About Me

- University of Southern Calif
- IBM Watson
- Airbnb Data Platform

3 Million

TOTAL HOMES ON AIRBNB

65K

CITIES

191+

COUNTRIES

Agenda

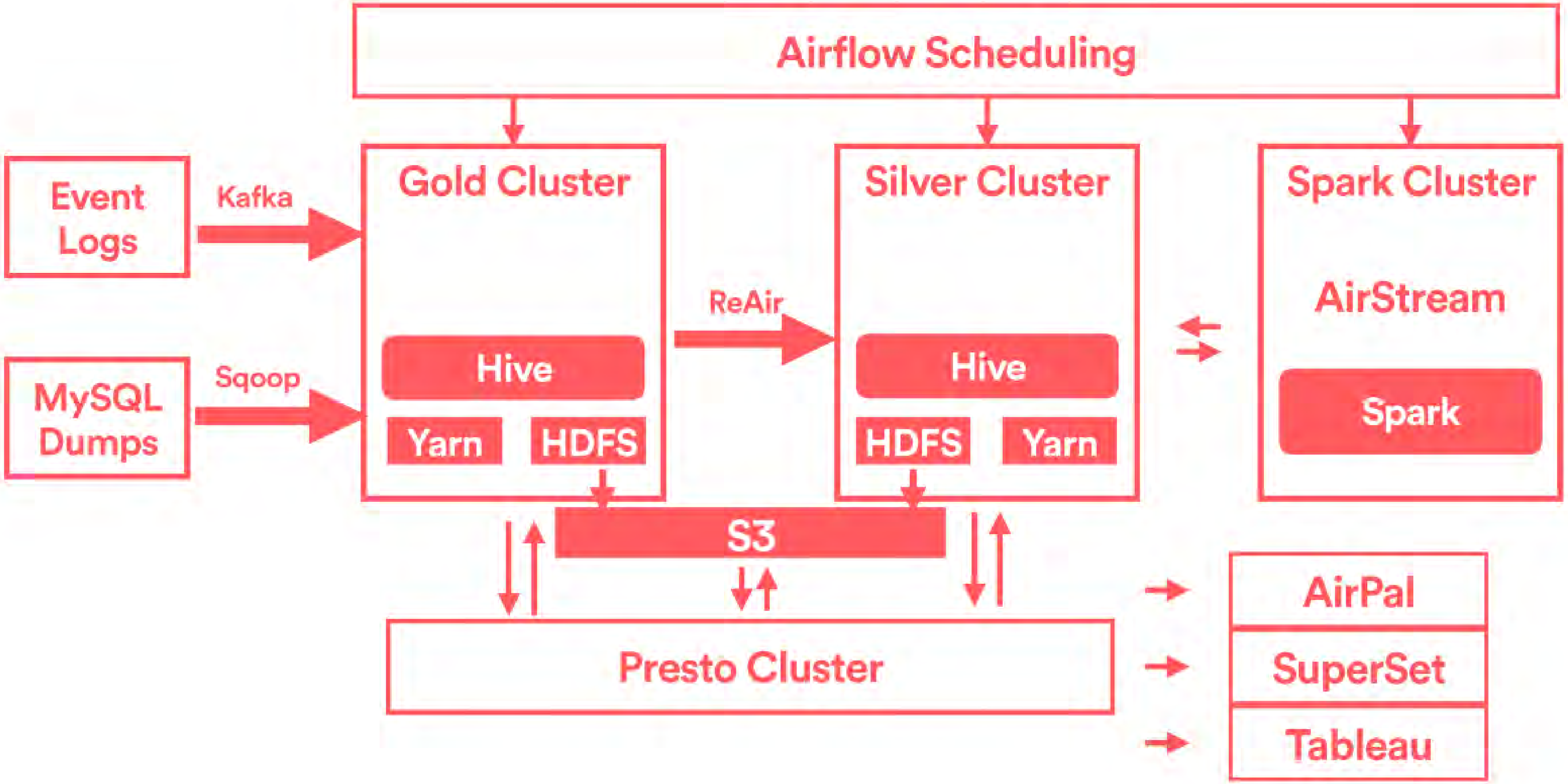
- Data Platform at Airbnb
  - High-level overview
  - ReAir: Incremental Data Replication
  - Airflow: Scheduling
  - Analytics: Superset
- AirStream: Unified Streaming and Batch Processing

AIRBNB DATA PLATFORM

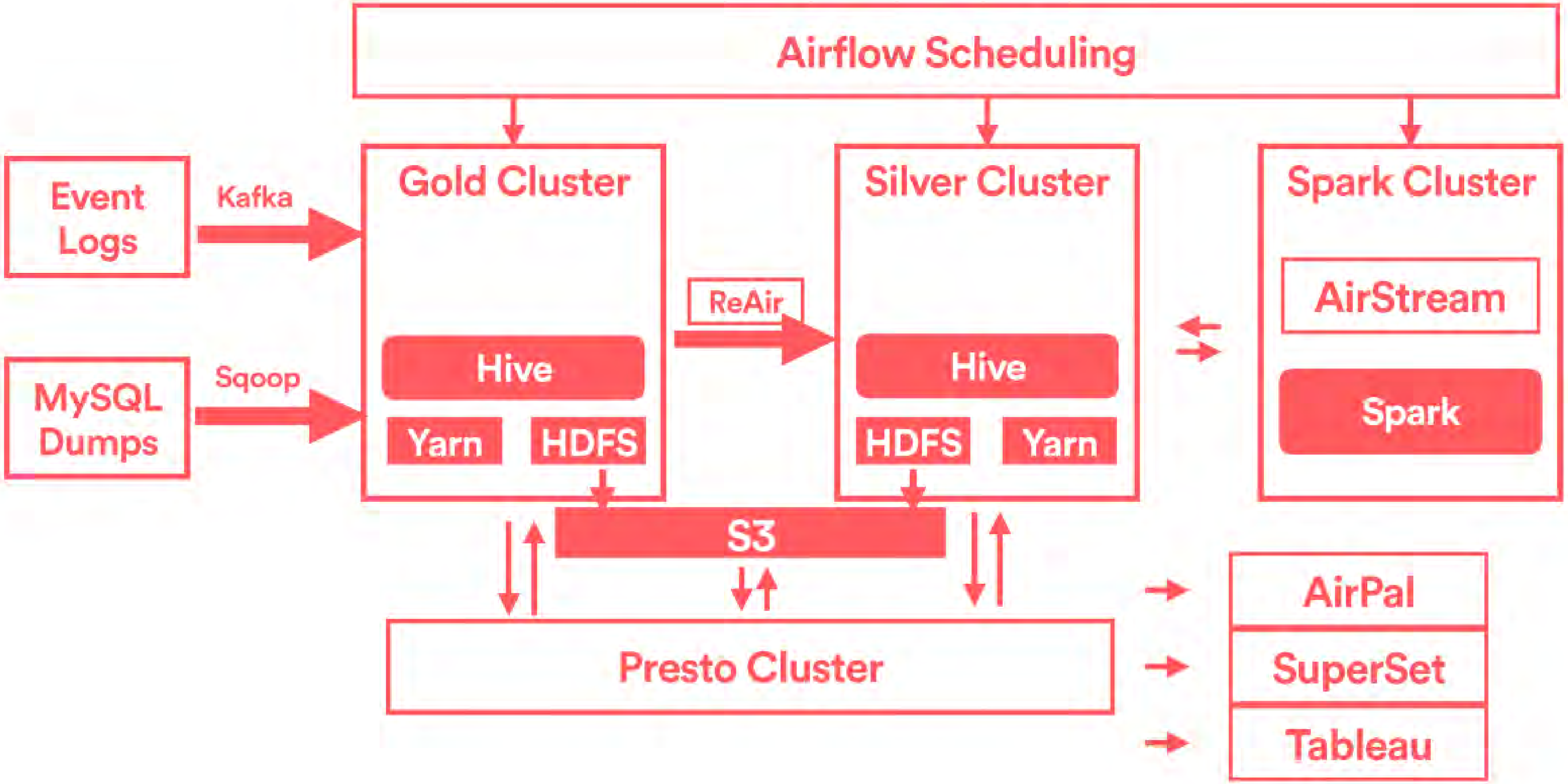
Scale of Data Infrastructure at Airbnb

>13B	>35PB	1400+	5x
#Events Collected	Warehouse Size	Machines Hadoop + Presto + Spark	YoY Data Growth

Data Platform

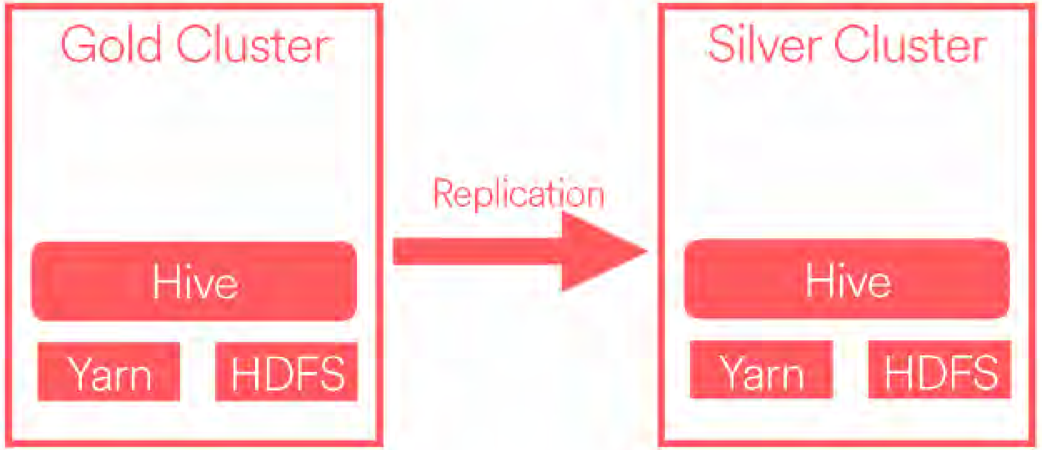


Data Platform



Two Clusters

- Two independent HDFS, MR, Hive metastores
- d2.8xlarge w/ 48TB local
- ~250 instances in final setup
- Replication of common / critical data - Silver is super of Gold
- For disaster recovery, separate AZ's



Multi-Cluster Trade-Offs

Advantages

- Failure isolation with user jobs
- Easy capacity planning
- Guarantee SLA’s
- Able to test new versions
- Disaster Recovery

Disadvantages

- Data synchronization
- User confusion
- Operational overhead

Multi-Cluster Trade-Offs

Advantages

- Failure isolation with user jobs
- Easy capacity planning
- Guarantee SLA’s
- Able to test new versions
- Disaster Recovery

Disadvantages

- **Data synchronization**
- User confusion
- Operational overhead

REAIR: INCREMENTAL DATA REPLICATION

Warehouse Replication Approaches

Batch

- Scan HDFS, metastore
- Copy relevant entries
- Simple, no state
- High latency

Incremental

- Record changes in source
- Copy/re-run operations on destination
- More complex, more state
- Low latency (seconds)

Incremental Replication

- Record Changes on Source
- Convert Changes to Replication Primitives
- Run Primitives on the Destination

Record Changes On Source

- Hive provides hooks API to fire at specific points
  - Pre-execute
  - Post-execute
  - Failure
- Use post-execute to log objects that are created into an audit log
- In critical path for queries

