FLiP Stack for Cloud Data Lakes

Tim Spann | Developer Advocate

- Introduction
- FLiP Stack
- Details
- Demo
- Q&A
- 50 Minutes

Tim Spann, Developer Advocate at StreamNative

Tim Spann
Developer Advocate

- FLiP(N) Stack = Flink, Pulsar and NiFI Stack
- Streaming Systems & Data Architecture Expert
- Experience:
  - 15+ years of experience with streaming technologies including Pulsar, Flink, Spark, NiFi, Big Data, Cloud, MXNet, IoT 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 BARNES & NOBLE FLIP STACK EY HORTONWORKS airis.DATA Hewlett Packard Enterprise

StreamNative

FLiP Stack Weekly

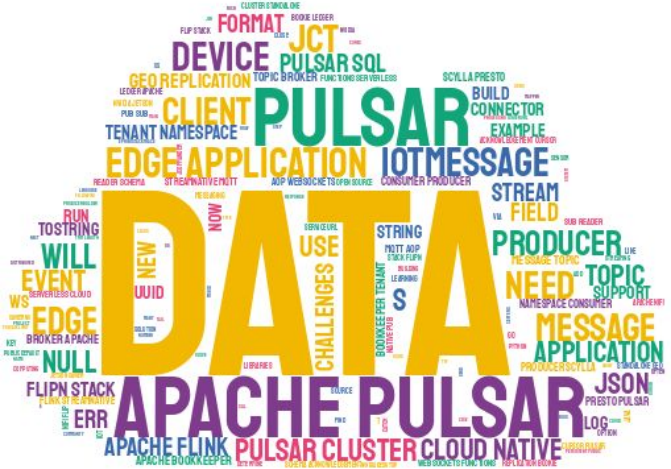
This week in Apache Flink, Apache Pulsar, Apache NiFi, Apache Spark and open source friends.

https://bit.ly/32dAJft

StreamNative

FLiP(N) Stack

- Apache **Fl**ink
- Apache **P**ulsar
- Pulsar Functions
- StreamNative's Flink Connector for Puls
- Apache **N**iFi
- Python, Java, Golang



StreamNative

Apache Pulsar

StreamNative

Apache Pulsar

Serverless computing framework.

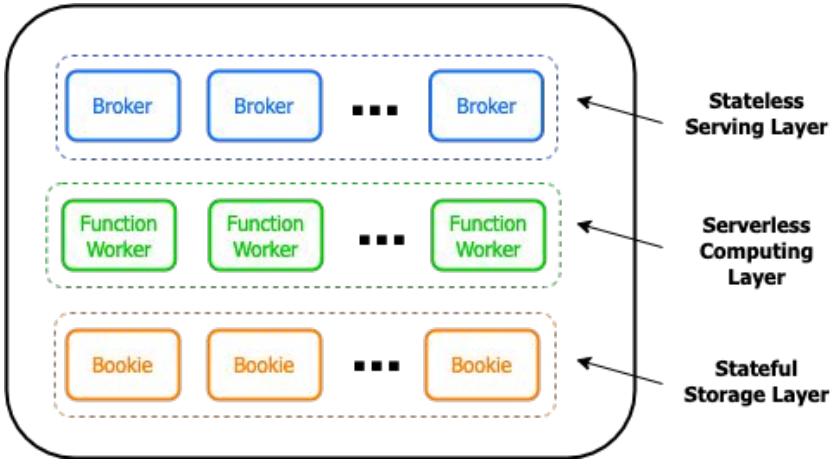
Unbounded storage, multi-tiered architecture, and tiered-storage.

Streaming & Pub/Sub messaging semantics.

Multi-protocol support.

Open Source

Cloud-Native

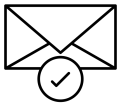


StreamNative

Why Apache Pulsar?

Unified

Messaging Platform



Guaranteed

Message Delivery

Resiliency

Infinite

Scalability

StreamNative

Pulsar Benefits

Unified Messaging Model

Simplify your data infrastructure and enable new use cases with queuing and streaming capabilities in one platform.

Multi-tenancy

Enable multiple user groups to share the same cluster, either via access control, or in entirely different namespaces.

Scalability

Decoupled data computing and storage enable horizontal scaling to handle data scale and management complexity.

Geo-replication

Support for multi-datacenter replication with both asynchronous and synchronous replication for built-in disaster recovery.

Tiered storage

Enable historical data to be offloaded to cloud-native storage and store event streams for indefinite periods of time.

Pulsar: Unified Messaging + Data Streaming

Messaging

Ideal for work queues that do not require tasks to be performed in a particular order—for example, sending one email message to many recipients.

RabbitMQ and Amazon SQS are examples of popular queue-based message systems.

Pulsar: Unified Messaging + Data Streaming

Messaging

Ideal for work queues that do not require tasks to be performed in a particular order—for example, sending one email message to many recipients.

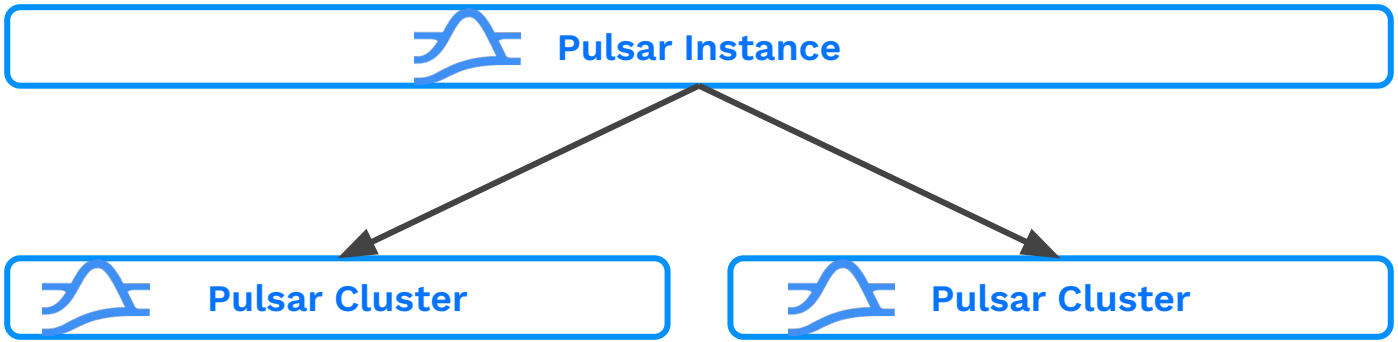
RabbitMQ and Amazon SQS are examples of popular queue-based message systems.

.. and Streaming

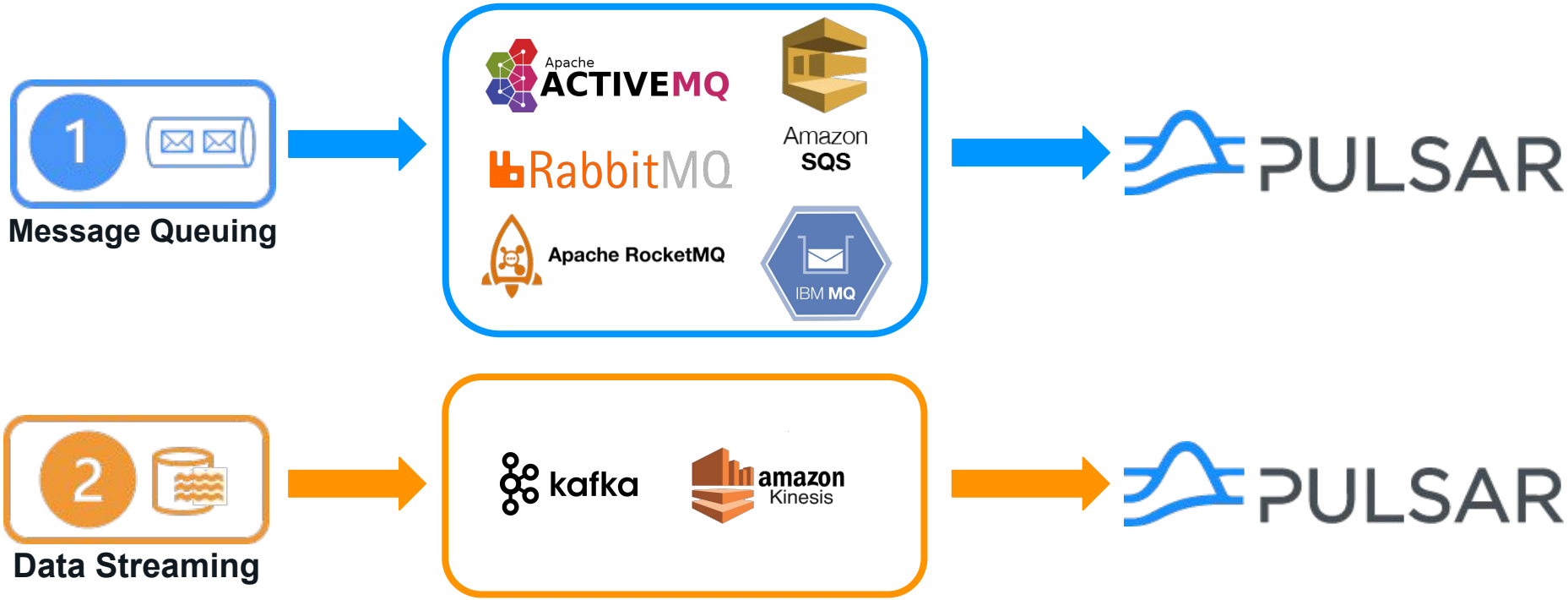
Works best in situations where the order of messages is important—for example, data ingestion.

