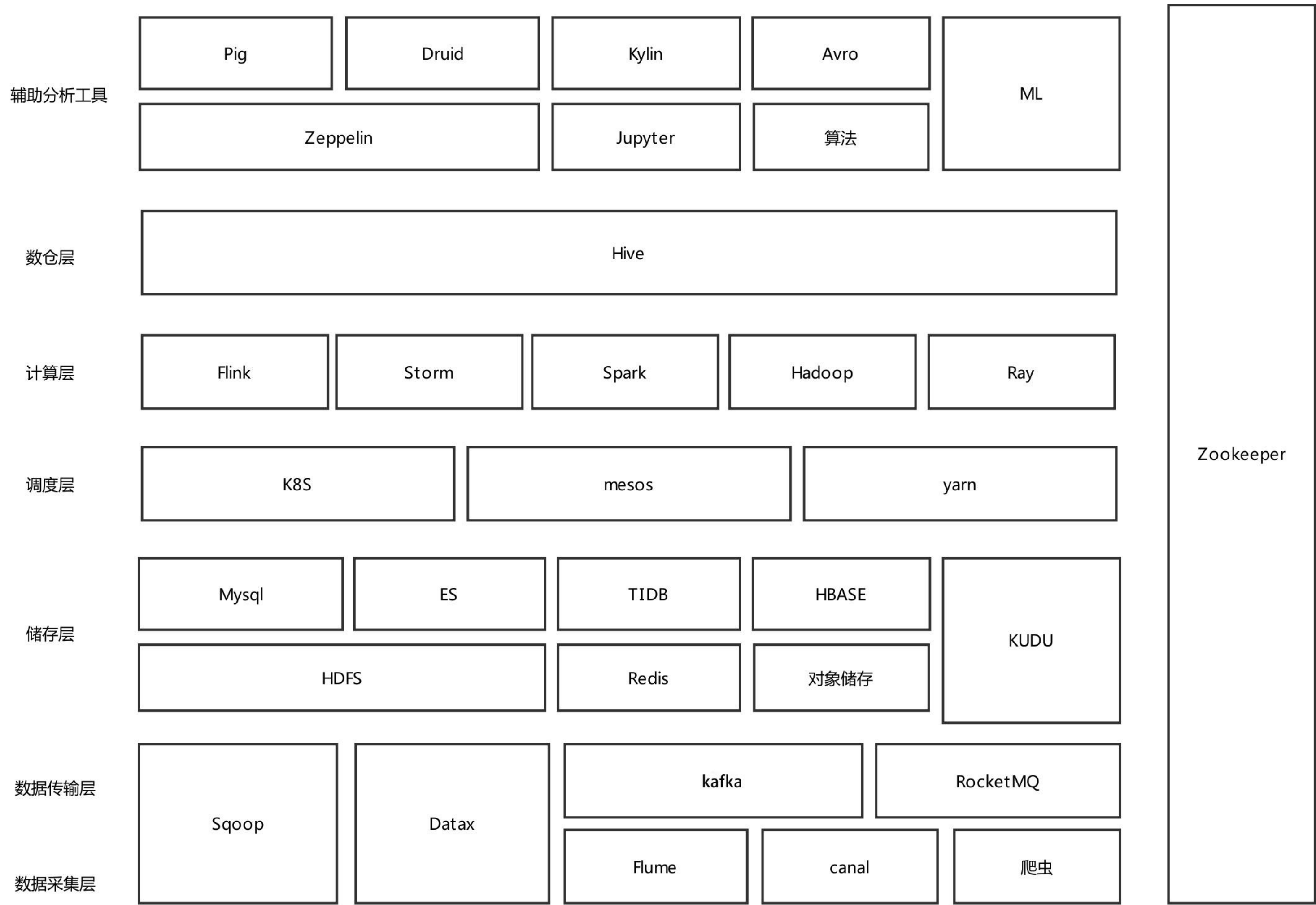
# 数据平台和组件



# 数据流动

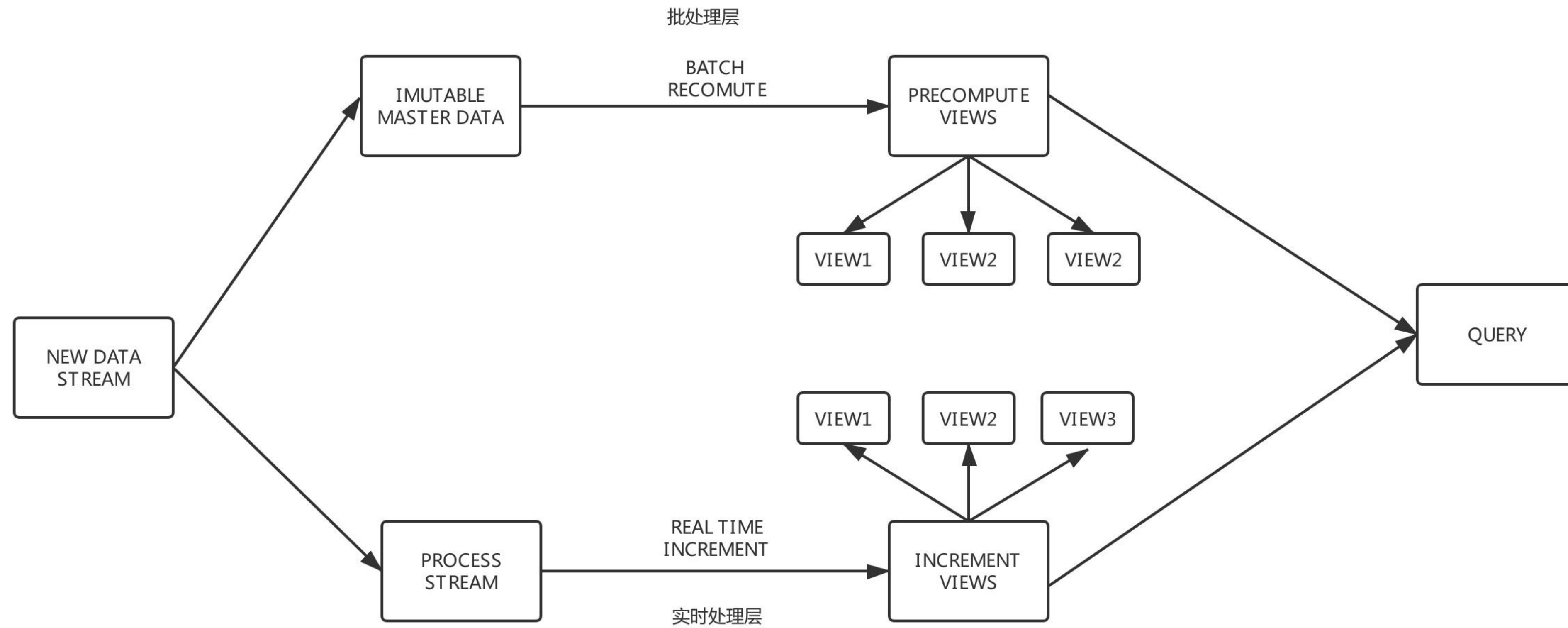


# 技术栈

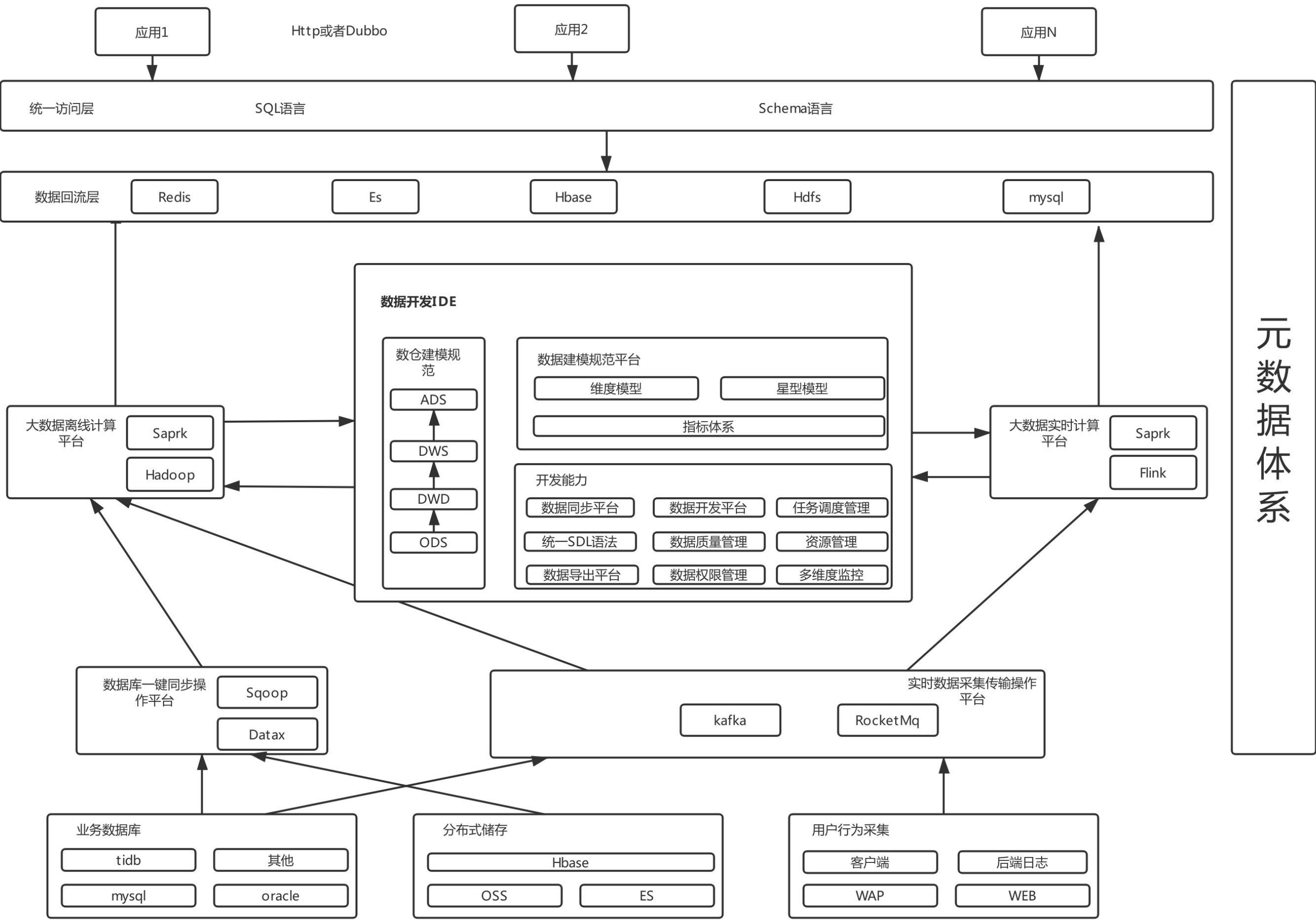




# Lambda 架构



# 真实的企业大数据架构

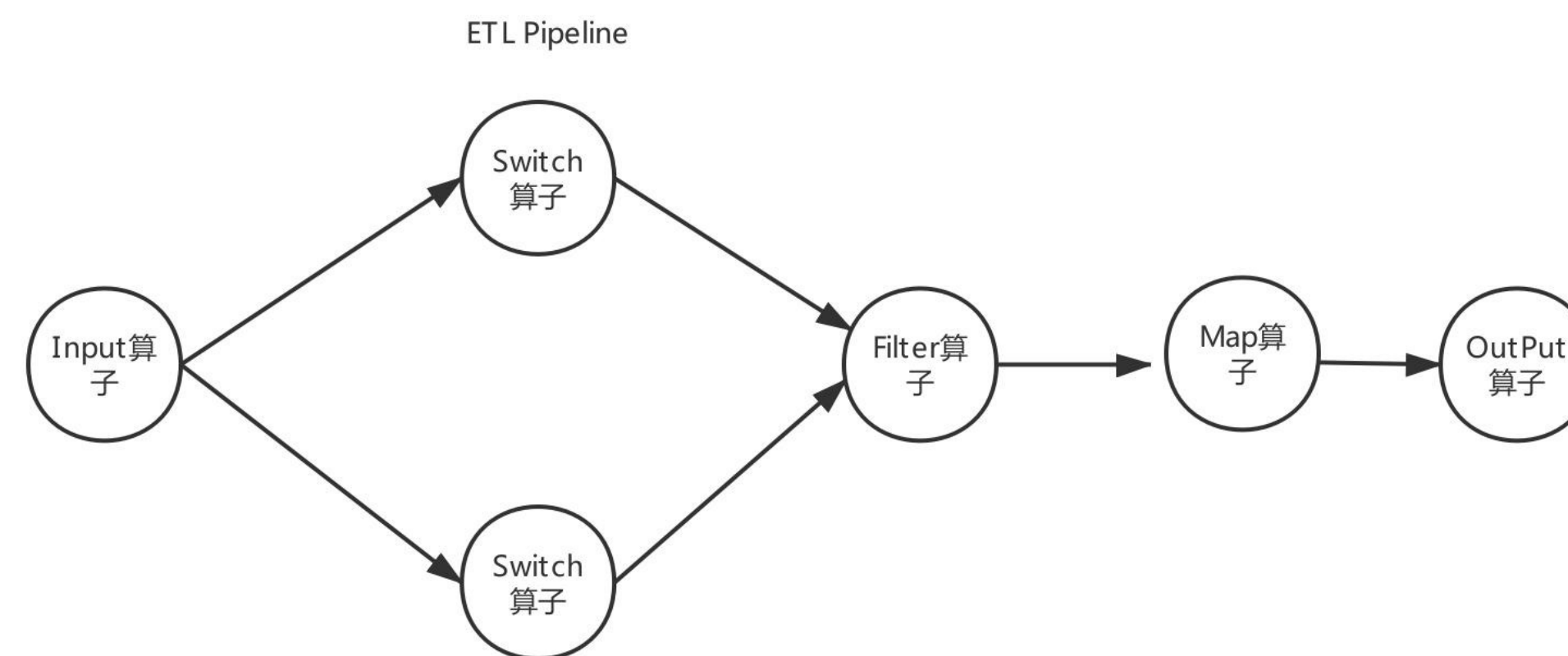
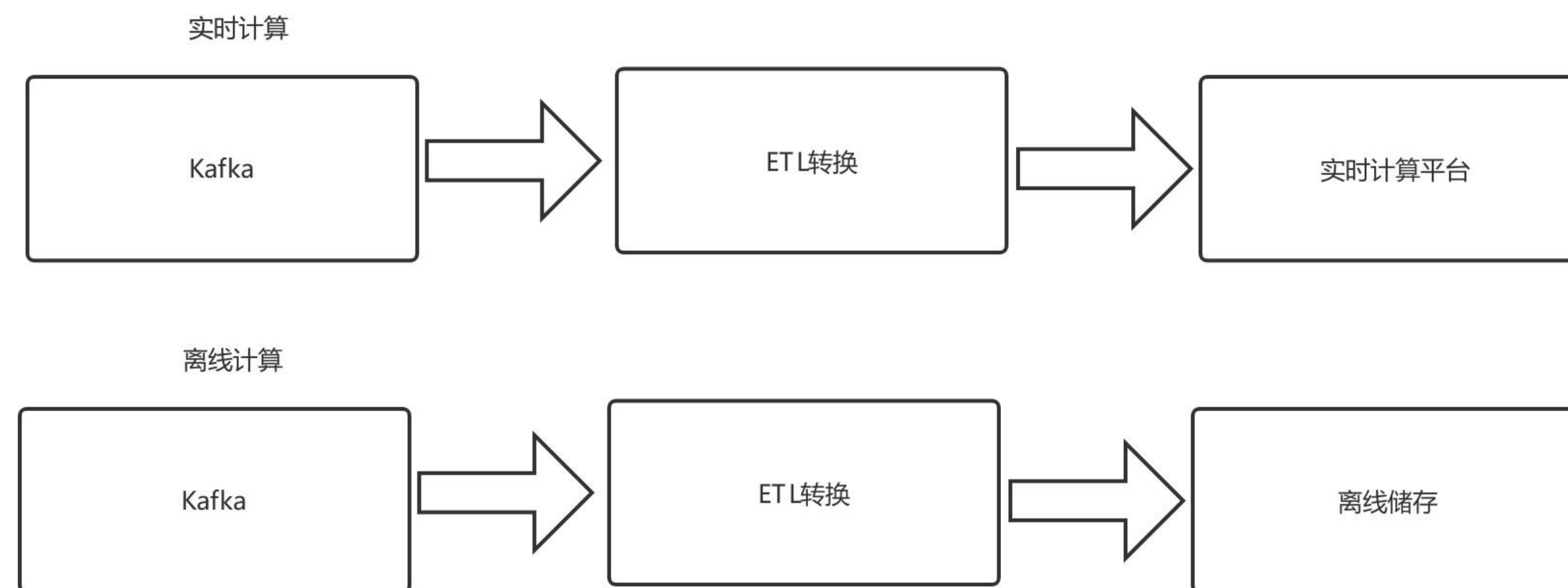




## 2. ETL 系统与技术

- ETL 即 Extract-Transform-Load，用来描述将数据从来源端经过抽取（extract）、转换（transform）、加载（load）至目的端的过程。ETL 一词较常用在数据仓库，但其对象并不限于数据仓库。
- 一般而言，ETL 平台在数据清洗、数据格式转换、数据补全、数据质量管理等方面有很重要的作用。

# 实时与离线 ETL





# 数据清洗步骤

## 预处理阶段：

- 将数据导入处理工具
- 看数据

## 分析处理阶段：

- 缺失值清洗
  - 去除不需要的字段
  - 填充缺失内容
  - 重新取数
- 格式内容清洗
  - 时间日期、数值、全半角显示不一致等
  - 内容中有不该存在的字符
  - 内容与该字段应有内容不符
- 逻辑错误清洗
  - 去重
  - 去除不合理值
  - 修正属性依赖冲突
- 非需求数据清洗
- 校验
  - 数据格式校验
  - 关联性校验

