Elastic Data Processing with Apache Flink and Apache Pulsar

Sijie Guo (sijieg)

2019-04-02

Who am I

- Apache Pulsar PMC Member
- Apache BookKeeper PMC Member
- Interested in technologies around Event Streaming

Agenda

- What is Apache Pulsar?
- A Pulsar View on Data - Segmented Stream
- Pulsar - Access Pattern & Tiered Storage
- Pulsar - Schema
- When Flink meets Pulsar

What is Apache Pulsar?

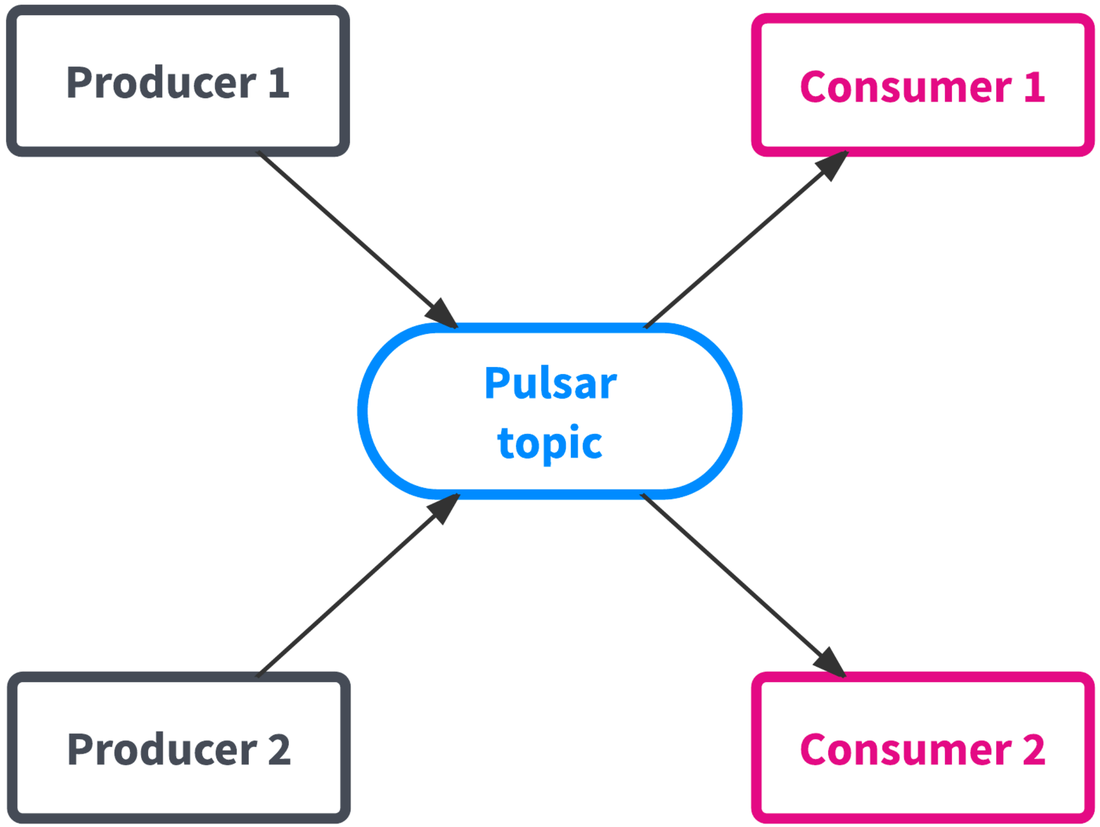
Pub/Sub Messaging

2003 2010 2012

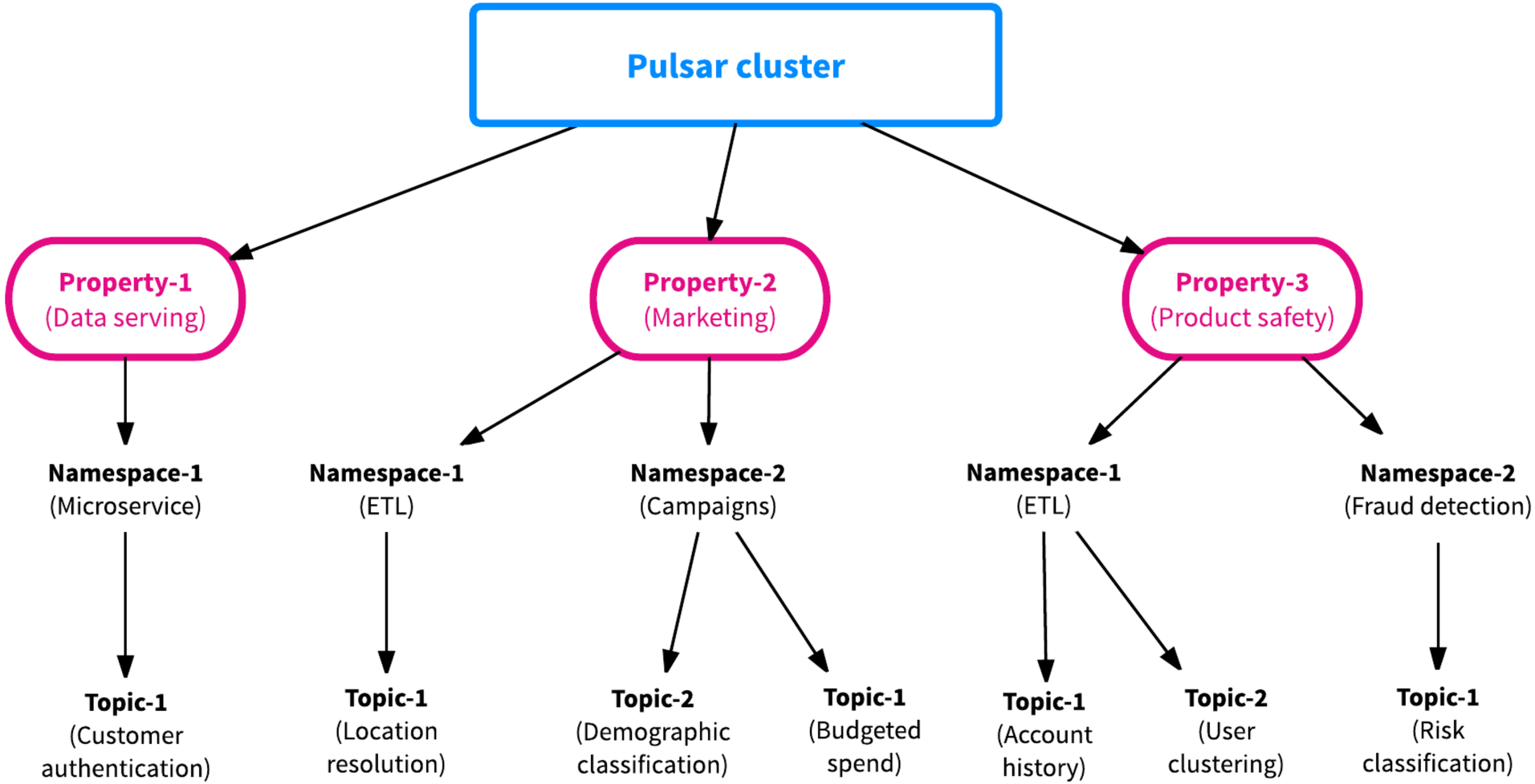
2006 2011

“Flexible Pub/Sub **messaging**
backed by durable log/stream **storage**”

