StreamNative

Why Spring Belongs In Your Data Stream (From Edge to Multi-Cloud)

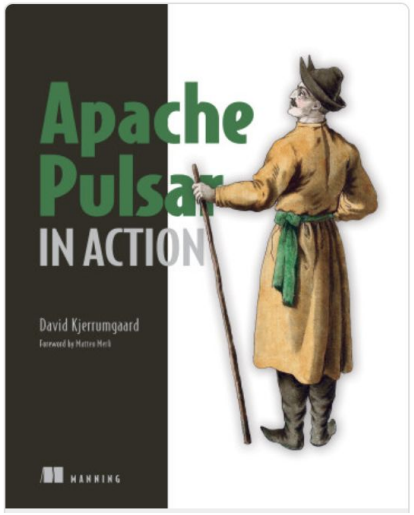
Tim Spann
Developer Advocate

David Kjerrumgaard
Developer Advocate

David Kjerrumgaard
Developer Advocate

- Apache Pulsar Committer | Author of *Pulsar In Action*
- Former Principal Software Engineer on Splunk's messaging team that is responsible for Splunk's internal Pulsar-as-a-Service platform.
- Former Director of Solution Architecture at Streamlio.

Westfield · BRISTOL WEST INSURANCE GROUP · splunk> · HORTONWORKS · FedEx · LOCKHEED MARTIN



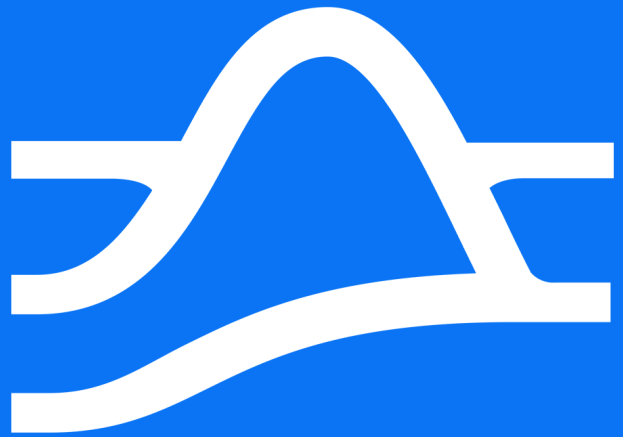
Apache Pulsar in Action

Please enjoy David's complete book which is the ultimate guide to Pulsar.

StreamNative

Agenda

- Introduction to Apache Pulsar
- Let’s Build an App!
- Demo
- Resources



Why Pulsar?

Pulsar Has a Built-in Super Set of OSS Features

Multi-Tenancy	Scalability	Geo-Replication
Unified Messaging Model	Durability	Functions

Open-Source Features → **Reduced Vendor Dependency**

Apache Pulsar Timeline

2012

CREATED

Originally developed inside Yahoo! as Cloud Messaging Service

2016

OPEN SOURCE

Pulsar committed to open source.

2018

APACHE TLP

Pulsar becomes Apache top level project.

TODAY

GROWTH

10x Contributors

10MM+ Downloads

Ecosystem Expands

Kafka on Pulsar

AMQ on Pulsar

Functions

...

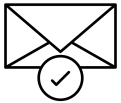
StreamNative

Key features	Apache Pulsar	Apache Kafka
Message retention (time-based)	✔	✔
Message replay	✔	✔
Message retention (acknowledge-based)	✔	✘
Built-in tiered storage	✔	✘
Processing capabilities (fully managed)	Pulsar Functions	KStream
Queue semantics (round robin)	✔	✘
Queue semantics (key based)	✔	✘
Dead letter queue	✔	✘

Key features	Apache Pulsar	Apache Kafka
Scheduled and delayed delivery	✔	✔
Rebalance-free scaling	✔	✔
Elastically scalable	✔	✘
Maximum number of topics	Millions	Up to 100k
Built-in multi-tenancy	✔	✘
Built-in geo-replication	✔	✘
Built-in schema management	✔	✘
End-to-end encryption	✔	✘