Kafka and Amazon Kinesis are examples of messaging systems that use streaming semantics for consuming messages.

Pulsar Instance

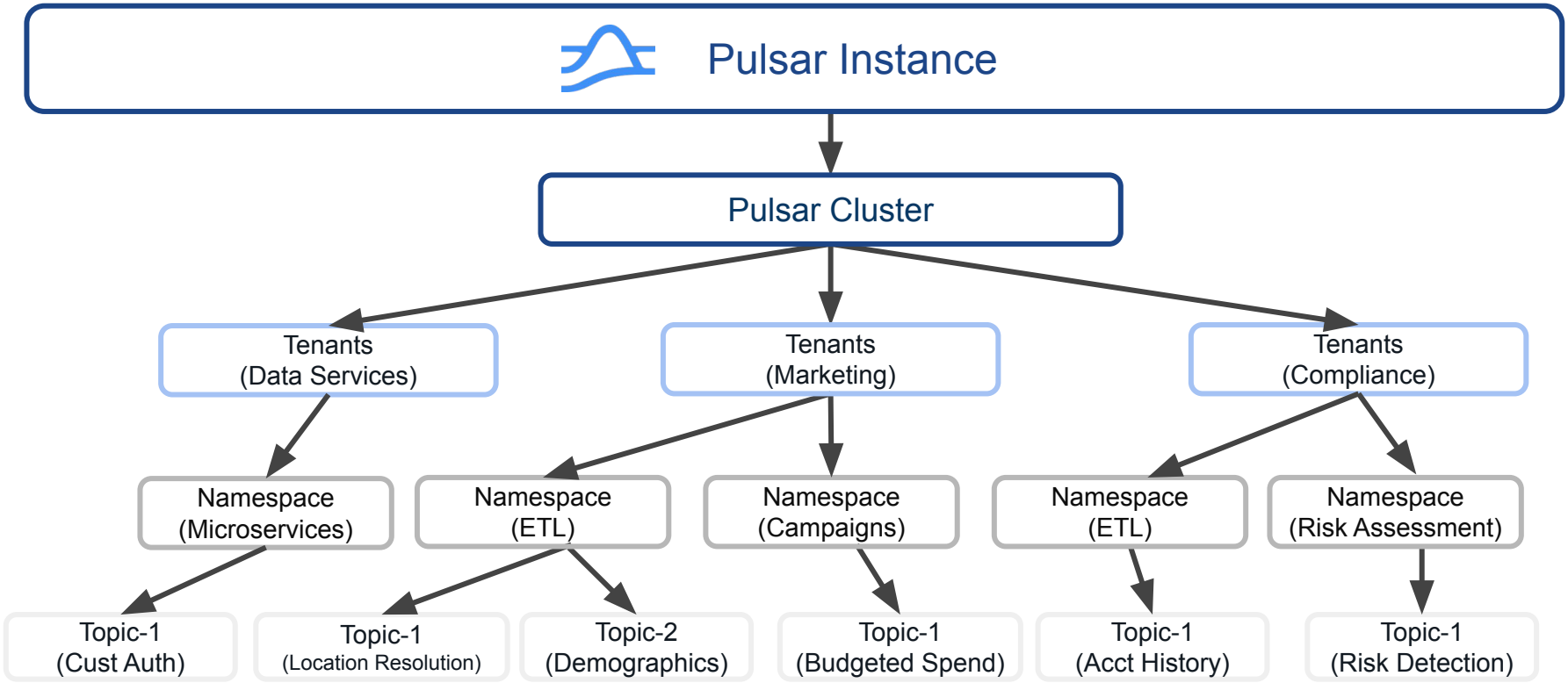


A Unified Messaging Platform

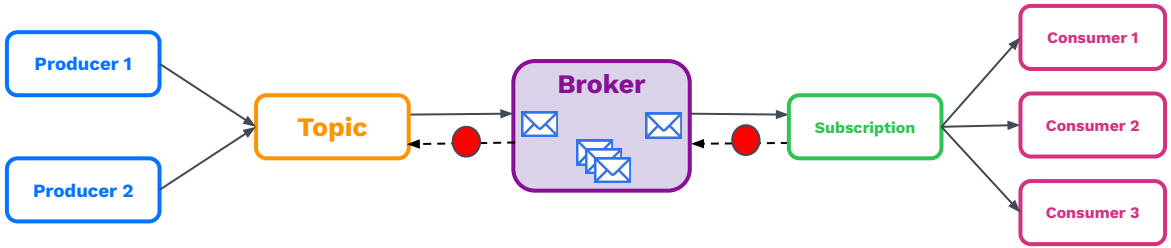


StreamNative

Topics


Pulsar Instance
Pulsar Cluster
Tenants
(Data Services)
Tenants
(Marketing)
Tenants
(Compliance)
Namespace
(Microservices)
Namespace
(ETL)
Namespace
(Campaigns)
Namespace
(ETL)
Namespace
(Risk Assessment)
Topic-1
(Cust Auth)
Topic-1
(Location Resolution)
Topic-2
(Demographics)
Topic-1
(Budgeted Spend)
Topic-1
(Acct History)
Topic-1
(Risk Detection)

Pulsar's Publish-Subscribe model



- **Producers** send messages.
- **Topics** are an ordered, named channel 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.

StreamNative

Messaging Ordering Guarantees

Topic Ordering Guarantees:

- Messages sent to a single topic or partition DO have an ordering guarantee.
- Messages sent to different partitions DO NOT have an ordering guarantee.

Subscription Mode Guarantees:

- A single consumer can receive messages from the same partition *in order* using an exclusive or failover subscription mode.
- Multiple consumers can receive messages from the same key *in order* using the key_shared subscription mode.

StreamNative

Messaging
Ordering Guarantees

Topic Ordering Guarantees:

- Messages sent to a single topic or partition DO have an ordering guarantee.
- Messages sent to different partitions DO NOT have an ordering guarantee.

Subscription Mode Guarantees:

- A single consumer can receive messages from the same partition *in order* using an exclusive or failover subscription mode.
- Multiple consumers can receive messages from the same key *in order* using the key_shared subscription mode.