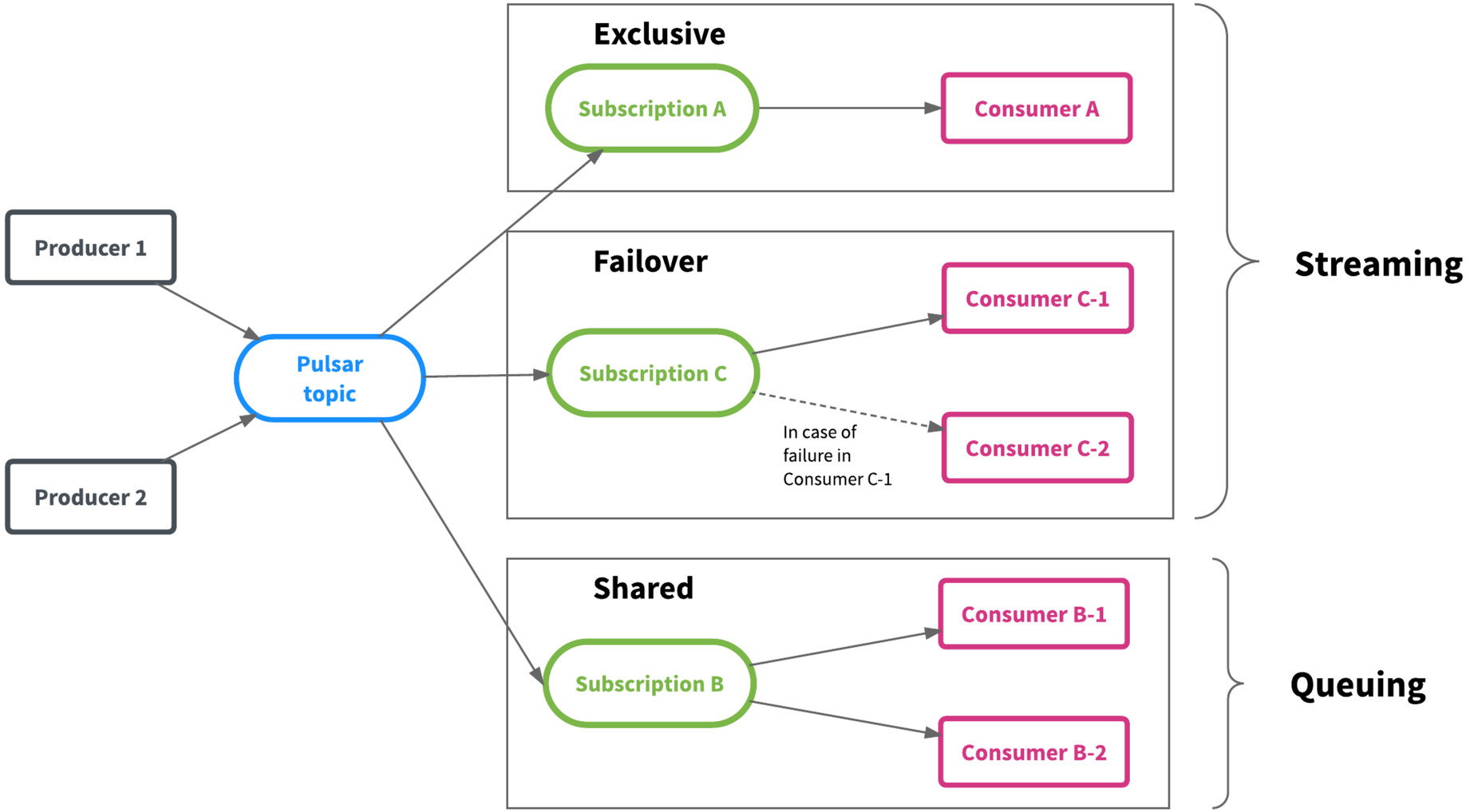
Pulsar - Pub/Sub



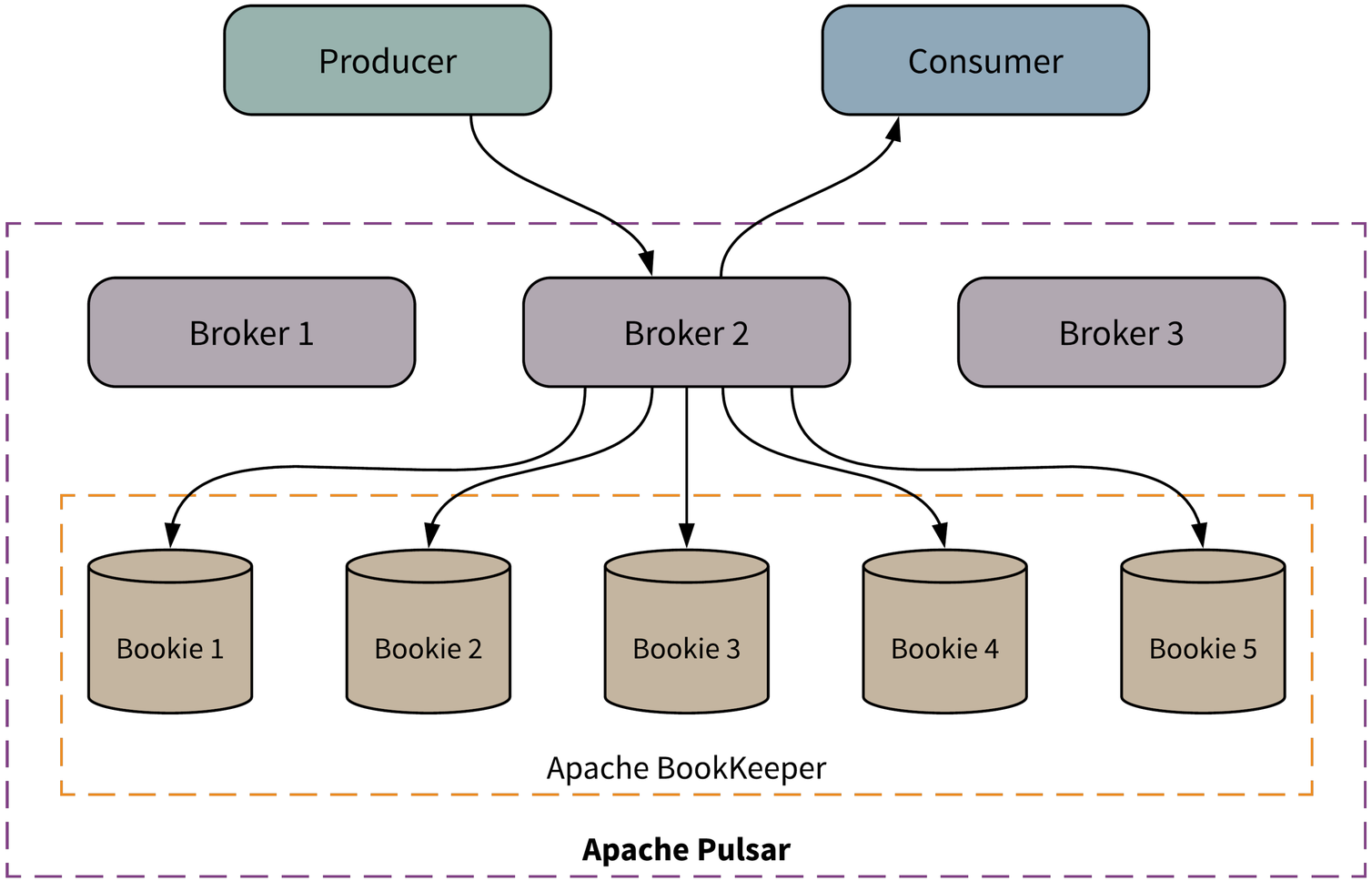
Pulsar - Multi Tenancy



Pulsar - Queue + Streaming



Pulsar - Cloud Native



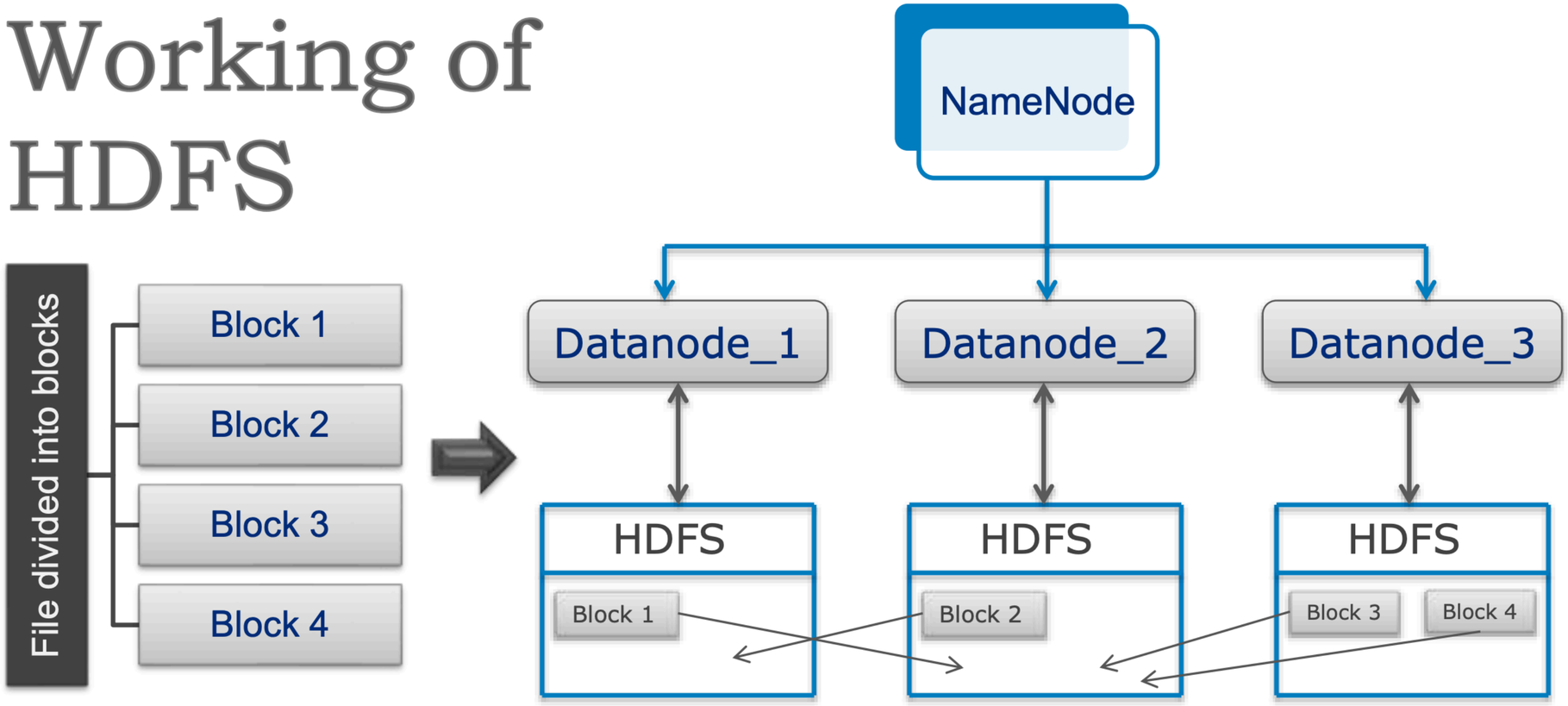
Layered Architecture

- Independent Scalability
- Instant Failure Recovery
- Balance–free on cluster expansions