The basics

Unified Messaging Platform

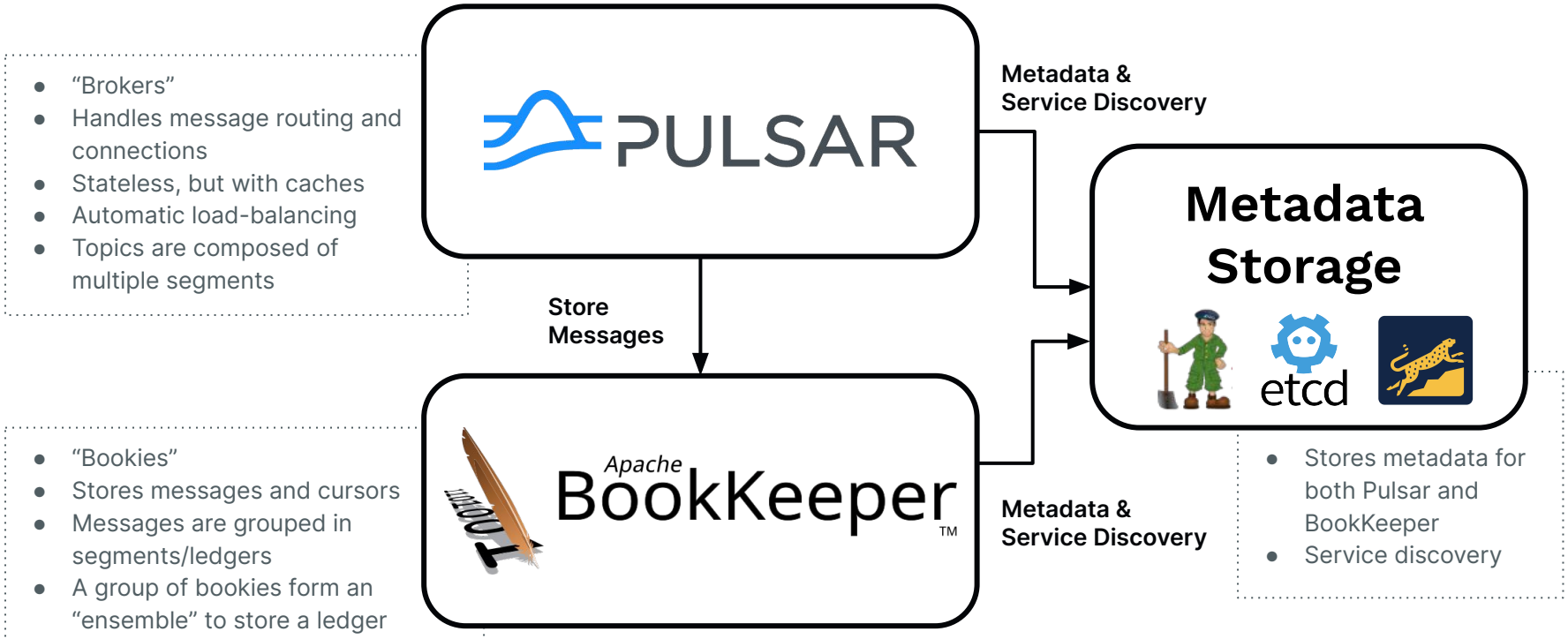


Guaranteed Message Delivery

Resiliency

Infinite Scalability

Pulsar Cluster

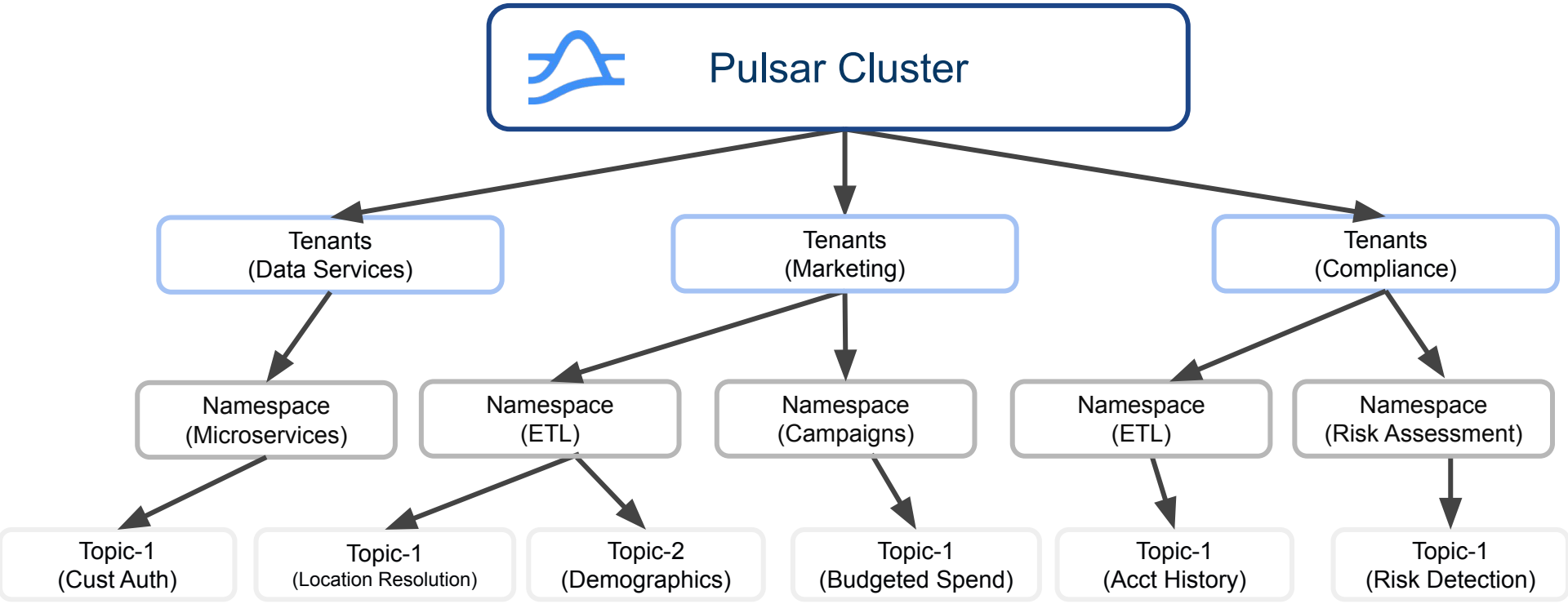


- "Brokers"
- Handles message routing and connections
- Stateless, but with caches
- Automatic load-balancing
- Topics are composed of multiple segments

- "Bookies"
- Stores messages and cursors
- Messages are grouped in segments/ledgers
- A group of bookies form an "ensemble" to store a ledger

- Stores metadata for both Pulsar and BookKeeper
- Service discovery

Tenant - Namespaces - Topics



Messages - the basic unit of Pulsar

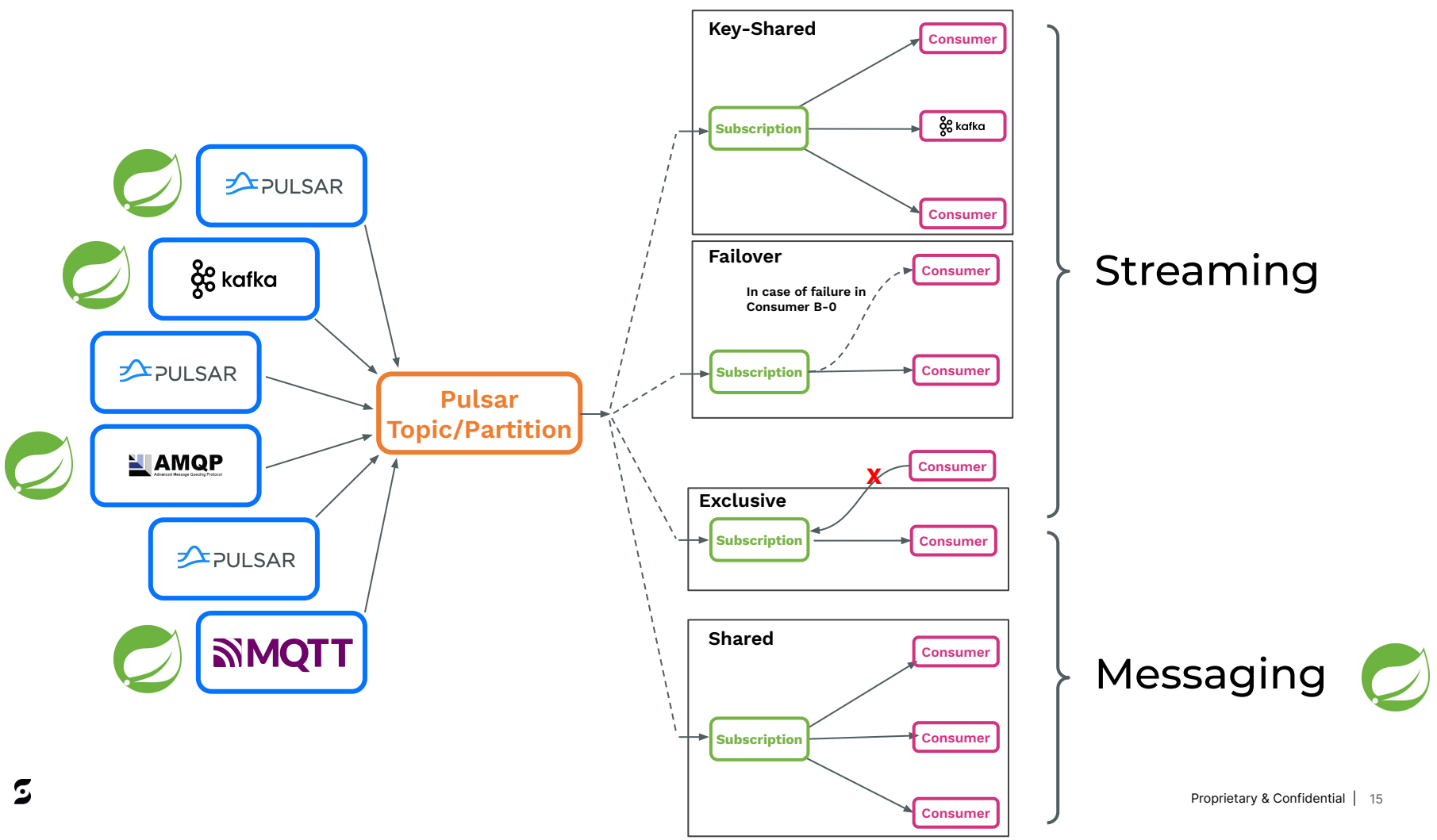
Component	Description
Value / data payload	The data carried by the message. All Pulsar messages contain raw bytes, although message data can also conform to data schemas.
Key	Messages are optionally tagged with keys, used in partitioning and also is useful for things like topic compaction.
Properties	An optional key/value map of user-defined properties.
Producer name	The name of the producer who produces the message. If you do not specify a producer name, the default name is used.
Sequence ID	Each Pulsar message belongs to an ordered sequence on its topic. The sequence ID of the message is its order in that sequence.

Producer-Consumer

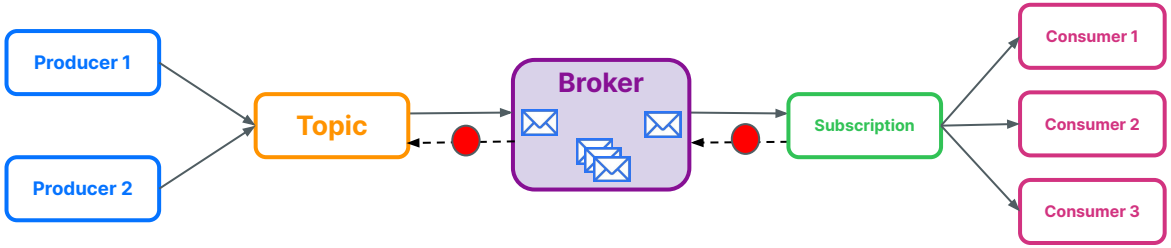
Publisher sends data and doesn't know about the subscribers or their status.

All interactions go through Pulsar and it handles all communication.

Subscriber receives data from publisher and never directly interacts with it.