StreamNative

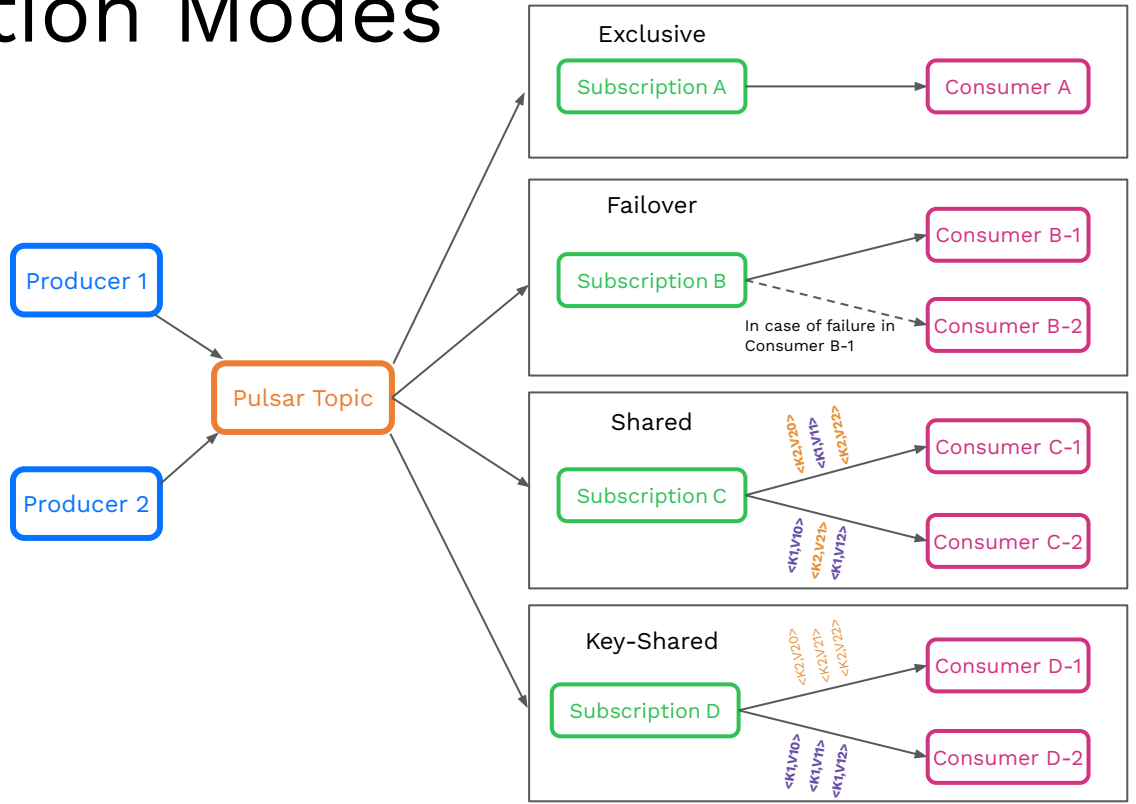
Pulsar Subscription Modes

Different subscription modes have different semantics:

Exclusive/Failover - guaranteed order, single active consumer

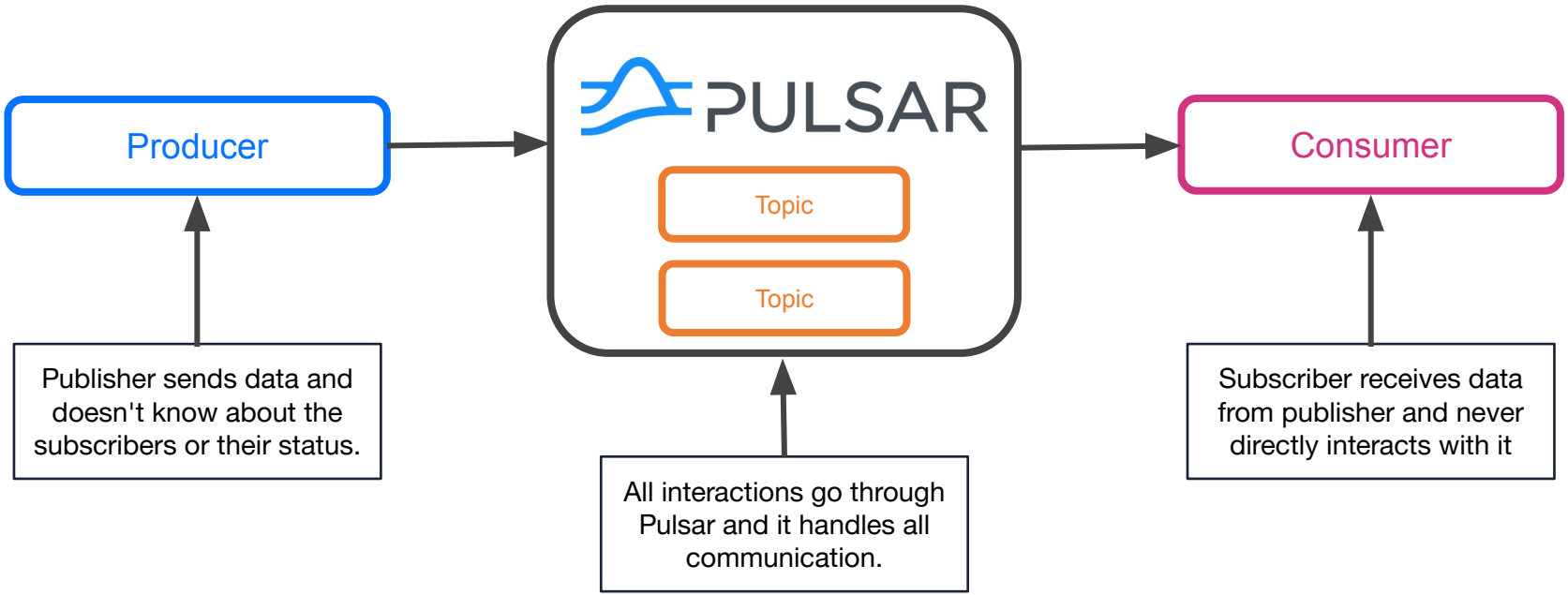
Shared - multiple active consumers, no order

Key_Shared - multiple active consumers, order for given key



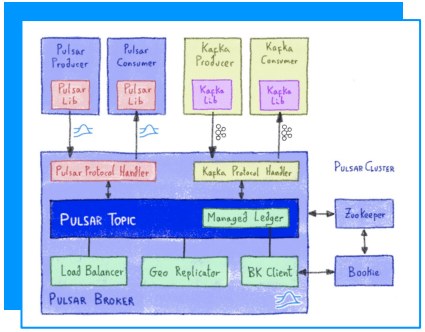
StreamNative

Producer-Consumer



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. Message De-Duplication.
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. Message De-Duplication.



Connectivity

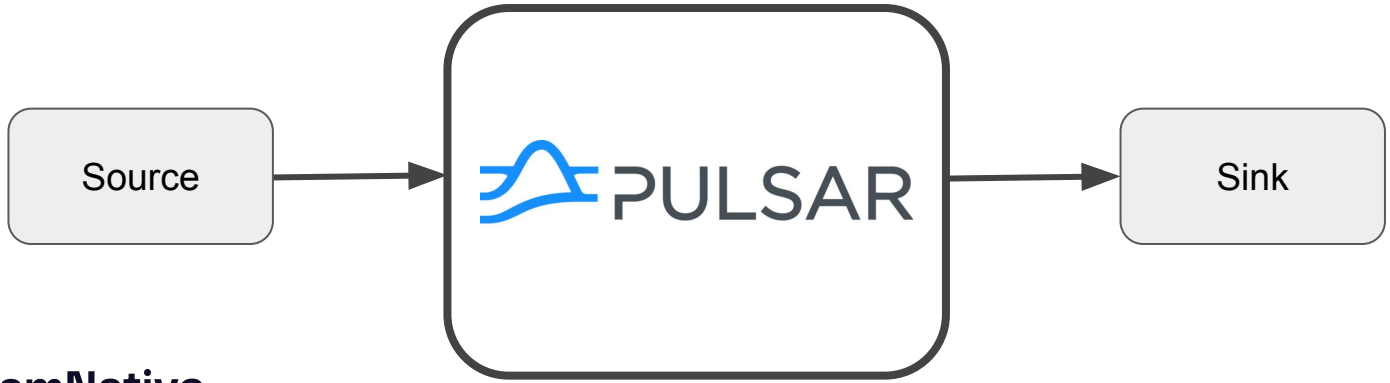
hub.streamnative.io

- **Libraries -** (Java, Python, Go, NodeJS, WebSockets, C++, C#, Scala, Rust,...)
- **Functions** - Lightweight Stream Processing (Java, Python, Go)
- **Connectors** - Sources & Sinks (Cassandra, Kafka, ...)
- **Protocol Handlers** - AoP (AMQP), KoP (Kafka), MoP (MQTT)
- **Processing Engines** - Flink, Spark, Presto/Trino via Pulsar SQL
- **Data Offloaders** - Tiered Storage - (S3)