Convert Changes to Primitive Operations

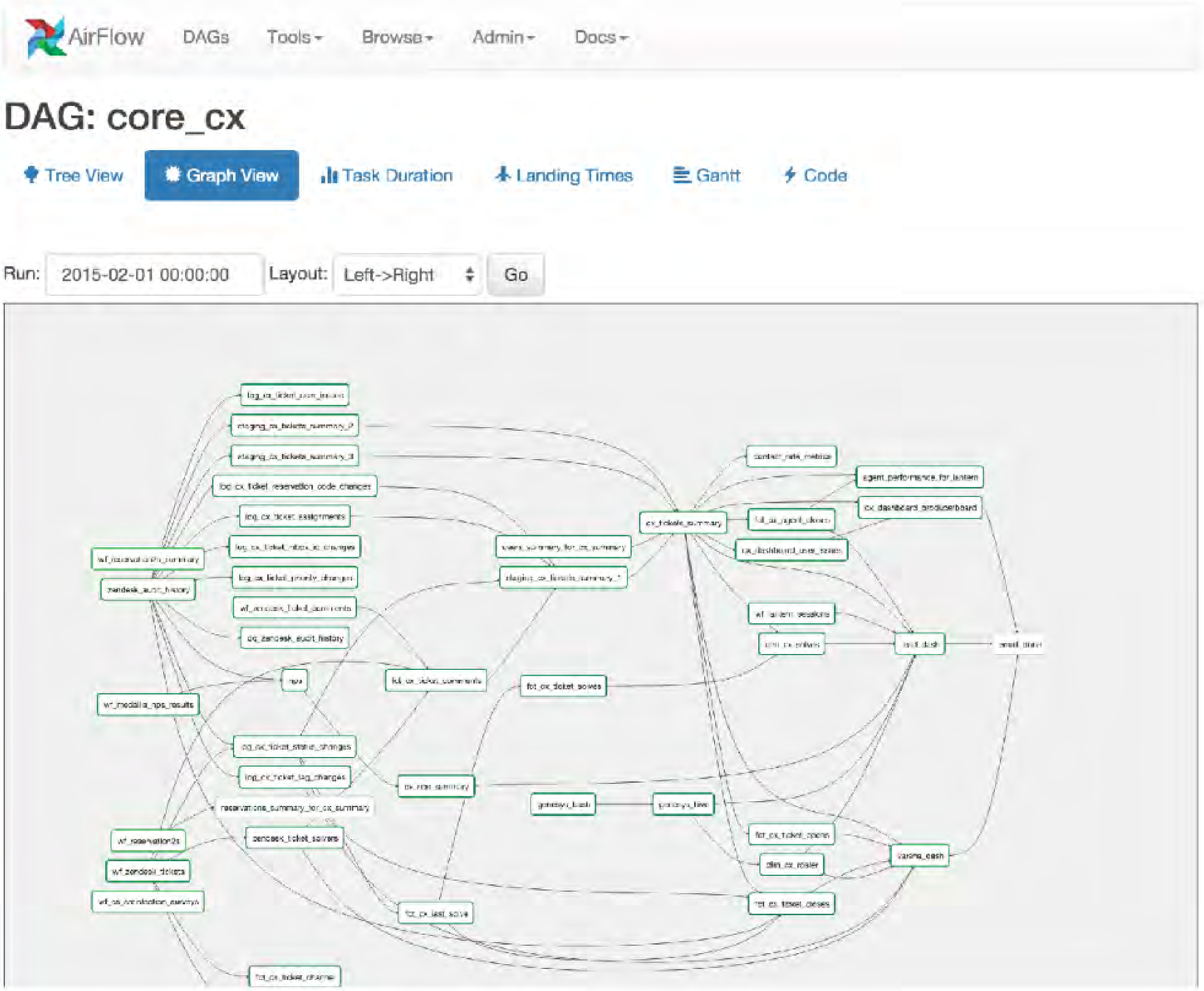
- 3 types of objects - DB, table, partition
- 3 types of operations - Copy, rename, drop
- 9 different primitive operations
- Idempotent

WORKFLOW & SCHEDULING

Airflow

A PLATFORM TO PROGRAMMATICALLY AUTHOR, SCHEDULE AND MONITOR WORKFLOWS

HTTP://AIRFLOW.APACHE.ORG



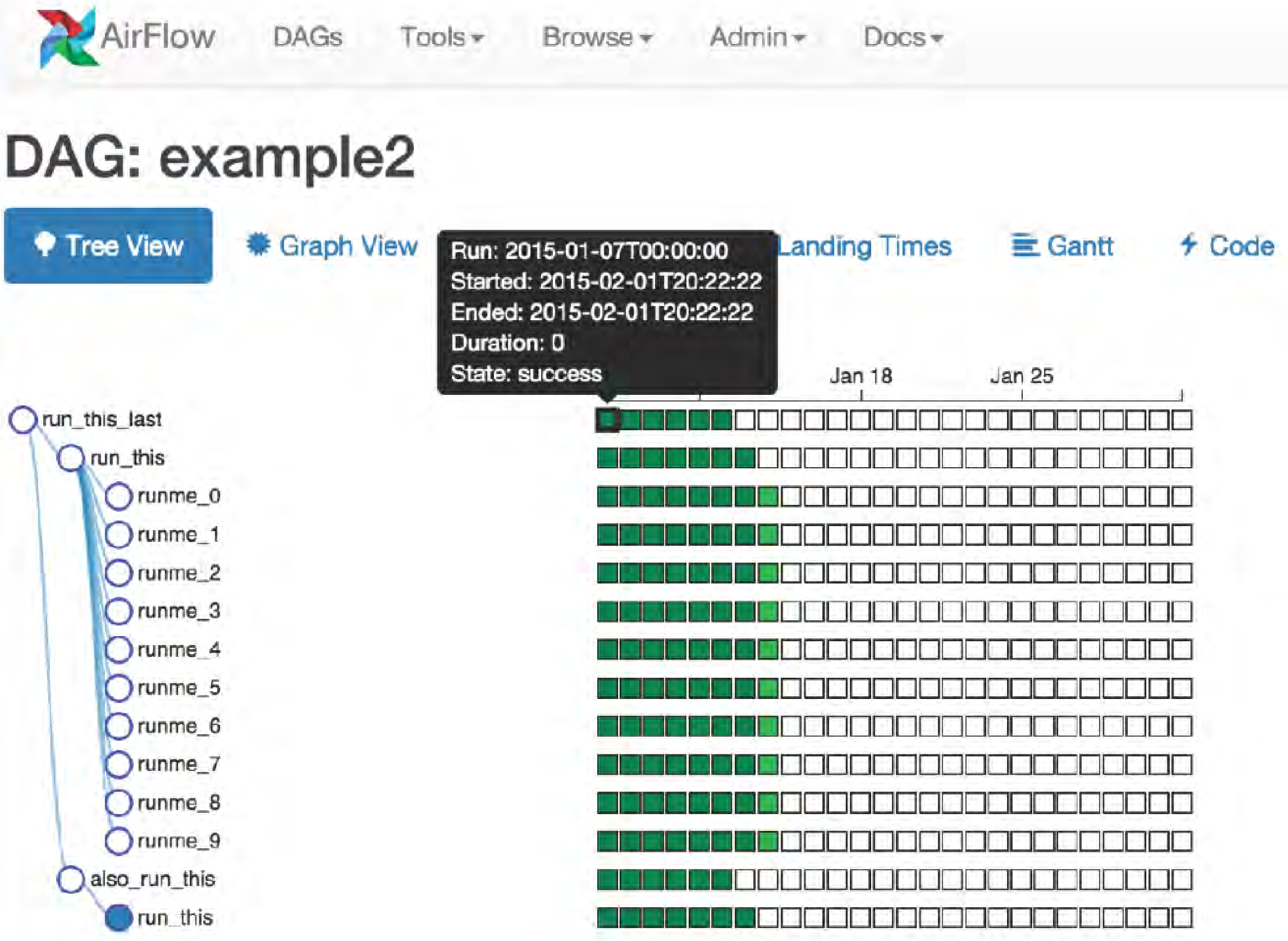
DAG: core_cx

Run: 2015-02-01 00:00:00 Layout: Left->Right

Airflow

A PLATFORM TO PROGRAMMATICALLY AUTHOR, SCHEDULE AND MONITOR WORKFLOWS

HTTP://AIRFLOW.APACHE.ORG



DAG: example2

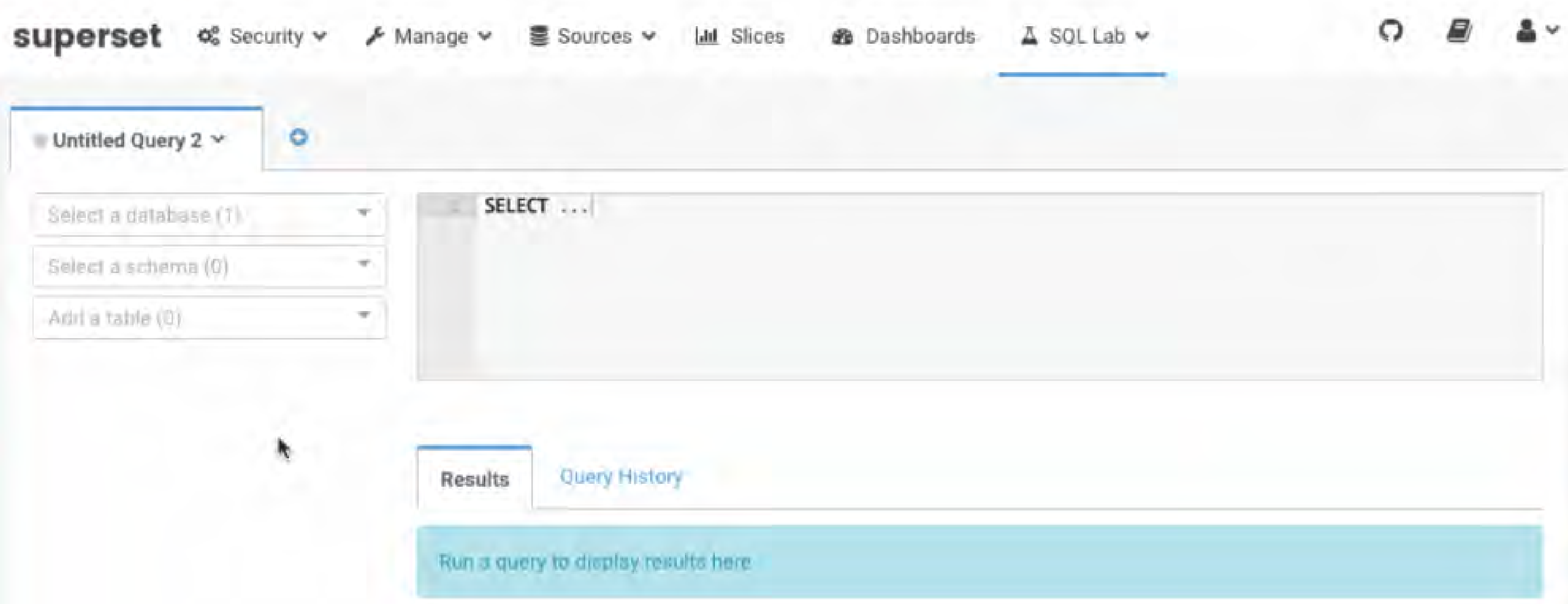
Run: 2015-01-07T00:00:00
Started: 2015-02-01T20:22:22
Ended: 2015-02-01T20:22:22
Duration: 0
State: success