A Pulsar View on Data

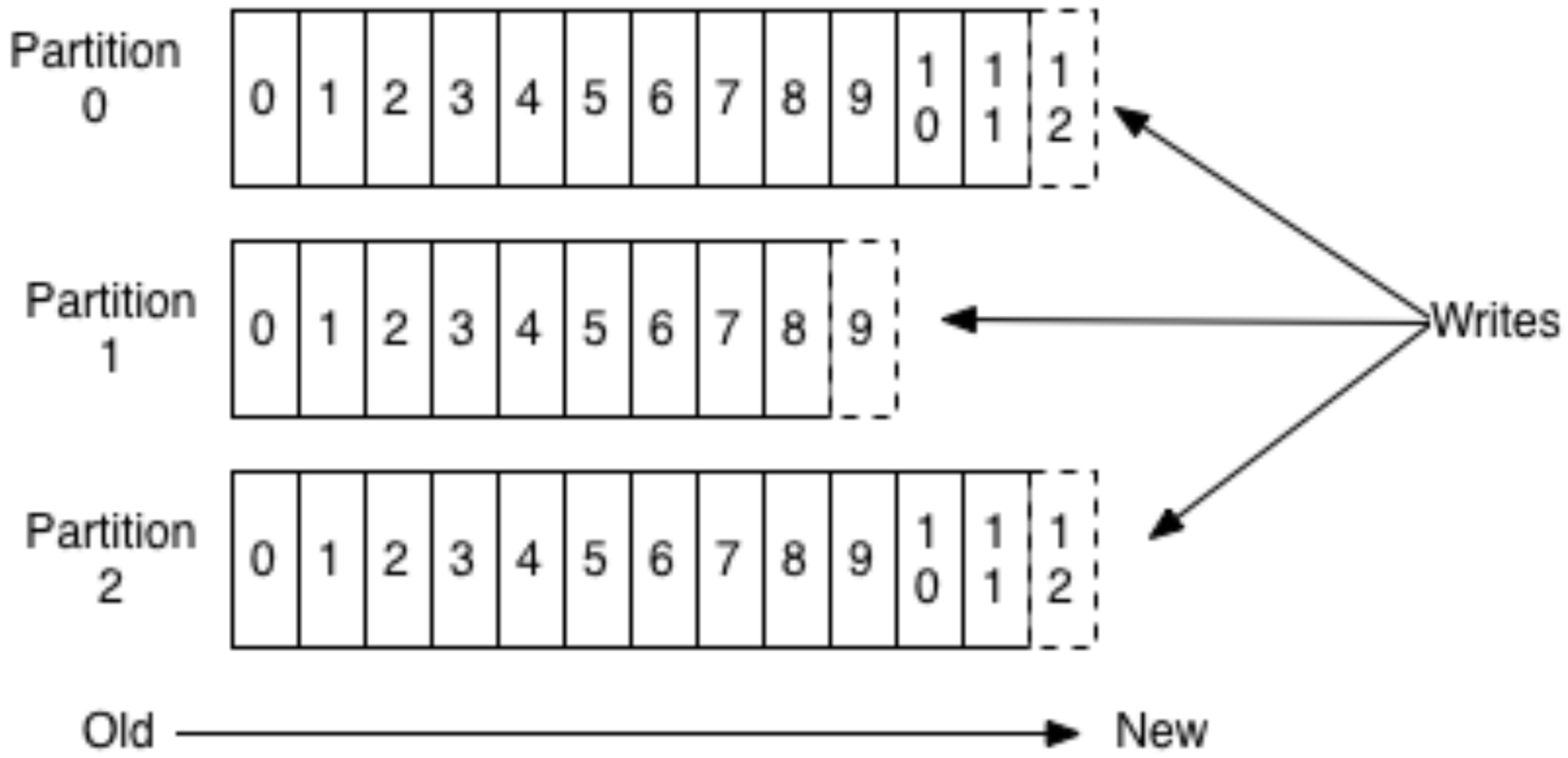
Batch - HDFS

Working of HDFS



Storage & Replication of Blocks in HDFS

Stream - Pub/Sub

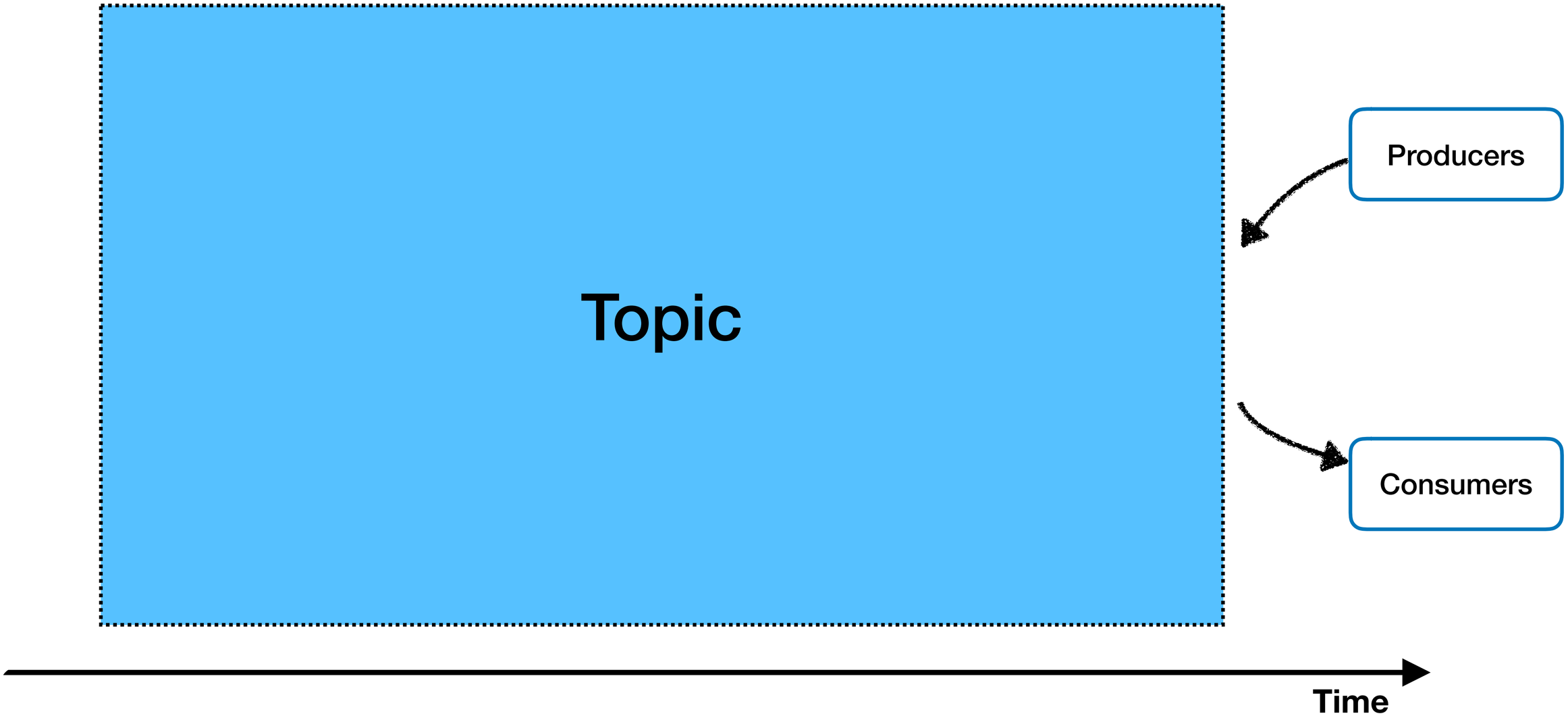


A Flink View on Computing

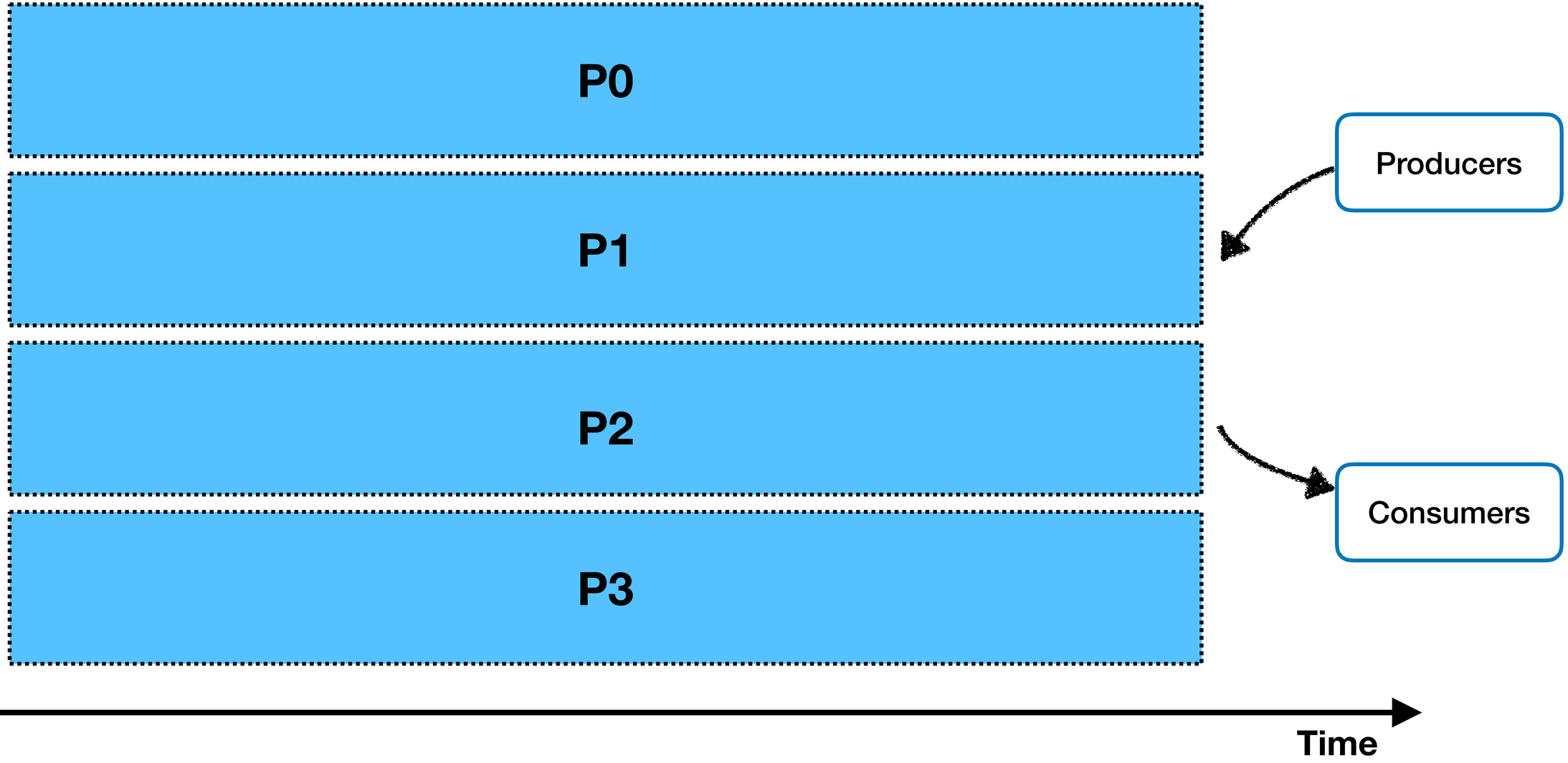
“Batch processing is a special case of
Stream processing”