Moving Data In and Out of Pulsar

IO/Connectors are a simple way to integrate with external systems and move data in and out of Pulsar. https://pulsar.apache.org/docs/en/io-jdbc-sink/

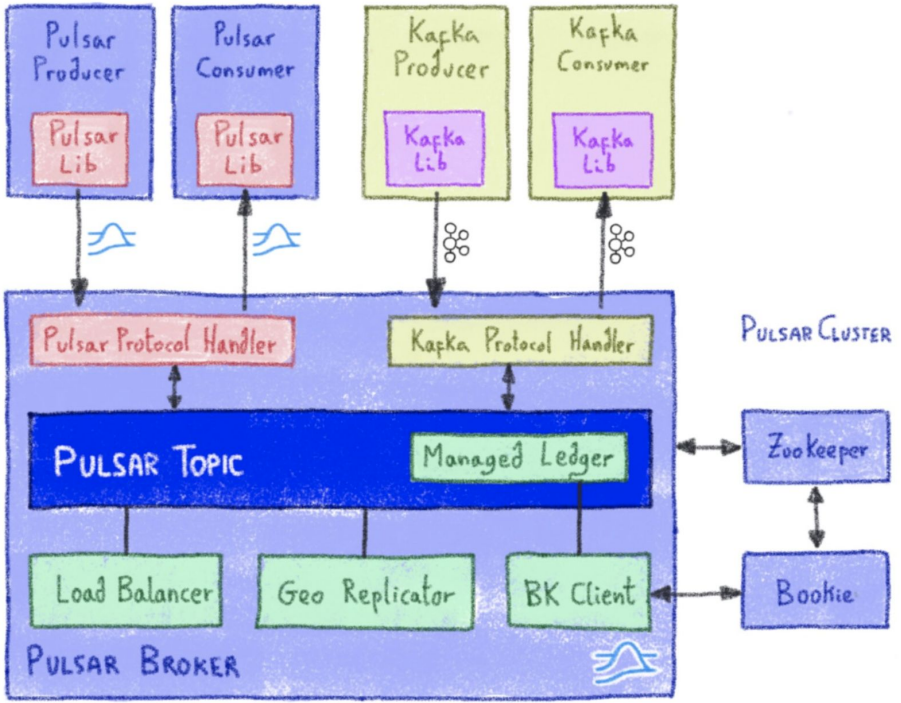
- Built on top of Pulsar Functions
- Built-in connectors - hub.streamnative.io



Kafka

On Pulsar

(KoP)

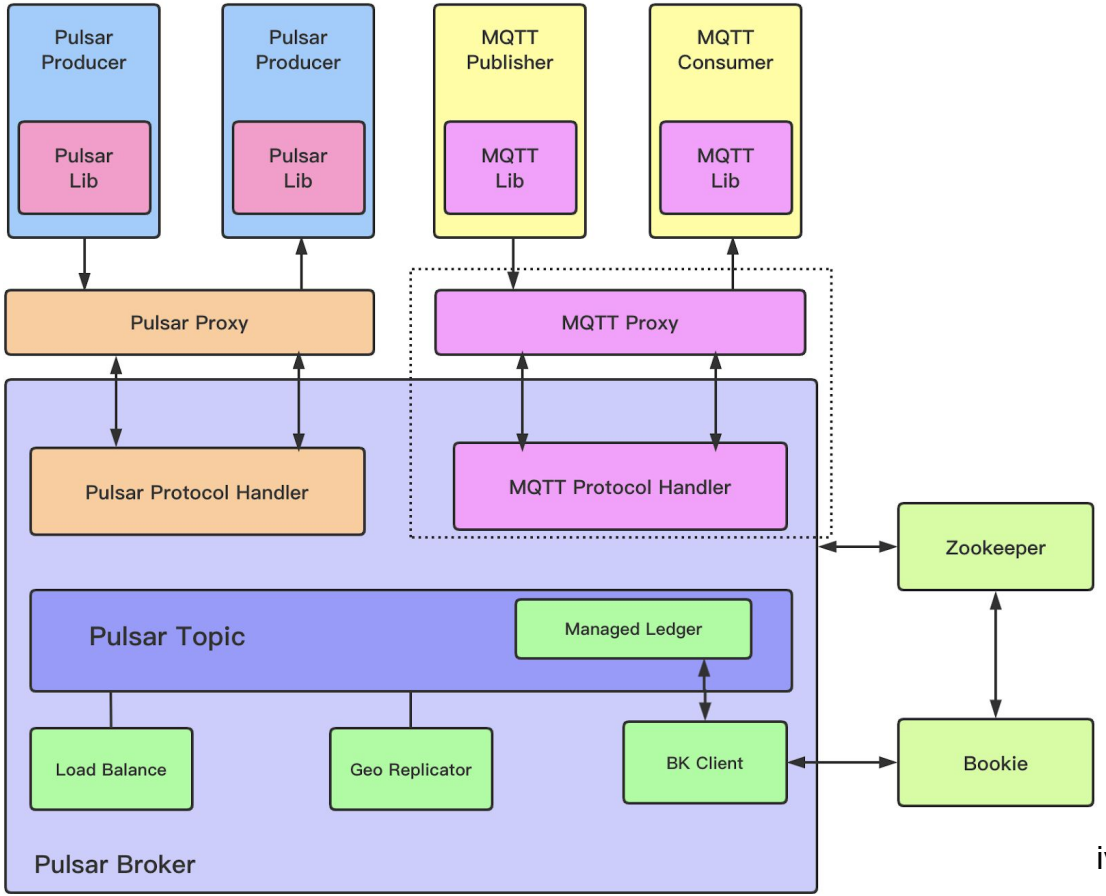


StreamNative

MQTT

On Pulsar

(MoP)



StreamNative

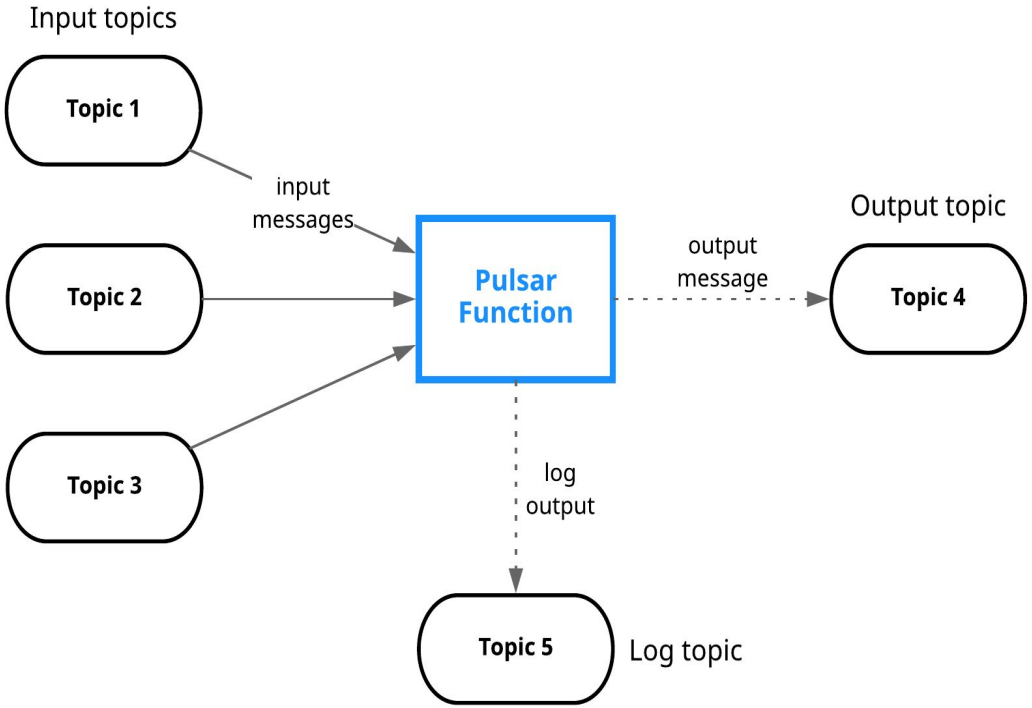
ive.io

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.

