基于 Pulsar 和 Flink 进行批流一体的弹性数据处理

Elastic data processing with Apache Pulsar and Apache Flink

申毅杰
Senior Software Engineer at StreamNative

FLINK FORWARD # ASIA

实时即未来 # Real-time Is The Future

FLINK FORWARD

Contents
目录

FLINK FORWARD

批流融合的弹性处理需求

Motivation on Elastic Stream and Batch Processing

01

对批流融合的弹性数据处理需求

Motivation on Elastic Stream and Batch Processing

无处不在的流数据

Sensors, logs from mobile app, IoT

Organizations got better at capturing data

快速发掘数据价值

Batch and interactive analysis, stream processing, machine learning, graph processing

计算引擎批流融合的趋势

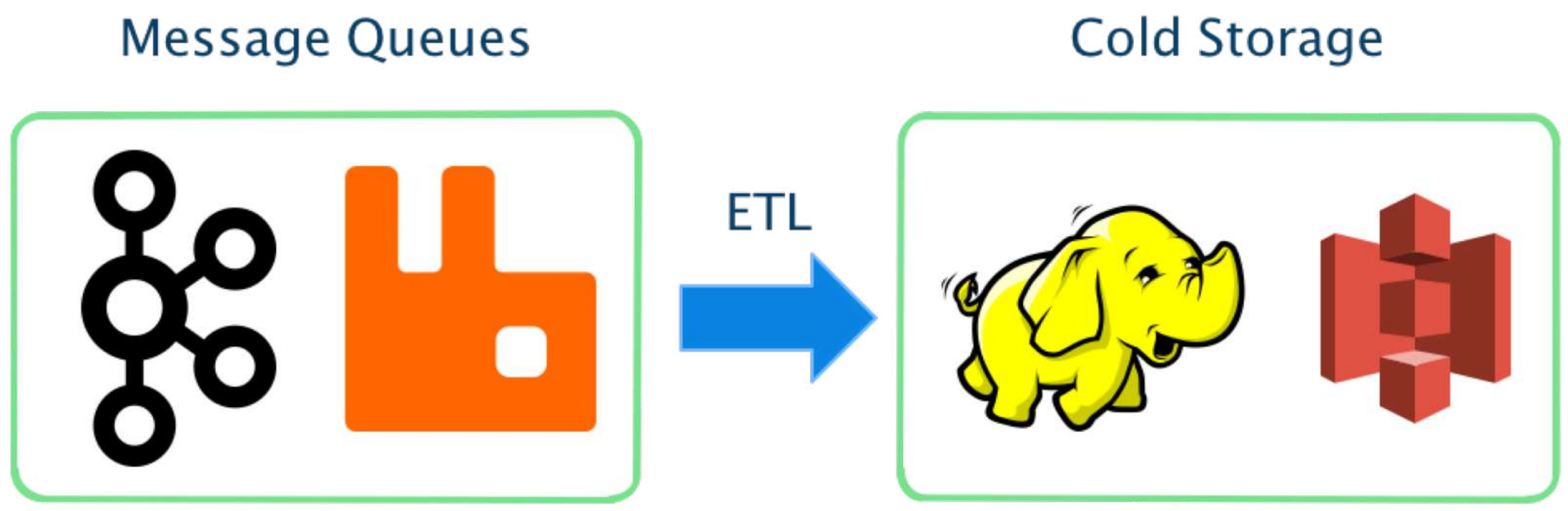
Unified / similar API for batch/interactive and stream processing