Pulsar = *Segmented* **Stream**

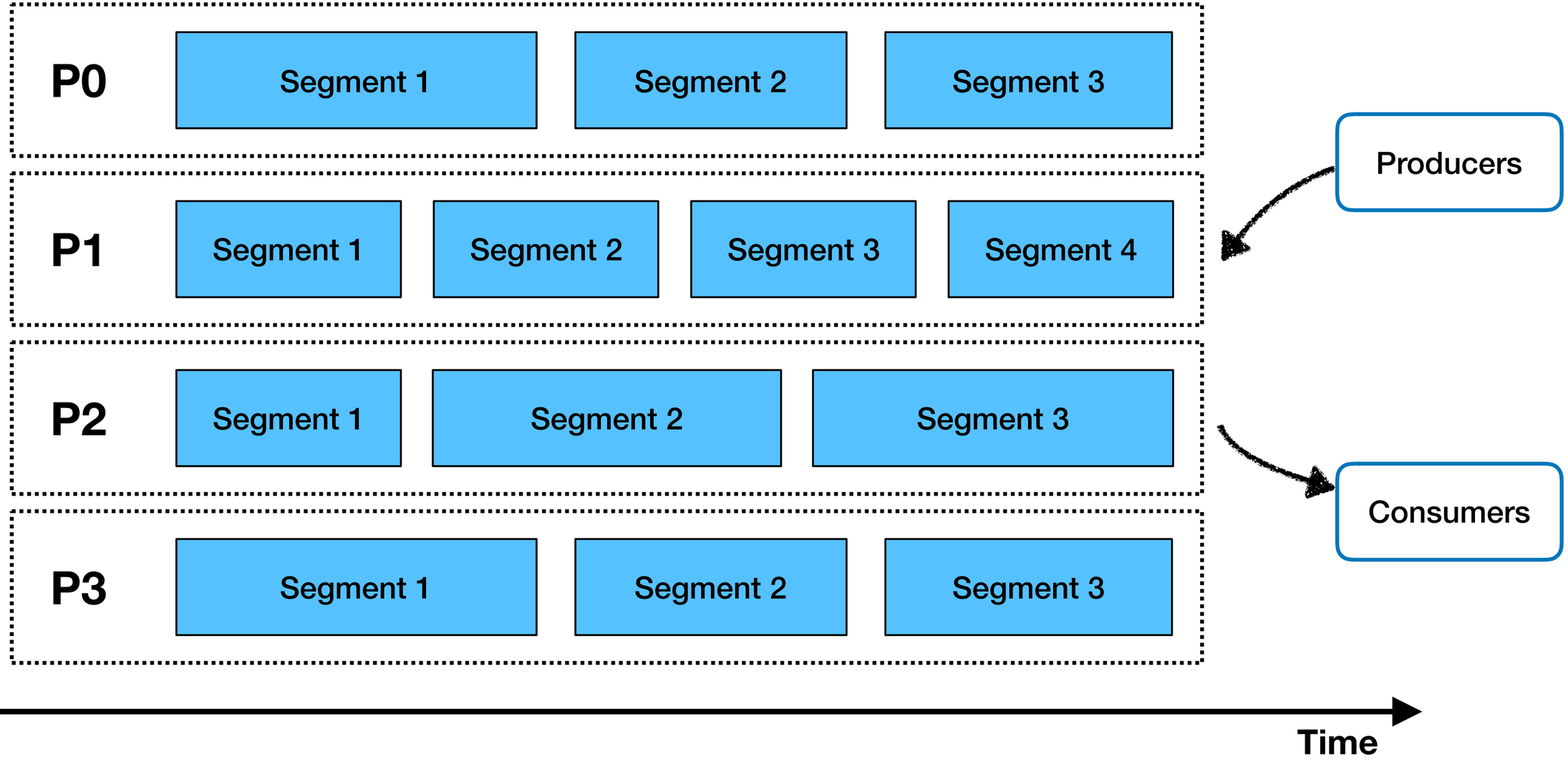
Topic



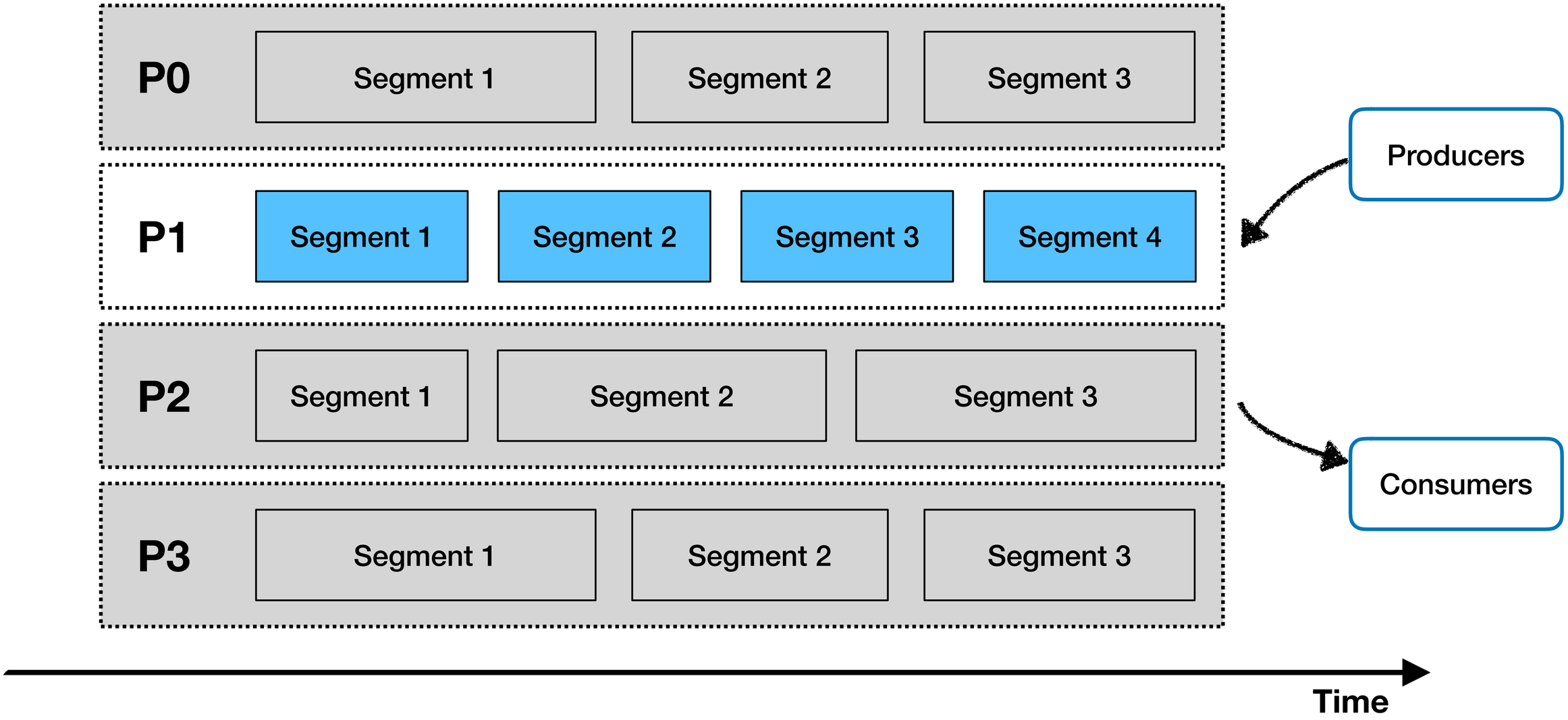
Partitions



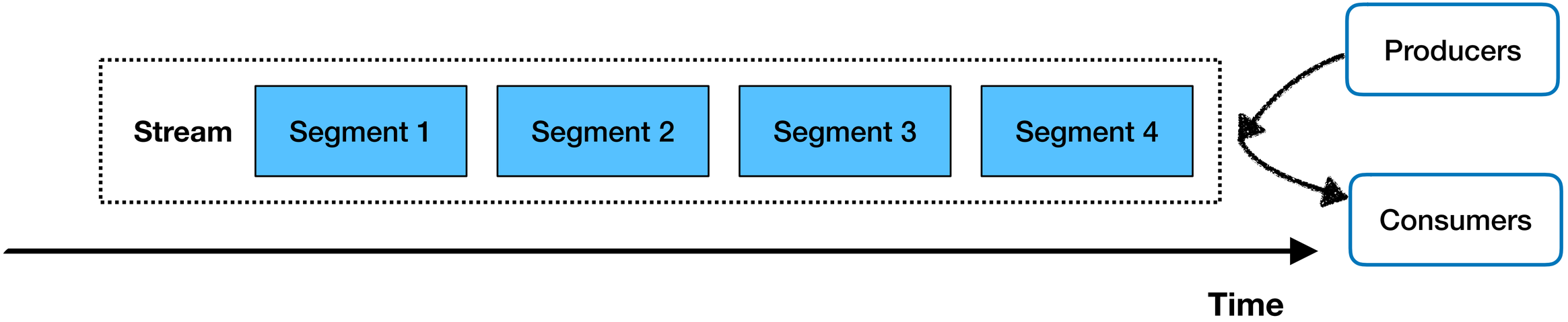
Segments


P0
Segment 1
Segment 2
Segment 3
P1
Segment 1
Segment 2
Segment 3
Segment 4
P2
Segment 1
Segment 2
Segment 3
P3
Segment 1
Segment 2
Segment 3
Producers
Consumers
Time

Stream


P0
Segment 1
Segment 2
Segment 3
P1
Segment 1
Segment 2
Segment 3
Segment 4
P2
Segment 1
Segment 2
Segment 3
P3
Segment 1
Segment 2
Segment 3
Producers
Consumers
Time

Stream

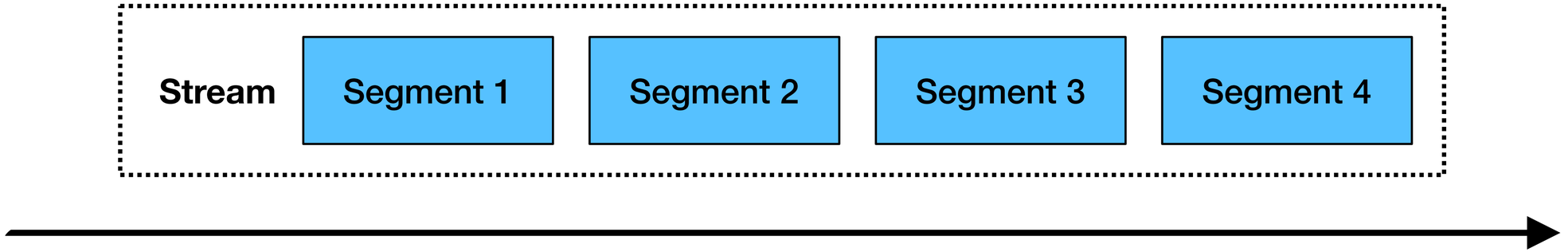

Stream
Segment 1
Segment 2
Segment 3
Segment 4
Producers
Consumers
Time

Segmented Stream

- Segmented Stream Systems
  - Apache Pulsar, Twitter EventBus, EMC Pravega
  - All Apache BookKeeper based
  - Used BK in a different way
    - Pulsar, EventBus - Uses BK as the segment store
    - Pravega - Uses BK as the journal only

Access Patterns

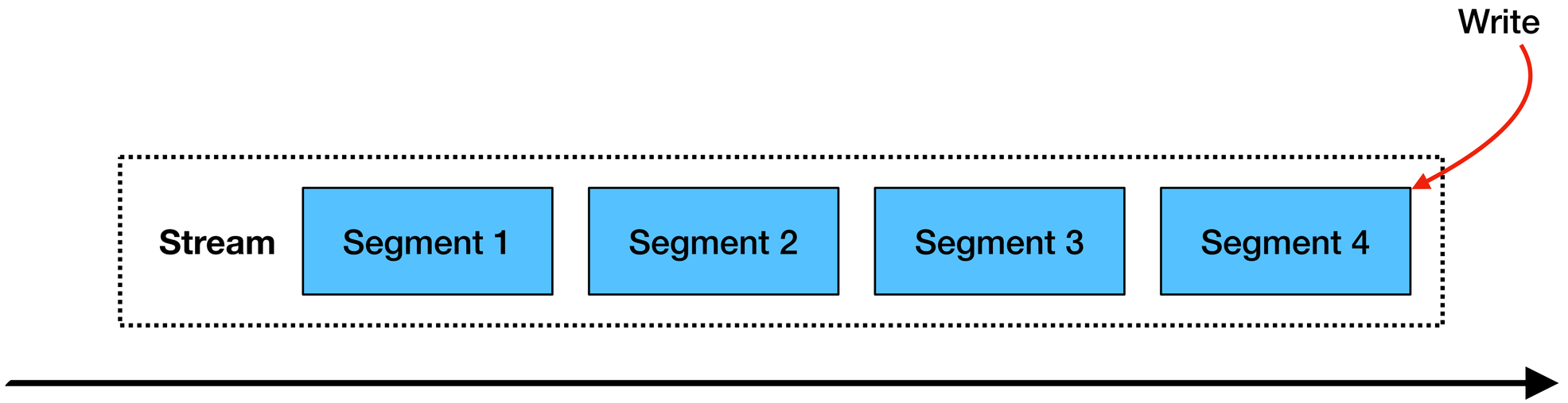
- ✓ Write
  写
- ✓ Tailing Read
  追尾
- ✓ Catchup Read
  追赶读