ANALYTICS

Superset

- QUERY AND VISUALIZE DATA
- CREATE AND SHARE DASHBOARDS
- ENTERPRISE-READY AUTHENTICATION
- INTEGRATION WITH DRUID

HTTP://SUPERSET.APACHE.ORG



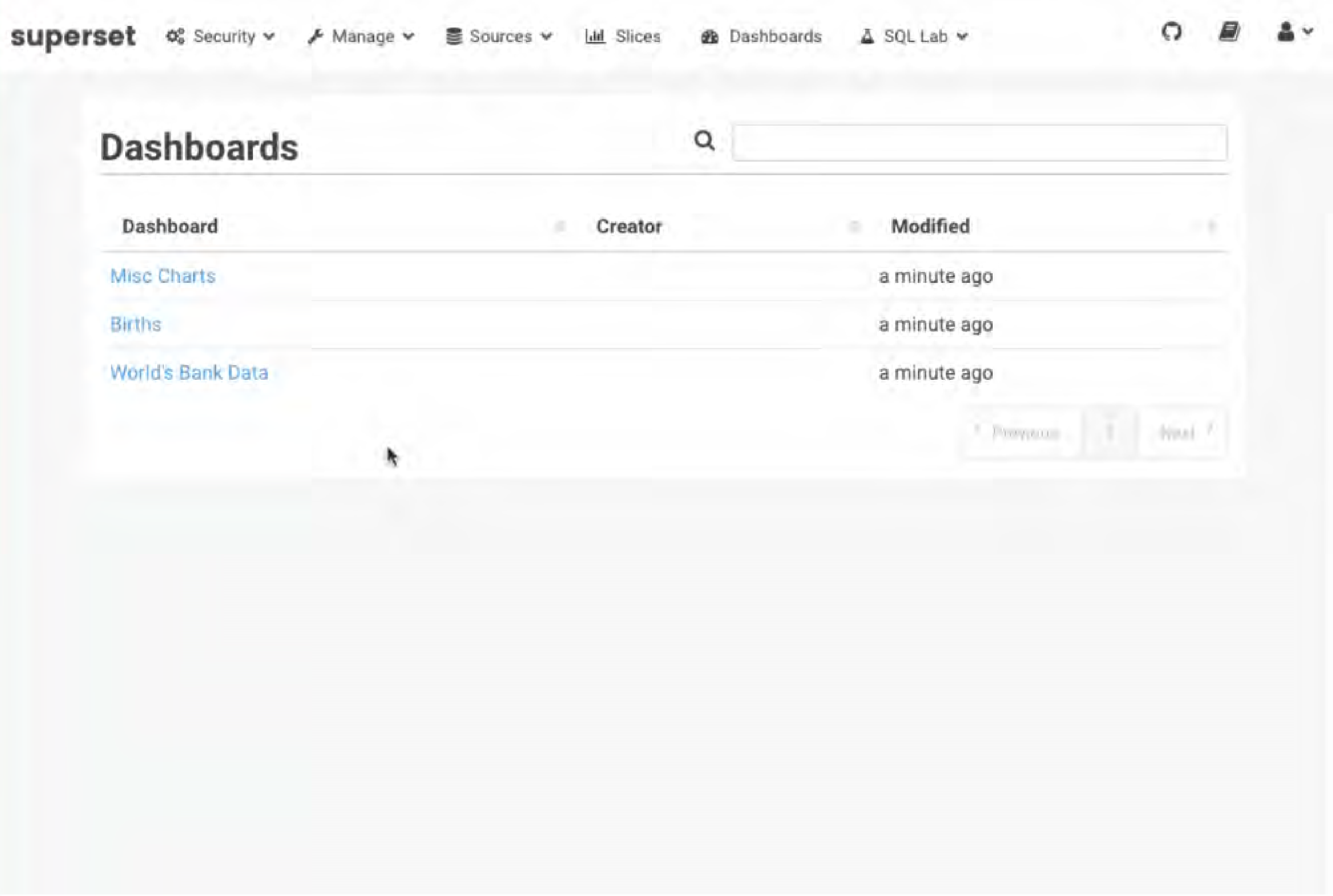
```
SELECT ...
```

Run a query to display results here

Superset

- QUERY AND VISUALIZE DATA
- CREATE AND SHARE DASHBOARDS
- ENTERPRISE-READY AUTHENTICATION
- INTEGRATION WITH DRUID

HTTP://SUPERSET.APACHE.ORG

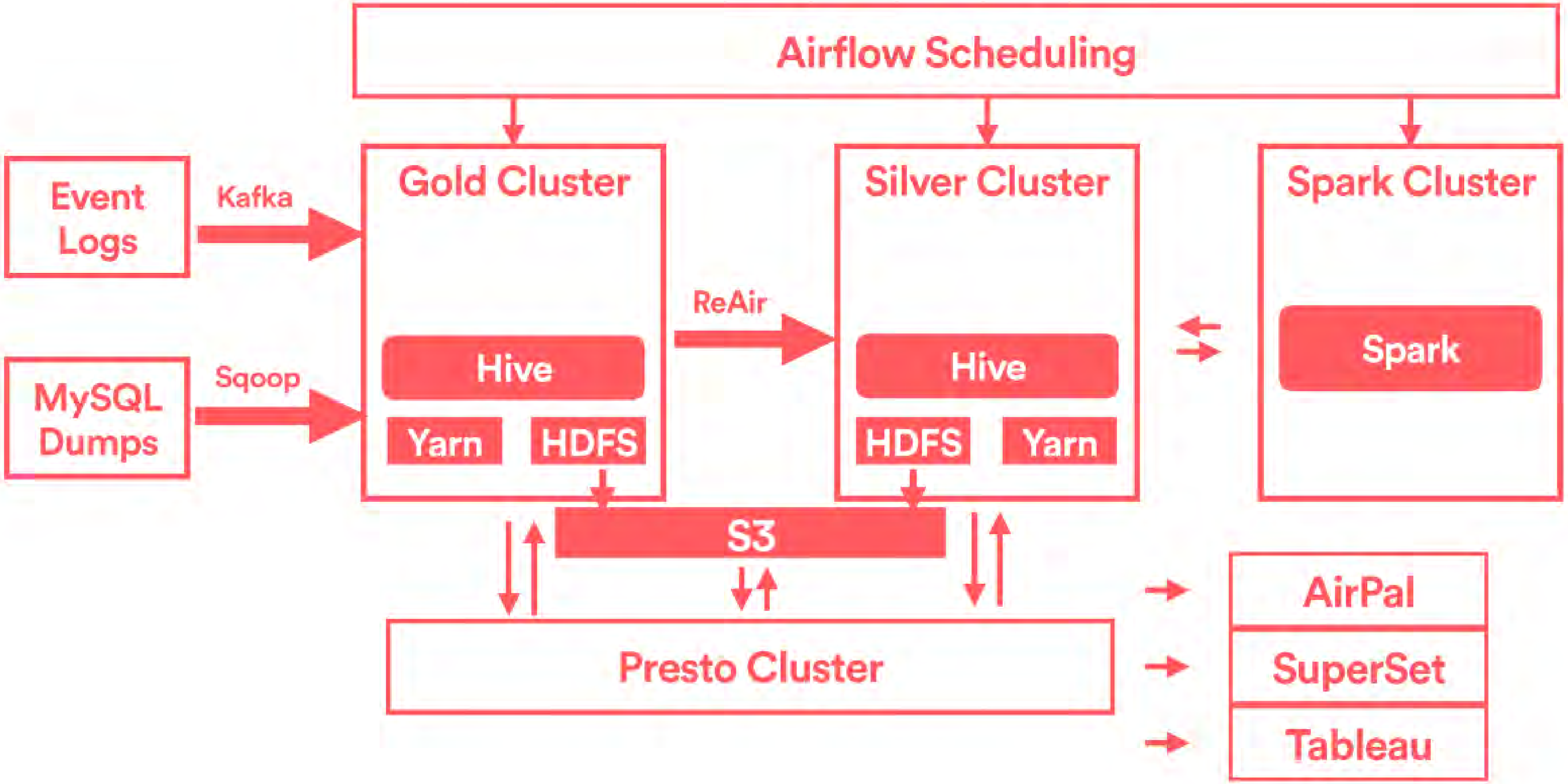


Dashboards

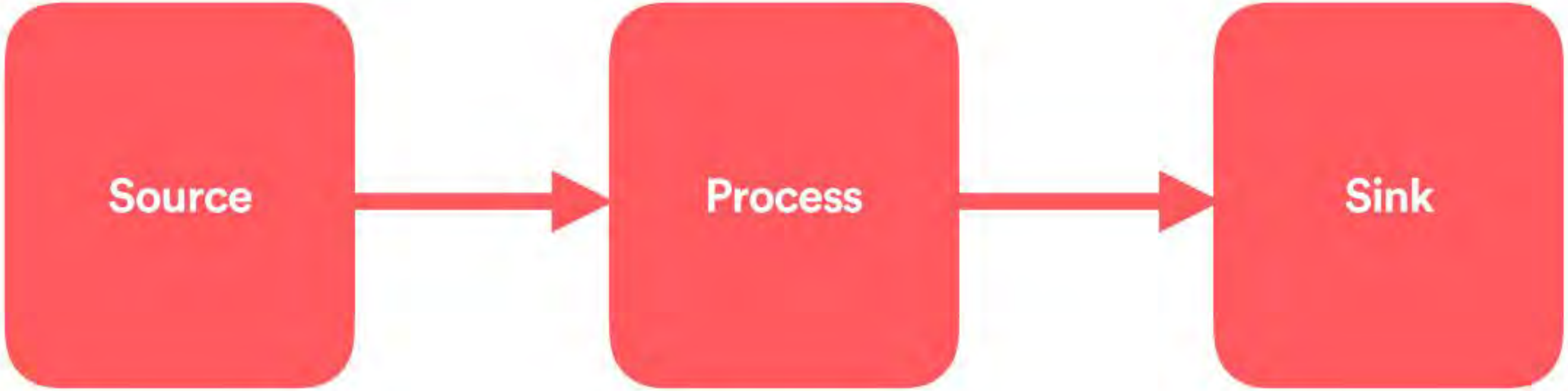
Dashboard	Creator	Modified
Misc Charts		a minute ago
Births		a minute ago
World's Bank Data		a minute ago