PULSAR
kafka
PULSAR
AMQP
PULSAR
MQTT
Pulsar Topic/Partition
Key-Shared
Subscription
Consumer
kafka
Consumer
Failover
In case of failure in Consumer B-0
Subscription
Consumer
Consumer
Exclusive
Consumer
X
Subscription
Consumer
Shared
Subscription
Consumer
Consumer
Consumer
Streaming
Messaging

Pulsar's Publish-Subscribe model

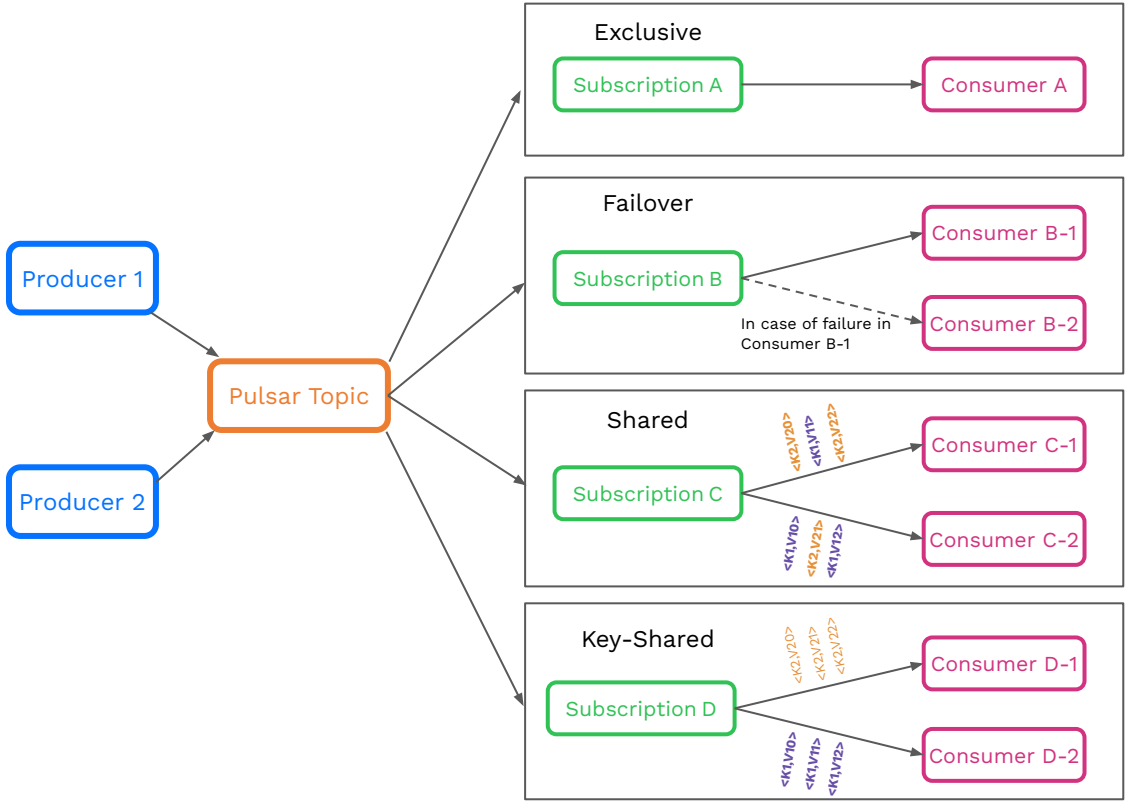


- **Producers** send messages.
- **Topics** are an ordered, named channels that producers use to transmit messages to subscribed consumers.
- **Messages** belong to a topic and contain an arbitrary payload.
- **Brokers** handle connections and routes messages between producers / consumers.
- **Subscriptions** are named configuration rules that determine how messages are delivered to consumers.
- **Consumers** receive messages.

Pulsar Subscription Modes

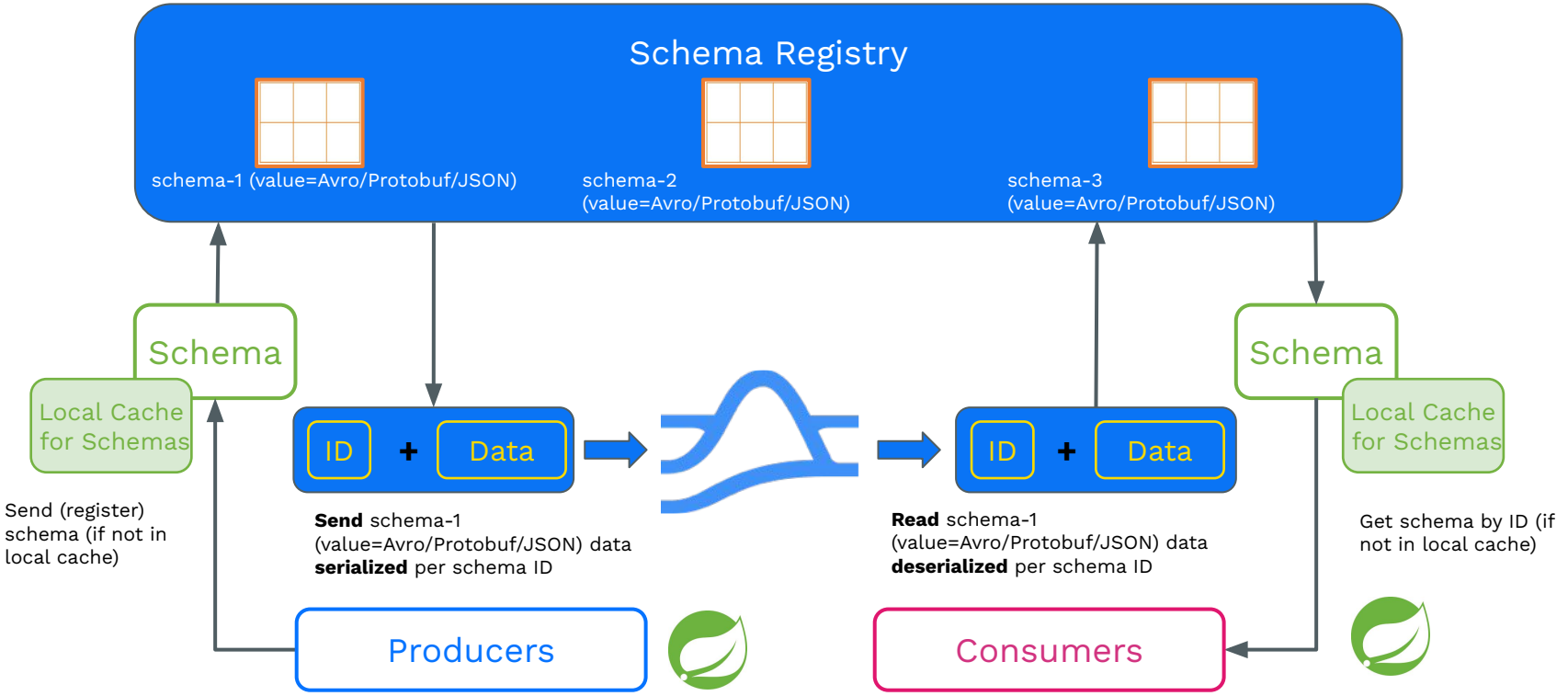
Different subscription modes have different semantics:

- **Exclusive/Failover** - guaranteed order, single active consumer
- **Shared** - multiple active consumers, no order
- **Key_Shared** - multiple active consumers, order for given key