Access Patterns

✓ **Write**

写



✓ Tailing Read

追尾

✓ Catchup Read

追赶读

Access Patterns

✓ Write

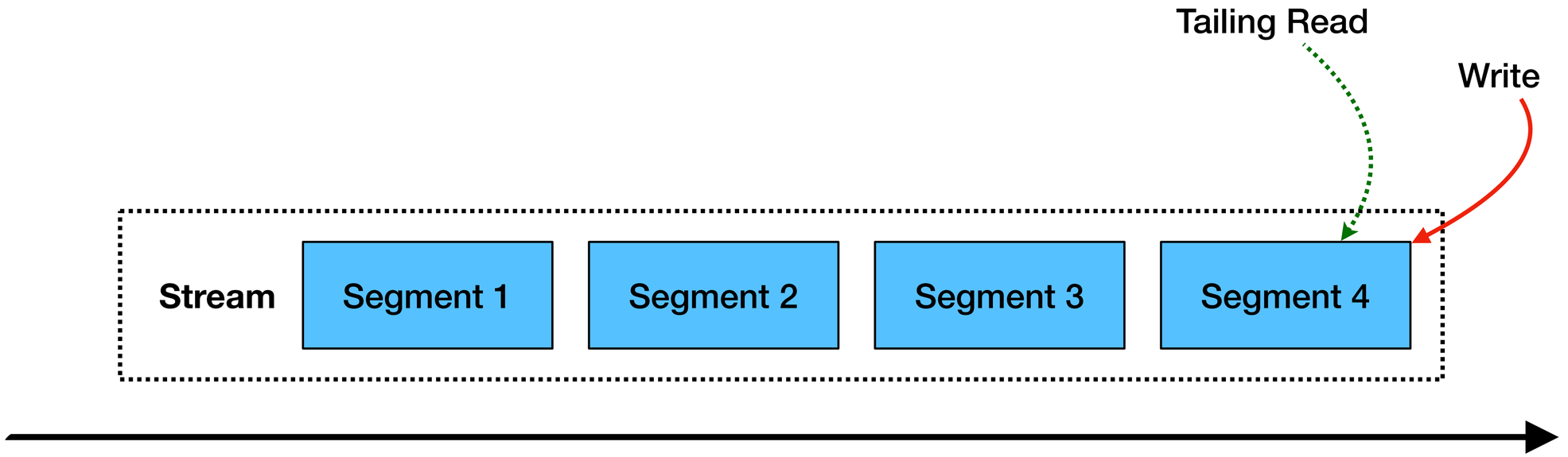
写

✓ **Tailing Read**

追尾

✓ Catchup Read

追赶读



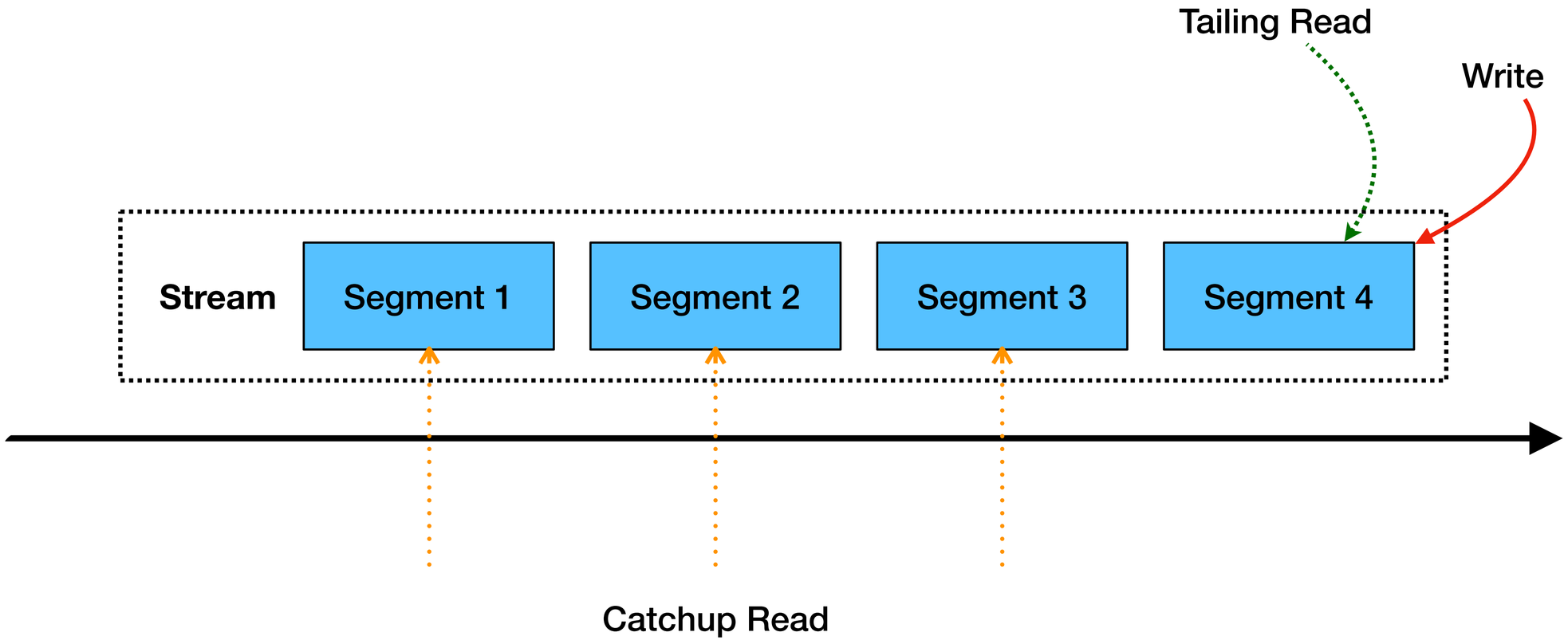
Access Patterns

✓ Write

写

✓ Tailing Read

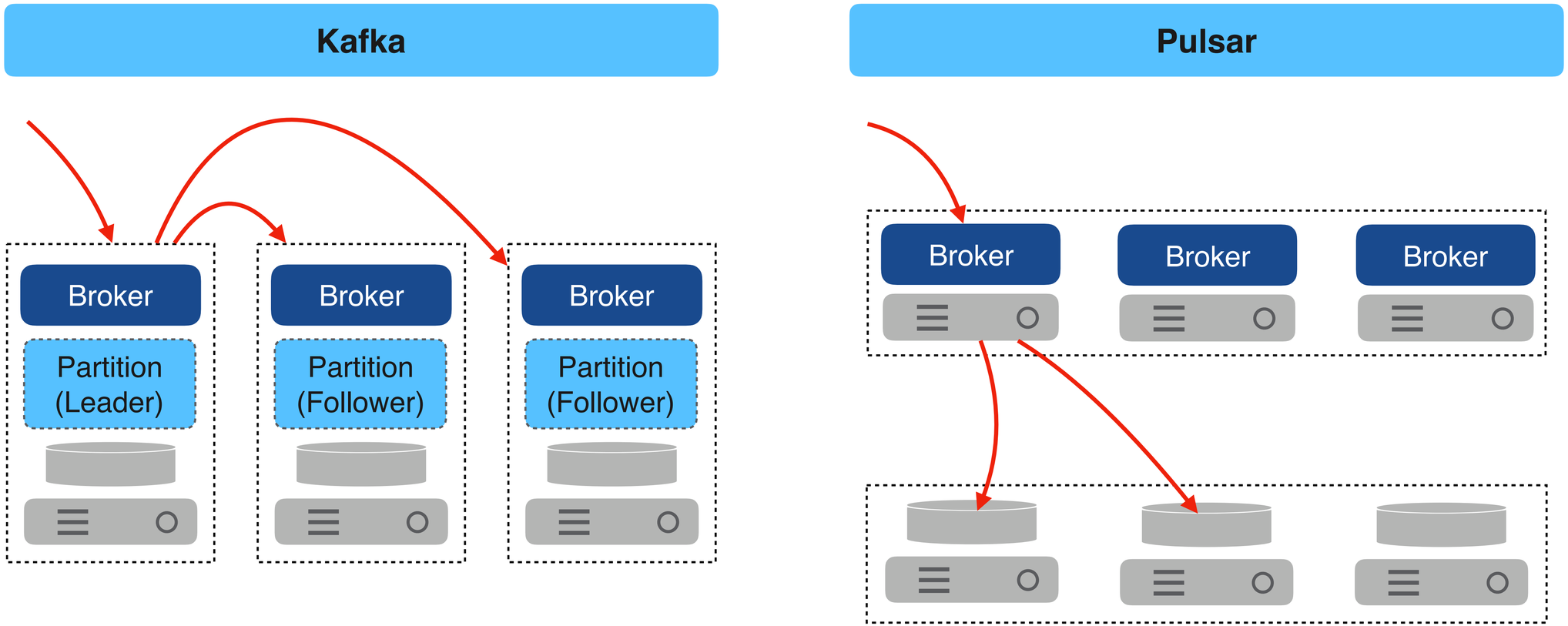
追尾



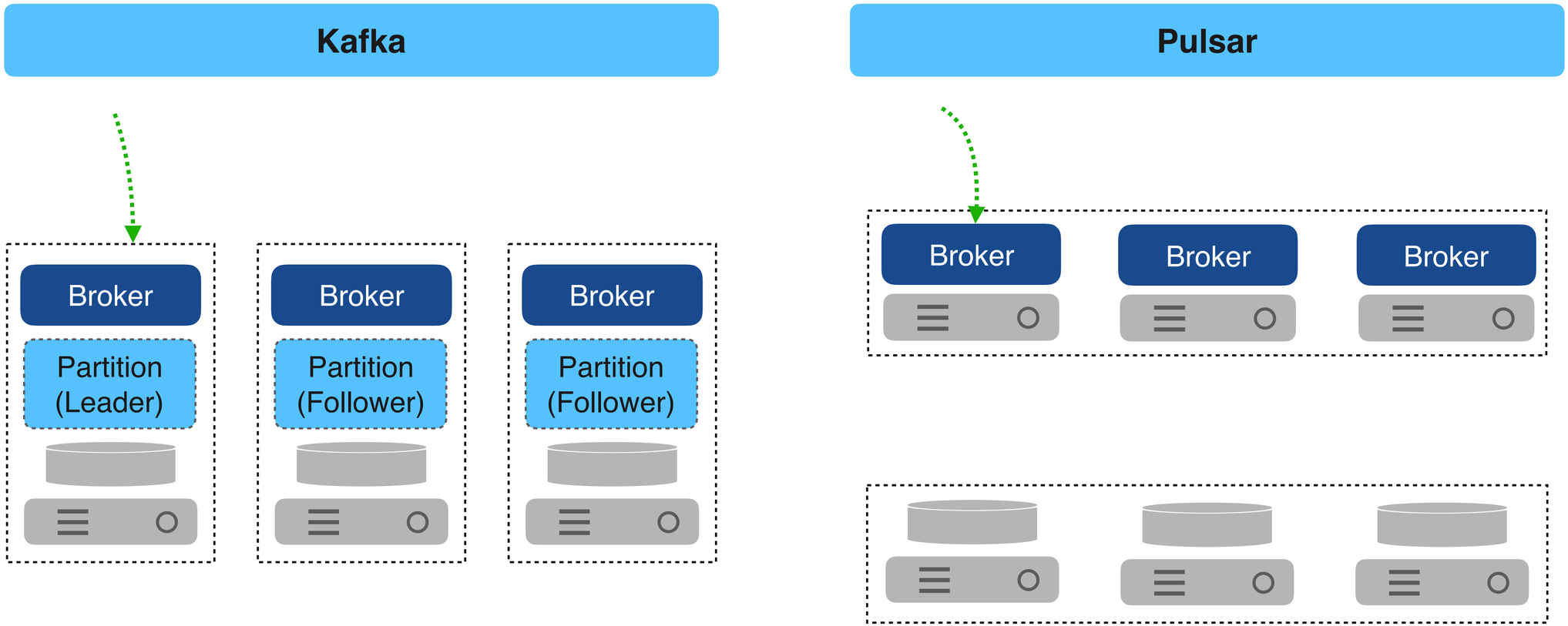
✓ **Catchup Read**

追赶读

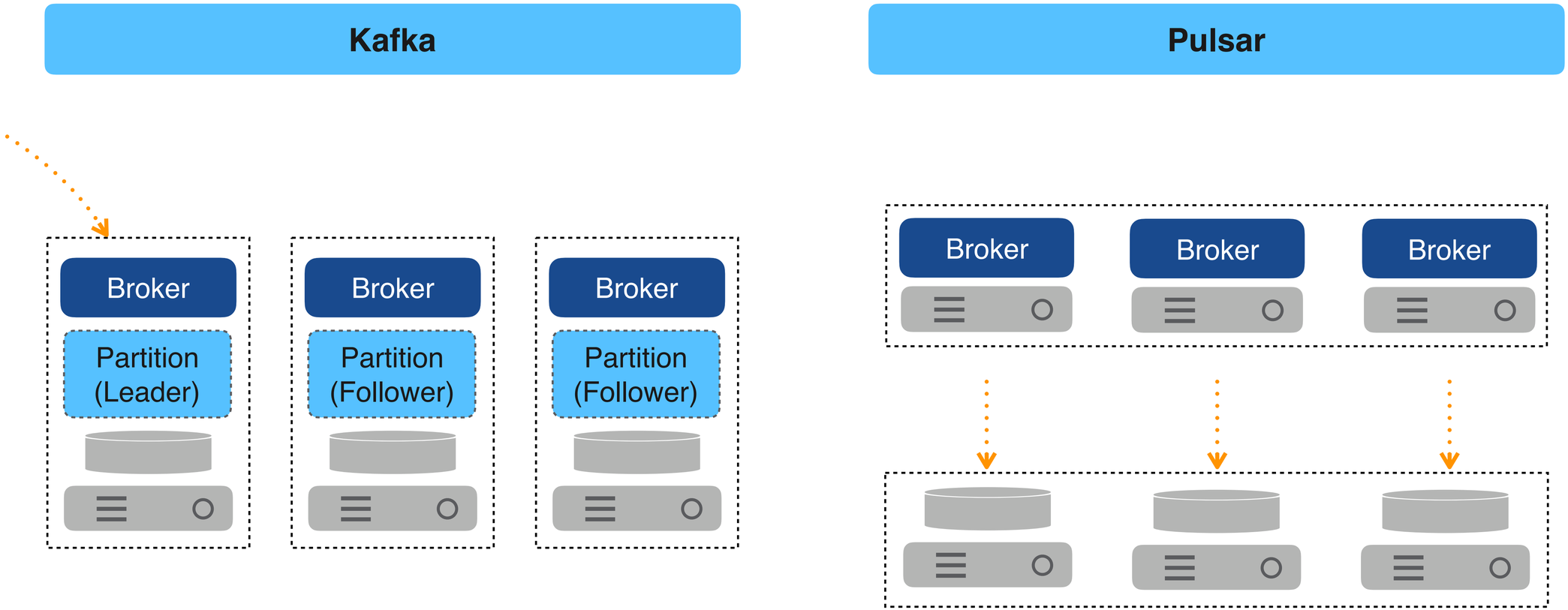
Write


Kafka
Broker
Partition (Leader)
Broker
Partition (Follower)
Broker
Partition (Follower)
Pulsar
Broker
Broker
Broker