AIRSTREAM: UNIFIED STREAMING AND BATCH PROCESSING

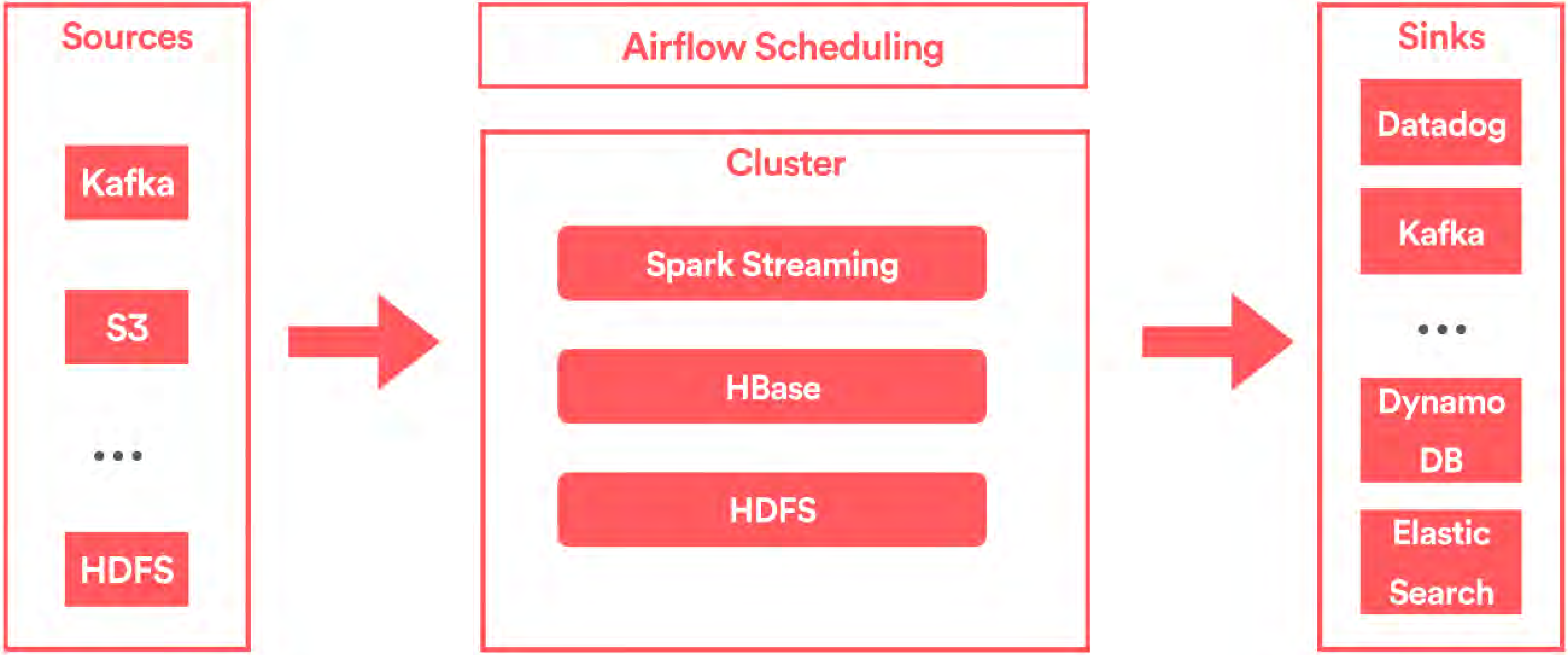
Batch Infrastructure



AirStream

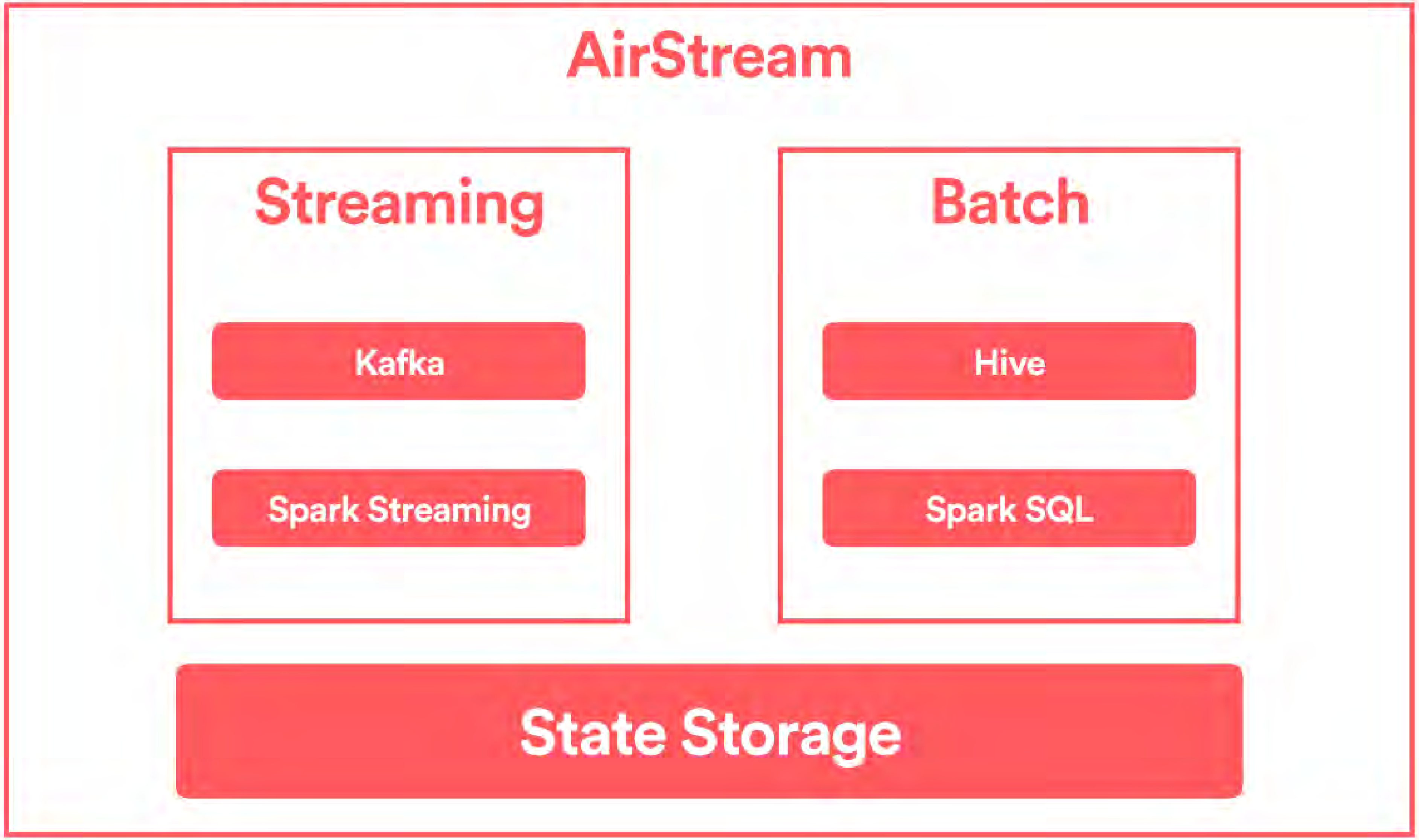


Streaming at Airbnb - AirStream



LAMBDA ARCHITECTURE

Lambda Architecture



AirStream

Streaming

Kafka

Spark Streaming

Batch

Hive

Spark SQL

State Storage

Sources

Streaming

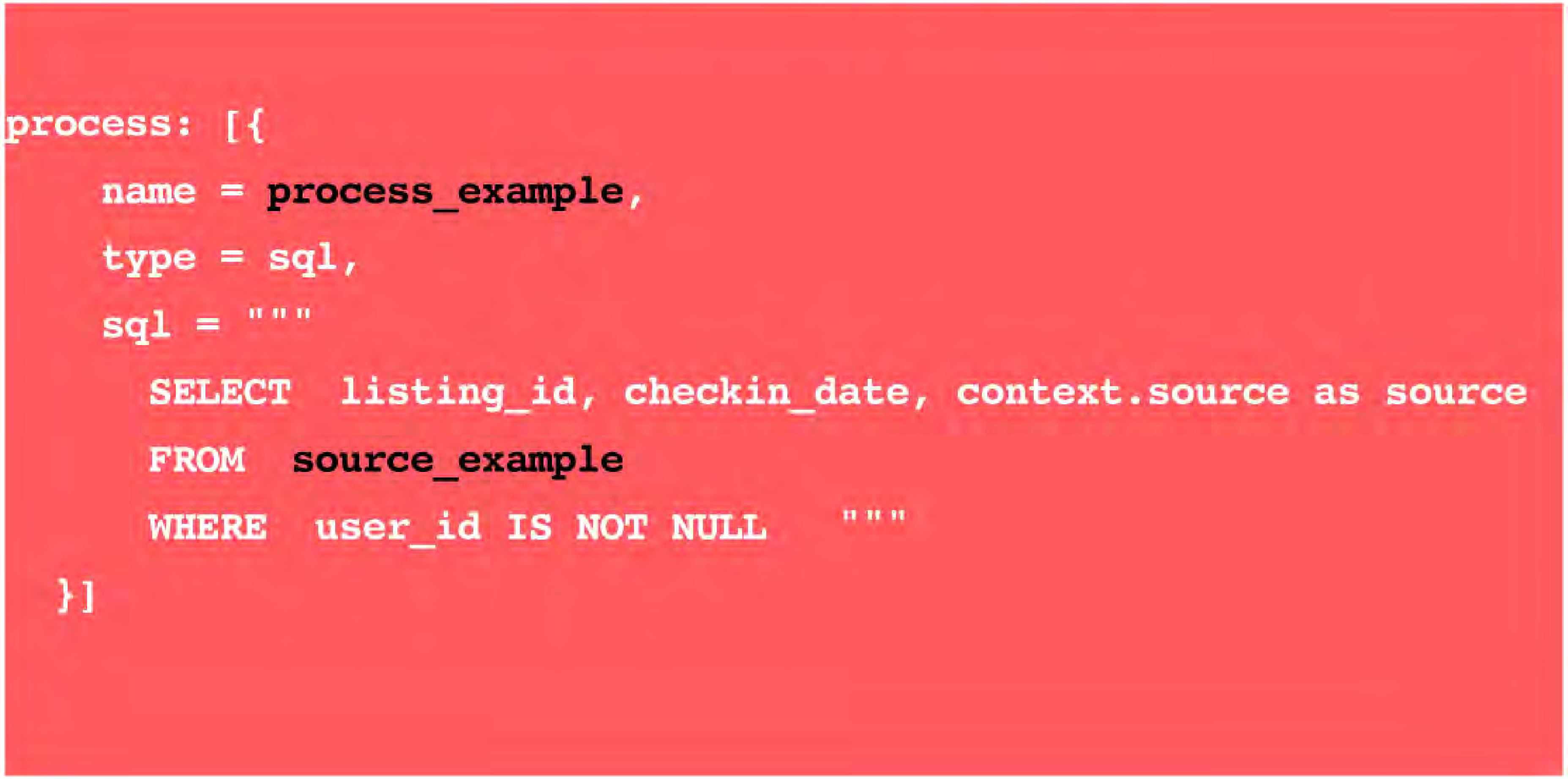
```
source: [
  {
    name: source_example,
    type: kafka,
    config: {
      topic: "example_topic",
    }
  }
]
```

Batch

```
source: [
  {
    name: source_example,
    type: hive,
    sql: {
     select * from db.table
where ds=‘2017-06-05’;
   }
  }
]
```

Computation

Streaming/Batch



```
process: [{
    name = process_example,
    type = sql,
    sql = """
      SELECT  listing_id, checkin_date, context.source as source
      FROM  source_example
      WHERE  user_id IS NOT NULL   """
  }]
```