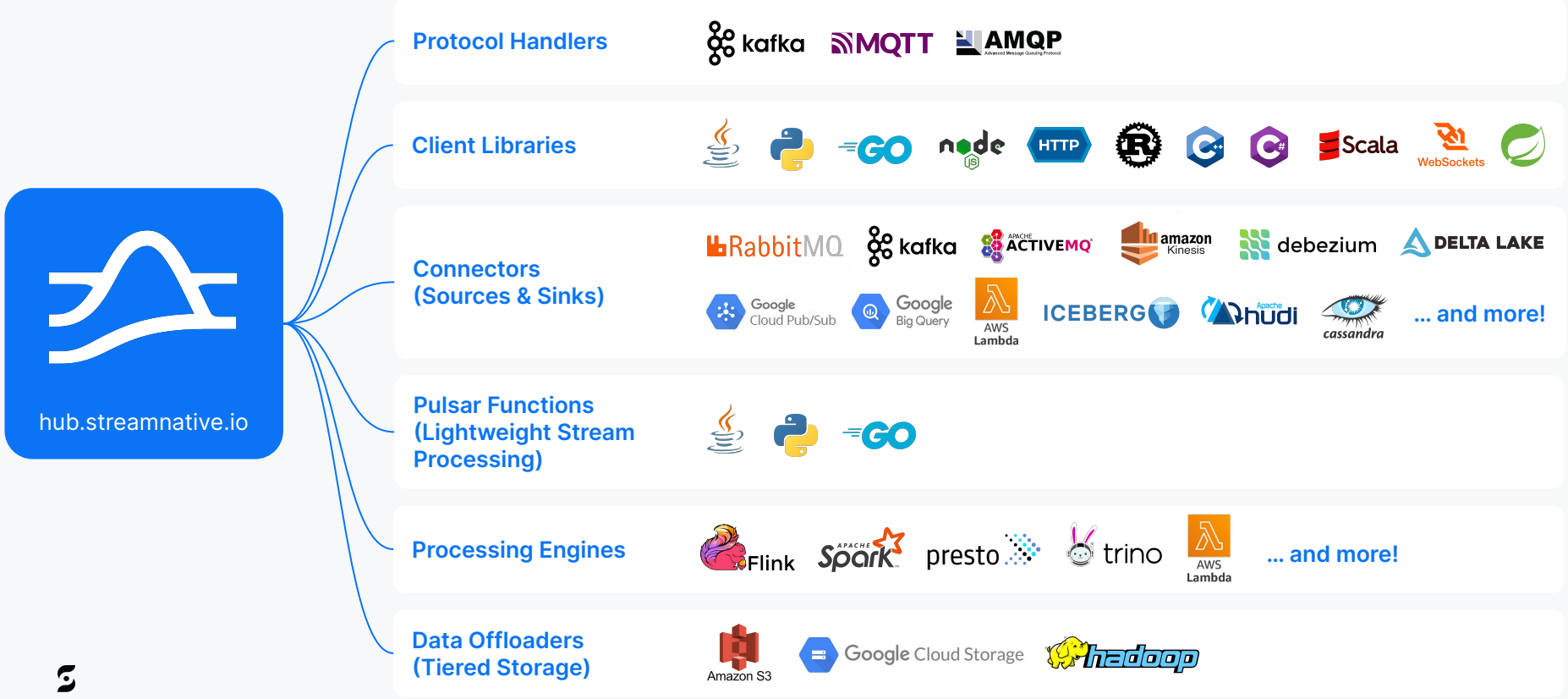



OPEN SOURCE UNIFIED MESSAGING
TAKE MY MONEY
imgflip.com

Integrated Schema Registry



StreamNative Pulsar ecosystem



The FliPN kitten crosses the stream 4 ways with Apache Pulsar

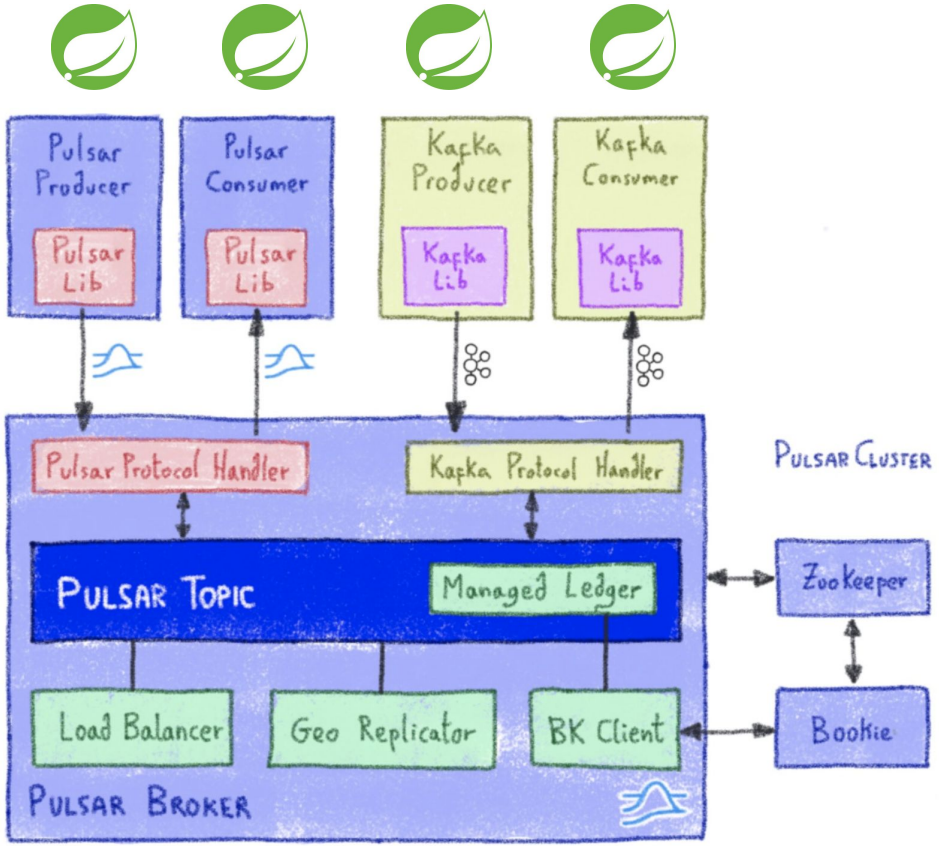
MoP

AoP

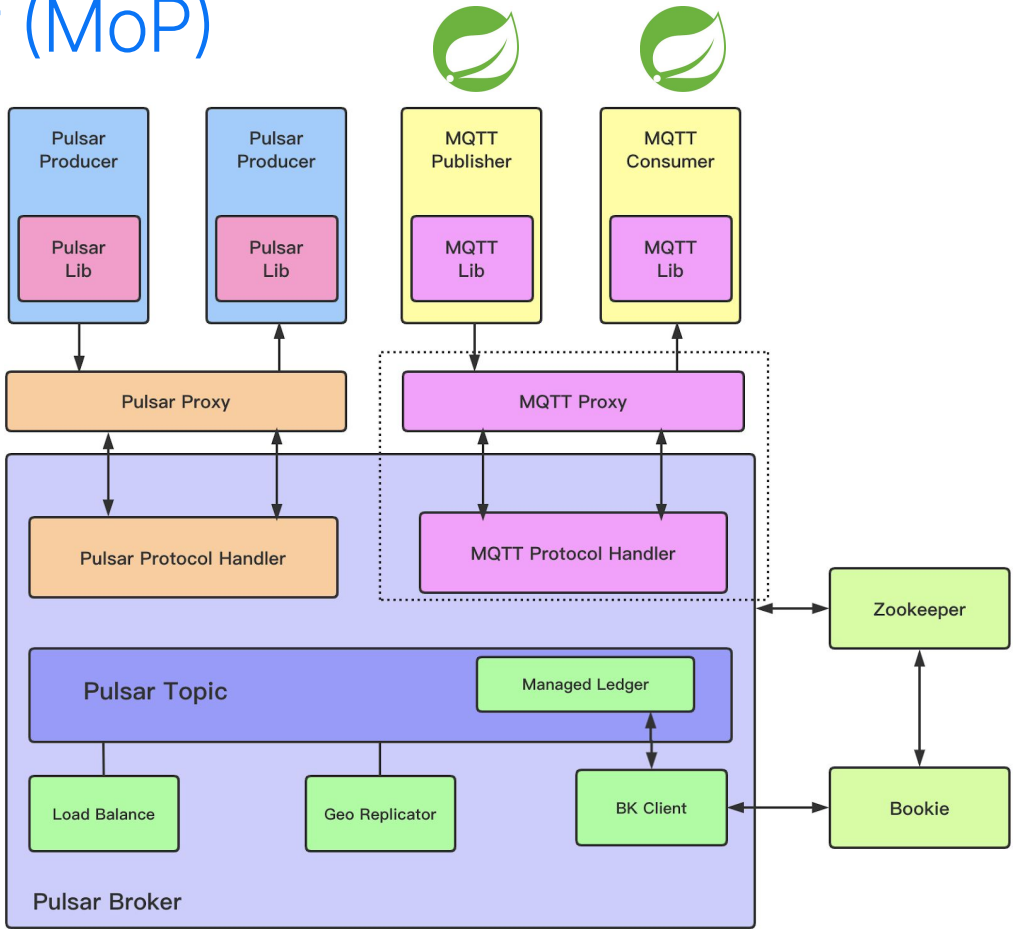
KoP

WebSockets

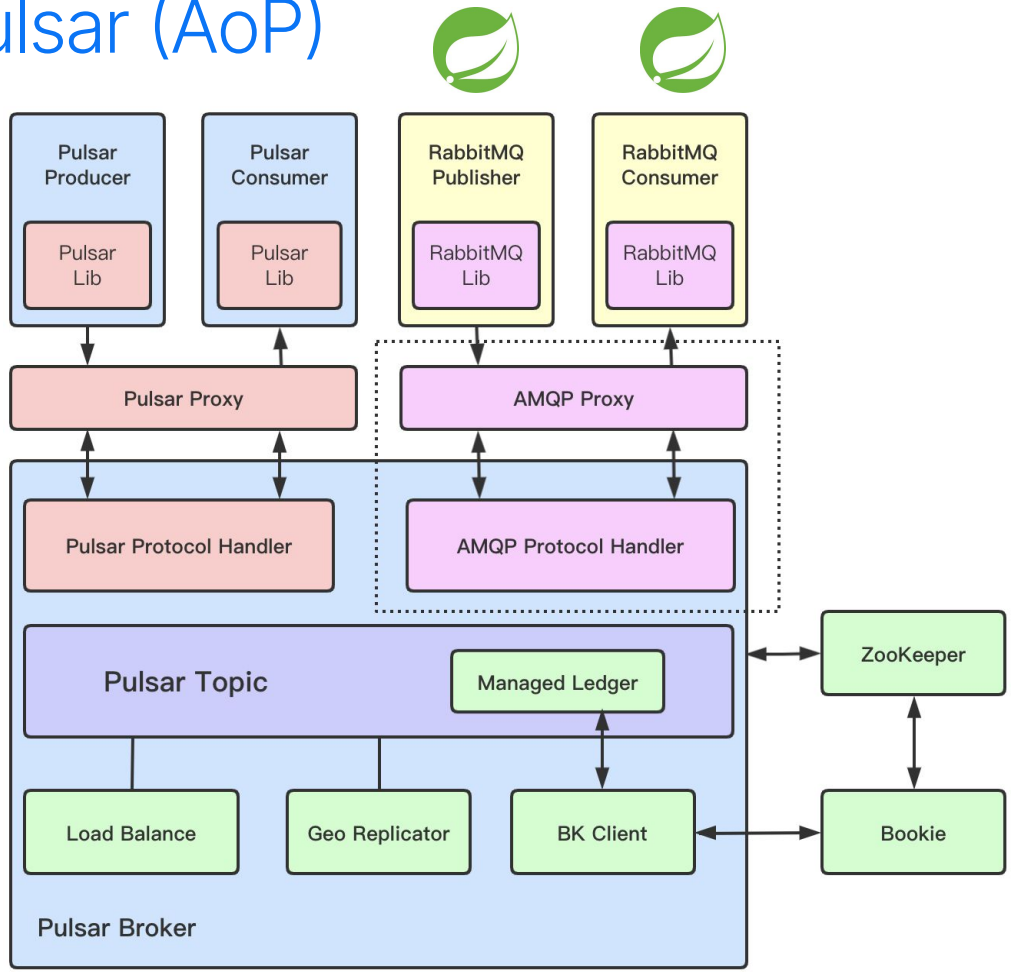
Kafka on Pulsar (KoP)