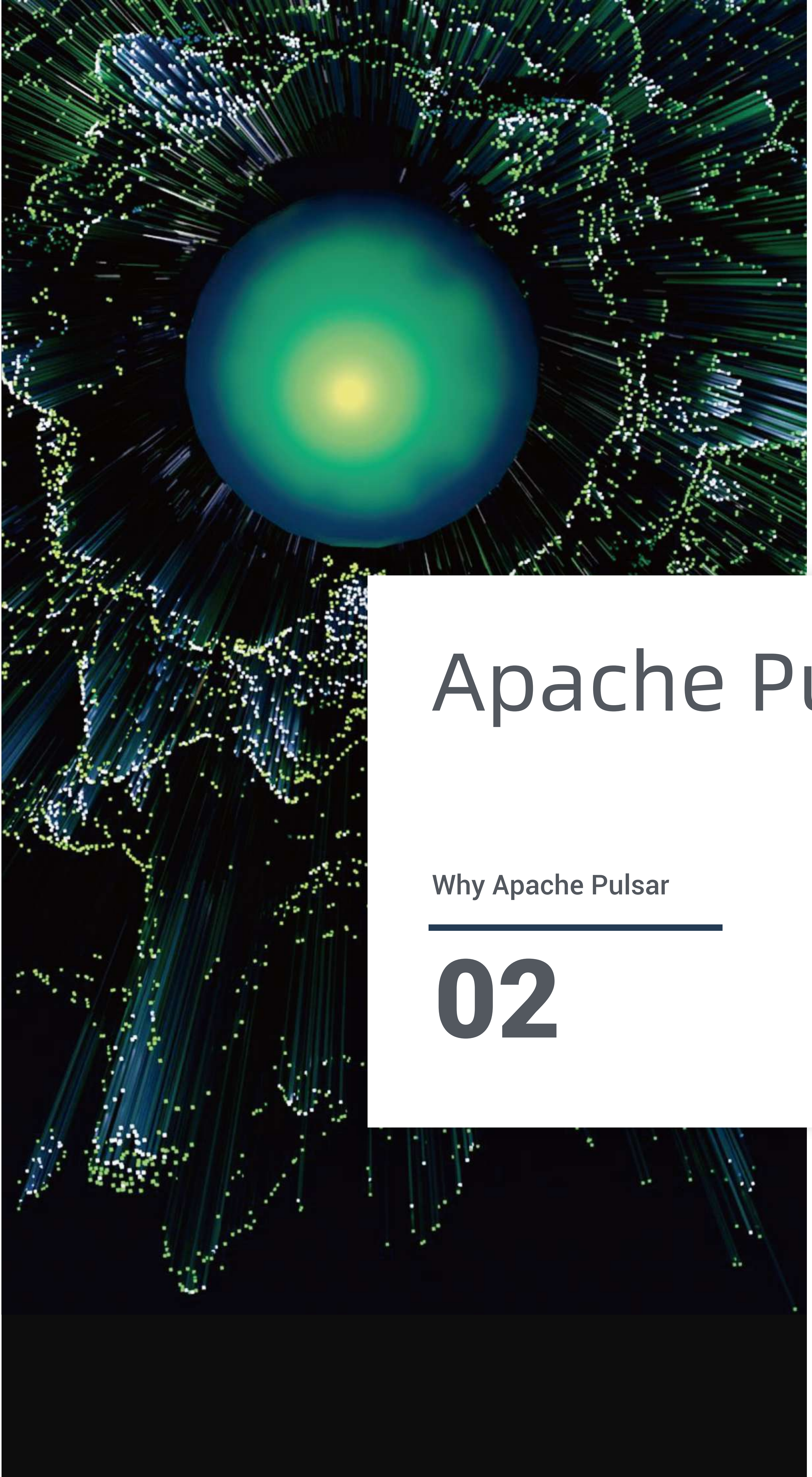
批流融合处理的挑战

Challenges for Traditional MQs or Log Storage Systems

- 云原生架构的兼容性
  Compatible with cloud native architecture
  - 多租户管理
    Multi-tenant management
  - 扩展性
    Scalability

- 数据存储组织的复杂度
  Complexity in a multi-system architecture
  - 多系统存储维护开销
    Maintenance as well as provisioning
  - 数据可见性问题
    Visibility of data