Sinks

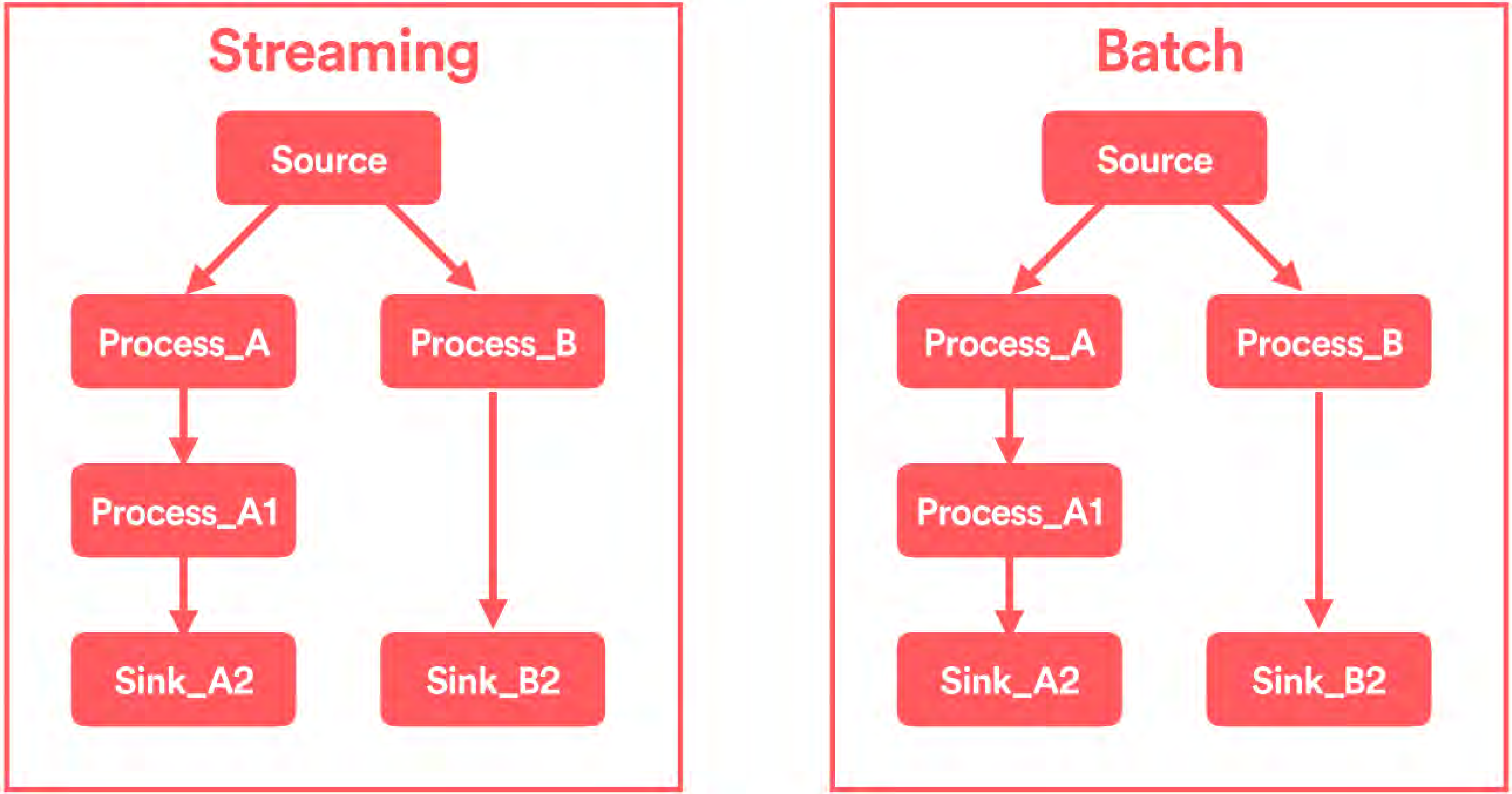
Streaming

```
sink: [
  {
    name = sink_example
    input = process_example
    type = hbase_update
    hbase_table_name = test_table
    bulk_upload = false
 }
]
```

Batch

```
sink: [
  {
    name = sink_example
    input = process_example
    type = hbase_update
    hbase_table_name = test_table
    bulk_upload = true
 }
]
```

Computation Flow

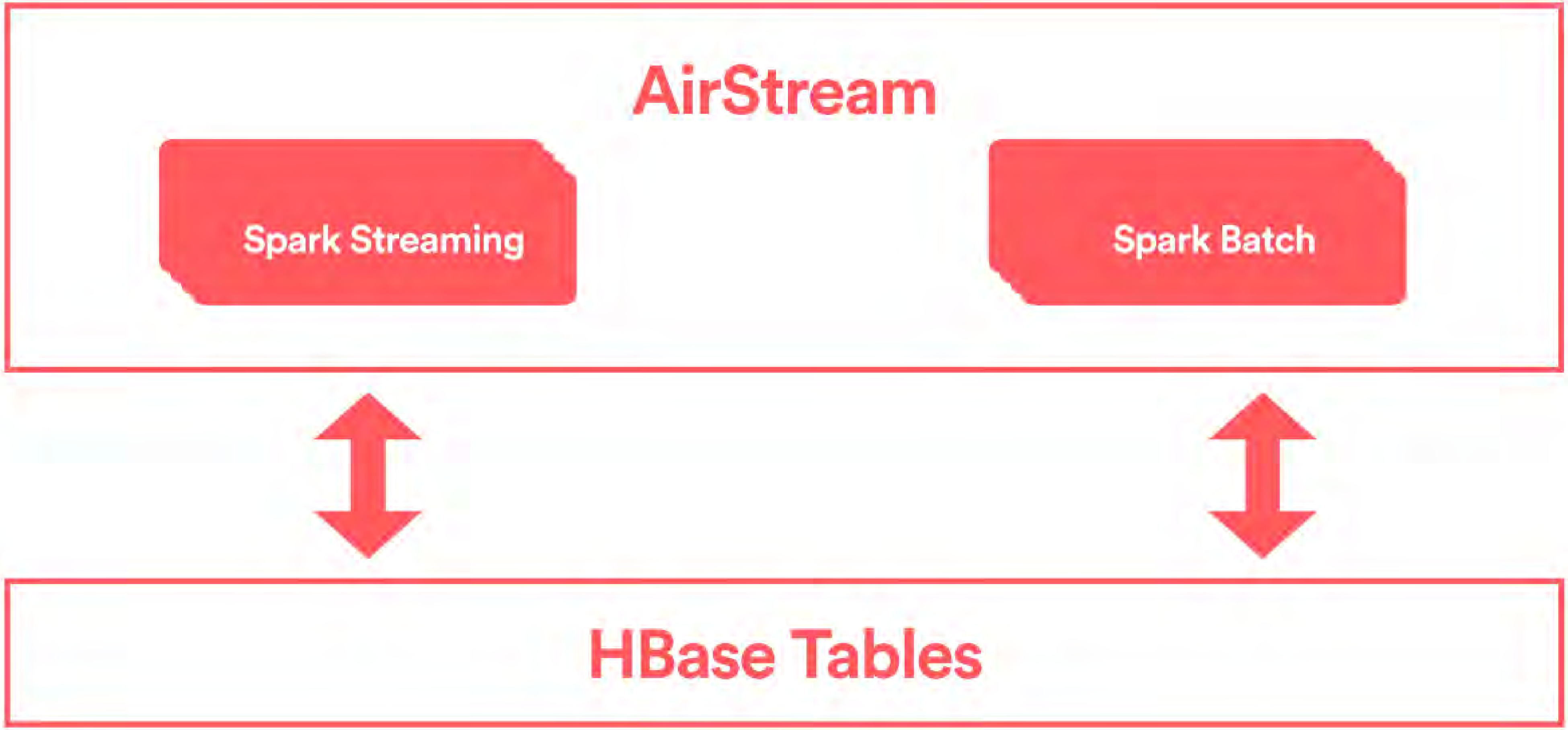


Unified API through AirStream

- Declarative job configuration
- Streaming source vs static source
- Computation operator or sink can be shared by streaming and batch job.
- Computation flow is shared by streaming and batch
- Single driver executes in both streaming and batch mode job

SHARED STATE STORAGE

Shared Global State Store



AirStream

Spark Streaming

Spark Batch

HBase Tables

Why HBase

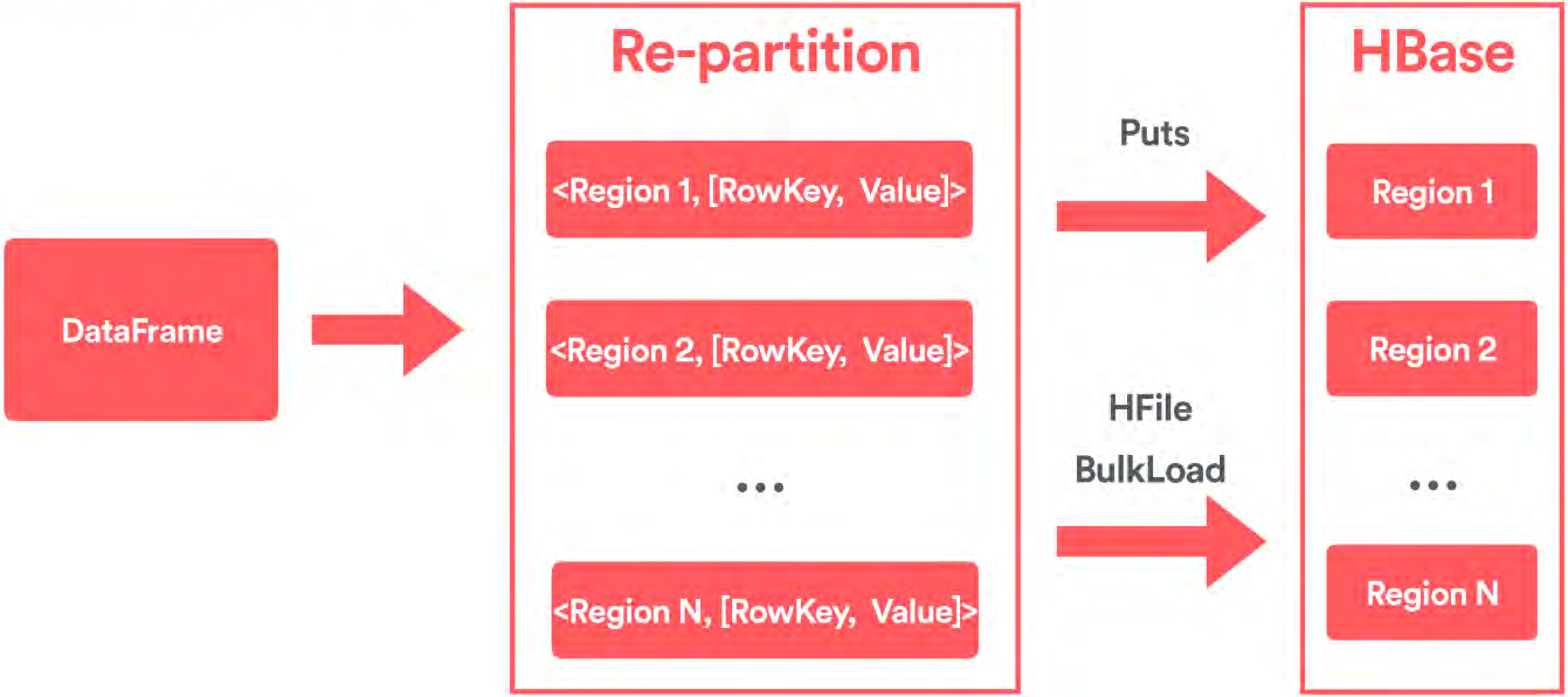
- Well integrated with Hadoop eco system