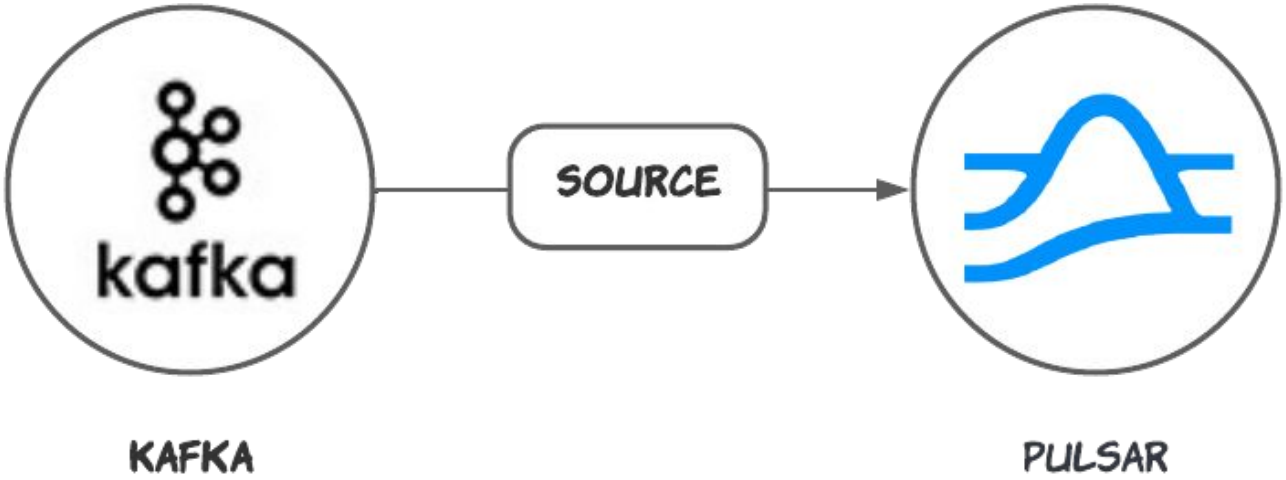
MQTT on Pulsar (MoP)



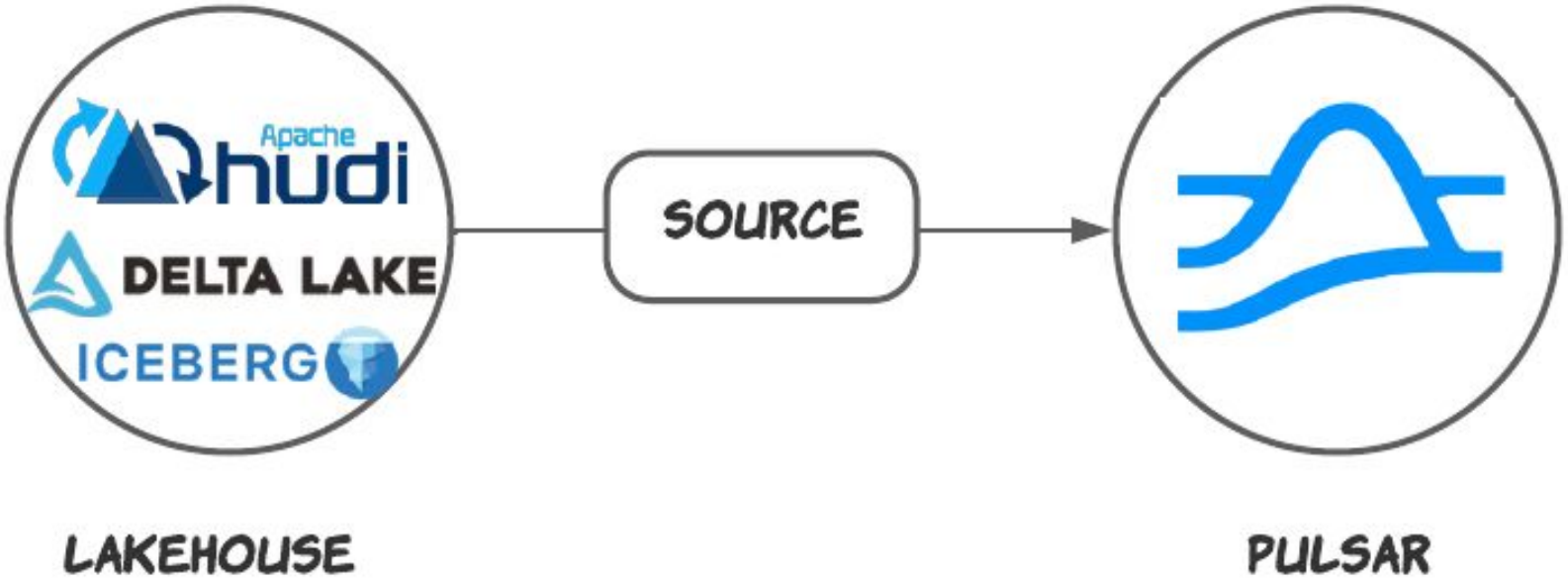
AMQP on Pulsar (AoP)