数据转换的任务主要是进行不一致的数据转换、数据粒度的转换和一些商务规则的计算。

- 数据类型转换
  - 增加“上下文”数据
  - 解码（Decoding）
  - 清洁和净化
- 多数据源整合
  - 字段映射（Mapping）
  - 代码变换（Transposing）
  - 合并（Merging）
  - 派生（Derivations）
- 数据粒度的转换
- 商务规则的计算

加载经转换和汇总的数据到目标数据仓库中，可实现 SQL 批量加载。数据加载（Load）经过数据转换生成的文件的结构与数据仓库数据表的结构完全一致。

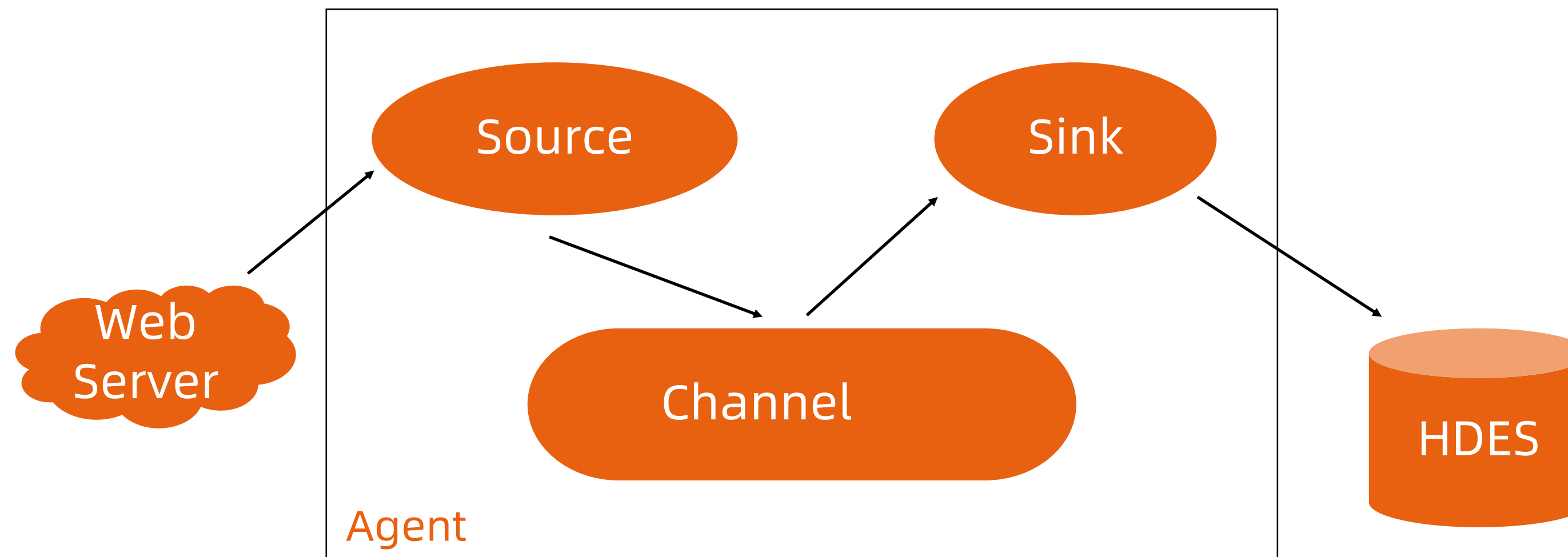
- 加载准备
  - 删除数据仓库中数据表的索引，提高加载效率
- 加载
  - Insert
  - Upsert
  - Refresh
- 加载后
  - 重新生成索引，在加载后阶段删除的索引需在此重建
  - 文件清理



### 3. 日志收集和相关开源框架

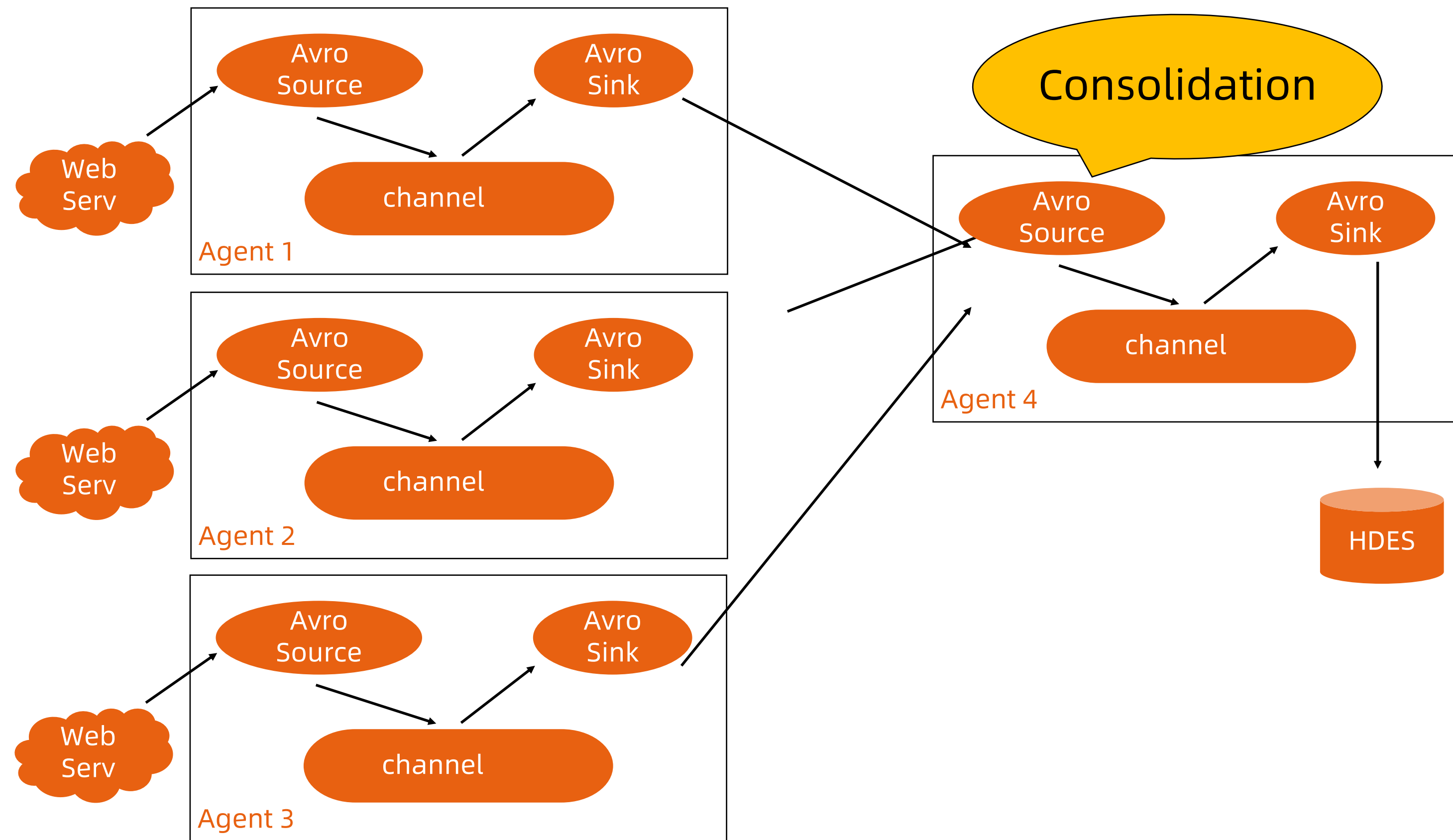
# 日志收集

- 每天会产生大量的日志（如搜索引擎的 PV、查询等），处理这些日志需要特定的日志系统，一般而言，这些系统需要具有以下特征：
  - 构建应用系统和分析系统的桥梁，并将它们之间的关联解耦；
  - 支持近实时的在线分析系统和类似于 Hadoop 之类的离线分析系统；
  - 具有高可扩展性，即当数据量增加时，可以通过增加节点进行水平扩展。
- 开源产品有 Apache Flume，由 Cloudera 开发的实时日志收集系统。