- Efficient API for streaming writes and bulk uploads

- Rich API for sequential scan and point-lookups

- Merged view based on version

Unified Write API



DataFrame

Re-partition

<Region 1, [RowKey, Value]>

<Region 2, [RowKey, Value]>

• • •

<Region N, [RowKey, Value]>

Puts

HFile

BulkLoad

HBase

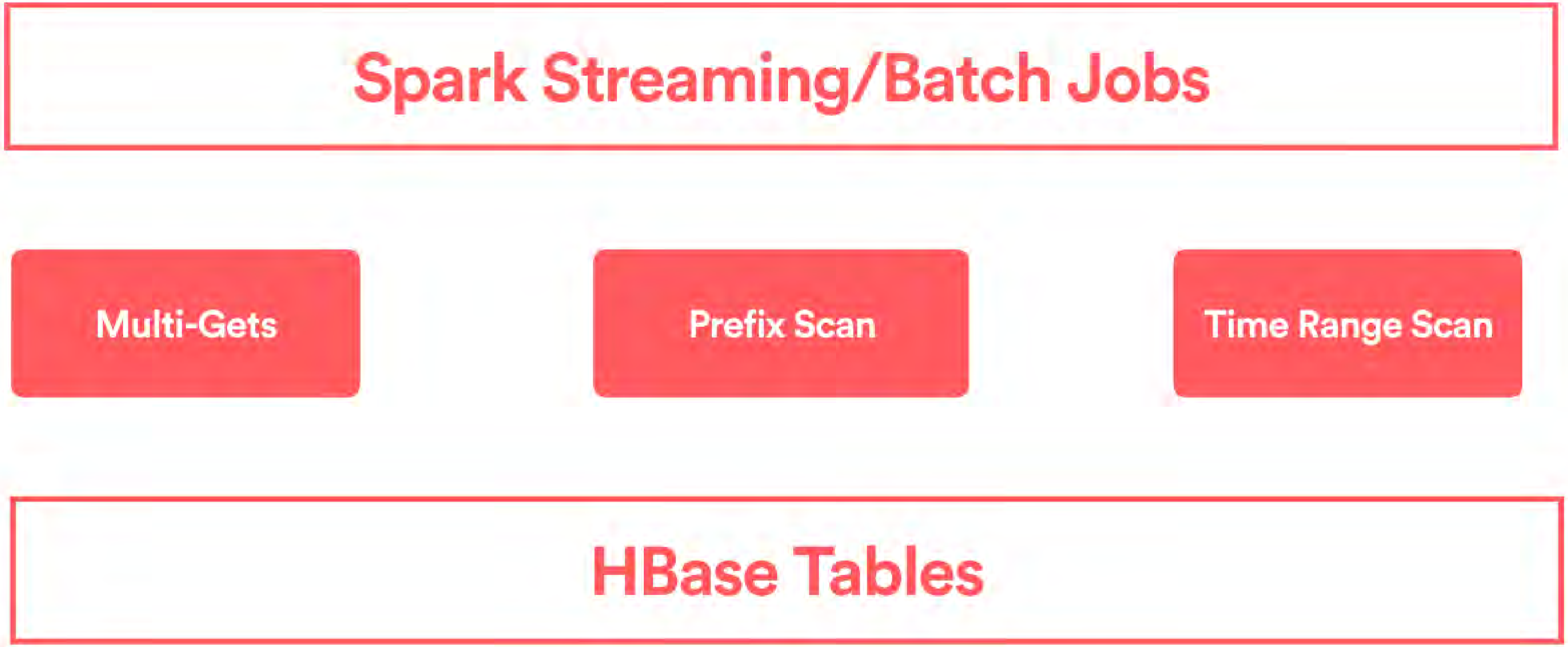
Region 1

Region 2

• • •

Region N

Rich Read API



Spark Streaming/Batch Jobs

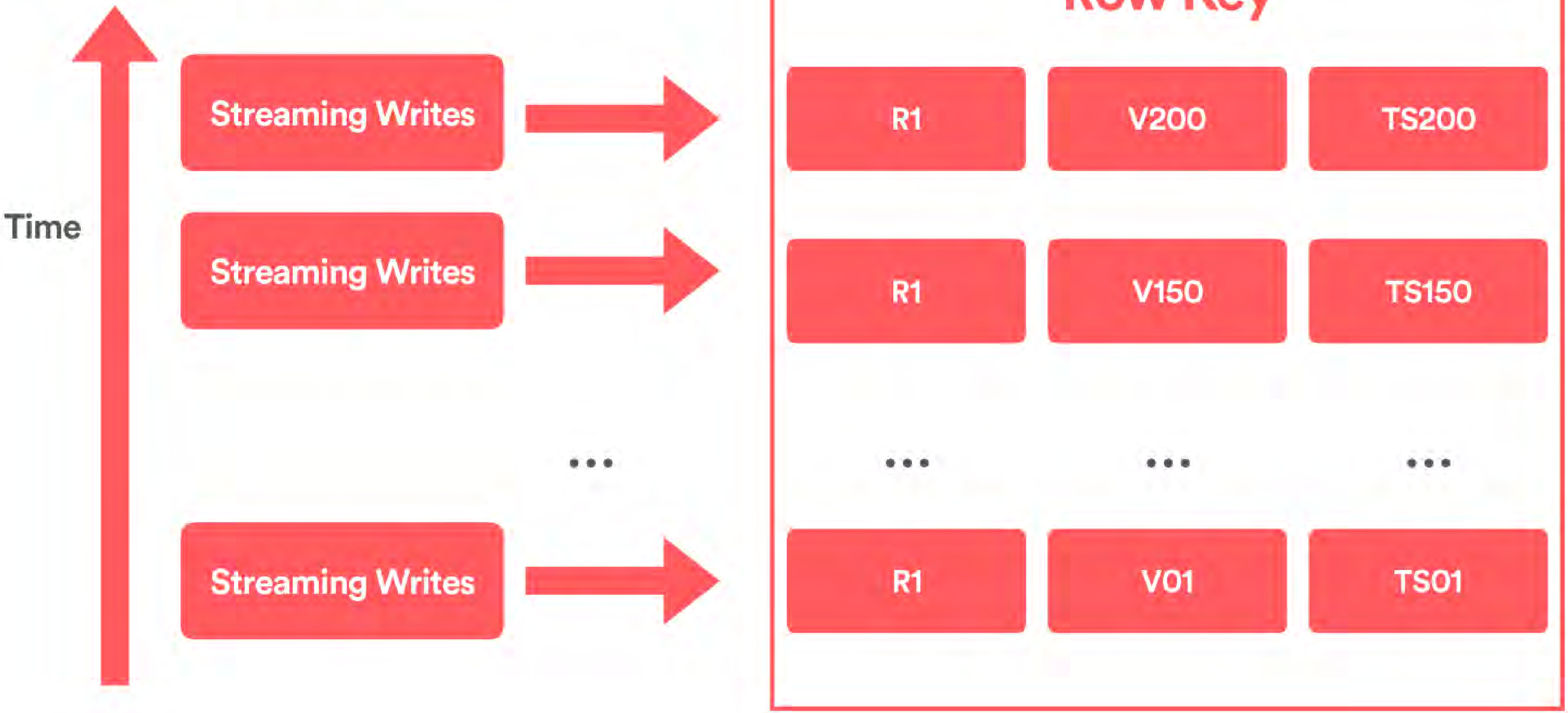
Multi-Gets

Prefix Scan

Time Range Scan

HBase Tables

Merged Views



Time

Streaming Writes

Streaming Writes

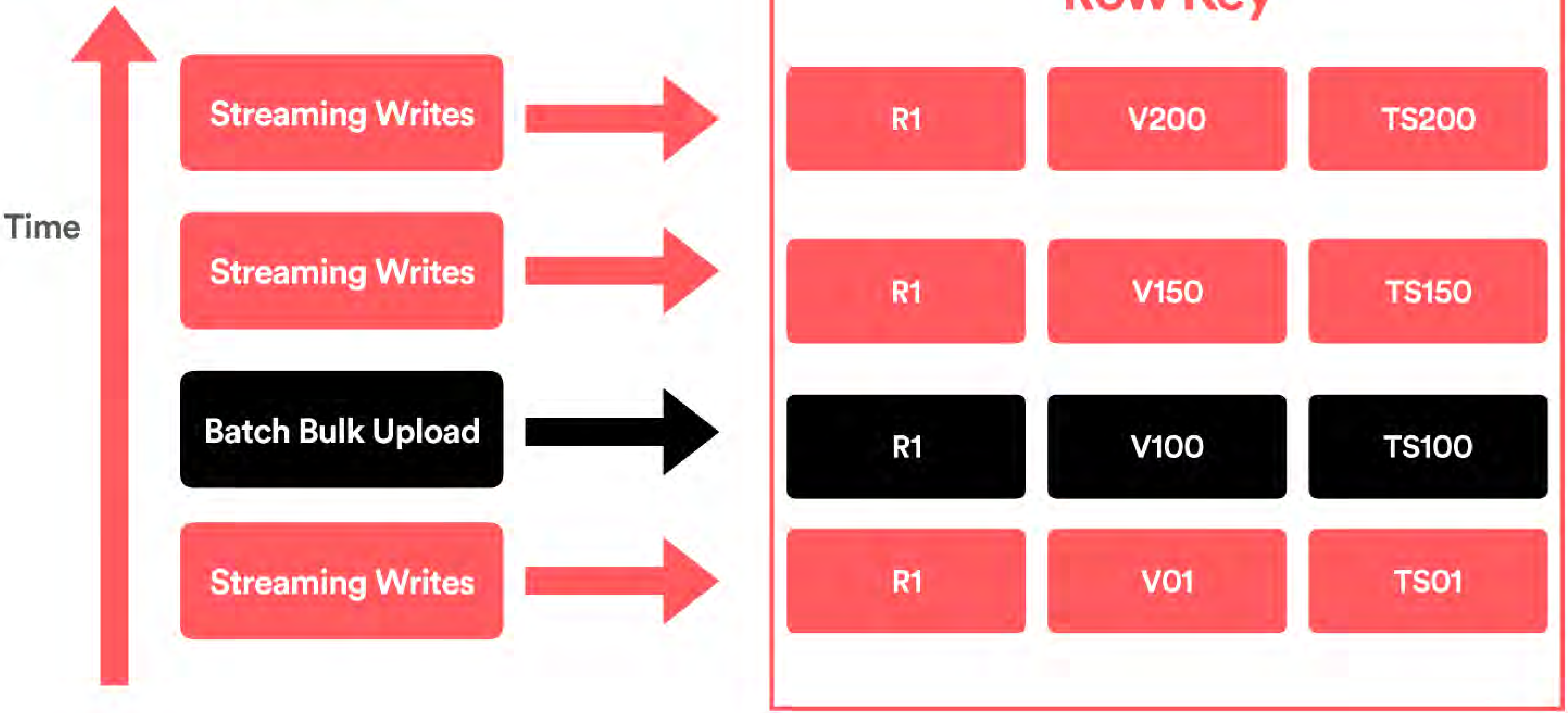
…

Streaming Writes

Row Key

R1	V200	TS200
R1	V150	TS150
…	…	…
R1	V01	TS01

Merged Views



Time

Streaming Writes

Streaming Writes

Batch Bulk Upload

Streaming Writes

Row Key

R1	V200	TS200
R1	V150	TS150