Pulsar Functions

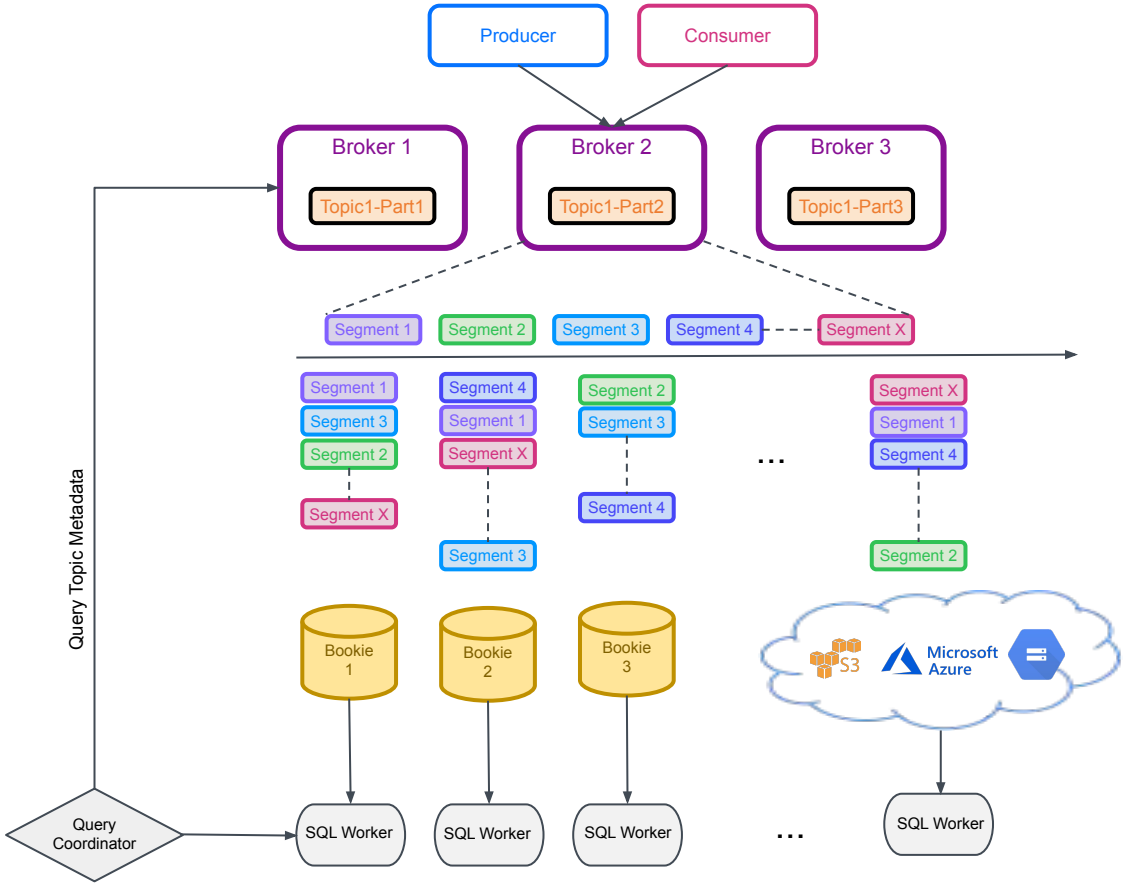


- Consume messages from one or more Pulsar topics.
- Apply user-supplied processing logic to each message.
- Publish the results of the computation to another topic.
- Support multiple programming languages (Java, Python, Go)
- Can leverage 3rd-party libraries to support the ***execution of ML models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

Pulsar SQL

Presto/Trino workers can read segments directly from bookies (or offloaded storage) in parallel.

trino



StreamNative

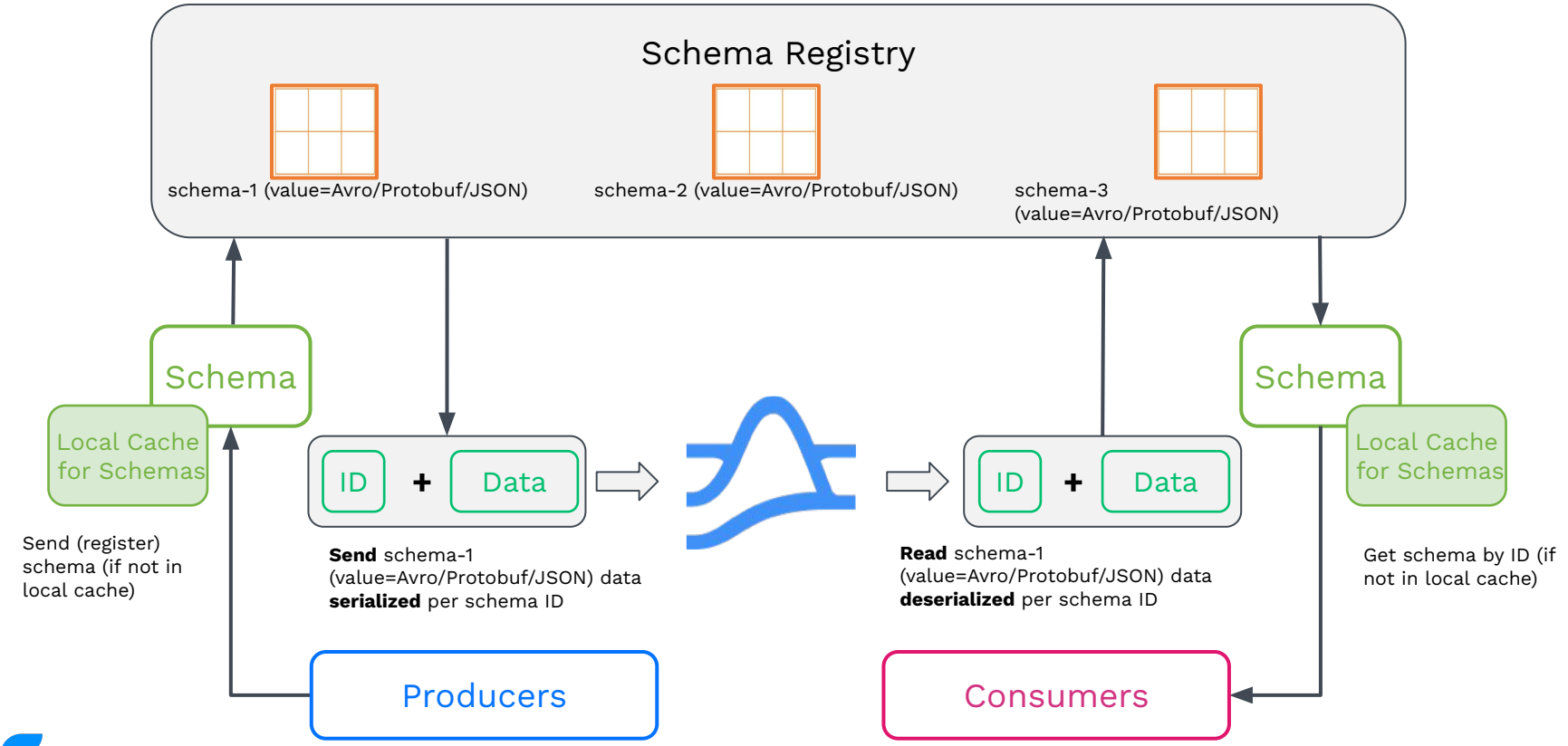
Query Your Topics with Pulsar SQL (Trino)

```
presto> select camera, cpu, cputempf, gputempf, memory, top1, top1pct, uuid, __publish_time__, __message_id__, __key__ from  pulsar."public/default".iotjetsonjson;
  camera    | cpu  | cputempf | gputempf | memory |       top1       |    top1pct     |                uuid                |    __publish_time__     | __message_id__ |
------------+------+----------+----------+--------+------------------+----------------+------------------------------------+-------------------------+----------------+
 /dev/video0 |  8.7 | 82       | 82       |   33.5 | microphone, mike | 18.85986328125 | xav_uuid_video0_lgl_20211001183019 | 2021-10-01 14:30:30.657 | (564,3,0)      |
 /dev/video0 |  8.7 | 82       | 82       |   33.6 | microphone, mike | 19.22607421875 | xav_uuid_video0_kpt_20211001183033 | 2021-10-01 14:30:44.380 | (564,4,0)      |
 /dev/video0 | 12.0 | 80       | 81       |   33.5 | microphone, mike | 12.53662109375 | xav_uuid_video0_gzd_20211001182930 | 2021-10-01 14:29:48.756 | (564,0,0)      |
 /dev/video0 |  8.5 | 82       | 82       |   33.6 | microphone, mike |        14.0625 | xav_uuid_video0_wlw_20211001182951 | 2021-10-01 14:30:02.919 | (564,1,0)      |
 /dev/video0 |  8.5 | 82       | 82       |   33.5 | microphone, mike |     29.8828125 | xav_uuid_video0_ulq_20211001183005 | 2021-10-01 14:30:16.787 | (564,2,0)      |
(5 rows)
(END)
```

```
presto> show tables in pulsar."public/default";
     Table
---------------
 generator_test
 iotjetsonjson
 mqtt-2
(3 rows)

Query 20211001_054538_00008_s8x23, FINISHED, 1 node
Splits: 19 total, 19 done (100.00%)
0:00 [3 rows, 105B] [14 rows/s, 493B/s]
```