# Flume

- Flume 提供了大量内置的 Source、Channel 和 Sink 类型，不同类型的 Source, Channel 和 Sink 可以自由组合，组合方式基于用户设置的配置文件，非常灵活。

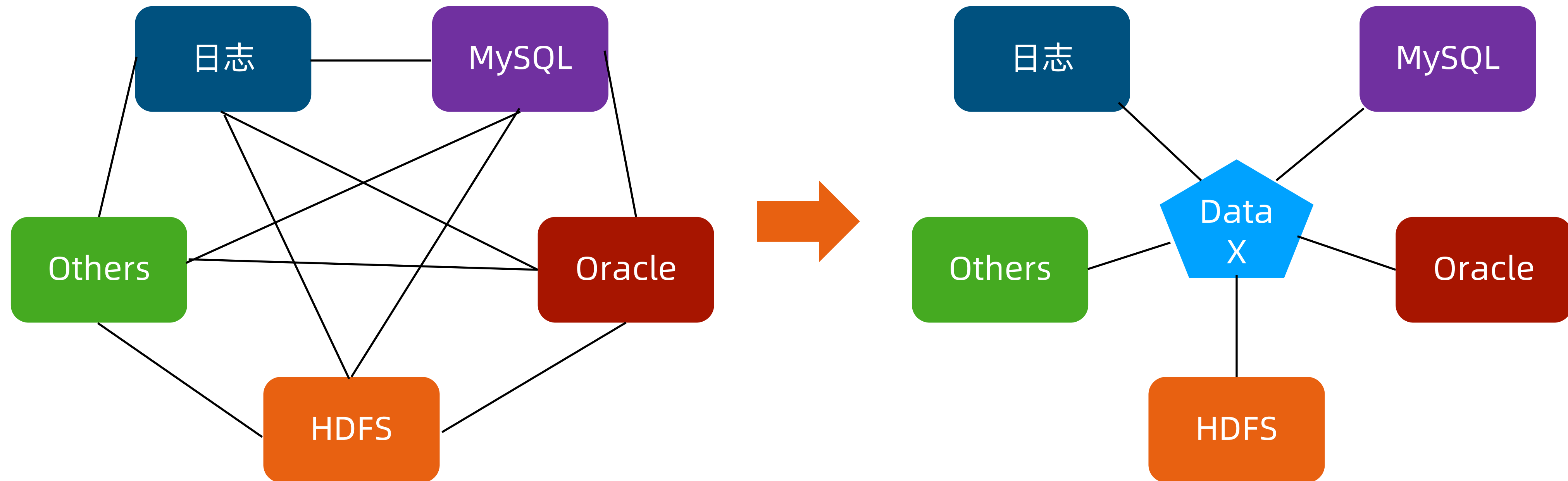




## 4. 数据库传输和相关开源框架

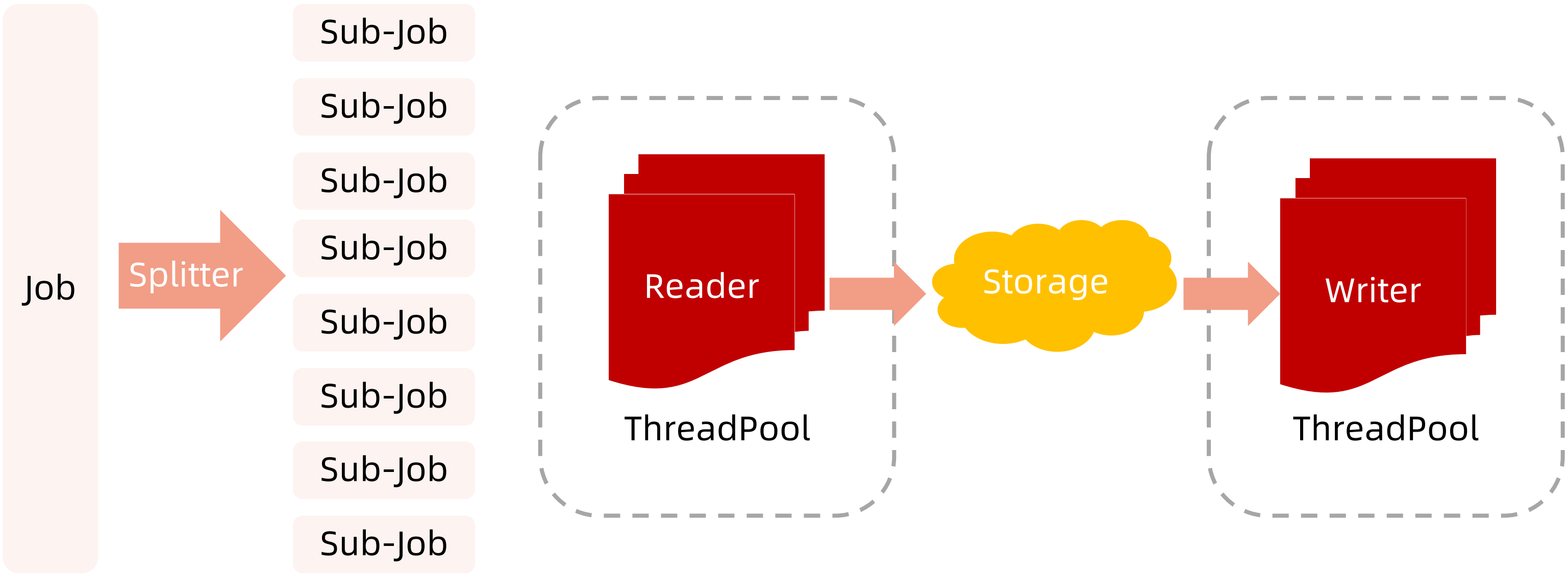
# 数据库传输

- DataX 是一个在异构的数据库/文件系统之间高速交换数据的工具，实现了在任意的数据处理系统（RDBMS/HDFS/Local Filesystem）之间的数据交换，由淘宝数据平台部门完成。



# DataX 的特点

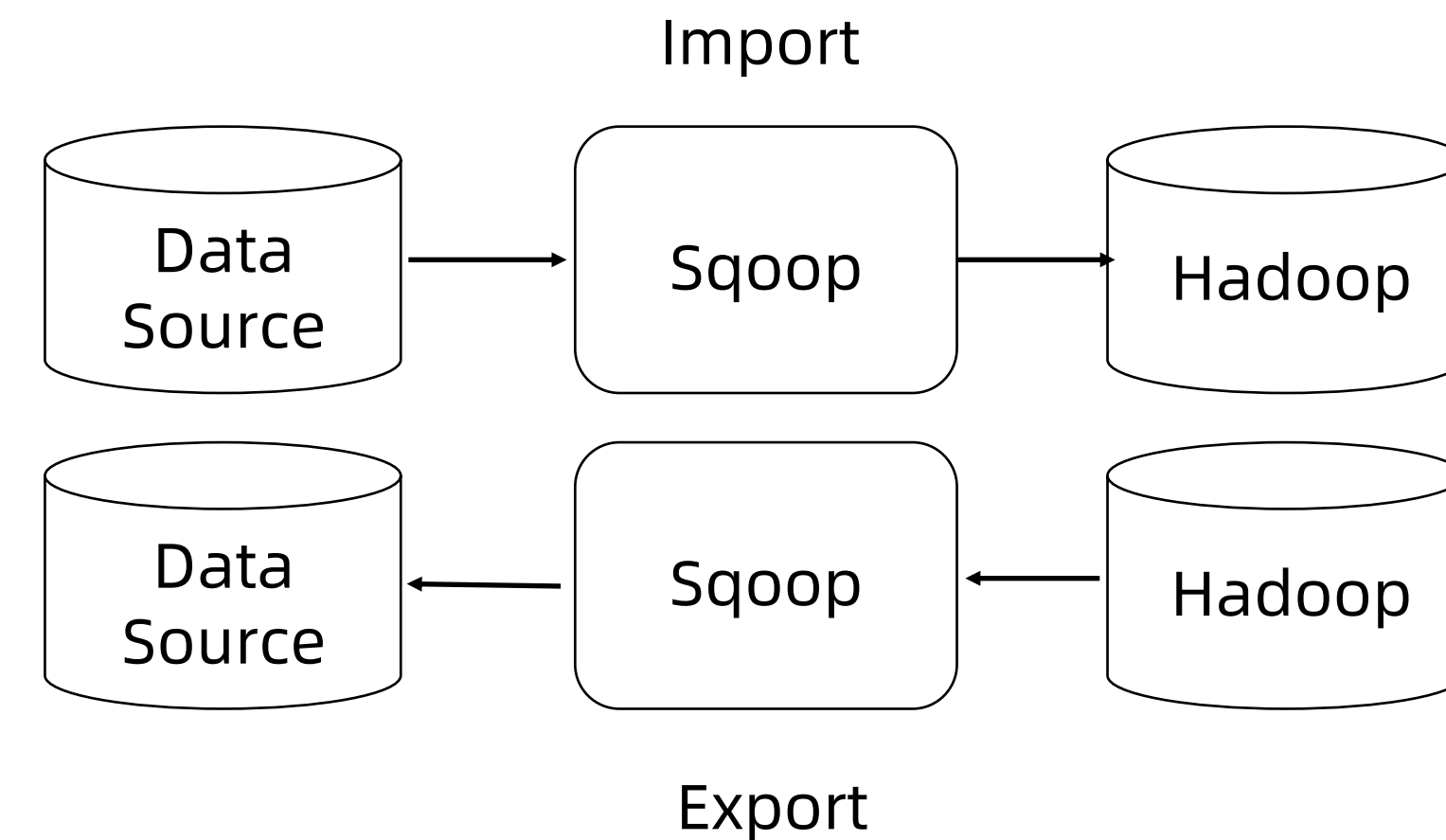
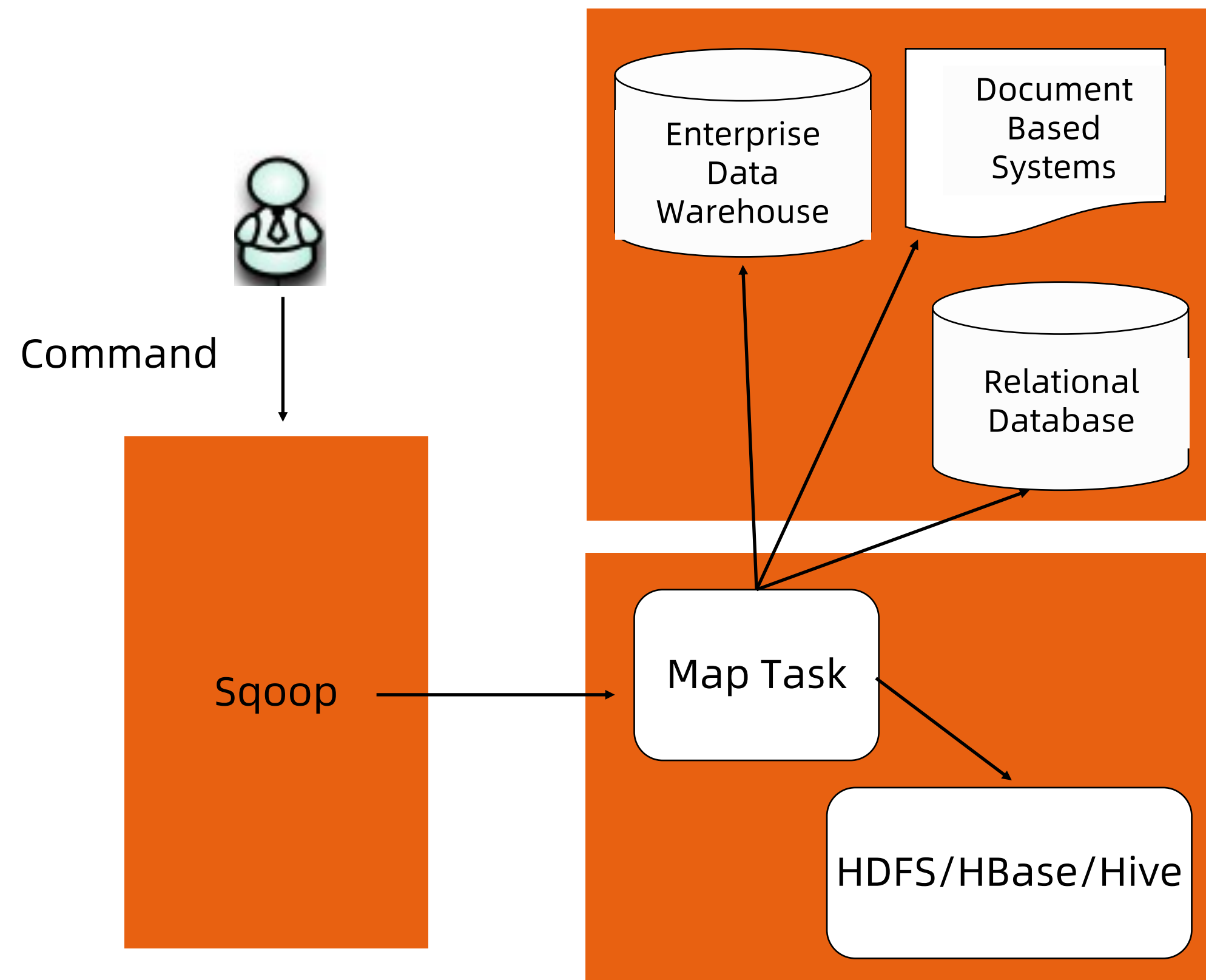
- 在异构的数据库/文件系统之间高速交换数据。
- 采用 Framework + plugin 架构构建，Framework 处理了缓冲、流控、并发、上下文加载等高速数据交换的大部分技术问题，提供了简单的接口与插件交互，插件仅需实现对数据处理系统的访问。
- 数据传输过程在单进程内完成，全内存操作，不读写磁盘，也没有 IPC。
- 开放式的框架，开发者可以在短时间内开发一个新插件以支持新的数据库/文件系统。