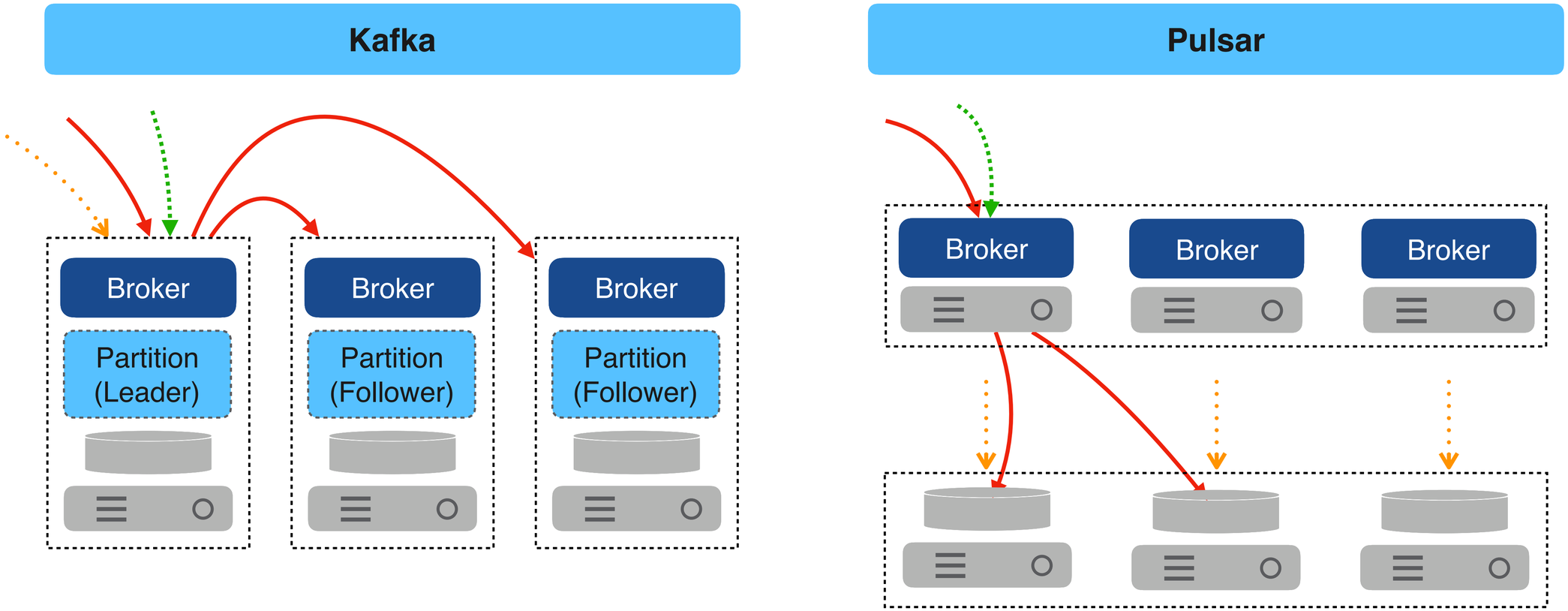
Tailing Read



Catchup Read

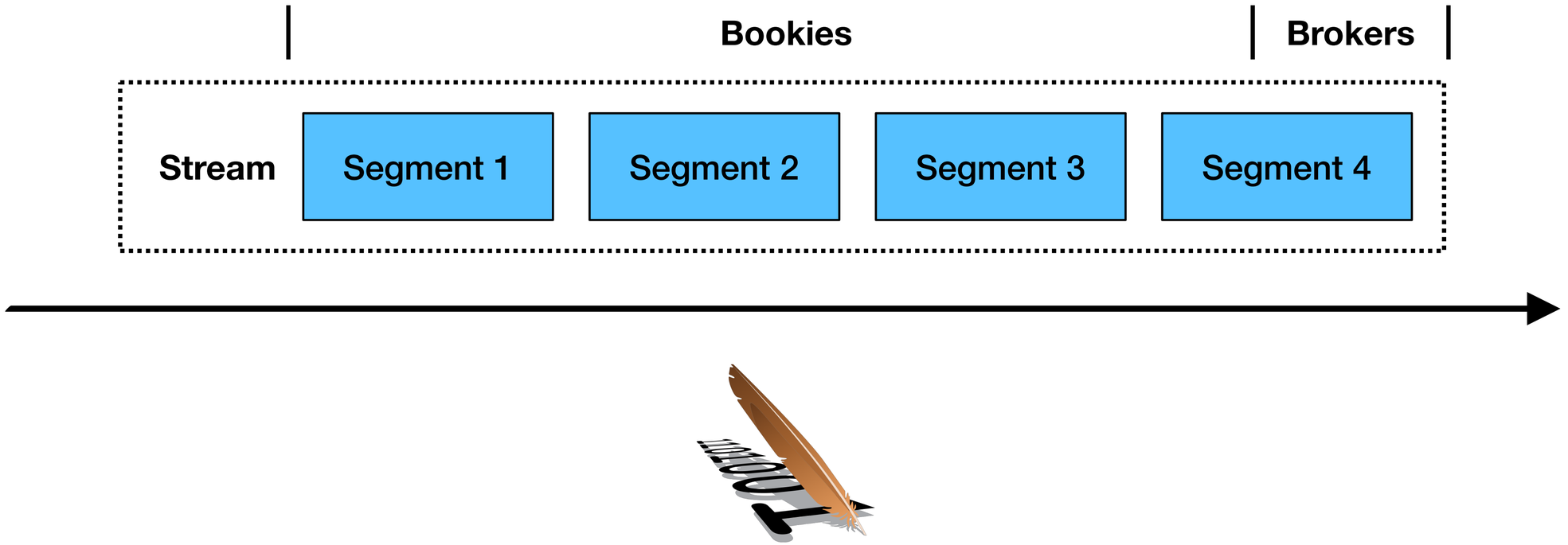


IO Isolation



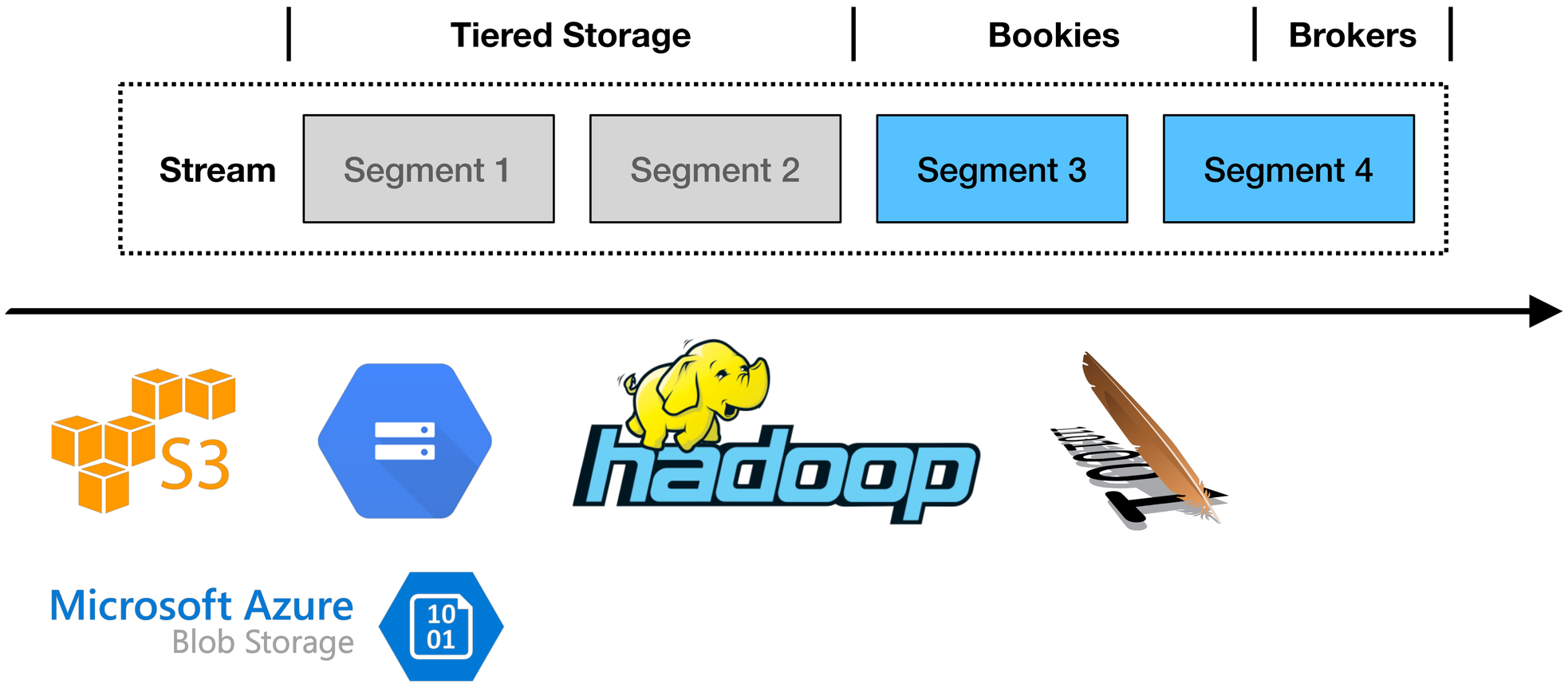
Infinite Stream

- ✓ Write
  写
- ✓ Tailing Read
  追尾
- ✓ Catchup Read
  追赶读



Infinite Stream

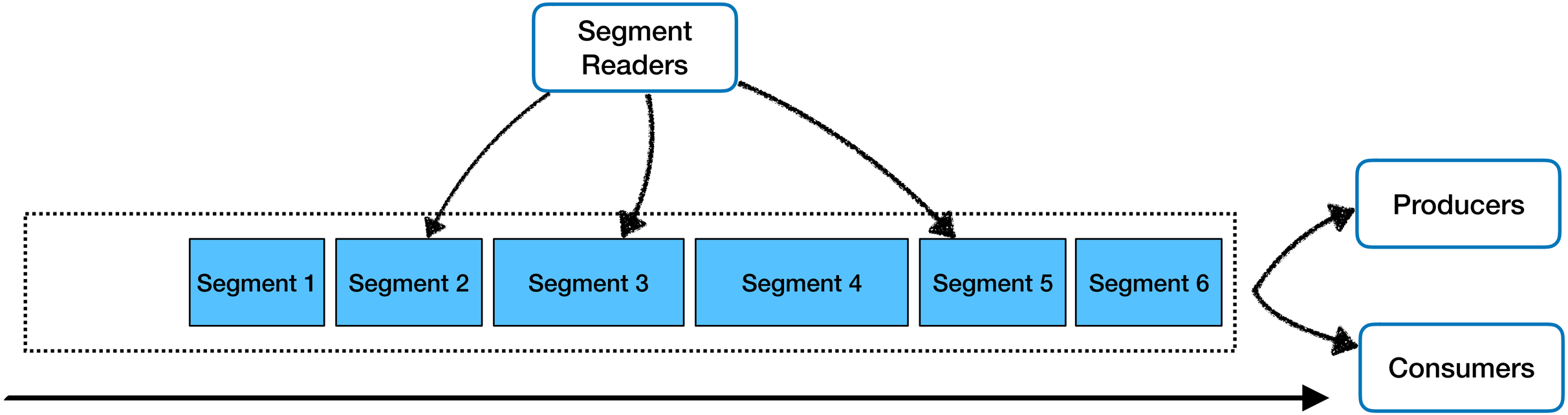
- ✓ Write
  写
- ✓ Tailing Read
  追尾
- ✓ Catchup Read
  追赶读



Tiered Storage

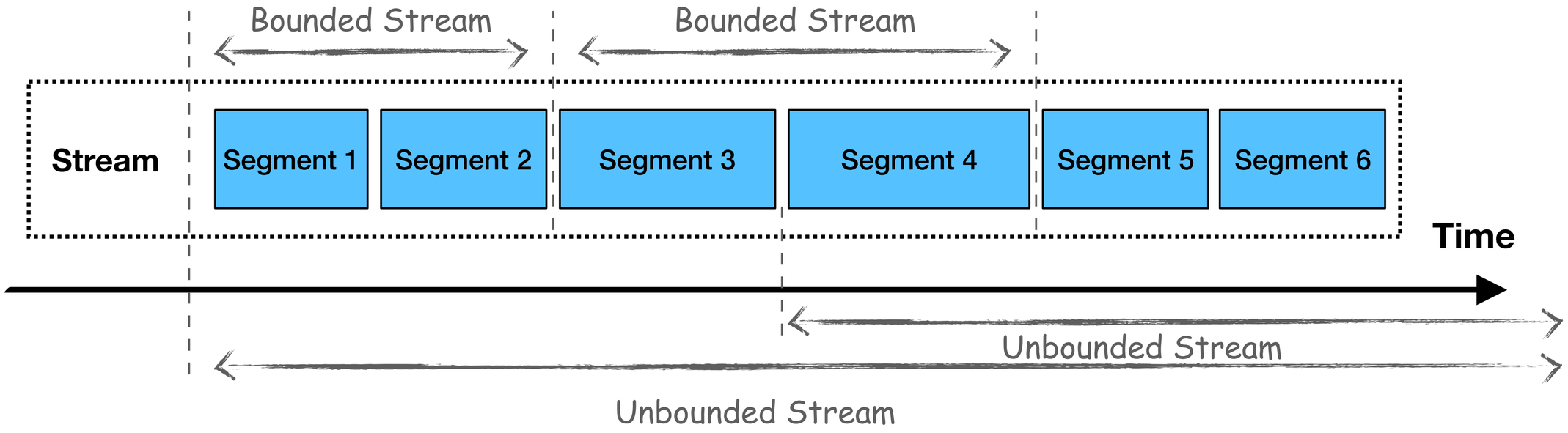
- Offloader
  - When: size-based, time-based, or triggered by pulsar-admin
  - How: copy a segment to tiered storage, and delete it from bookkeeper
  - Access: broker knows how to read the data back, or bypass read the offloaded segments directly
- Available Offloaders
  - Cloud Offloder : AWS, GCS, Azure, …
  - HDFS, Ceph, …

Stream as a Unified View on Data



Microsoft Azure
Blob Storage

Data Processing on Pulsar



When Flink meets Pulsar

Goals

- Flink + Pulsar
  - Streaming Connectors
  - Source Connectors
  - PulsarCatalog: Schema Integration
  - PulsarStateBackend
- Pulsar for the unified view of Data, Flink for the unified view of Computing

Done

Streaming Source -> Streaming Sink

```
PulsarSourceBuilder<String> builder = PulsarSourceBuilder.builder(new SimpleStringSchema())
    .serviceUrl(serviceUrl)
    .topic(inputTopic)
    .subscriptionName(subscription);
SourceFunction<String> src = builder.build();
DataStream<String> input = env.addSource(src);

DataStream<WordWithCount> wc = input
    .flatMap((FlatMapFunction<String, WordWithCount>) (line, collector) -> {
        for (String word : line.split("\\s")) {
            collector.collect(new WordWithCount(word, 1));
        }
    })
    .returns(WordWithCount.class)
    .keyBy("word")
    .timeWindow(Time.seconds(5))
    .reduce((ReduceFunction<WordWithCount>) (c1, c2) ->