Schema Registry



StreamNative

Streaming FLiPS Apps

SCYLLA <-> Events <-> PULSAR <-> Events <-> Flink

Edge

Gateway

PULSAR

Unified Batch and Stream **STORAGE**

PULSAR

(Queuing + Streaming)

Streaming

Unified Batch and Stream **COMPUTING**

Flink

(Batch + Stream)

Offload

Batch

S3

Tiered Storage

hadoop HDFS

StreamNative Cloud

APACHE Spark

Protocols

Pulsar

KoP

MoP

Websocket

Apps

Pulsar Sink

StreamNative Hub

CDC

APACHE nifi

StreamNative

PULSAR Functions

Use Cases

Unified Messaging Platform

AdTech

Fraud Detection

Connected Car

IoT Analytics

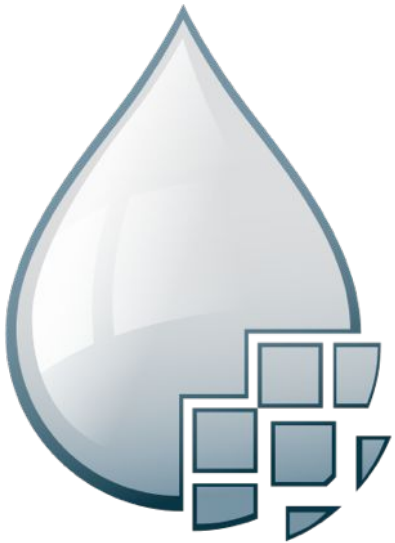
Microservices Development

StreamNative

Apache NiFi

StreamNative

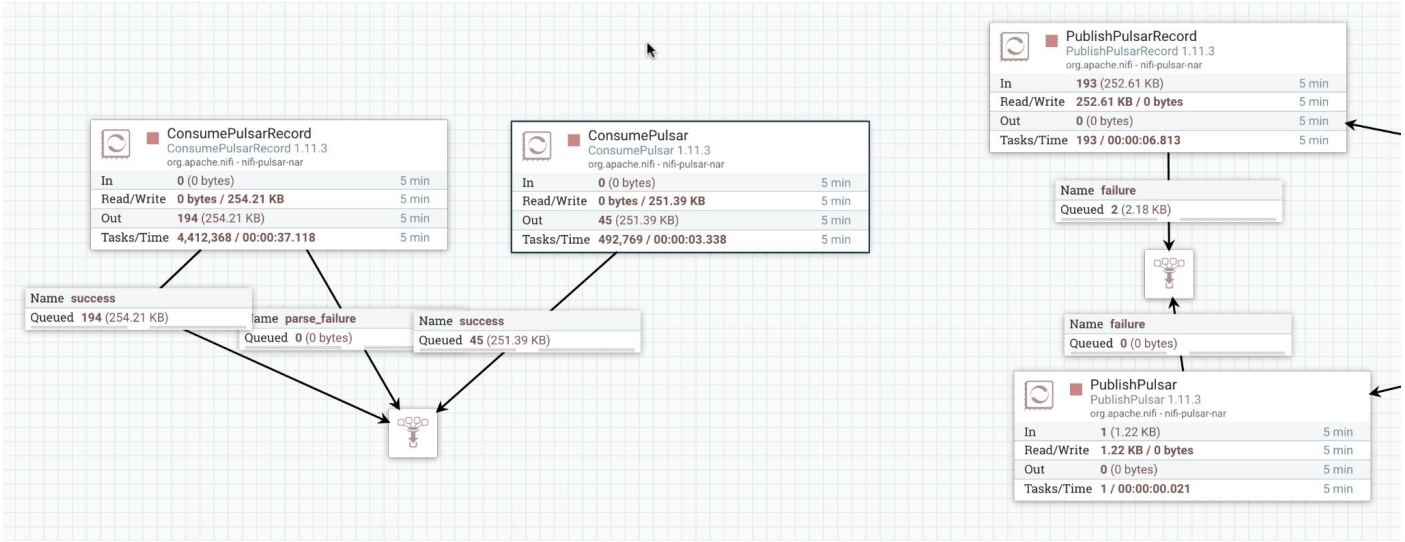
Why Apache NiFi?



- Guaranteed delivery
- Data buffering
  - Backpressure
  - Pressure release
- Prioritized queuing
- Flow specific QoS
  - Latency vs. throughput
  - Loss tolerance
- Data provenance
- Supports push and pull models

- Hundreds of processors
- Visual command and control
- Over a 300 sources
- Flow templates
- Pluggable/multi-role security
- Designed for extension
- Clustering
- Version Control

Apache NiFi Pulsar Connector



https://github.com/david-streamlio/pulsar-nifi-bundle

StreamNative

Apache NiFi Pulsar Connector

Displaying 12 of 339

pulsar

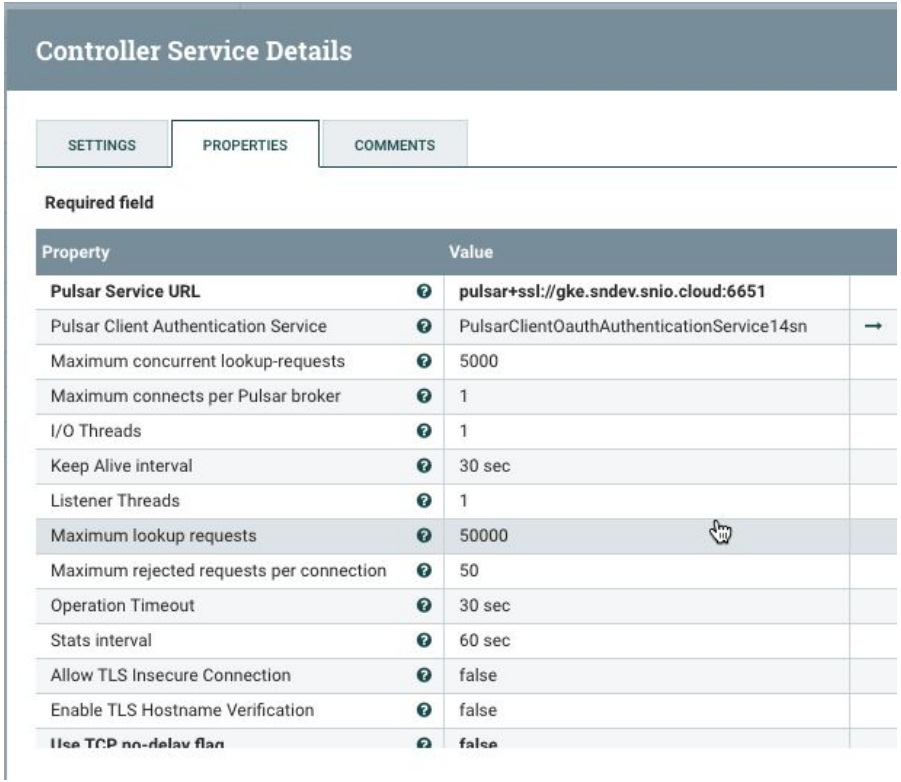
Type	Version ▲	Tags
ConsumePulsar	1.11.0	PubSub, Consume, Ingest, Get, I...
ConsumePulsarRecord	1.11.0	PubSub, Consume, Ingest, Get, ...
PublishPulsar	1.11.0	PubSub, Message, Pulsar, Apac...
PublishPulsarRecord	1.11.0	PubSub, 1.0, Message, csv, json...
ConsumePulsar	1.11.3	PubSub, Consume, Ingest, Get, I...
ConsumePulsarRecord	1.11.3	PubSub, Consume, Ingest, Get, ...
PublishPulsar	1.11.3	PubSub, Message, Pulsar, Apac...
PublishPulsarRecord	1.11.3	PubSub, 1.0, Message, csv, json...
ConsumePulsar	1.14.0	PubSub, Consume, Ingest, Get, I...
ConsumePulsarRecord	1.14.0	PubSub, Consume, Ingest, Get, ...
PublishPulsar	1.14.0	PubSub, Message, Pulsar, Apac...
PublishPulsarRecord	1.14.0	PubSub, 1.0, Message, csv, json...