# Sqoop

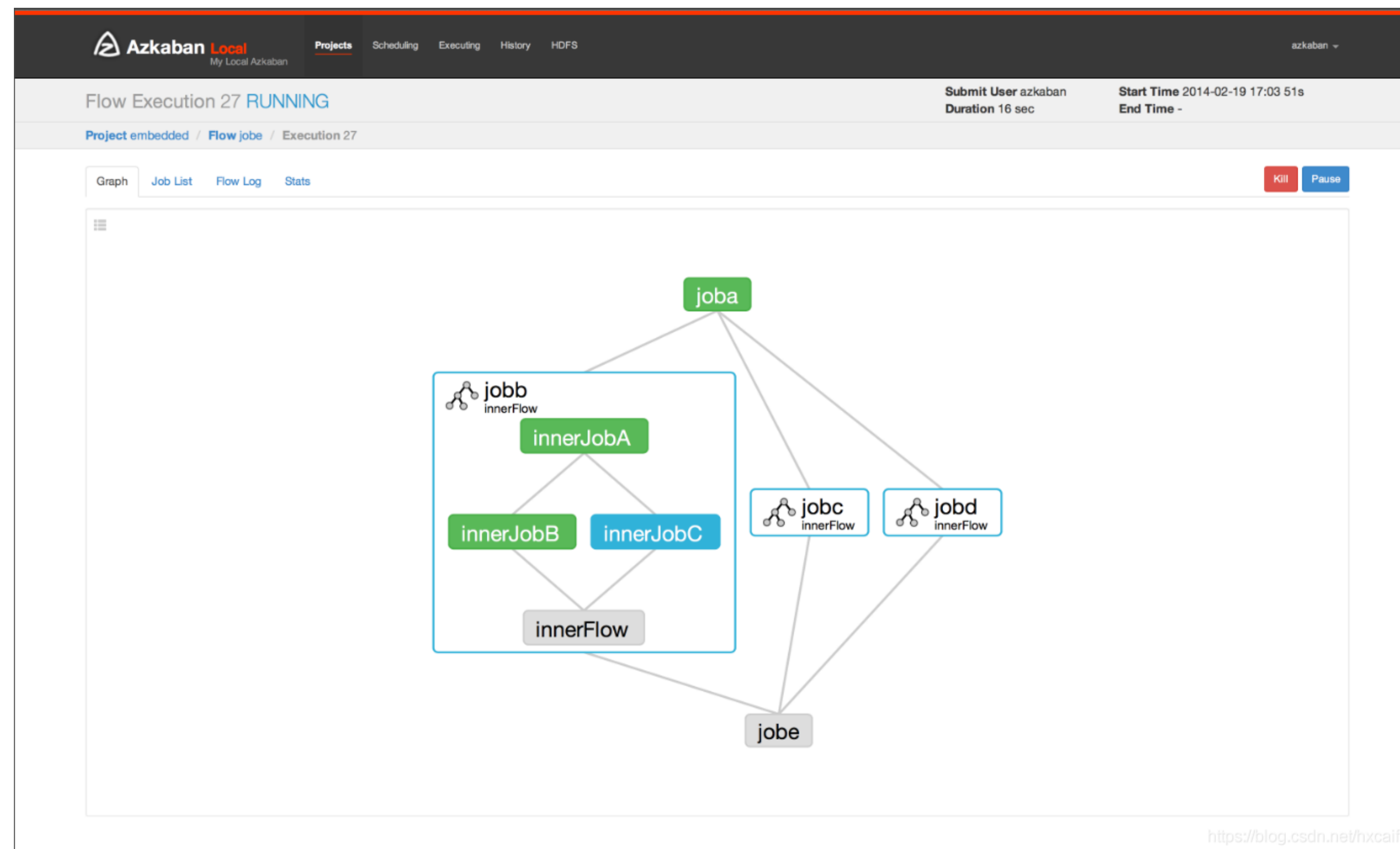
- Sqoop 中一大亮点就是可以通过 Hadoop 的 MapReduce 把数据从关系型数据库中导入数据到 HDFS。Sqoop 架构非常简单，其整合了 Hive、HBase 和 Oozie，通过 map-reduce 任务来传输数据，从而提供并发特性和容错。



## 5. 监控与调度系统

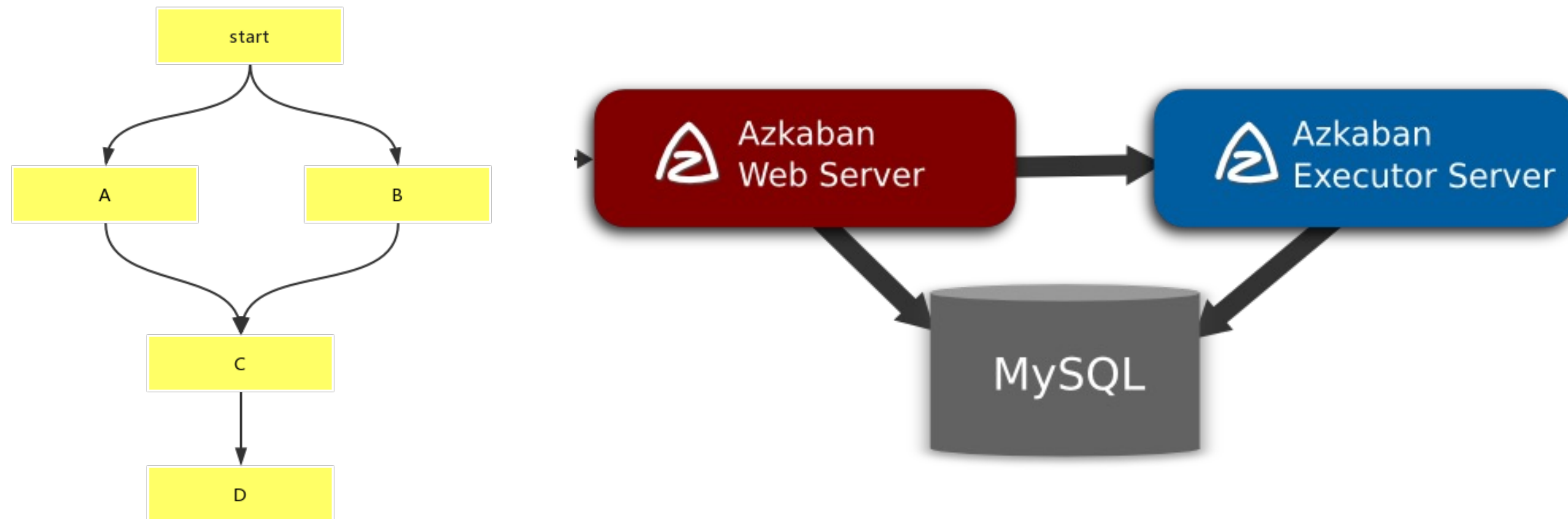
# Azkaban UI

- Azkaban 是由 Linkedin 公司推出的一个批量 workflow 任务调度器，主要用于在一个 workflow 内以一个特定的顺序运行一组工作和流程。Azkaban 使用 job 配置文件建立任务之间的依赖关系，并提供一个易于使用的 Web 用户界面维护和跟踪你的 workflow。



# Azkaban 特点

- 提供功能清晰，简单易用的 Web UI 界面
- 提供 job 配置文件快速建立任务和任务之间的依赖关系
- 提供模块化和可插拔的插件机制，原生支持 command、Java、Hive、Pig、Hadoop
- 基于 Java 开发，代码结构清晰，易于二次开发
- 提供了 RESTful 接口，方便我们平台定制化





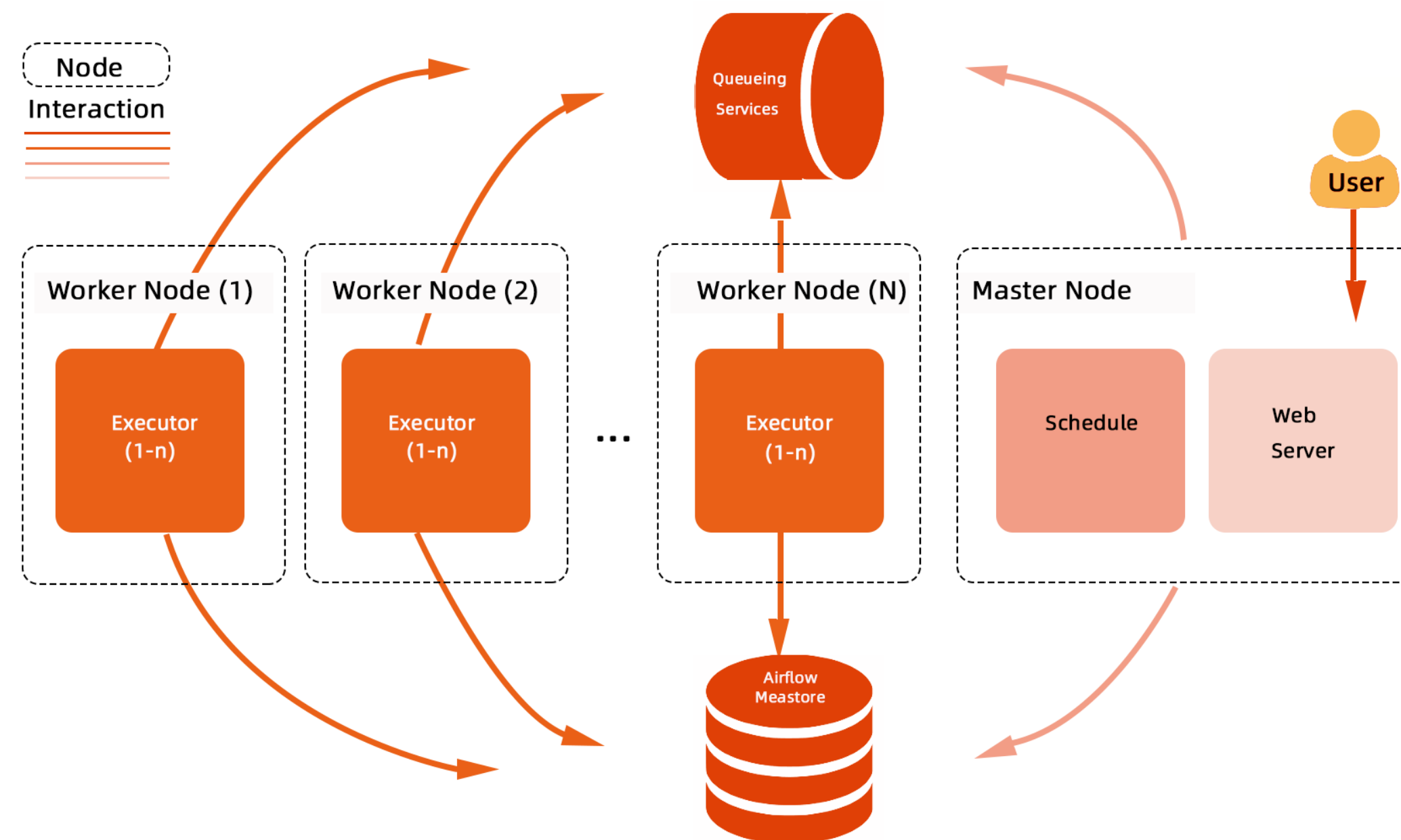
# Airflow

- Airbnb 开源的一款调度工具，核心特征是分布式、有向无环图（DAG），此外还提供了 UI 方便管理和监控，可看做 crontab 的升级版，功能更加丰富和完善。

The screenshot displays the Airflow web interface. At the top, there's a navigation bar with links for DAGs, Data Profiling, Browse, Admin, Docs, and About. The current time is 2018-09-07 22:29:47 UTC. Below the navigation bar, the main header shows 'On DAG: example\_bash\_operator' with a 'schedule: 0 0 \*\*\*' indicator. A toolbar contains various view options: Graph View (selected), Tree View, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, Refresh, and Delete. Below the toolbar, there are filters for 'success' status, a 'Base date' of '2018-09-06 00:00:01', 'Number of runs' set to '25', a 'Run' dropdown showing 'scheduled\_\_2018-09-06T00:00:00+00:00', a 'Layout' dropdown set to 'Left->Right', and a 'Go' button. A search bar is also present. Below the filters, there are tabs for 'BashOperator' and 'DummyOperator'. A legend shows status indicators: success (green), running (blue), failed (red), skipped (pink), retry (yellow), queued (grey), and no status (white). The main area displays a DAG graph with nodes: 'runme\_0', 'runme\_1', 'runme\_2', 'run\_after\_loop', 'also\_run\_this', and 'run\_this\_last'. The graph shows 'runme\_0' leading to 'run\_after\_loop', 'runme\_1' leading to 'run\_after\_loop', and 'runme\_2' leading to 'also\_run\_this'. Both 'run\_after\_loop' and 'also\_run\_this' lead to 'run\_this\_last'. A refresh button is in the top right corner of the graph area.