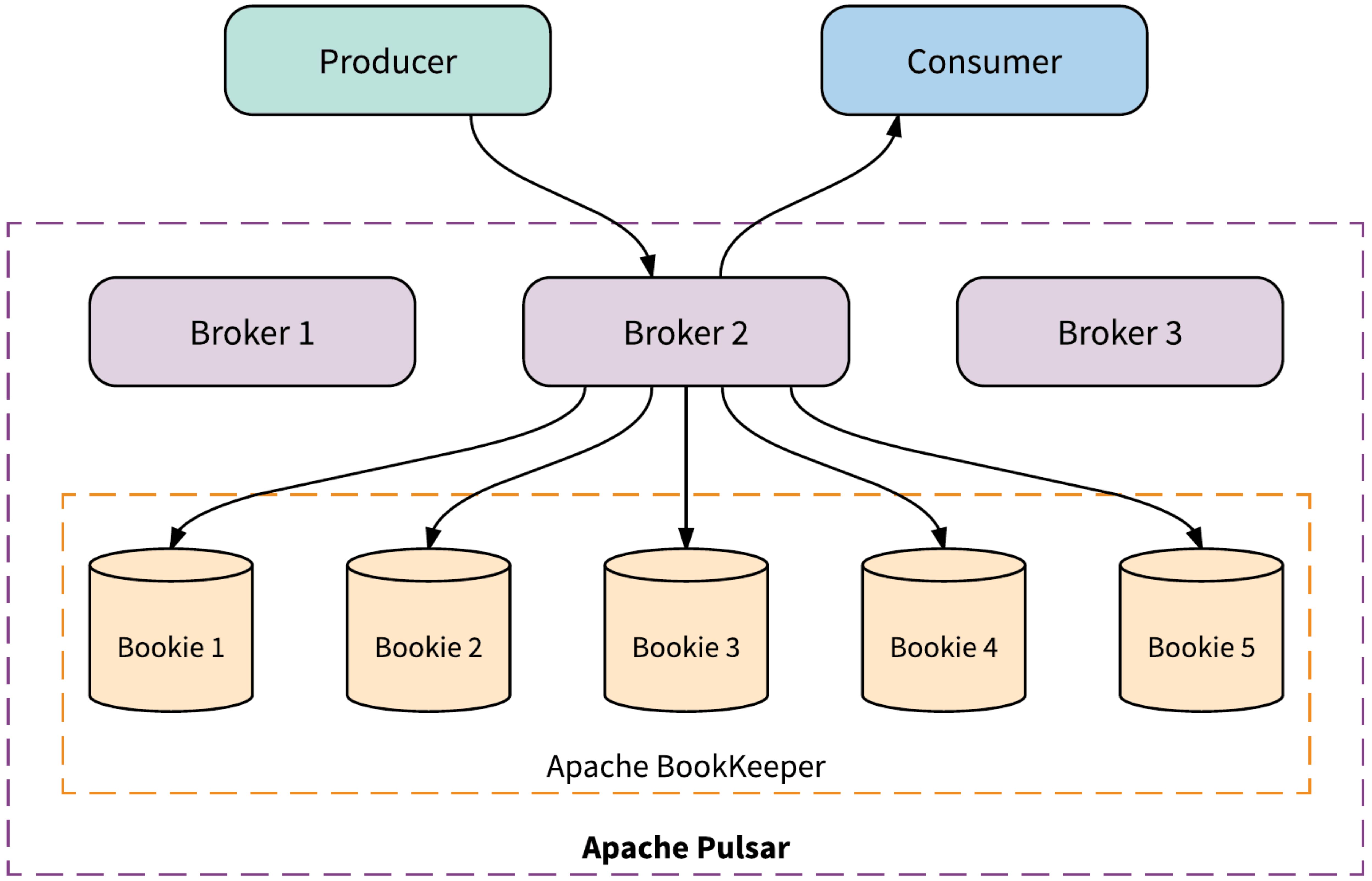
Apache Pulsar 简介

Why Apache Pulsar

02

云原生的架构

Pulsar -- Cloud Native Architecture



无状态服务层

Stateless serving