ConsumePulsar 1.11.3 org.apache.nifi - nifi-pulsar-nar

Consumes messages from Apache Pulsar. The complementary NiFi processor for sending messages is PublishPulsar.

StreamNative

Apache NiFi Pulsar Connector



StreamNative

Apache NiFi Pulsar Connector

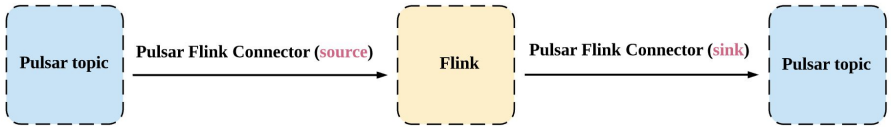
StreamNative

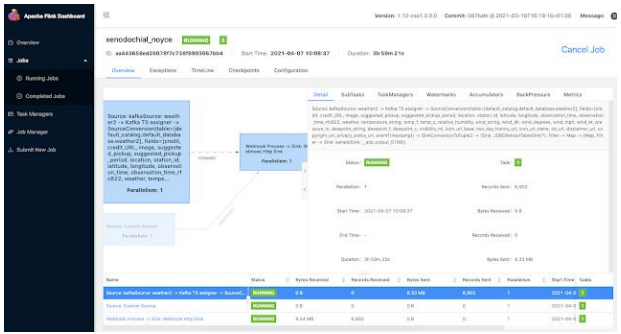
Apache Flink

Flink

StreamNative

Why Apache Flink?

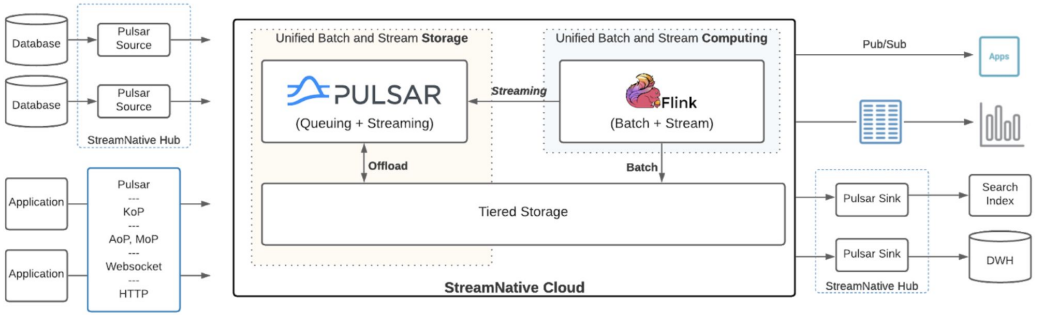


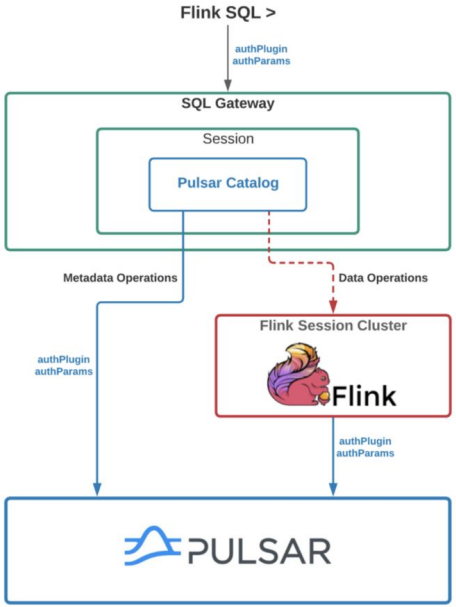


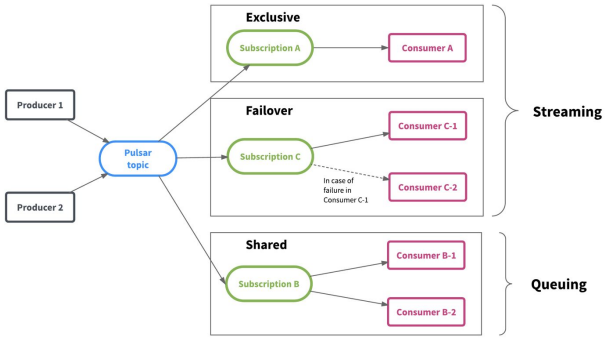
- Unified computing engine
- Batch processing is a special case of stream processing
- Stateful processing
- Massive Scalability
- Flink SQL for queries, inserts against Pulsar Topics
- Streaming Analytics
- Continuous SQL
- Continuous ETL
- Complex Event Processing
- Standard SQL Powered by Apache Calcite

StreamNative

Flink + Pulsar







https://flink.apache.org/2019/05/03/pulsar-flink.html
https://github.com/streamnative/pulsar-flink
https://streamnative.io/en/blog/release/2021-04-20-flink-sql-on-streamnative-cloud

StreamNative

StreamNative Cloud

Stream Native

Founded by the original developers of Apache Pulsar.

Passionate and dedicated team.

StreamNative helps teams to **capture**, **manage**, and **leverage data** using Pulsar's unified messaging and streaming platform.

streamnative.io

StreamNative Cloud

Powered by Apache Pulsar, StreamNative provides a cloud-native, real-time messaging and streaming platform to support multi-cloud and hybrid cloud strategies.

Cloud Native

Built for Containers

Flink SQL

StreamNative

StreamNative

Products Open Source Resources Contact Login

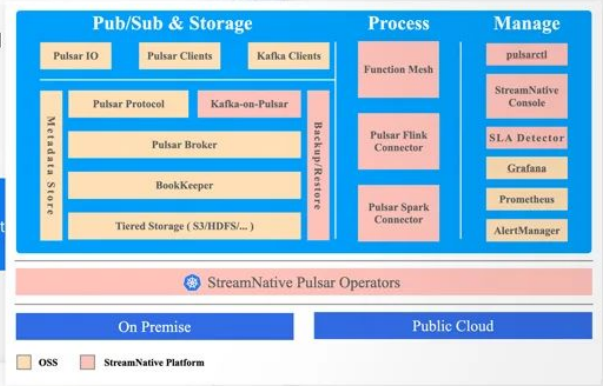
The unified messaging and streaming platform made by the creators of Apache Pulsar.

Built for Kubernetes. Made for the cloud. Enables multi-cloud and hybrid

Take A Tour Contact Sales

Product Update

StreamNative Cloud on AWS Marketplace. Leverage Pulsar on the largest cloud provider with StreamNat



StreamNative, Powered by Apache Pulsar

StreamNative Cloud

Apache Pulsar as a service, StreamNative Cloud delivers a resilient and scalable messaging and event streaming service deployable in minutes.

StreamNative Platform

StreamNative Platform is a cloud-native messaging and event streaming platform built by the original creators for Apache Pulsar.

StreamNative Pro Services

Accelerate your messaging and streaming platform development and drive business results with help from StreamNative's Pulsar experts.