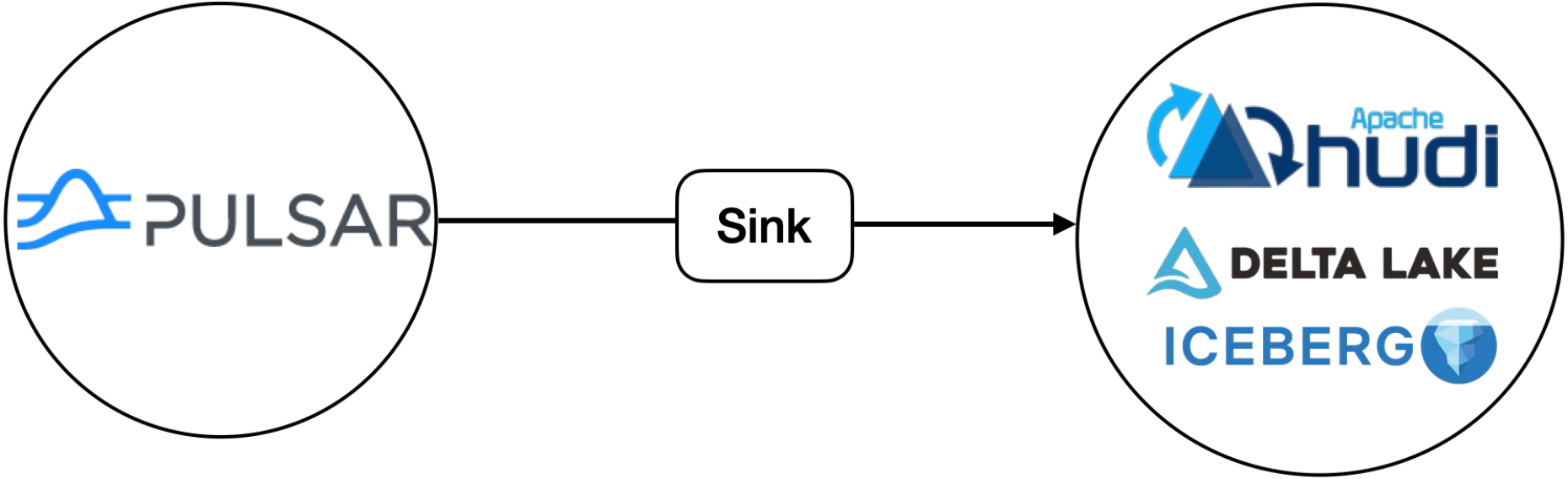


Kafka to Pulsar



Lakehouse to Pulsar





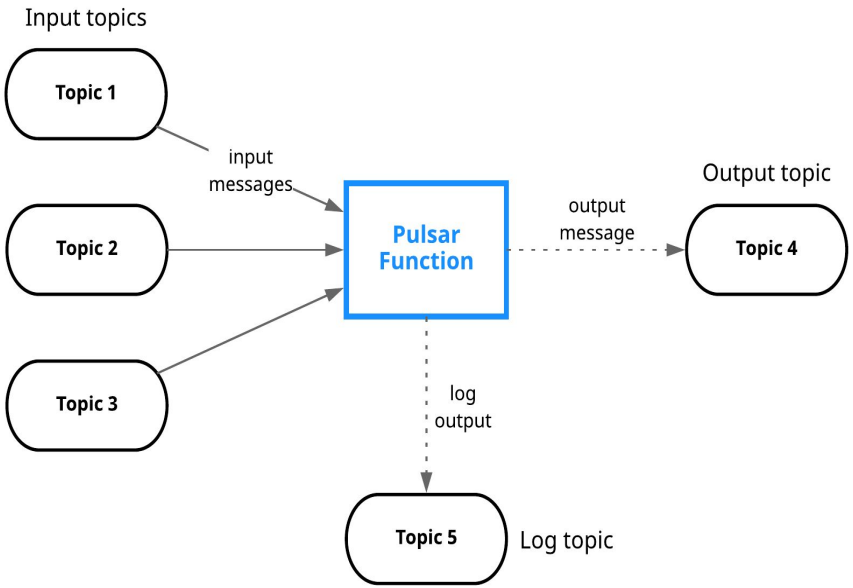
humidity	co2	datetimestamp	cputempf	ts	uuid
36.56	1127.0	2022-07-15 13:56: ...	106	1657893362	thrml_xlh_2022071 ...
36.69	1127.0	2022-07-15 13:56: ...	107	1657893367	thrml_cuv_2022071 ...

Pulsar Functions

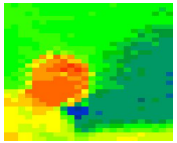
A serverless event streaming framework

- Lightweight computation similar to AWS Lambda.
- Specifically designed to use Apache Pulsar as a message bus.
- Function runtime can be located within Pulsar Broker.
- **Java** Functions

Pulsar Functions



- Consume messages from one or more Pulsar topics.
- Apply user-supplied processing logic to each message.
- Publish the results of the computation to another topic.
- Support multiple programming languages (**Java**, Python, Go)
- Can leverage 3rd-party libraries to support the **execution of ML models on the edge.**



Tim Spann

Developer Advocate at StreamNative

FLiP(N) Stack = **Flink**, **Pulsar** and **NiFi** Stack

Streaming Systems & Data Architecture Expert

Experience:

- 15+ years of experience with streaming technologies including Pulsar, Kafka, Flink, Spark, NiFi, Spring, Big Data, Cloud, ML, IoT, Python and more.
- Today, he helps to grow the Pulsar community sharing rich technical knowledge and experience at both global conferences and through individual conversations.

CLOUDERA

Pivotal

HORTONWORKS

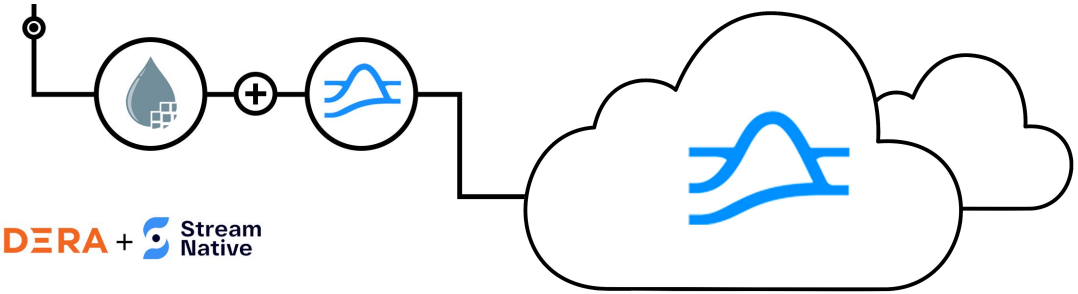
EY

Hewlett Packard Enterprise

Building Real-Time Requires a Team

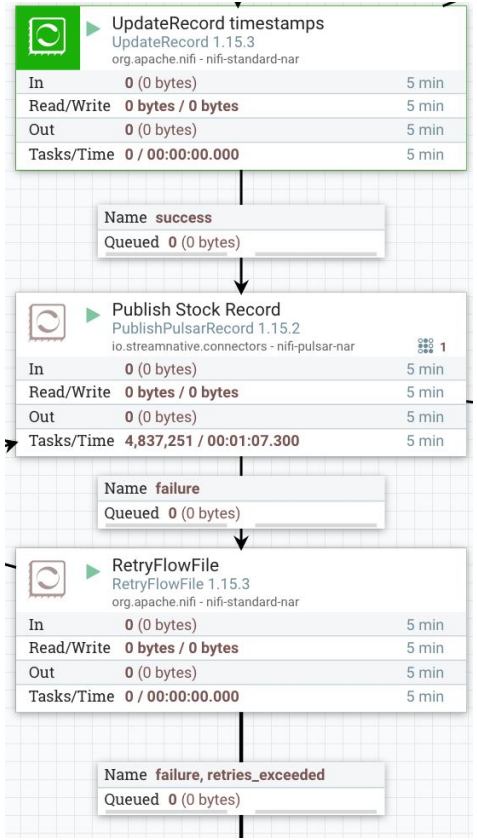
NiFi-Pulsar Two-Way Connector

Apache NiFi - Apache Pulsar Connector



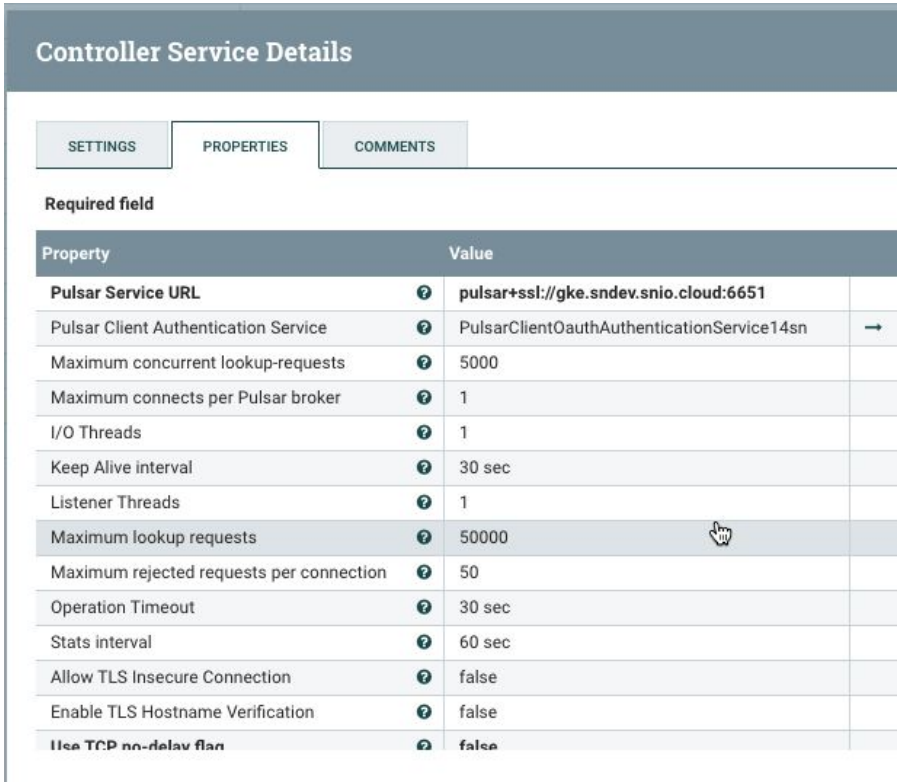
CLOUDERA + Stream Native

Announcing the Integration of
Apache NiFi and Apache Pulsar



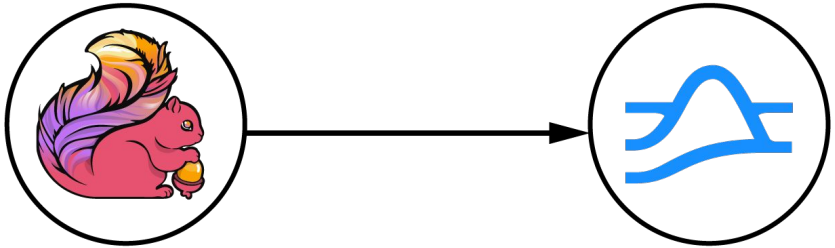
https://github.com/streamnative/pulsar-nifi-bundle