        new WordWithCount(c1.word, c1.count + c2.count));

if (null != outputTopic) {
    wc.addSink(new FlinkPulsarProducer<>(
        serviceUrl,
        outputTopic,
        wordWithCount -> wordWithCount.toString().getBytes(UTF_8),
        wordWithCount -> wordWithCount.word
    )).setParallelism(parallelism);
} else {
    // print the results with a single thread, rather than in parallel
    wc.print().setParallelism(1);
}
```

Streaming Source -> Streaming Table Sink

```
PulsarSourceBuilder<String> builder = PulsarSourceBuilder.builder(new SimpleStringSchema())
        .serviceUrl(serviceUrl)
        .topic(inputTopic)
        .subscriptionName(subscription);
SourceFunction<String> src = builder.build();
DataStream<String> input = env.addSource(src);


DataStream<WordWithCount> wc = input
        .flatMap((FlatMapFunction<String, WordWithCount>) (line, collector) -> {
            for (String word : line.split("\\s")) {
                collector.collect(
                        WordWithCount.newBuilder().setWord(word).setCount(1).build()
                );
            }
        })
        .returns(WordWithCount.class)
        .keyBy(ROUTING_KEY)
        .timeWindow(Time.seconds(5))
        .reduce((ReduceFunction<WordWithCount>) (c1, c2) ->
                WordWithCount.newBuilder().setWord(c1.getWord()).setCount(c1.getCount() + c2.getCount()).build()
        );

tableEnvironment.registerDataStream("wc",wc);
Table table = tableEnvironment.sqlQuery("select word, `count` from wc");
table.printSchema();
TableSink sink = null;
if (null != outputTopic) {
    sink = new PulsarAvroTableSink(serviceUrl, outputTopic, ROUTING_KEY, WordWithCount.class);
} else {
    // print the results with a csv file
    sink = new CsvTableSink("./examples/file",  "|");
}
table.writeToSink(sink);
```