R1	V100	TS100
R1	V01	TS01

Our Foundations

- Unify streaming and batch process
- Shared global state store

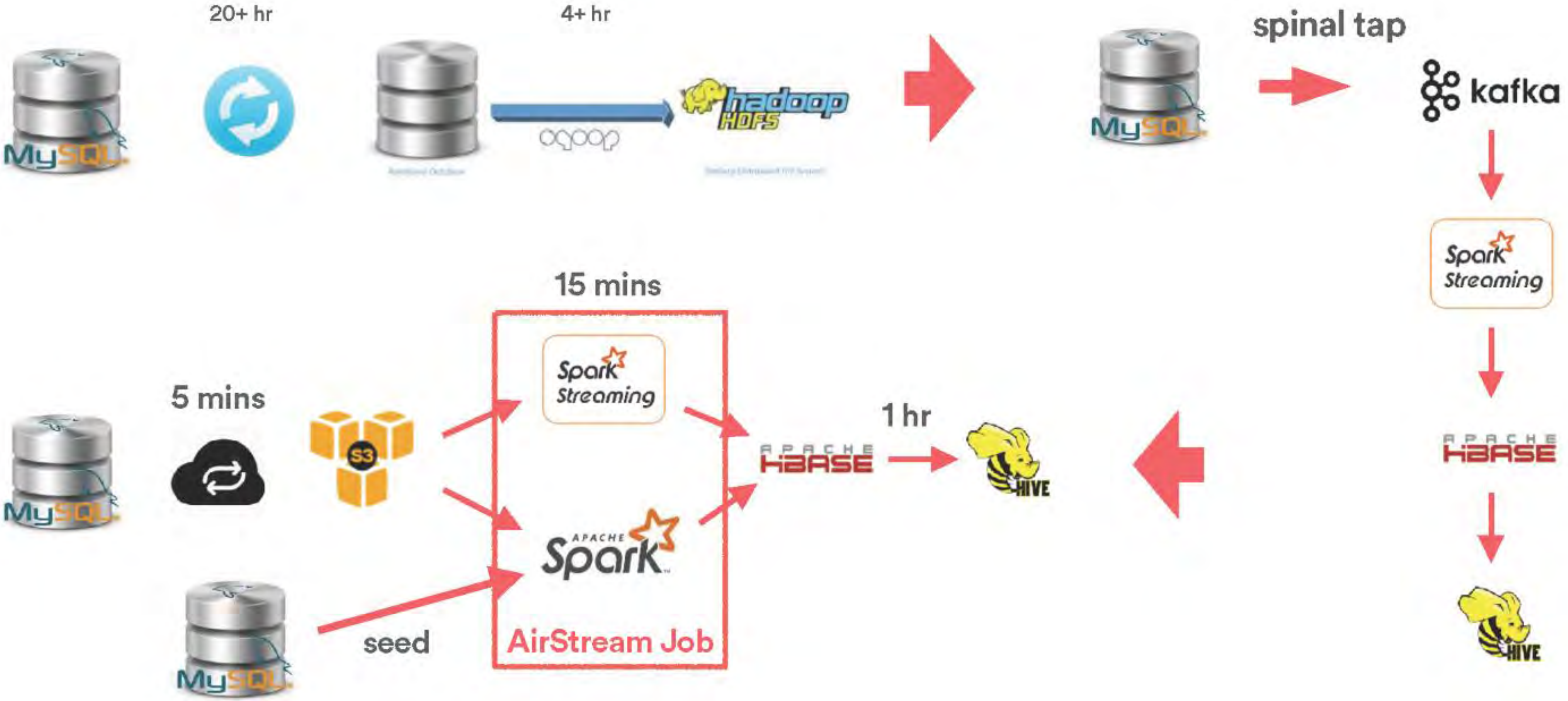
MYSQL DB SNAPSHOT USING BINLOG REPLAY

Move Elephant

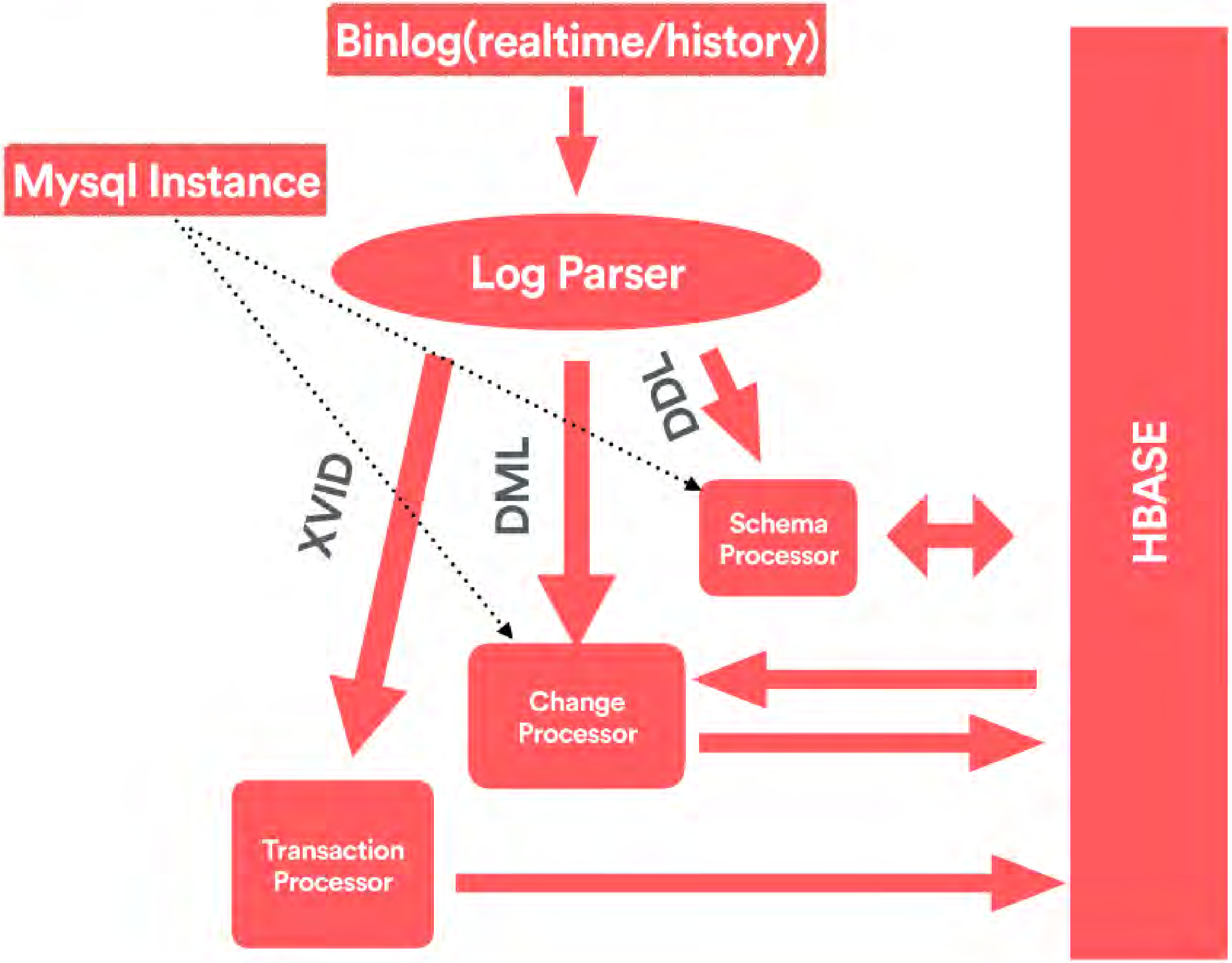
Database Snapshot

- **Large amount of data:** Multiple large mysql DBs
- **Realtime-ness:** minutes delay/ hours delay
- **Transaction :** Need to keep transaction across different tables
- **Schema change:** Table schema evolves

Binlog Replay on Spark



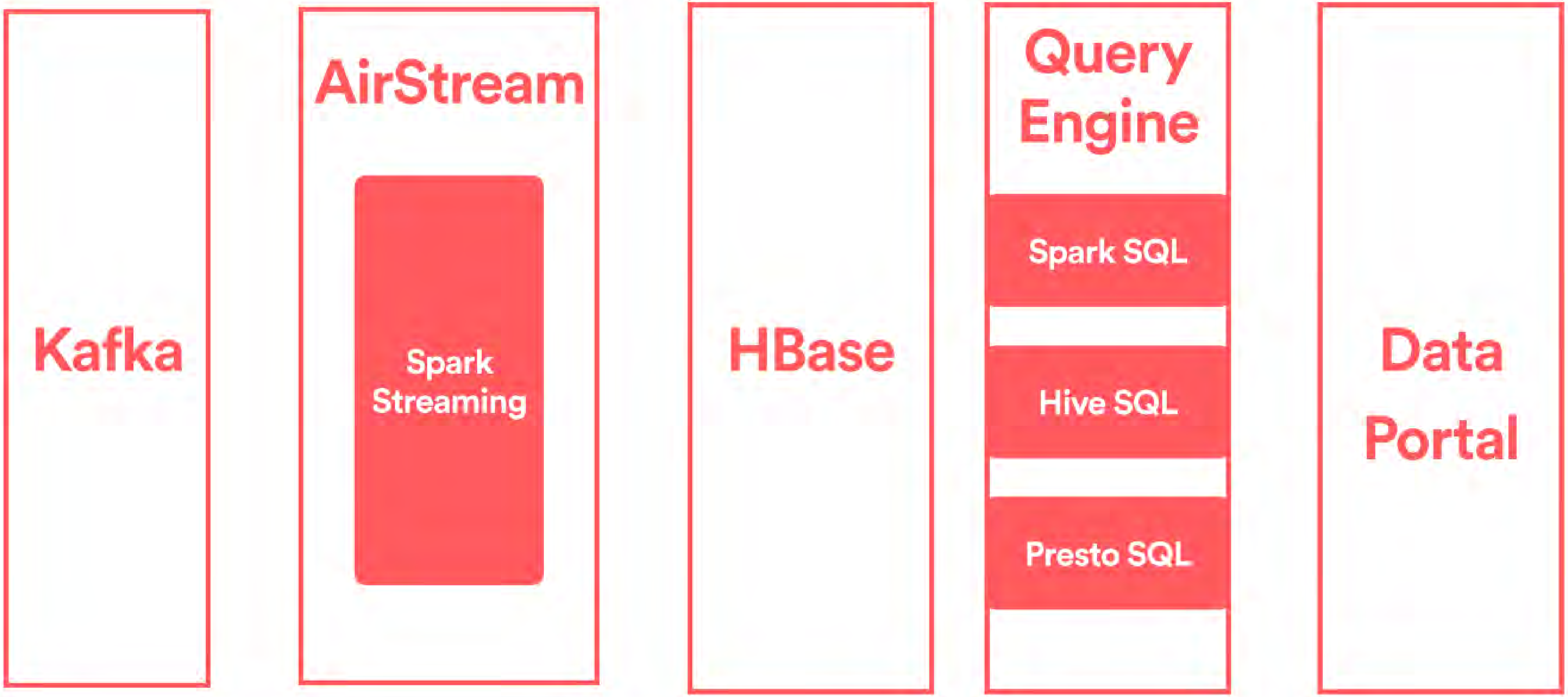
Lambda Architecture



- **Streaming and Batch shares Logic:** Binlog file reader, DDL processor, transaction processor, DML processor.
- **Merged by binlog position:** <filenum, offset>
- **Idempotent:** Log can be replayed multiple times.
- **Schema changes:** Full schema change history.