Apache NiFi - Apache Pulsar Connector



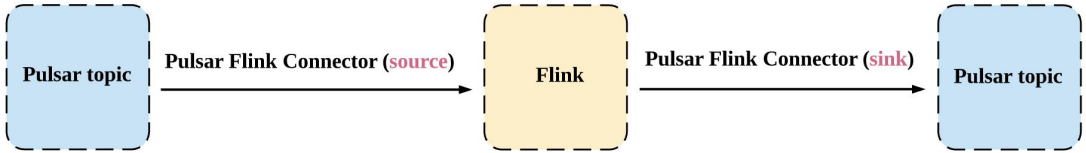
Apache NiFi - Apache Pulsar Connector

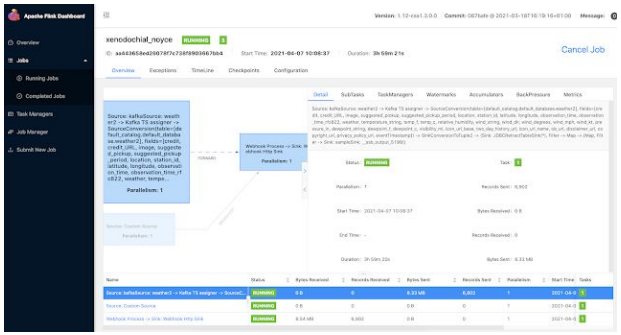
https://github.com/david-streamlio/pulsar-nifi-bundle/releases/tag/v1.14.0



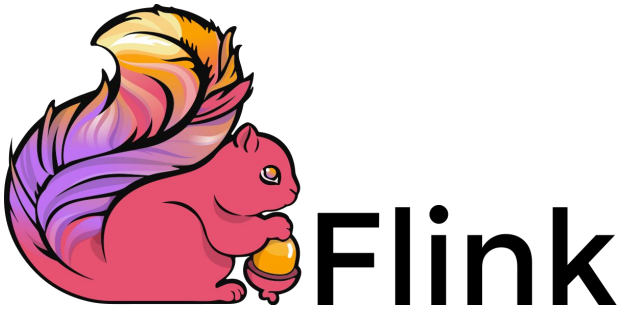
Flink-Pulsar Sink Connector

Apache Flink





- Unified computing engine
- Batch processing is a special case of stream processing
- Stateful processing
- Massive Scalability
- Flink SQL for queries, inserts against Pulsar Topics
- Streaming Analytics
- Continuous SQL
- Continuous ETL
- Complex Event Processing
- Standard SQL Powered by Apache Calcite



StreamNative

Flink SQL


```
select aqi, parameterName, dateObserved, hourObserved, latitude,
longitude, localTimeZone, stateCode, reportingArea from
airquality;

select max(aqi) as MaxAQI, parameterName, reportingArea from
airquality group by parameterName, reportingArea;

select max(aqi) as MaxAQI, min(aqi) as MinAQI, avg(aqi) as
AvgAQI, count(aqi) as RowCount, parameterName, reportingArea
from airquality group by parameterName, reportingArea;
```

Flink SQL

SQL Query Result (Table)

Refresh: 1 s Page: Last of 1 Updated: 15:52:48.189

uuid	ipaddress	cputempf	runtime
snr_20220323195238	192.168.1.229	99	453
snr_20220323195243	192.168.1.229	100	458

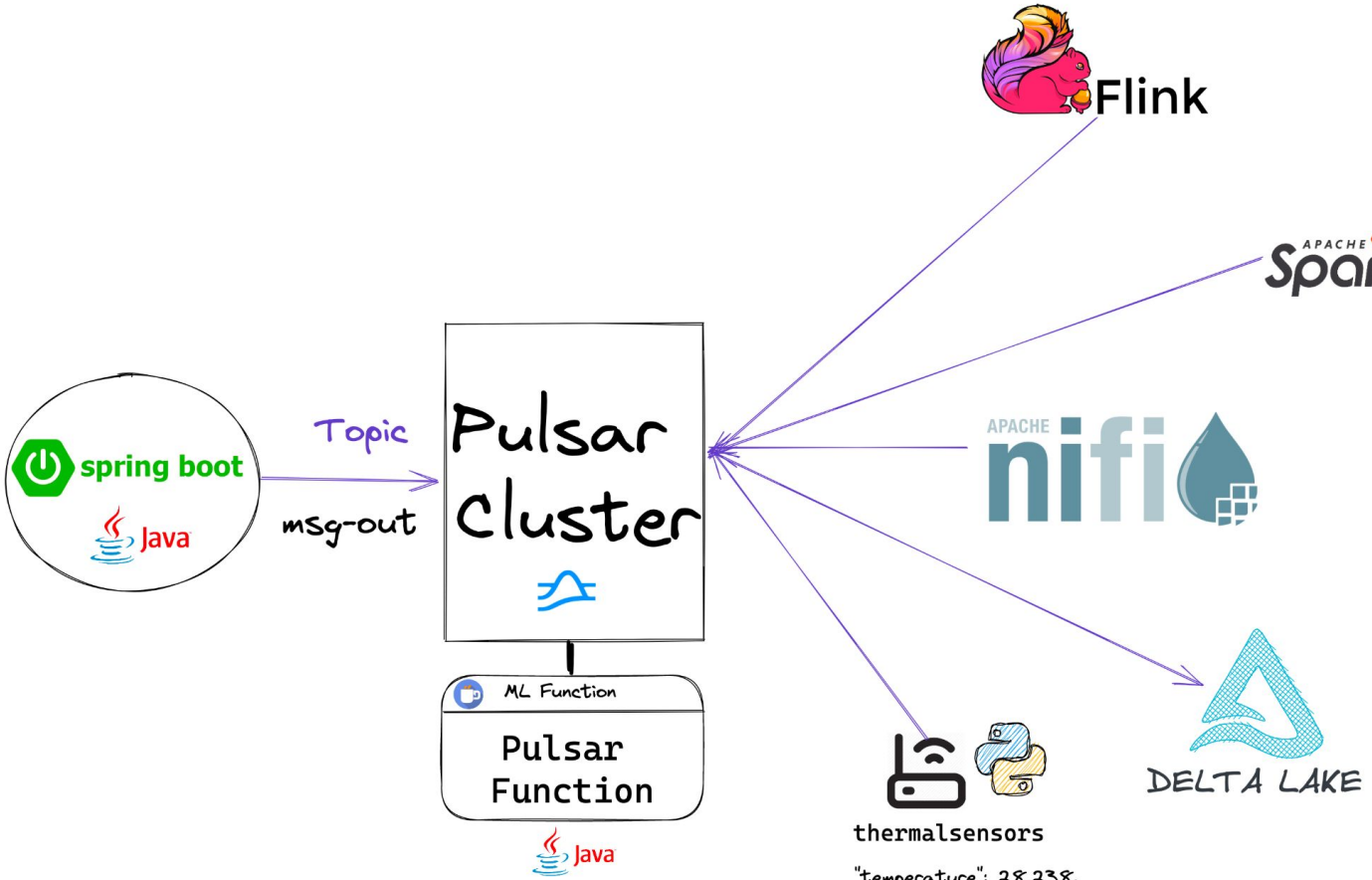
Q Quit
R Refresh
+ Inc Refresh
_ Dec Refresh
G Goto Page
L Last Page
N Next Page
P Prev Page
O Open Row