# Airflow 架构

- Master-Slave 架构，存在两种主要角色 Scheduler 和 Executor，以及辅助角色 WebServer：
  - Scheduler：调度器，周期性地轮询元数据库，筛选需要被执行的 DAG（同一时间只能存在一个）；
  - Executor：执行器，负责任务的执行；
  - WebServer：Web 服务器，为前端提供监控管理调度任务的能力。



## 6. 元数据系统

# 什么是元数据？

- 元数据 Metadata 狭义的解释是用来描述数据的数据。
- 广义来看，除了业务逻辑直接读写处理的那些业务数据，所有其它用来维持整个系统运转所需的信息 / 数据都可以叫作元数据。
  - 比如数据表格的 Schema 信息、任务的血缘关系、用户和脚本 / 任务的权限映射关系信息等等。
- 管理 Metadata 信息的目的：一方面是为了让用户能够更高效地挖掘和使用数据，另一方面是为了让平台管理人员能更加有效地做好系统维护管理工作。



# 收集元数据

- 数据的表结构 Schema 信息
- 数据的空间存储、读写记录、权限归属和其它各类统计信息
- 数据的血缘关系信息
- 数据的业务属性信息

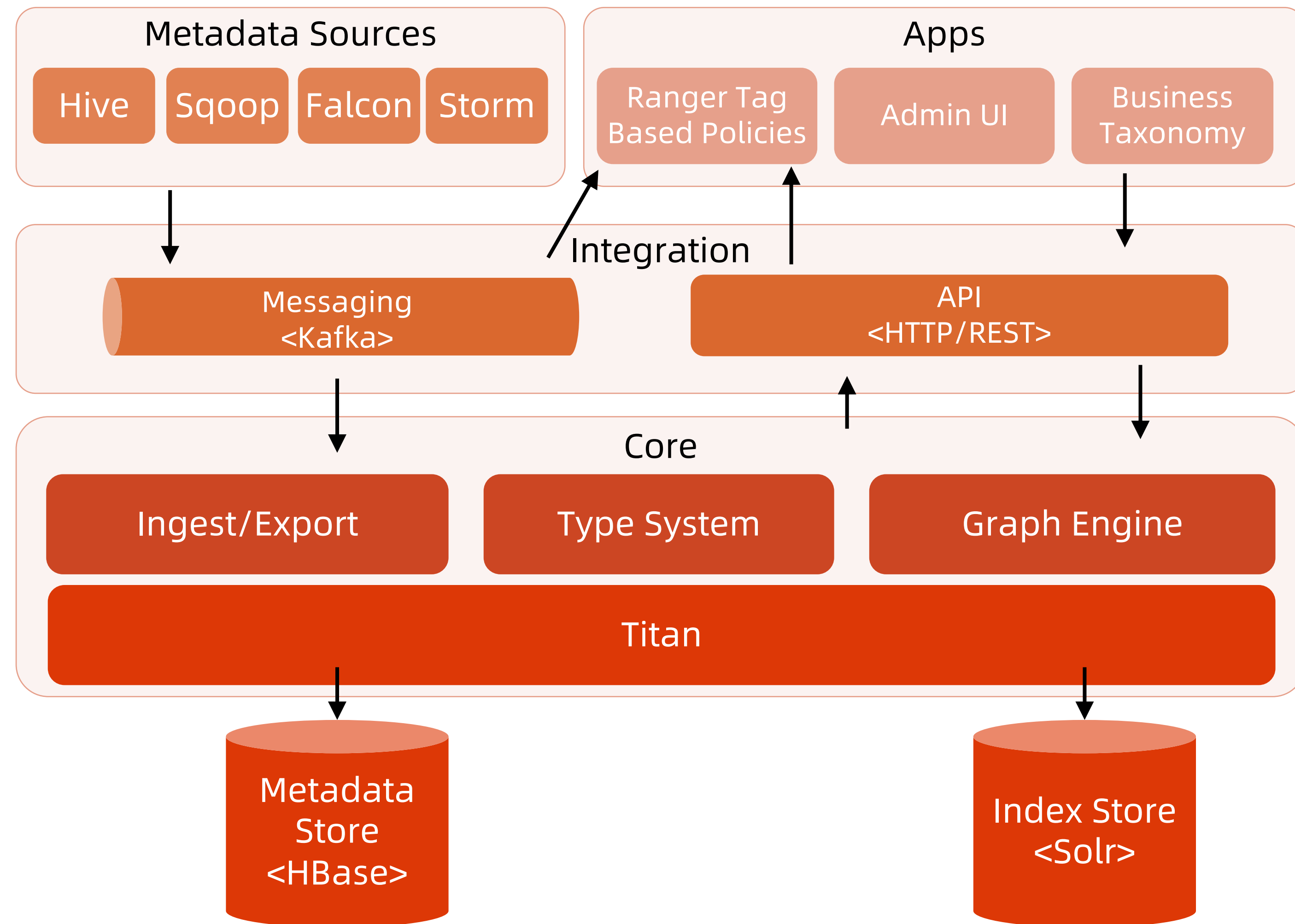
# 开源元数据管理：Apache Atlas

- Hadoop 的数据管理和元数据框架。
- Apache Atlas 为组织提供开放的元数据管理和治理能力，以建立其数据资产的目录，对这些资产进行分类和管理，并为数据科学家、分析师和数据治理团队提供围绕这些数据资产的协作能力。

# Atlas 主要功能

- 数据分类
  - 定义、注释和自动捕获数据集和底层之间的关系元素包括源、目标和派生过程
- 安全审计
  - 数据访问的日志审计
- 搜索和血缘关系
  - 元数据信息及数据之间的血缘
- 安全与策略引擎
  - 结合 ApacheRanger 来设置数据的访问权限

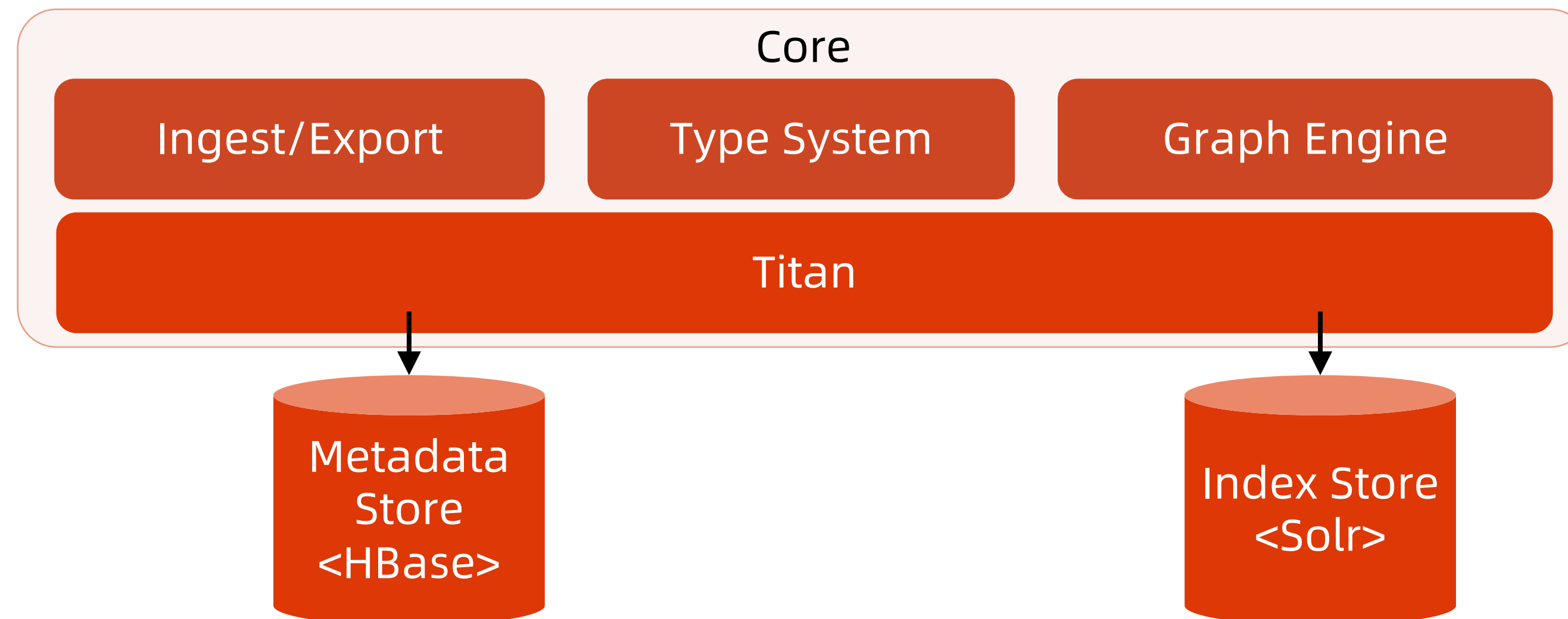
# Atlas 的架构



# Atlas 的架构

- Core 层

- 类型系统（Type System）：用户为他们想要管理的元数据对象定义模型。Type System 称为“实体”的“类型”实例，表示受管理的实际元数据对象。
- 图形引擎（Graph Engine）：Atlas 在内部使用 Graph 模型持久保存它管理的元数据对象。
- 采集/导出（Ingest/Export）：采集组件允许将元数据添加到 Atlas。同样，“导出”组件将 Atlas 检测到的元数据更改公开为事件。

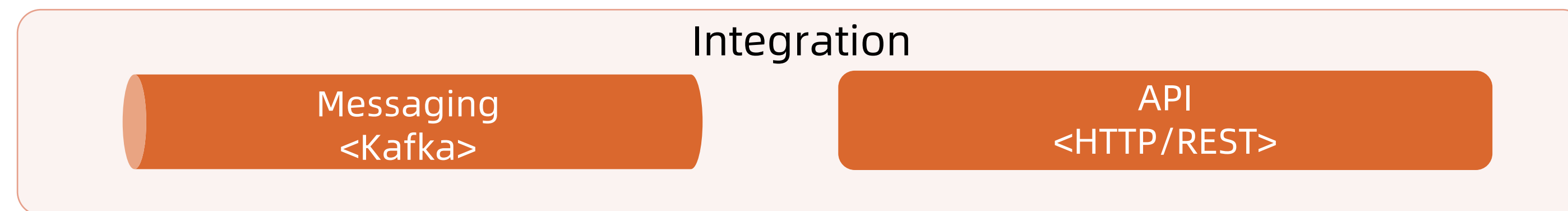




# Atlas 的架构

- Integration 层

- API: Atlas 的所有功能都通过 REST API 向最终用户暴露, 该 API 允许创建、更新和删除类型和实体。它也是查询和发现 Atlas 管理的类型和实体的主要机制。
- Messaging: 除了 API 之外, 用户还可以选择使用基于 Kafka 的消息传递接口与 Atlas 集成。