PULSAR

Batch Sink

```
// create PulsarOutputFormat instance
final OutputFormat pulsarOutputFormat =
        new PulsarOutputFormat(serviceUrl, topic, wordWithCount -> wordWithCount.toString().getBytes());

// create DataSet
DataSet<String> textDS = env.fromElements(EINSTEIN_QUOTE);

// convert sentences to words
textDS.flatMap(new FlatMapFunction<String, WordWithCount>() {
    @Override
    public void flatMap(String value, Collector<WordWithCount> out) throws Exception {
        String[] words = value.toLowerCase().split(" ");
        for(String word: words) {
            out.collect(new WordWithCount(word.replace(".", ""), 1));
        }
    }
})

// filter words which length is bigger than 4
.filter(wordWithCount -> wordWithCount.word.length() > 4)

// group the words
.groupBy(new KeySelector<WordWithCount, String>() {
    @Override
    public String getKey(WordWithCount wordWithCount) throws Exception {
        return wordWithCount.word;
    }
})

// sum the word counts
.reduce(new ReduceFunction<WordWithCount>() {
    @Override
    public WordWithCount reduce(WordWithCount wordWithCount1, WordWithCount wordWithCount2) throws Exception {
        return  new WordWithCount(wordWithCount1.word, wordWithCount1.count + wordWithCount2.count);
    }
})

// write batch data to Pulsar
.output(pulsarOutputFormat);
```

Case Study - Zhaopin.com

Zhaopin.com

Zhaopin.com is the biggest online recruitment service provider in China

Zhaopin.com provides job seekers a comprehensive resume service, latest employment, and career development related information, as well as in-depth online job search for positions throughout China

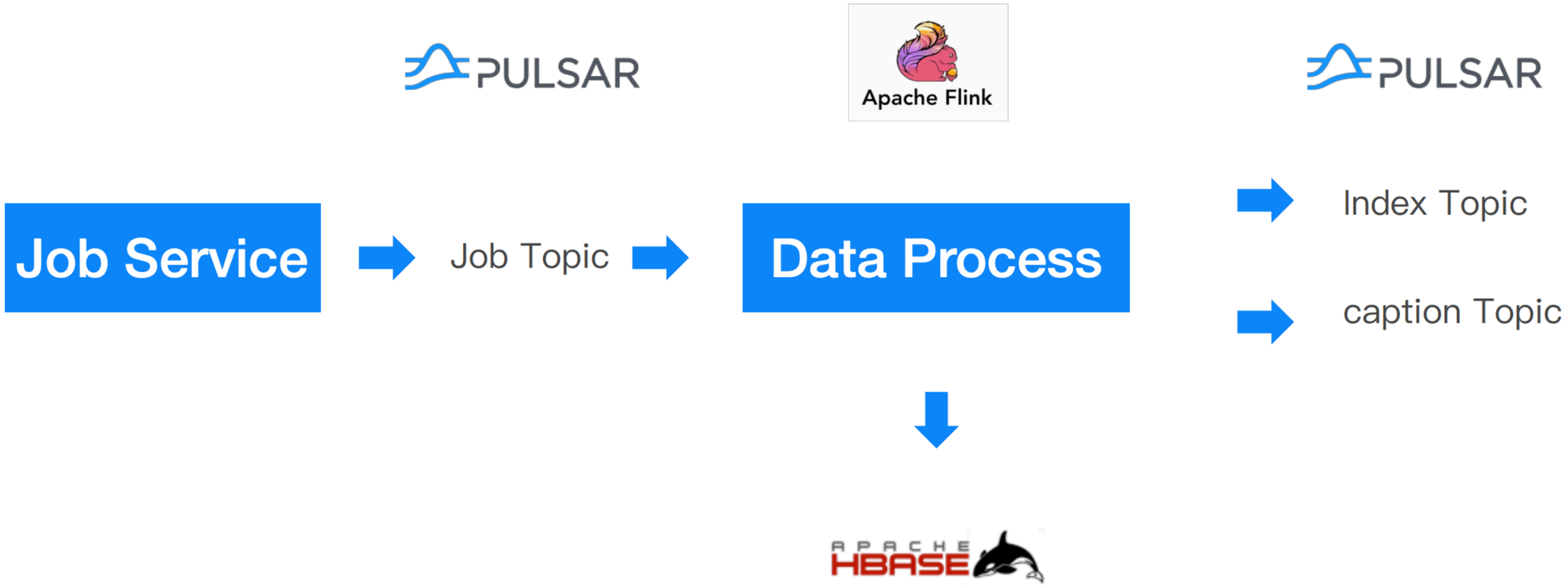
Zhaopin.com provides professional HR services to over 2.2 million clients and its average daily page views are over 68 million.

Job Search


智联招聘
zhaopin.com
职位
java
首页
北京站
校园招聘
高端职位
海外招聘
智联教育
地点:
北京
行政区
地铁沿线
设置家的地址
行业:
不限
互联网/IT
金融
房地产/建筑
商业服务
贸易/批发/零售
教育/艺术
服务业
文化/传媒/娱乐
制造业
物流运输
政府/非盈利
农林牧渔
月薪:
不限
1K以下
1K–2K
2K–4K
4K–6K
6K–8K
8K–10K
10K–15K
15K–25K
25K–35K
35K–50K
50K–70K
其他:
职位类别
工作经验
学历要求
公司性质
职位类型
智能匹配
薪酬最高
最新发布
java开发工程师（支付方向）
顶
急
10K–18K
北京
1–3年
本科
14薪
节日福利
中金支付有限公司
ZP名企
国企
100–499人
置顶
云计算/虚拟化运维工程师
顶
急
10K–20K
北京
1–3年
本科
绩效奖金
年终分红
加班补助
房补
带薪年假
新华网股份有限公司
最佳雇主
国企
1000–9999人
置顶


java
批量投递
全部行业
薪资范围
工作地点
其他要求
云计算/虚拟化运维工程师
置顶
1万-2万/月
北京 | 1-3年 | 本科
00:51
绩效奖金
年终分红
加班补助
房补
带薪年假
新华网股份有限公司
Java开发工程师(006283)
置顶
2万-4万/月
北京 | 5-10年 | 大专
12月12日
健身俱乐部
五险一金
年底双薪
餐补
带薪年假
神州优车股份有限公司
查看此公司其他2个相似职位
学会一口流利的英语，机遇总比别人多
1
想学习
要培训
到培训园
可免费试听
零基础教学
纯正外教口语
模拟场景教学
培训园-美联英语
智联教育
广告
java高级开发工程师
1万-2万/月
北京 | 3-5年 | 本科
绩效奖金
交通补助
创业公司
每年多次调薪
ERP
Java
Node.js
J2EE
数据库
Linux
北京琪佳淼网络科技有限公司

Data Processing



Metrics

50+ Namespaces

3000+ Topics

6+ billion Messages per day

3TB Storage per day

20+ Core Services

Roadmap

Batch Source

- Read Segments in Parallel
- Bypass Brokers
- Access tiered storage directly
- Scan Trimmer
  - Select Segments by Publish Time

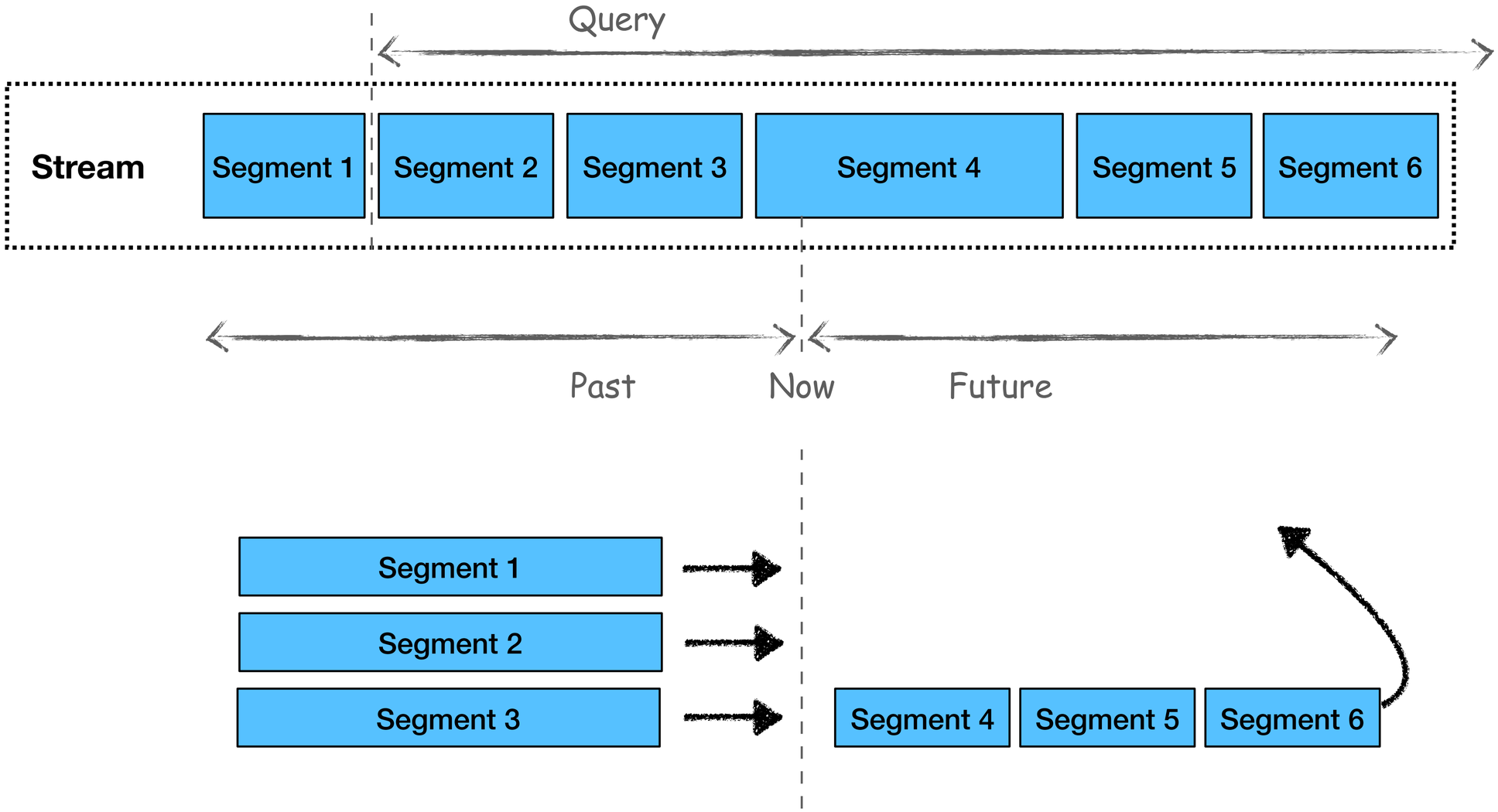
Schema Integration

- Pulsar has builtin schema registry
  - Primitive types, Avro, Json, Protobuf, …
- Schema Evolution & Multi-versioning schemas
- PulsarCatalog

State Backend

- BookKeeperStateBackend
  - Save State as Segments to BookKeeper

Unified Data Processing



Community

- ✓ Twitter: **@apache_pulsar**
- ✓ Wechat Subscription: **ApachePulsar**
- ✓ Mailing Lists

  dev@pulsar.apache.org, users@pulsar.apache.org
- ✓ Slack

  https://apache-pulsar.slack.com
- ✓ Localization

  https://crowdin.com/project/apache-pulsar
- ✓ Github

  https://github.com/apache/pulsar

  https://github.com/apache/bookkeeper