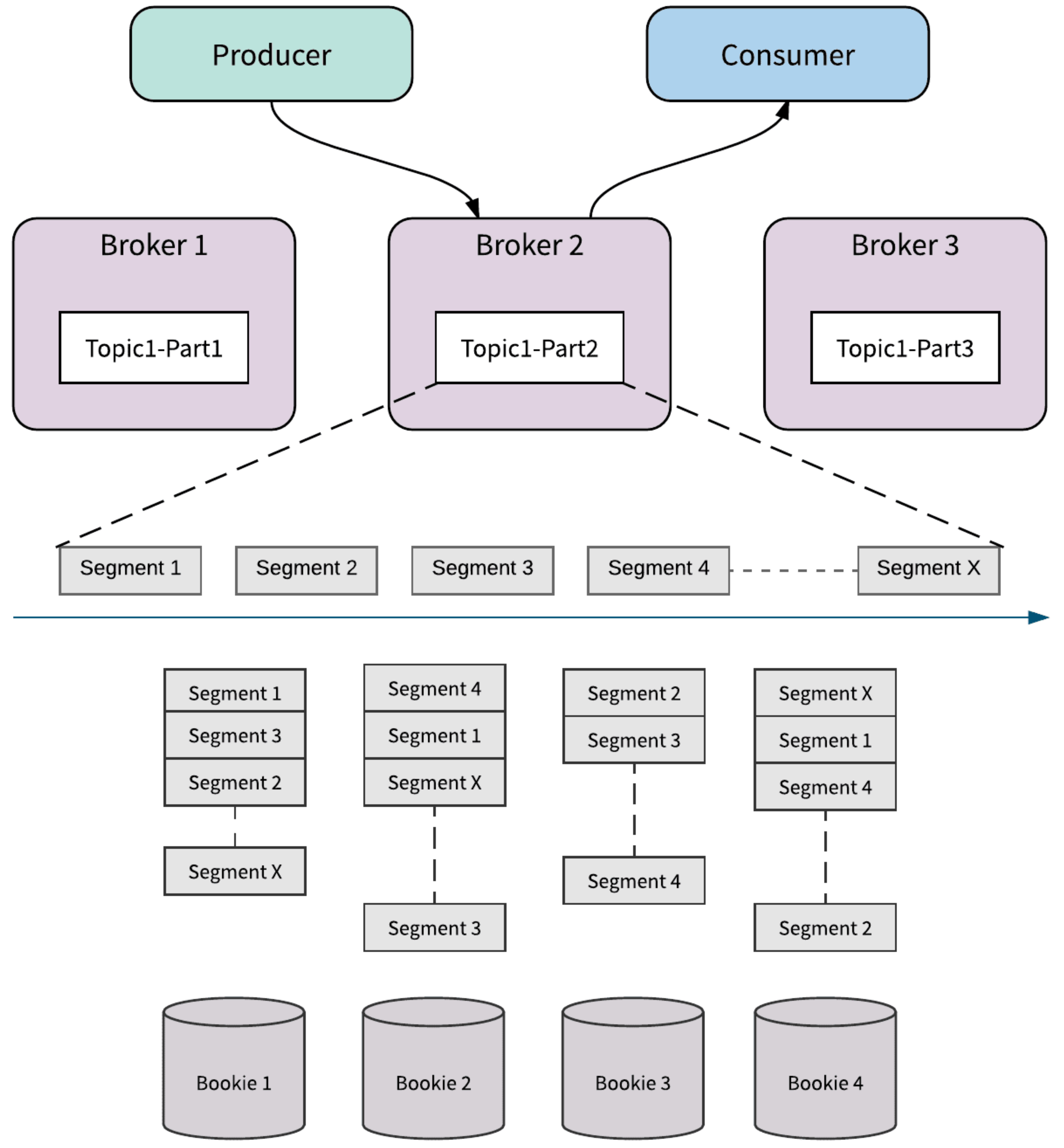
数据持久层

Durable storage

基于分片的数据存储

Pulsar -- Segment-based Storage

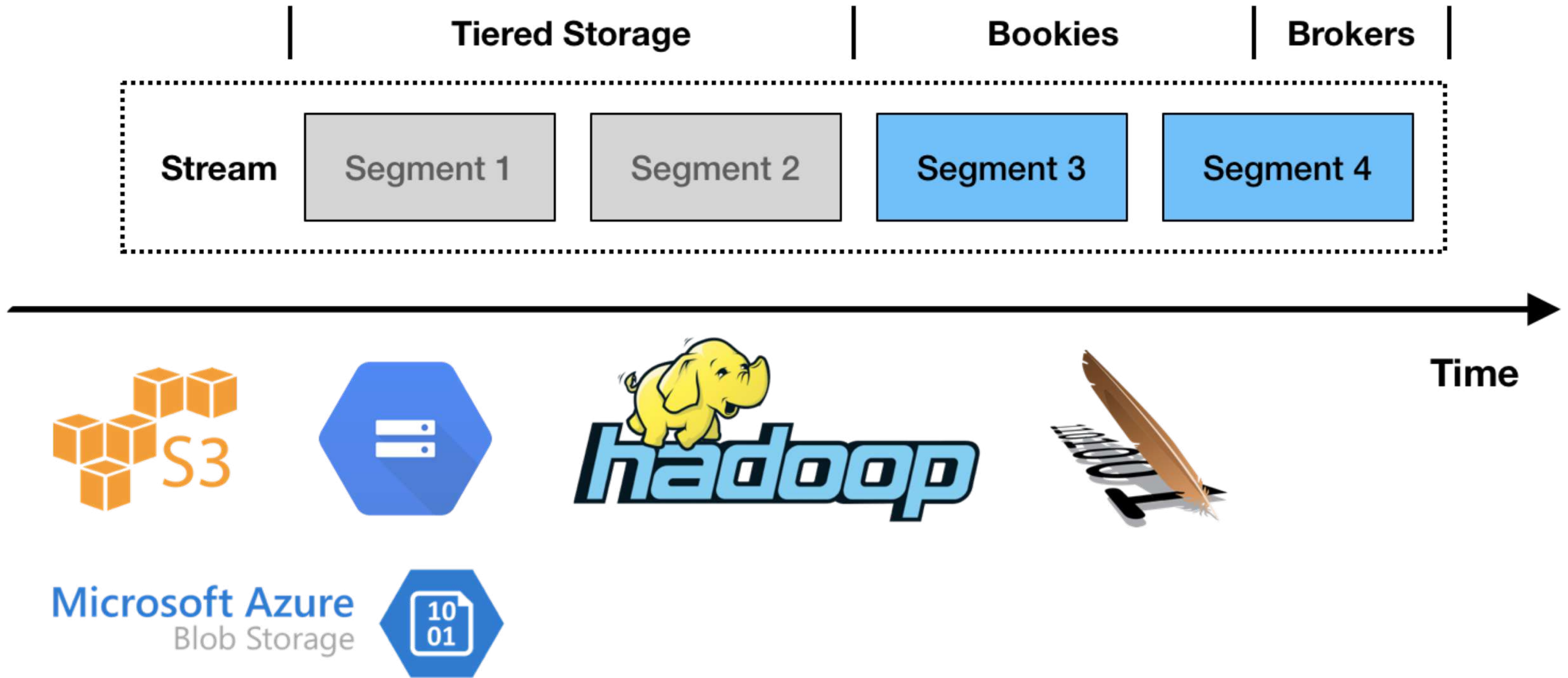
- **Managed Ledger**
  - **Topic** 的存储抽象
    Storage layer for a single topic
- **Ledger**
  - 单写者，追加写
    Single writer, append-only
  - 被复制到多个 **bookie** 节点上
    Replicated to multiple bookies