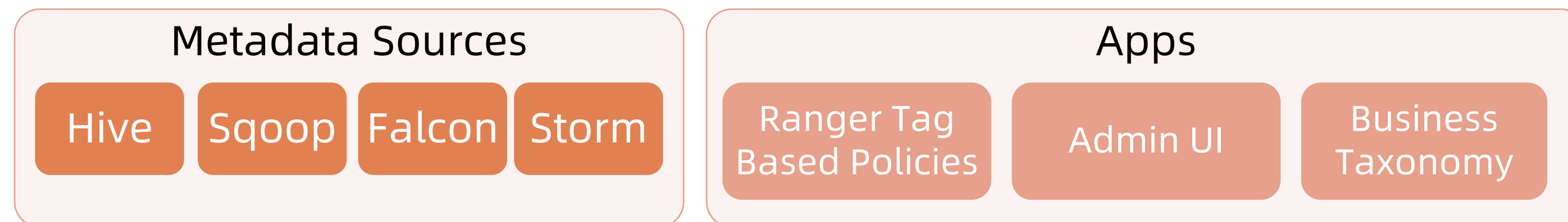
# Atlas 的架构

- Metadata Sources 层

- 目前，Atlas 支持从以下来源提取和管理元数据：HBase、Hive、Sqoop、Storm、Kafka。

- Applications 层

- Atlas Admin UI：该组件是一个基于 Web 的应用程序，允许数据管理员和科学家发现和注释元数据。这里最重要的是搜索界面和类似 SQL 的查询语言，可用于查询 Atlas 管理的元数据类型和对象。
- Ranger Tag Based Policies：权限管理模块。



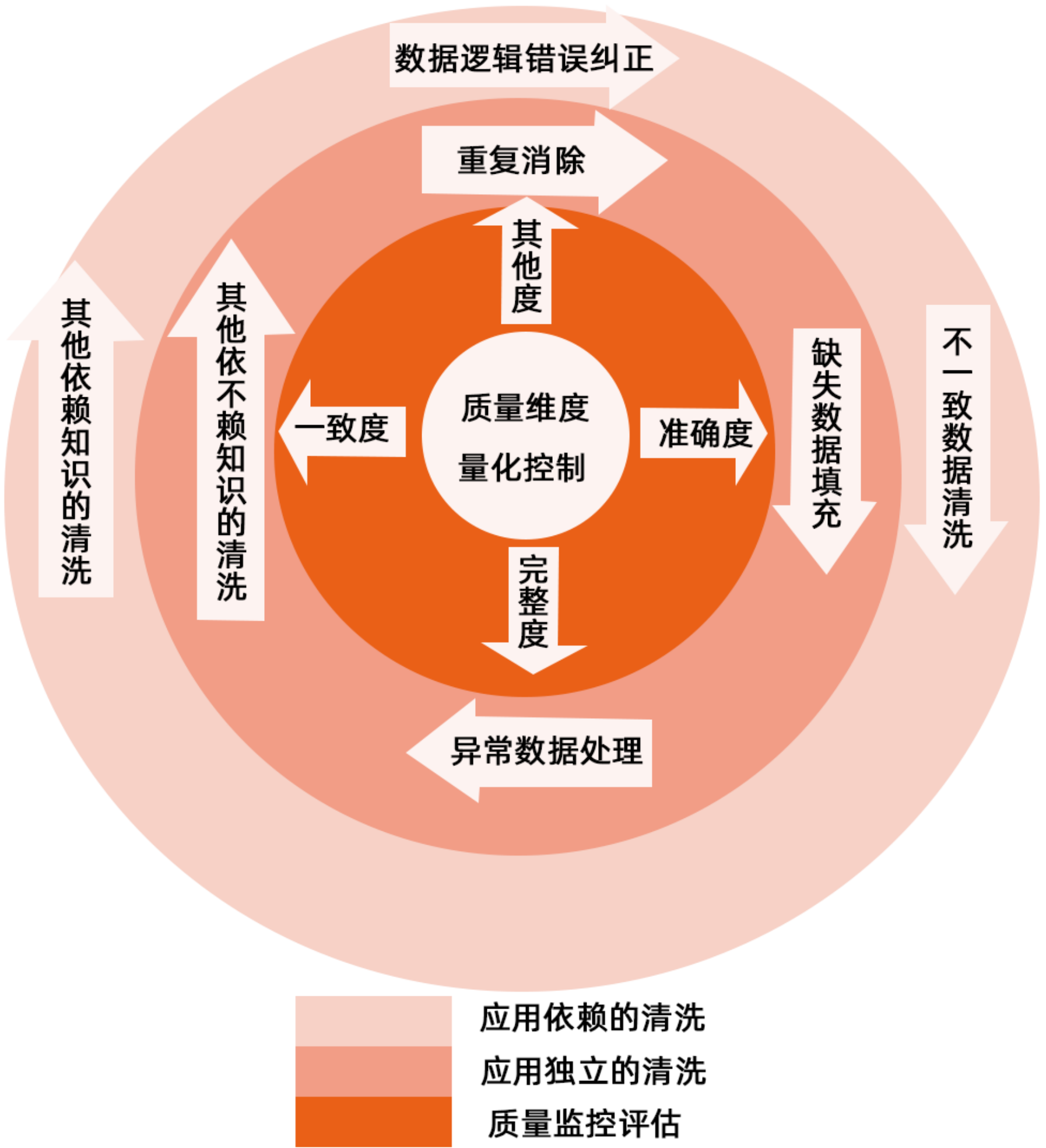
## 7. 数据质量系统

# 数据质量的意义

数据质量的好坏是决定数据仓库成功的关键，可以从下列方面衡量系统中的数据质量：

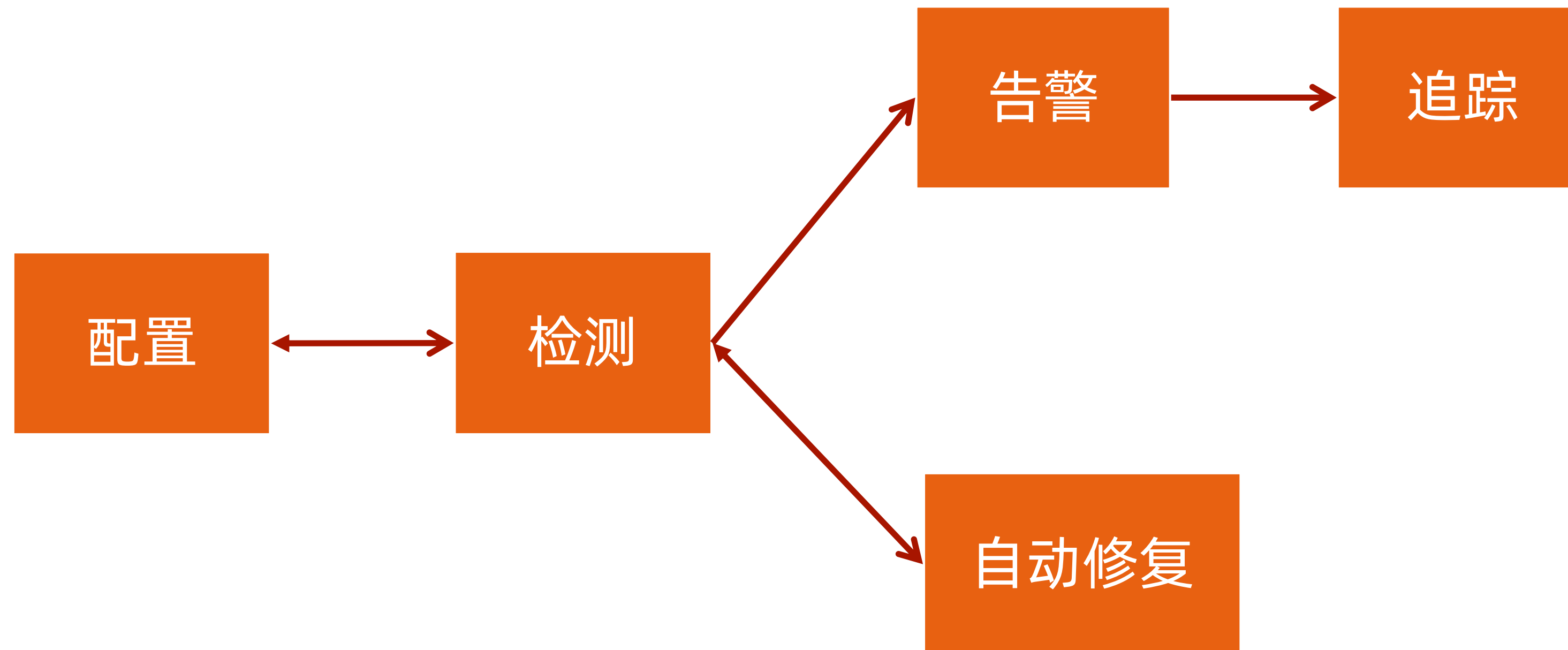
- **准确性**：存储在系统中的关于一个数据元素的值是这个数据元素的正确值；
- **域完整性**：一个属性的数值在合理且预定义的范围之内；
- **数据类型**：一个数据属性的值通常是根椐这个属性所定义的数据类型来存储的；
- **一致性**：一个数据字段的形式和内容在多个源系统之间是相同的；
- **冗余性**：相同的数据在一个系统中不能存储在超过一个地方；
- **完整性**：系统中的属性不应该有缺失的值。

# 解决框架





# 系统基本结构



# 数据质量系统 Demo

数据质量详情 dpdim\_bi\_top\_middle\_shopnum

负责人

阚冉冉

元数据探查【最新】 高优先级0 中优先级0 低优先级0

详细数据探查 高优先级0 中优先级0 低优先级0

2015-07-31



探查【统计分析】

记录数



列名	列类型	NULL	最小值   最小长度	平均值   平均长度	最大值   最大长度	0   空字符串	特殊字符	重复率	自定义	配置
city_id	bigint									
city_name	string									
top_number	bigint									
middle_number	bigint									
dw_add_ts	string									

自定义规则 高优先级0 中优先级0 低优先级0

# 数据质量系统 Demo



## 数据质量监控

数据日期

2015-07-31

📅

状态

有失败的

▼

优先级

全部

▼

负责人

全部

▼

数据源

全部

▼

数据库

全部

▼

表名

模糊搜索

🔍 查询

▶ hive - bi - dpdim\_dpid\_backup 失败1 [ 负责人: 张磊.s, 邹玉静 ]

▶ hive - bi - dpdim\_dpid\_cnt 失败1 成功1 [ 负责人: 张磊.s, 邹玉静 ]

▼ hive - bi - dpdim\_dp\_shop 失败4 成功4 [ 负责人: 曹一帆 ]

状态	优先级	数据质量类型	取值范围	实际值	操作
失败	低	shop_name: NULL值数	[0,10]	40	
失败	低	cat0_name: NULL值数	[0,1000]	3,295	
失败	低	自定义探查: CASE WHEN ( not cat0_id between 1 and 100 ) THEN ( 1 ) ELSE ( 0 ) END	[0,1000]	1,388	
失败	中	自定义SQL	{0}		
成功	低	shop_id: NULL值数	{0}	0	
成功	中	shop_id: 重复率	{0}	0	
成功	低	city_id: NULL值数	{0}	0	
成功	低	自定义探查: CASE WHEN ( city_tier not in('K','A','B','C','D','NOTSET')) THEN ( 1 ) ELSE ( 0 ) END	{0}	0	

▶ 调度重跑

▶ 数据质量重跑

▶ hive - bi - dpdim\_trade\_product\_group 失败1 成功3 [ 负责人: 郭闻铭, 曹杰 ]

▶ hive - bi - dpdm\_consume\_base 失败5 成功8 [ 负责人: 张敬云, 赵宏 ]

# Apache Griffin

- Griffin 起源于 eBay 中国，并于 2016 年 12 月进入 Apache 孵化器，Apache 软件基金会于 2018 年 12 月 12 日正式宣布 Apache Griffin 毕业成为 Apache 顶级项目。
- Griffin 数据质量监控工具正是为了解决数据质量问题而诞生的开源解决方案。
- Griffin 是属于模型驱动的方案，基于目标数据集合或者源数据集（基准数据），用户可以选择不同的数据质量维度来执行目标数据质量的验证。
- 支持两种类型的数据源：batch 数据和 streaming 数据。对于 batch 数据，可以通过数据连接器从 Hadoop 平台收集数据；对于 streaming 数据，可以连接到诸如 Kafka 之类的消息系统来做近似实时数据分析。在拿到数据之后，模型引擎将在 Spark 集群中计算数据质量。