STREAMING INGESTION & REALTIME INTERACTIVE QUERY

Realtime Ingestion and Interactive Query



Kafka

AirStream

Spark Streaming

HBase

Query Engine

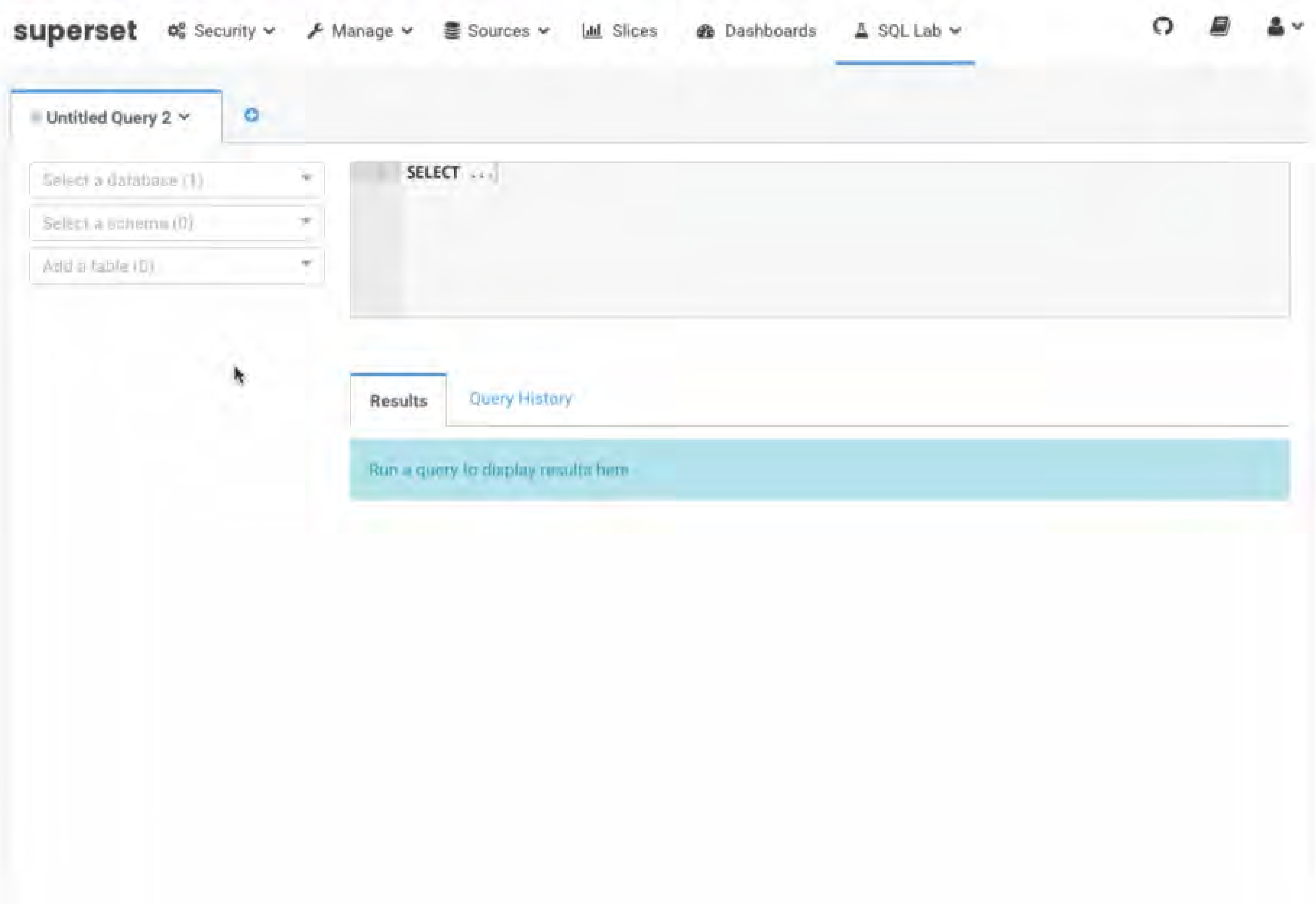
Spark SQL

Hive SQL

Presto SQL

Data Portal

Interactive Query in SqlLab



```
SELECT ...
```

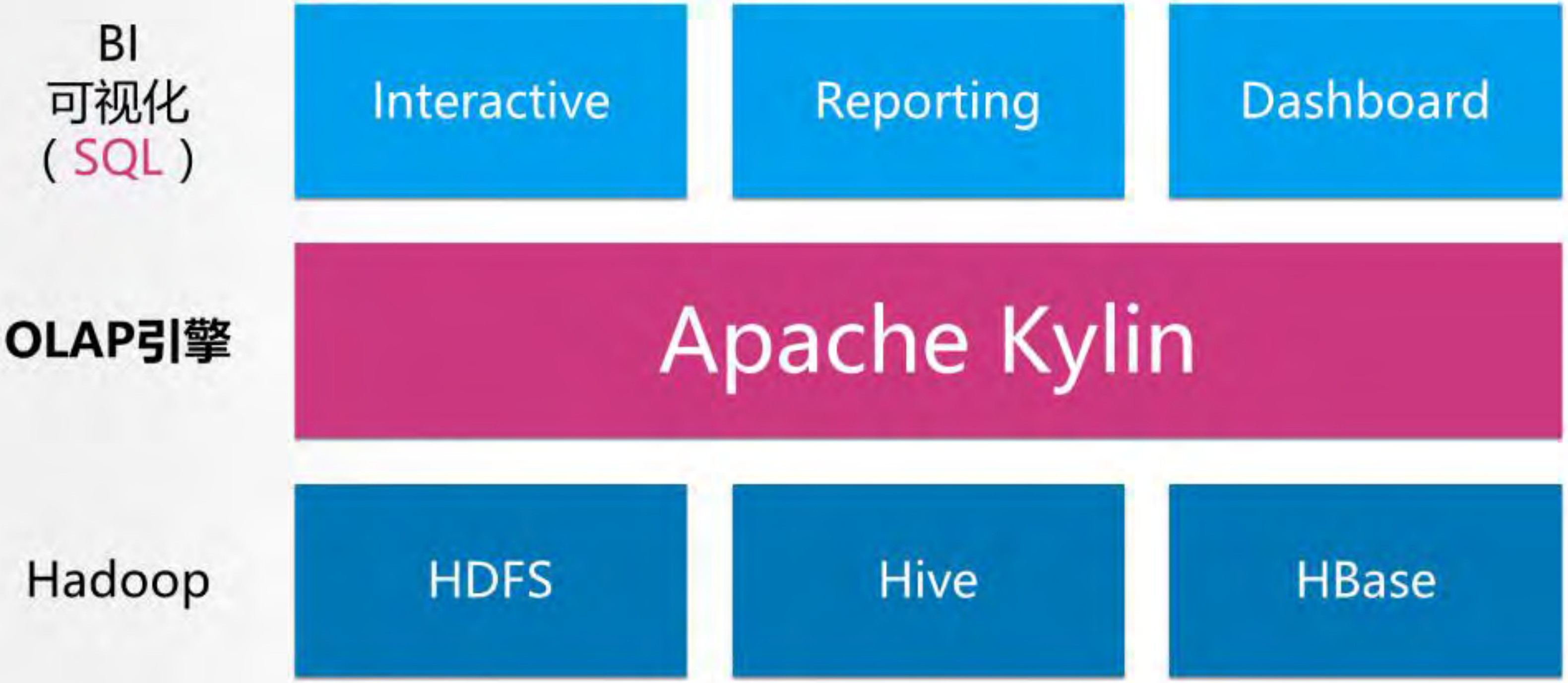
Run a query to display results here

Thank you!

Apache Kylin v2
从OLAP引擎到实时数据仓库的演进

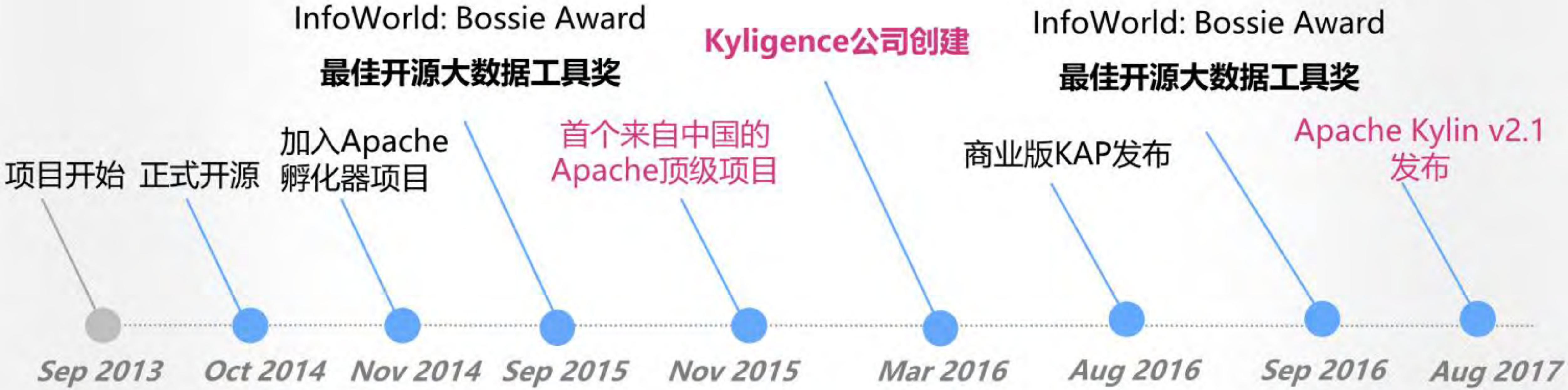
史少锋
Apache Kylin committer & PMC
Kyligence 技术合伙人 & 高级架构师

Apache Kylin是什么？



- 3 万亿条数据，
  < 1 秒 查询延迟
  @今日头条，国内第一新闻资讯app
- 60+ 维度的Cube
  @CPIC
- JDBC / ODBC / RestAPI
- BI 集成
  Tableau、Excel、Cognos、Superset

Apache Kylin 历史



ebay

Apache Kylin

InfoWorld 2015 Awards BOSSIE Best of Open Source Software Big Data

THE APACHE SOFTWARE FOUNDATION

Kyligence Unleash big data productivity

InfoWorld BOSSIE 2016 AWARDS Big Data