无限、廉价的数据存储

Pulsar -- Infinite Data Storage

- 使用廉价存储，持久化无限数据
  Reduce storage cost
  - 按照分片粒度将数据卸载到廉价存储中
    Offloading segment to tiered storage one-by-one



有结构的数据

Pulsar - Structured Data

- 内置的 **Schema** 注册
  Built-in schema registry
  - 在服务器端的消息结构共识
    Consensus of data at server-side
  - **Topic** 级别的消息结构
    Data schema on a per-topic basis

- 直接产生、消费有结构的数据
  Send and receive typed message directly
  - **Pulsar** 进行消息验证
    Validation
  - 支持消息版本的演化
    Multi-version

Pulsar Flink 连接器

Pulsar Flink Connector

03

连接器API

Flink Pulsar Connector -- API

- Read

```
val props = new Properties()
props.setProperty("service.url", ...)
props.setProperty("admin.url", ...)
props.setProperty("partitionDiscoveryInterval
Millis", "5000")
props.setProperty("startingOffsets", "earliest")
props.setProperty("topic", "test-source-topic")
val source = new FlinkPulsarSource(props)
val dataStream = env.addSource(source)
```

- Write

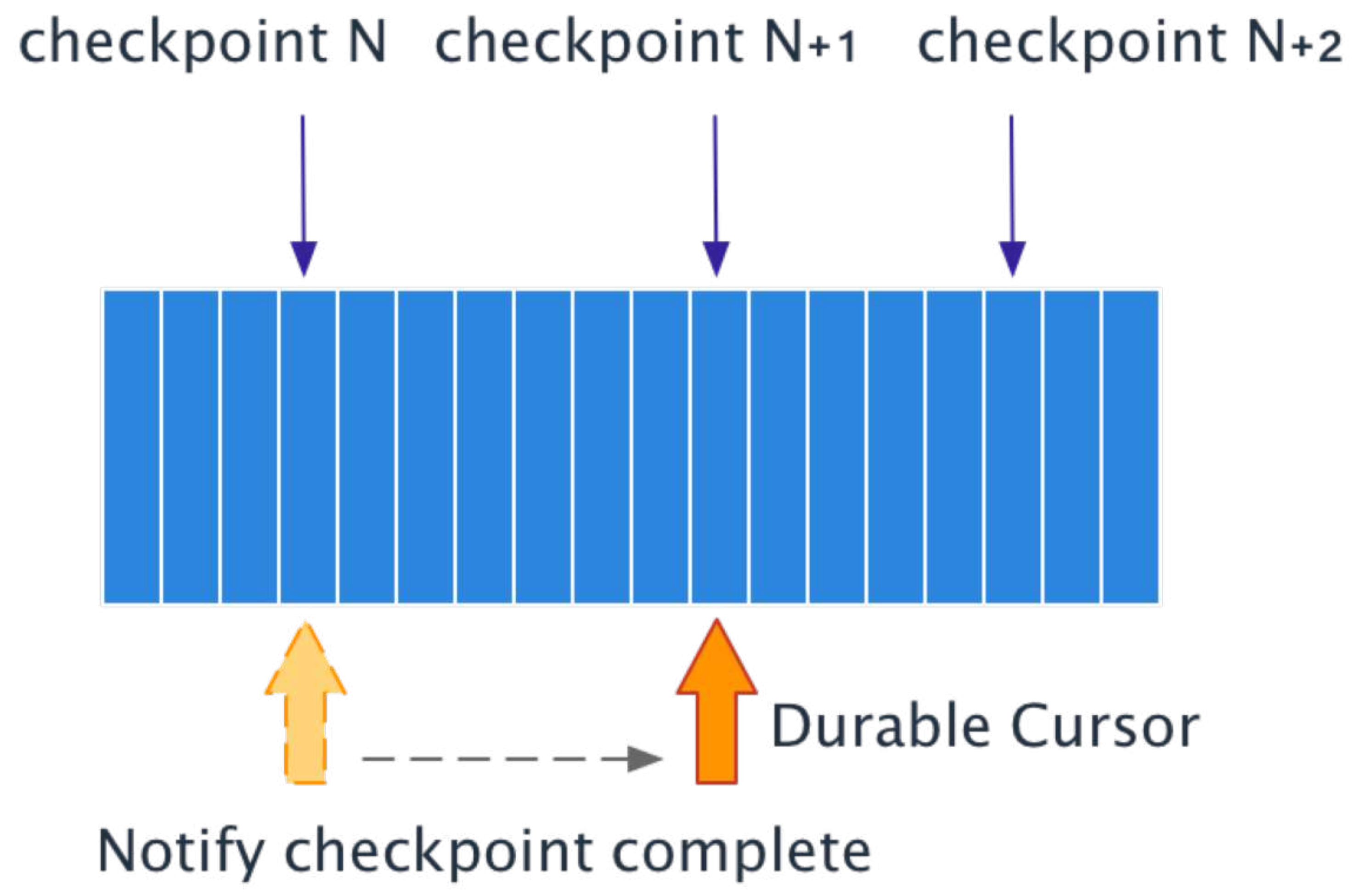
```
val prop = new Properties()
prop.setProperty("service.url", ...)
prop.setProperty("admin.url", ...)
prop.setProperty("flushOnCheckpoint", "true")
prop.setProperty("failOnWrite", "true")
props.setProperty("topic", "test-sink-topic")
stream.addSink(new FlinkPulsarSink[Row](prop,
DummyTopicKeyExtractor))
```

```
tEnv
  .connect(new Pulsar().properties(props))
  .inAppendMode()
  .registerTableSource("pulsar-test-table")
```

持久化、可重放的数据源

Durable and ordered source

- 故障无法避免
  Failures are inevitable for engines
  - **Task** 从 **checkpoint** 中恢复
    Tasks recover from checkpoint
- **Exactly-once**
  - 基于 **topic** 内消息有序的特性
    Based on message order in topic
  - 通过 **Seek & read** 实现
    Implement based on seek and read
- 通过额外的订阅避免消息被删除
  Messages ”keep-alive” by subscription
  - 在得到 **checkpoint** 完成通知时移动订阅游标
    Move sub cursor on commit



结构化数据存取

Processing typed records

- 将 **Pulsar topic** 看作是一张有结构的表
  Regard Pulsar as structured storage
- 在任务调度期获取表 **Schema** 定义
  Fetching schema as the first step
- 将 **Pulsar message** （反）序列化成**Row**
  SerDe your messages into Row
  - 支持 **avro/json/protobuf** 的消息转换
    Avro schema and avro/json/protobuf Message
- 消息元数据转化为表的内部列
  Message metadata as metadata fields
  - **__key, __publishTime, __eventTime, __messageId, __topic**