# Griffin 特点

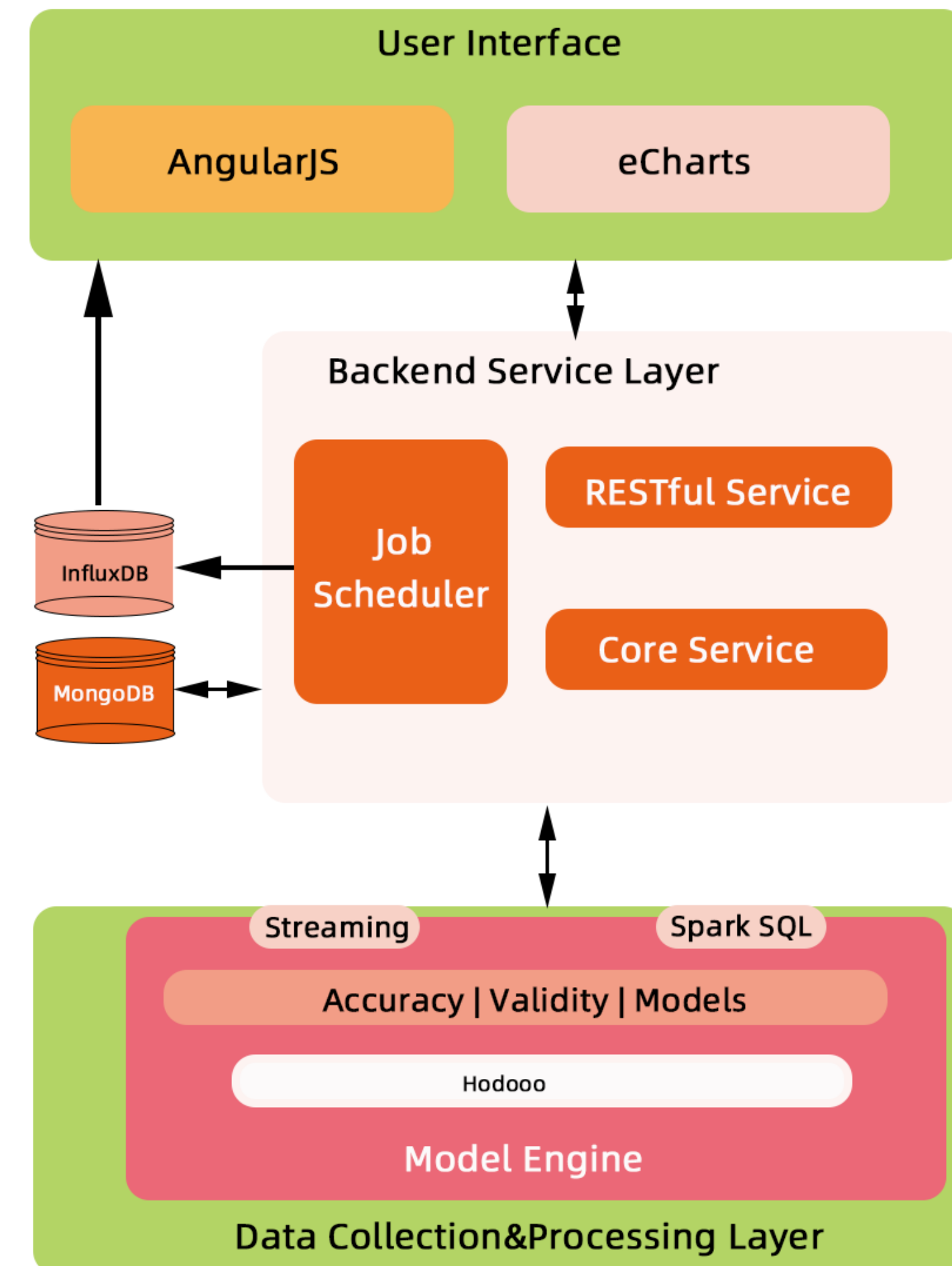
- 可配置、可自定义的数据质量验证。
- 基于 Spark 的数据分析，可以快速计算数据校验结果。
- 历史数据质量趋势可视化。



# Griffin 架构

Griffin 系统主要分为：

- 数据收集处理层（Data Collection & Processing Layer）
- 后端服务层（Backend Service Layer）
- 用户界面（User Interface）



# Griffin Demo: 选择数据源

GriFFinHealthMeasuresJobsMy Dashboard

Create DQ ModelBack

Accuracy

Definition: Measured by how the values agree with an identified source of truth

1

Choose Source

2

Choose Target

3

Mapping Source and Target

4

Partition Configuration

5

1. Select the source dataset and fields which will be used for comparison
2. Select the target dataset and fields which will be used for comparison
3. Mapping the target fields with source
4. Set partition configuration for source dataset and target dataset
5. Set basic configuration for your model (name, system, threshold, etc.)

**Example:** suppose source table A has 1000 records and target table B only has 999 records perfectly matched with A for selected fields, then Accuracy Rate(%) =  $999/1000 * 100\% = 99.9\%$ 

Data Profiling

Definition: Data profiling is the process of examining the data available in an existing data set and collecting statistics and information about that data

1

Choose Target

2

Define/Select Models

3

4

1. Select the target dataset and fields which want to be checked
2. Define your syntax check logic which will be applied on the selected fields
3. Set partition configuration for target dataset
4. Set basic configuration for your model(name, system, threshold, etc.)

**Example:** Check the data range(minimum, maximum) within a set of allowable values

Publish

Definition: Publish is the process of storing user's own quality data and visualizing it

1

Configuration

1. Set basic configuration for your model(name, system, threshold, etc.)

**Example:** any data ...

# Griffin Demo: 选择账单明细源表字段

Griffin

HealthMeasuresJobsMy Dashboard

Create Measure

Back

1

2

3

4

5

Choose Source

Choose Target

Mapping Source and Target

Partition Configuration

Configuration

This step let you choose the single source of truth for data quality comparision with target. Currently you can only select the attributes from one schema

Please select schema

▼ default

bill\_detail\_source

bill\_detail\_target

Select attributes

View schema: default.bill\_detail\_source

<input type="checkbox"/>	Column Name	Type	Comment
<input checked="" type="checkbox"/>	id	bigint	
<input checked="" type="checkbox"/>	vendor_code	string	
<input type="checkbox"/>	vendor_name	string	
<input checked="" type="checkbox"/>	order_num	string	
<input checked="" type="checkbox"/>	total_amount	decimal(6,2)	

Next

# Griffin Demo：选择账单明细目标表字段

GriFFin

HealthMeasuresJobsMy Dashboard

Create Measure

Back

1

Choose Source

2

Choose Target

3

Mapping Source and Target

4

Partition Configuration

5

Configuration

This step let you choose the target for data quality comparision with source

Please select schema

▼ default

bill\_detail\_source

bill\_detail\_target

Select attributes

View schema: bill\_detail\_target

<input type="checkbox"/>	Column Name	Type	Comment
<input checked="" type="checkbox"/>	id	bigint	
<input checked="" type="checkbox"/>	vendor_code	string	
<input type="checkbox"/>	vendor_name	string	
<input checked="" type="checkbox"/>	order_num	string	
<input checked="" type="checkbox"/>	total_amount	decimal(6,2)	

Back

Next



# Griffin Demo: 设置源表和目标表的校验字段映射关系

GriFFin

Health

Measures

Jobs

My Dashboard

Create Measure

Back

✓

Choose Source

✓

Choose Target

3

Mapping Source and Target

4

Partition Configuration

5

Configuration

This step let you map the target data fields to source fields, you can choose the related fields from dropdown list of source

Map the fields

Target Fields	Map To	Source Fields
default.bill_detail_target.id	=	default.bill_detail_source.id
default.bill_detail_target.vendor_code	=	default.bill_detail_source.vendor_code
default.bill_detail_target.order_num	=	default.bill_detail_source.order_num
default.bill_detail_target.total_amount	=	default.bill_detail_source.total_amount

Accuracy Calculation Formula as Below:

Accuracy Rate(%) =  $\frac{\text{Total Count of Matched records between 4 bill_detail_target and 4 bill_detail_source fields}}{\text{Total Count of records in default.bill_detail_source}}$  x 100%

Back

Next



# Griffin Demo：选择数据分区、条件和输出结果文件



Griffin

Health

Measures

Jobs

My Dashboard

Create Measure

Back

✓

Choose Source

✓

Choose Target

✓

Mapping Source and Target

4

Partition Configuration

5

Configuration

Please complete the partition configuration for bill\_detail\_source and bill\_detail\_target

Required Information

default.bill\_detail\_source

Where:

dt=#YYYYMMdd# AND hour=#HH#

Partition Size:

1

day

Time Zone:

has Done file

please write the Done file path relative to hdfs://griffin:9000/user/hive/warehouse/bill\_detail\_source

/dt=#YYYYMMdd#/hour=#HH#/\_DONE

default.bill\_detail\_target

Where:

dt=#YYYYMMdd# AND hour=#HH#

Partition Size:

1

day

Time Zone:

has Done file

please write the Done file path relative to hdfs://griffin:9000/user/hive/warehouse/bill\_detail\_target

/dt=#YYYYMMdd#/hour=#HH#/\_DONE

Back

Next

# Griffin Demo：设置验证项目名称和描述

Griffin

HealthMeasuresJobsMy Dashboard

Create Measure

Back

✓

✓

✓

✓

5

Choose SourceChoose TargetMapping Source and TargetPartition ConfigurationConfiguration

Please setup the measure required information

Required Information

Measure Name✓:

bill\_detail\_check

Measure Description:

bill detail sync check

Measure Type\*::

accuracy

Source\*::

bill\_detail\_source

Target\*::

bill\_detail\_target

Owner:


test

After submitted, please go to "Measures" to check the measure status

BackSubmit

# Griffin Demo: 提交后在列表看到度量的信息











 GriFFin

[Health](#)[Measures](#)[Jobs](#)[My Dashboard](#)

+ Create Measure

Back

Measure Name	Measure Type	Description	Action
bill_detail_miss_order	accuracy	check bill detail miss order sync	 
bill_detail_vendor_profiling	profiling	bill_detail_vendor_profiling	 
bill_detail_profiling	profiling	detail data accuracy	 
bill_detail_check	accuracy	detail data accuracy	 

# Griffin Demo: 创建 spark job

GriFFin

HealthMeasuresJobsMy Dashboard

Create Job

Back

Please setup the job required information

Required Information

Job Name✔:

bill\_detail\_check\_job

Measure Name\*:

bill\_detail\_check

Cron Expression\*:

0 0/20 \*\*\* ?

please select data range for default.bill\_detail\_source

One step means a partition size,and default.bill\_detail\_source partition size = 1day

begin : -1

end : 0

please select data range for default.bill\_detail\_target

One step means a partition size,and default.bill\_detail\_target partition size = 1day

begin : -1

end : 0

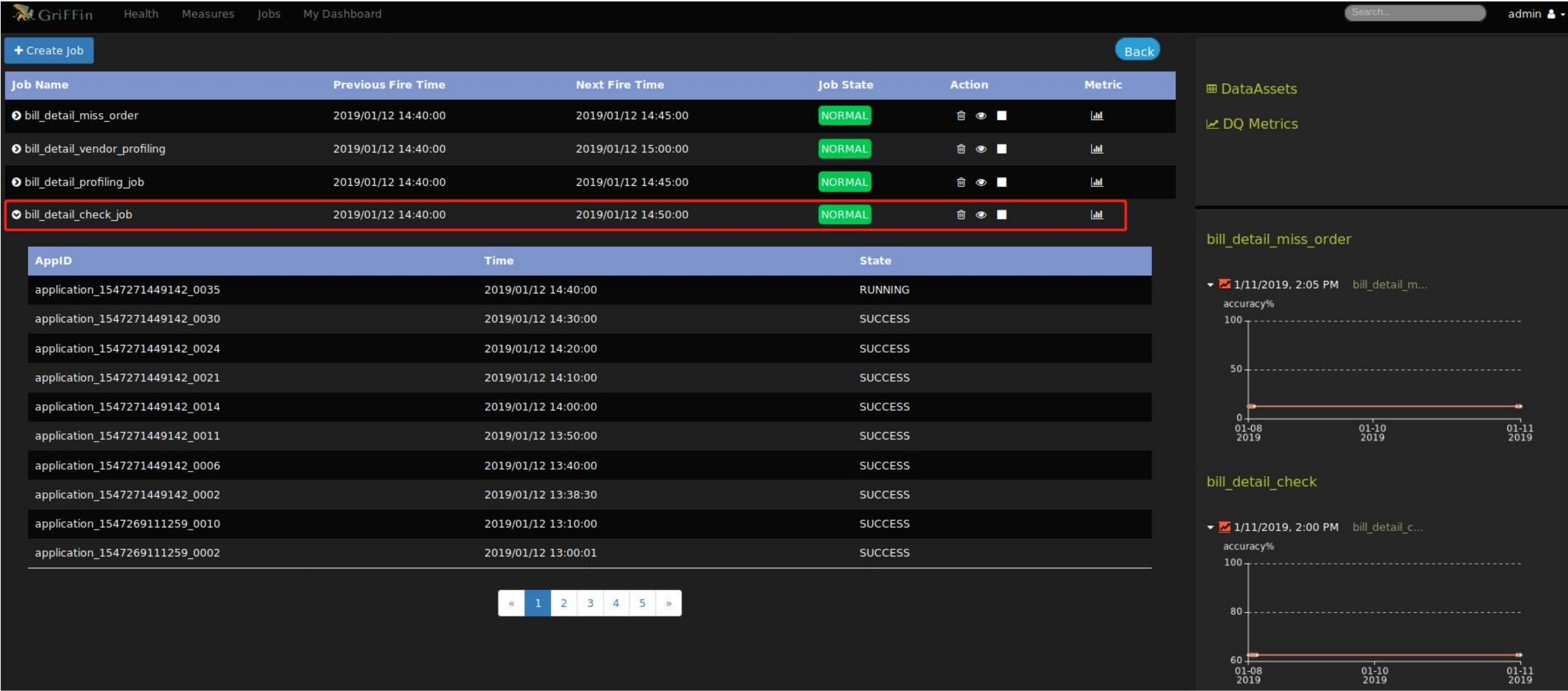
After submitted, please go to "Jobs" to check the job status