StreamNative

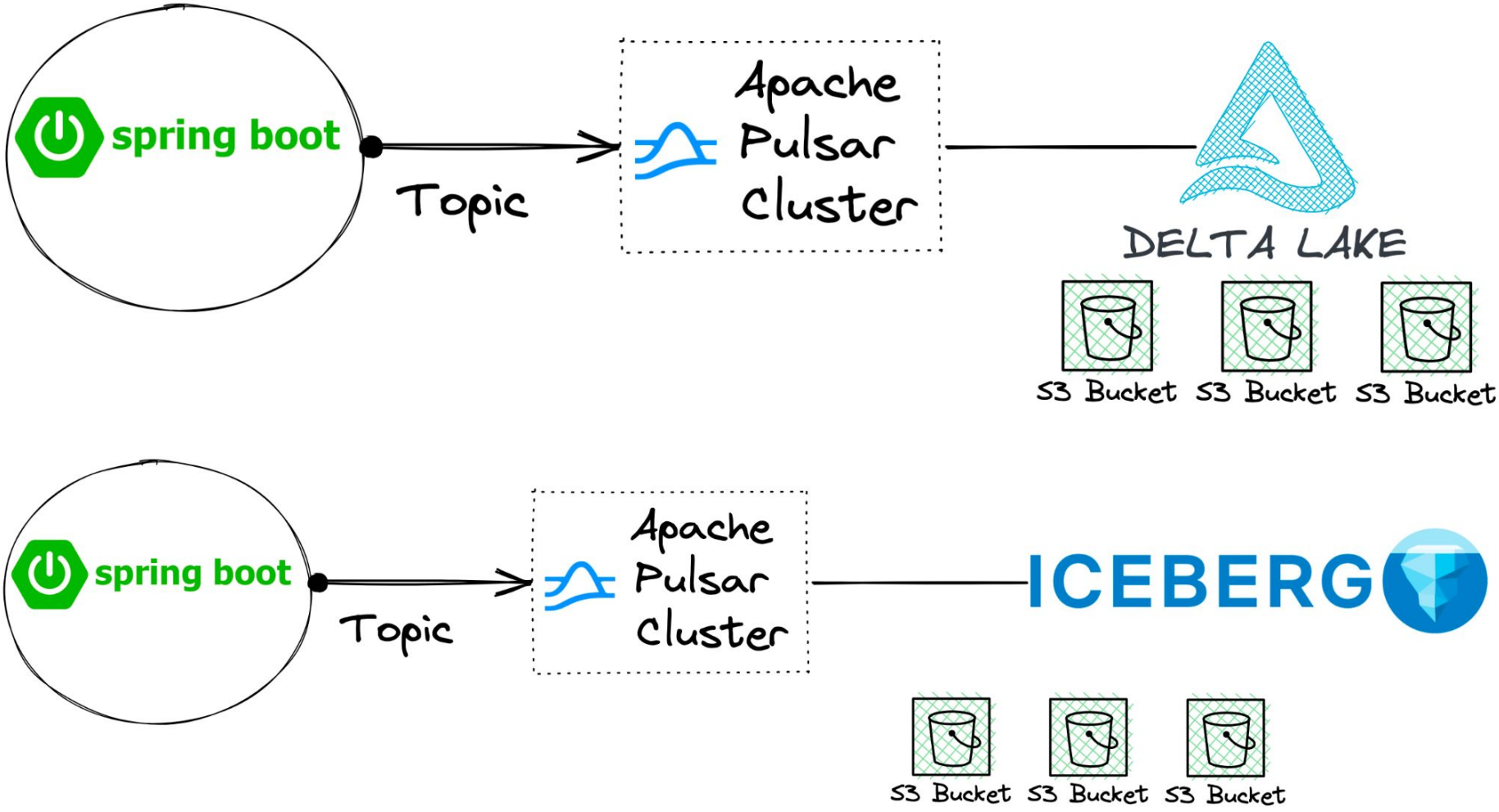
streamnative.io

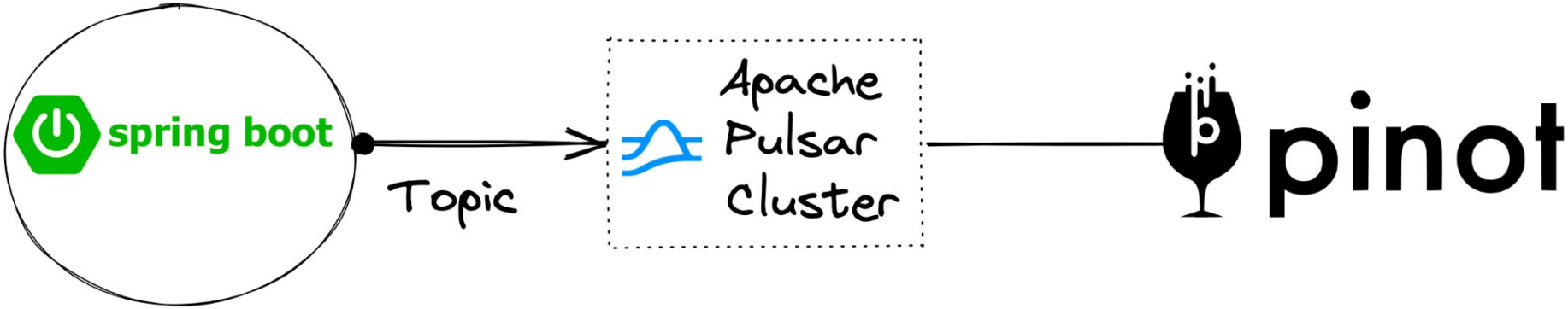
Demo

StreamNative

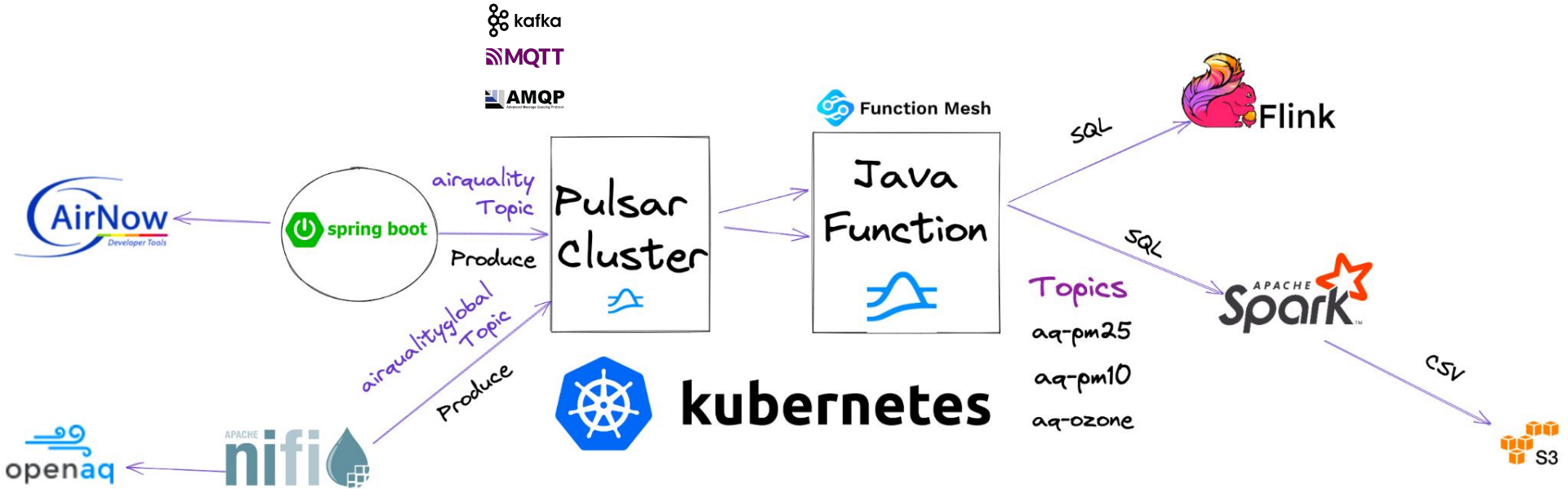


StreamNative


spring boot
Topic
Apache
Pulsar
Cluster
DELTA LAKE
S3 Bucket
S3 Bucket
S3 Bucket
spring boot
Topic
Apache
Pulsar
Cluster
ICEBERG
S3 Bucket
S3 Bucket
S3 Bucket


spring boot
Topic
Apache
Pulsar
Cluster
pinot

REST + Spring Boot + Pulsar + Friends



StreamNative

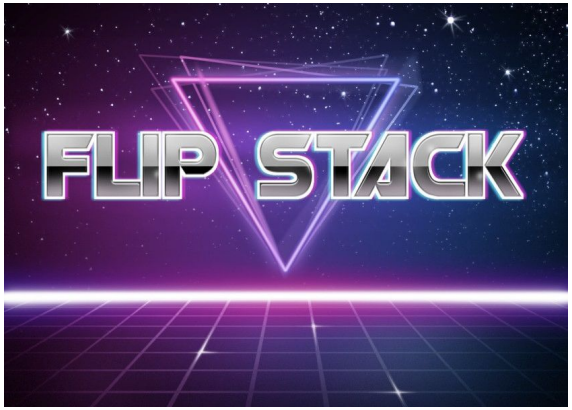
Introduction to Spring with Apache Pulsar

https://streamnative.io/blog/engineering/2022-05-26-spring-into-pulsar/

Spring into Pulsar Part 2

Spring into Pulsar Part 2

Spring-based Microservices for Multiple Protocols with Apache Pulsar



FLiP Stack Weekly

This week in Apache Flink, Apache Pulsar, Apache NiFi, Apache Spark and open source friends.

https://bit.ly/32dAJft

Tim Spann

Developer Advocate
at StreamNative

https://streamnative.io/pulsar-python/

https://www.linkedin.com/in/timothyspann

@PaaSDev

https://github.com/tspannhw

StreamNative