Topic 和 Partition 发现

Topic/Partition discovery

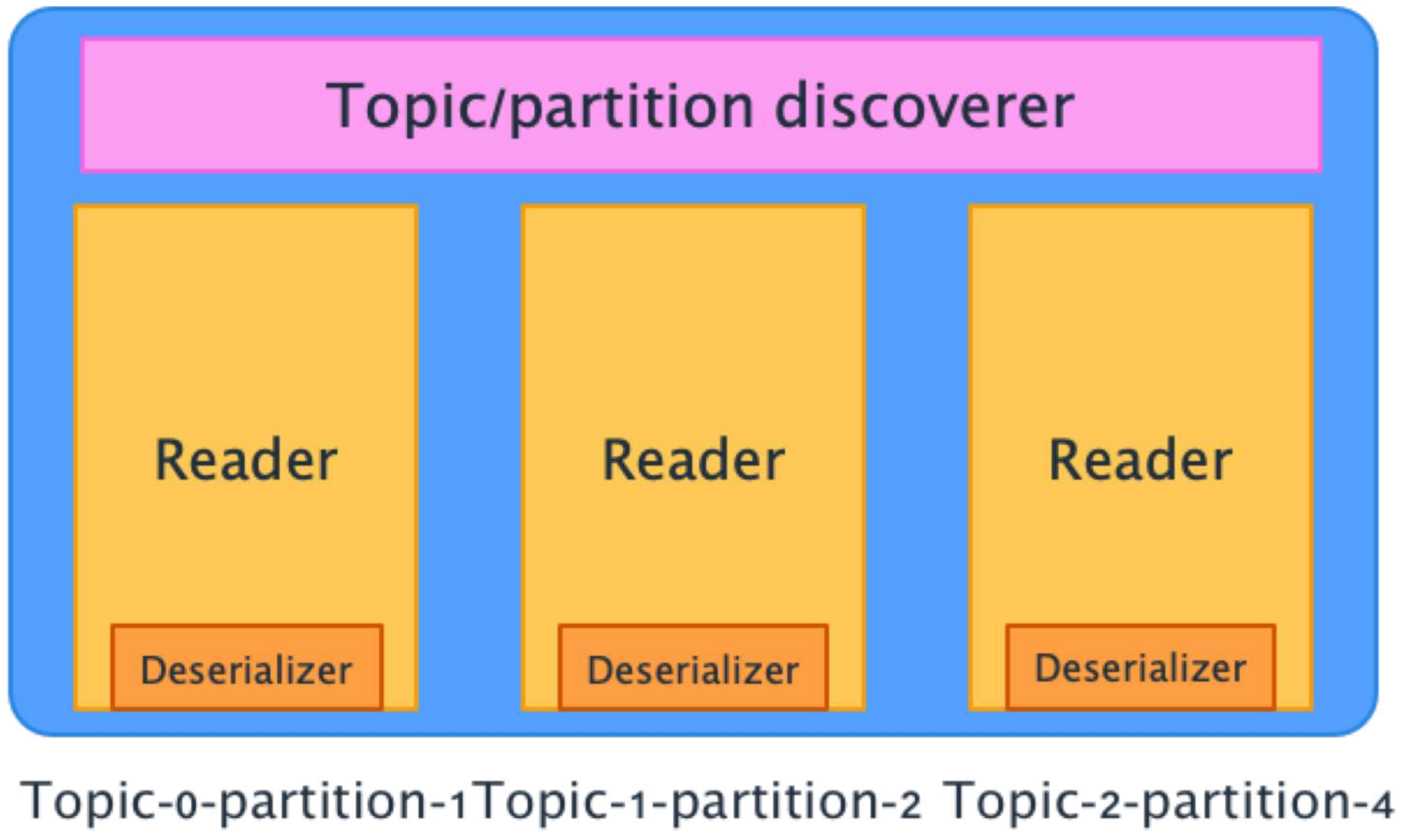
- 流处理作业是长时间运行的

  Streaming jobs are long running
- 在作业执行期间，**topic** 可能被添加或删除

  Topics & partitions may be added on removed during a job
- 阶段性检查 **topic** 状态

  Periodically check topic for status
  - 每个 **task** 内部一个用于监控的线程

    With a monitoring thread in each task



未来方向

Future Directions

04

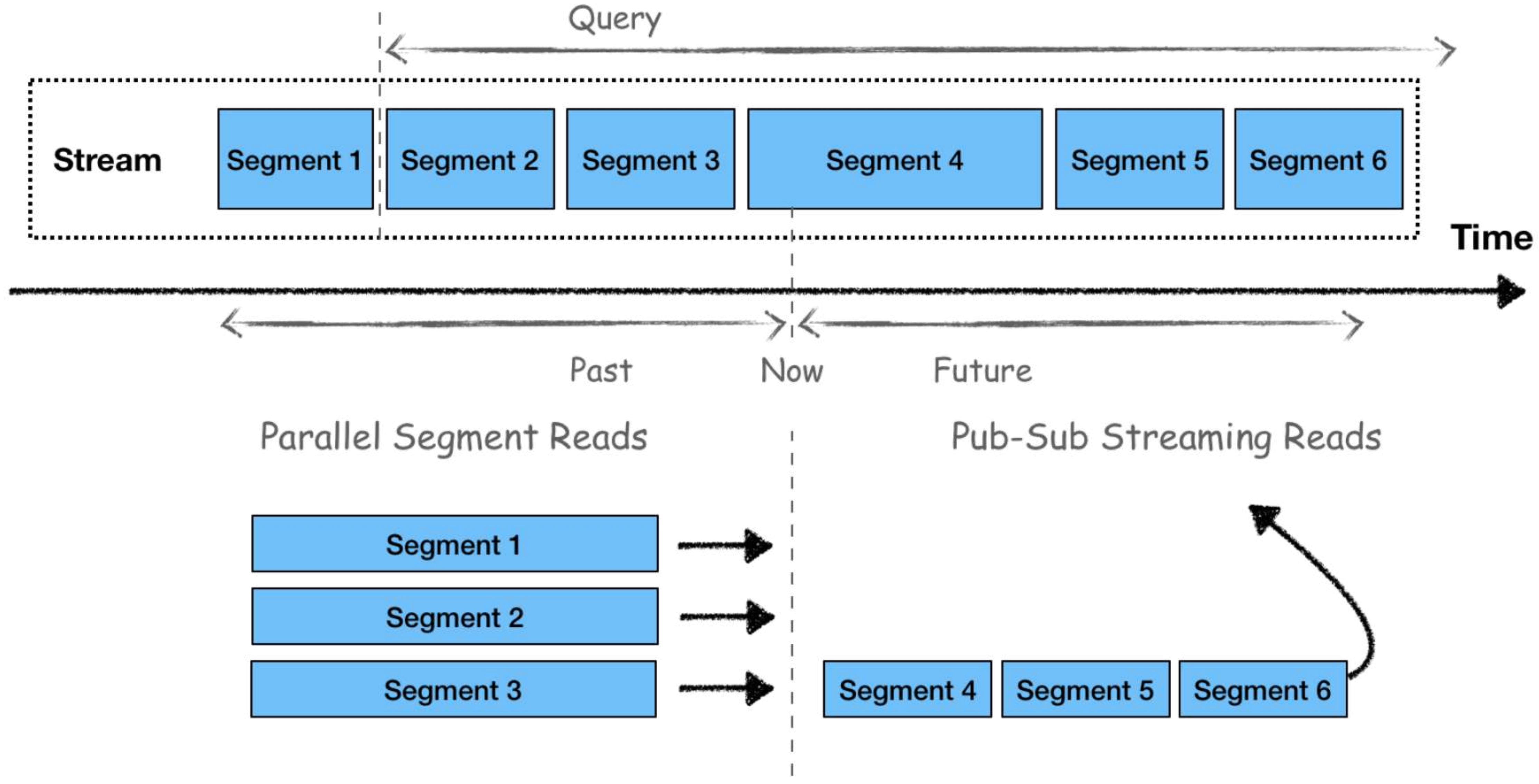
分析友好的数据组织、访问

Analytical-friendly data organizations and access method

- 谓词下推 + 粗粒度索引
  Filter push down & coarse-grained index
  - Segment 级别的 max、
    Max/min at segment level
    min
    Generated by brokers
  - Broker 收集，写入
    Indices are generated during broker put
    Segment 的元数据

- 列形式组织 Segment
  Organize segment data in columnar format
  - 针对分析型负载
    Target at analytical workloads
  - 节约磁盘带宽
    Save disk-bandwidth / network IO
  - 节约 CPU 时间
    Save CPU time

- 更灵活的数据消费模式
  More flexible data consumption mode