Back

Submit

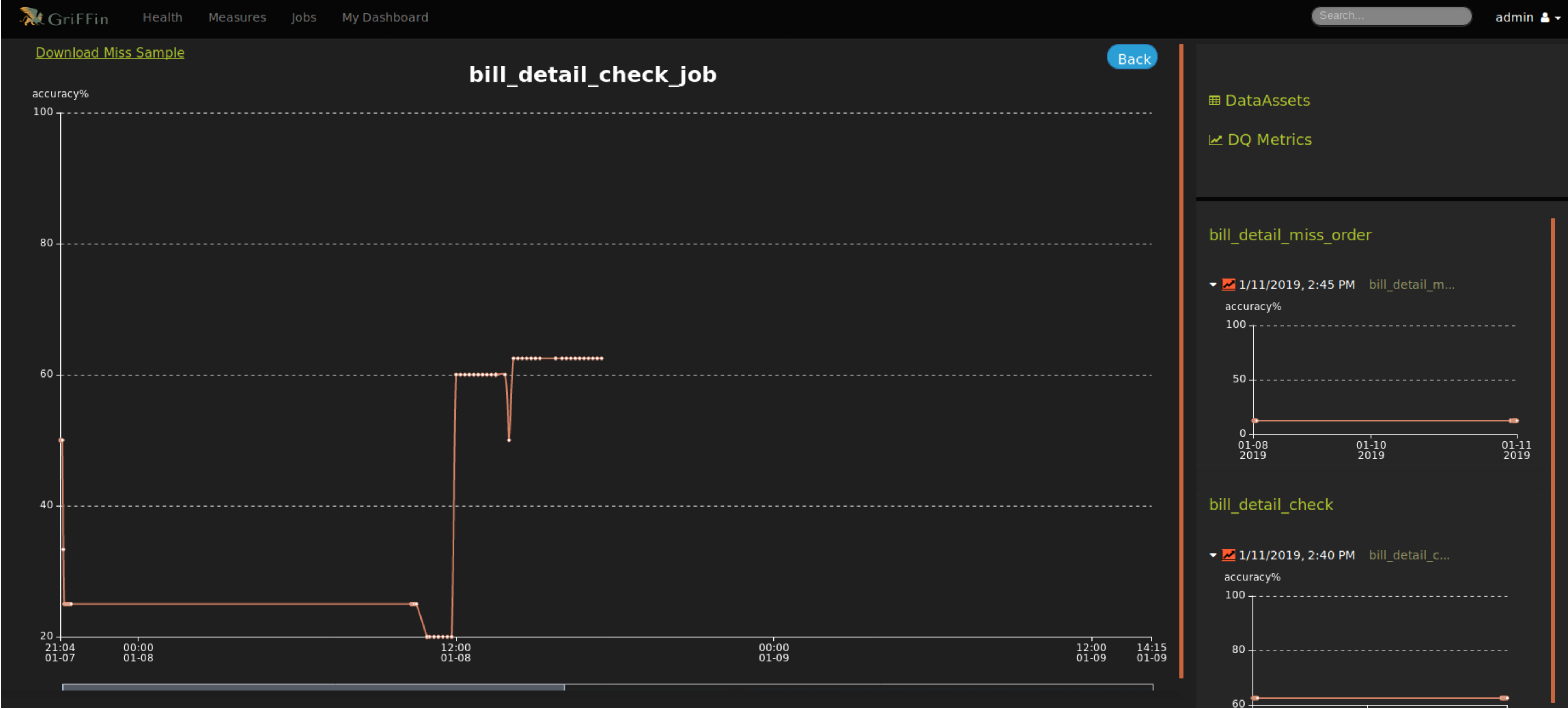


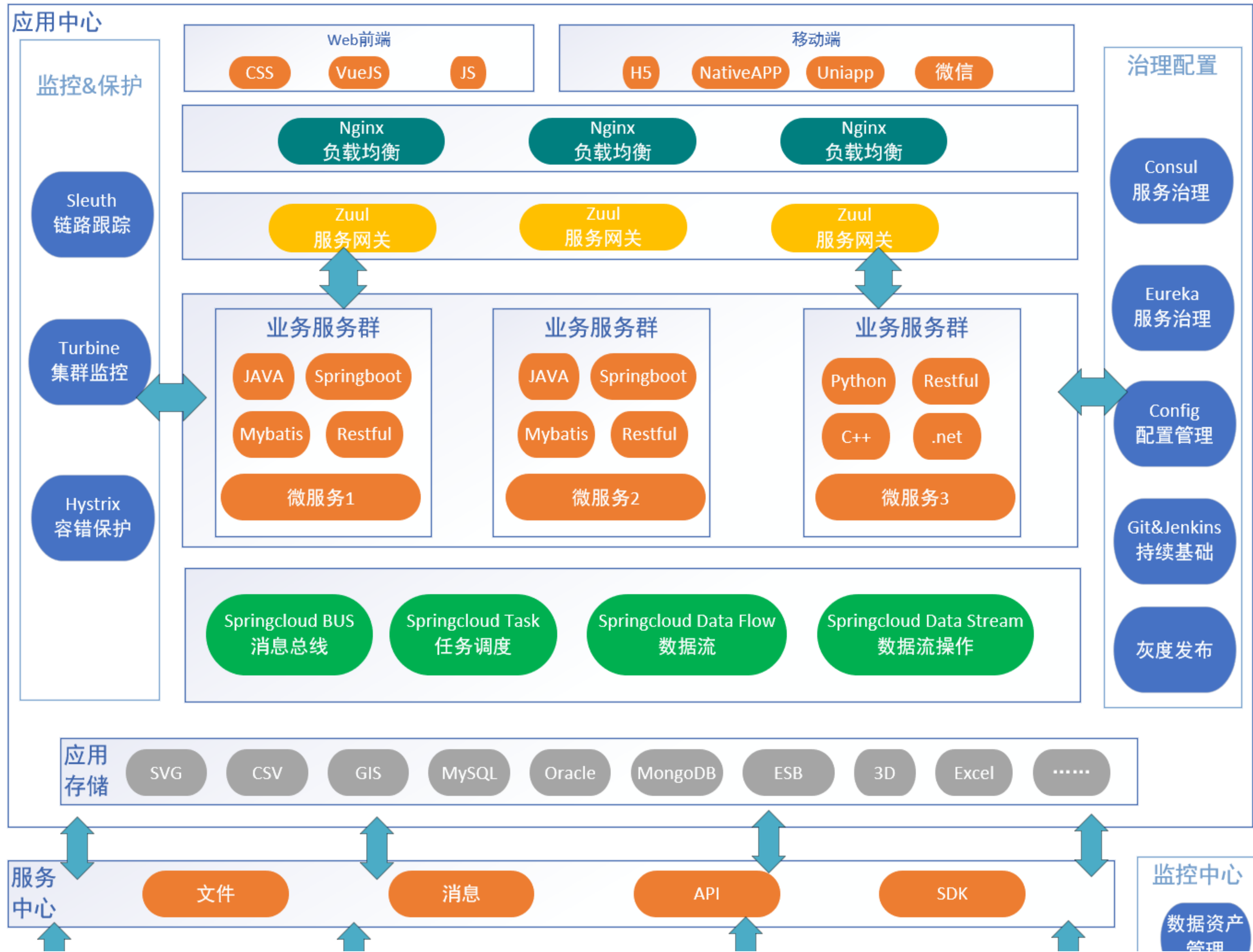
# Griffin Demo：选择源表和目标表数据范围

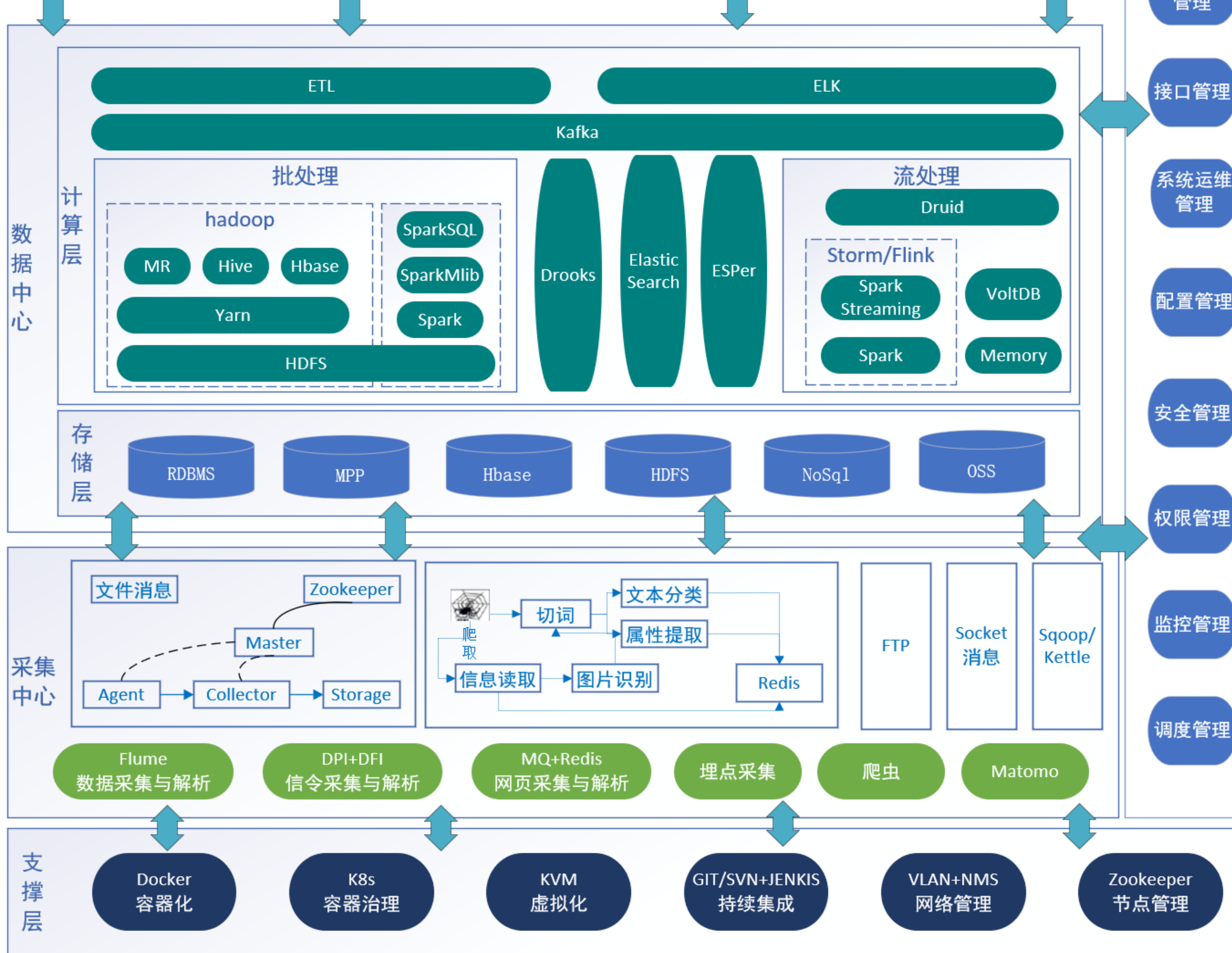




# Griffin Demo：在控制面板上监控数据质量







# Reference

<https://blog.csdn.net/jishulaozhuanjia/article/details/104816371>

[https://blog.csdn.net/Li\\_\\_Sir/article/details/102721293](https://blog.csdn.net/Li__Sir/article/details/102721293)



# THANKS

 极客时间 | 训练营