Pulsar 社区

Pulsar Community

智联招聘 zhaopin.com

Tencent 腾讯

VIPKID

中国移动 China Mobile

tuya 涂鸦智能

EMQ

中国电信 CHINA TELECOM

天翼物联

360 TOTAL SECURITY

OKCoin

甜橙金融 ORANGE FINANCE

拉卡拉

晓多 · 知晓更多

虎牙直播 Huya.com

环球易购 Globalegrow E-Commerce

维护大师

十虹

Pulsar 社区

Pulsar Community

narvar gsk MISO BIRD netskope RetailMeNot inc.

MACQUARIE Cdiscount GoSecure POWERED BY COUNTERTACK iterable Engagio

verizon media group comcast BOSE NUTANIX overstock SIEMENS

HSL HRT Giggso proxyclick YAHOO! JAPAN THEHUTGROUP

One Click Retail ONLINE SIMPLIFIED GM IoTium mercado libre sticorp SOLUÇÕES INTELIGENTES TaxiStartup

</ndustrial.io> toast

TARGET Disney Capital One Pinterest UNITED Spotify Booking.com ERICSSON Honeywell

Danske Bank Digital Asset hulu MOTORQ Accionlabs ThoughtWorks Picatic HomeAway PEX box

here CoolFront REFRIGERATION MACROMETA Gridwise Mediture ReLOPS FORMULATE chime Implisense INSTRUCTURE exenSa PETUUM

THANKS