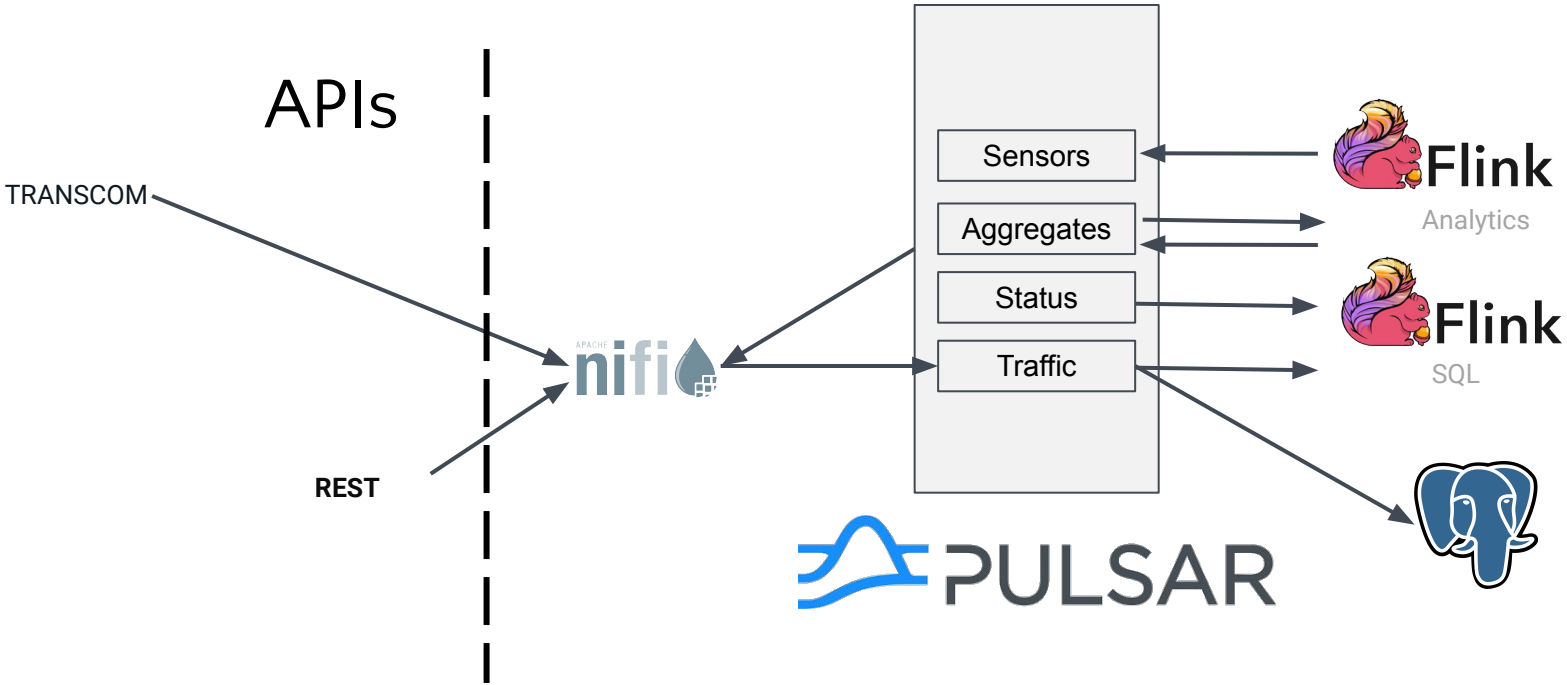
InfoWorld

BOSSIE 2021 AWARDS

Demo

StreamNative

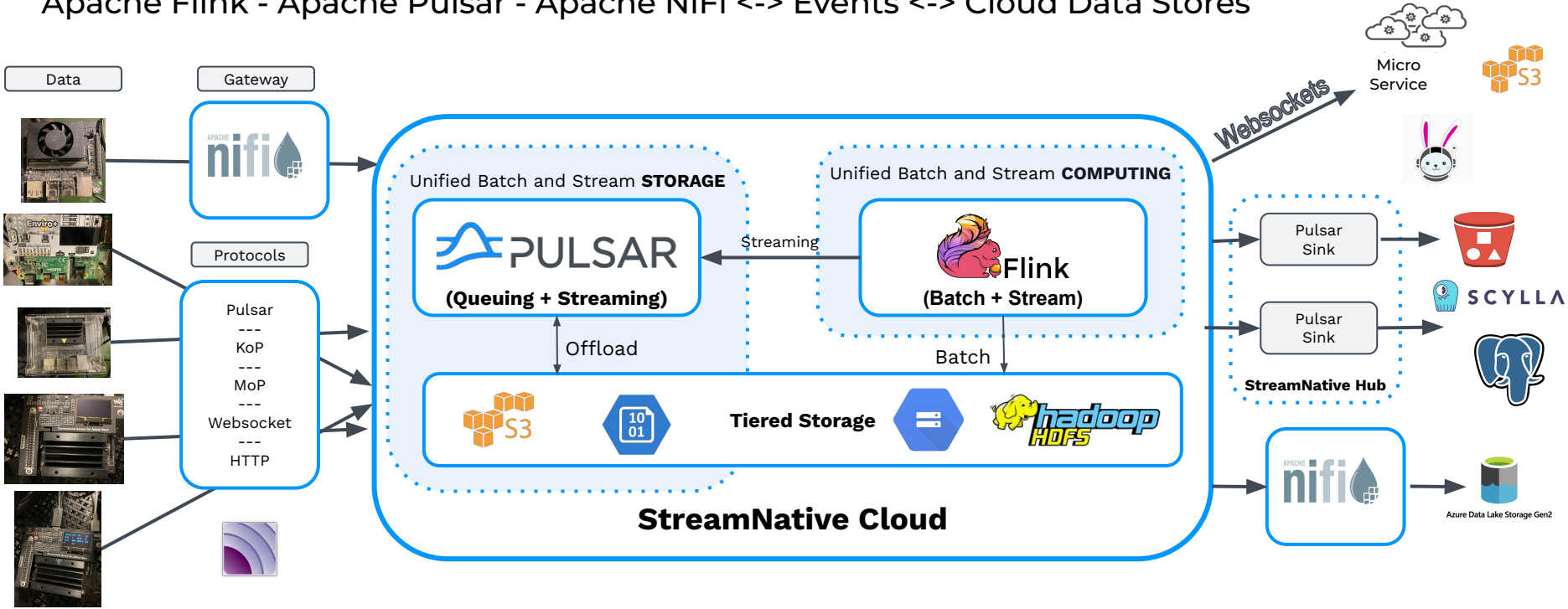
Real-Time Cloud Streaming Pipeline



StreamNative

Data to Cloud Data Lake

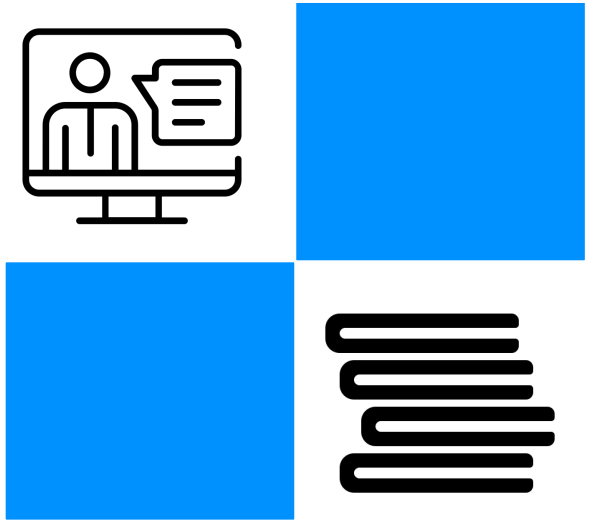
Apache Flink - Apache Pulsar - Apache NiFi <-> Events <-> Cloud Data Stores

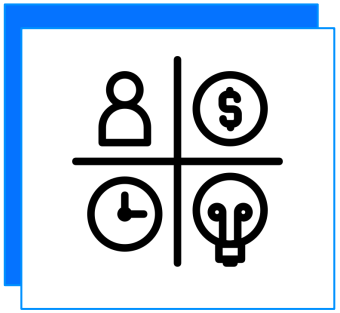


StreamNative

What's Next?

Here are resources to continue your journey with Apache Pulsar



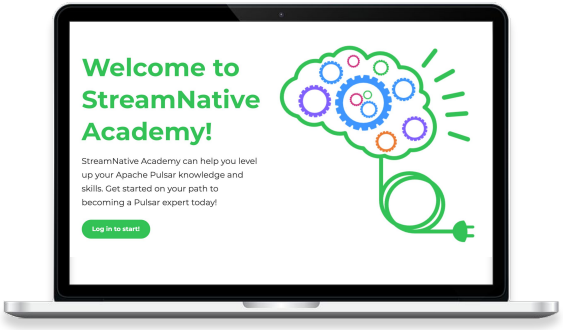


Pulsar Resources Page

Learn More

Tell us about your Pulsar experience and what improvements you would like to see!

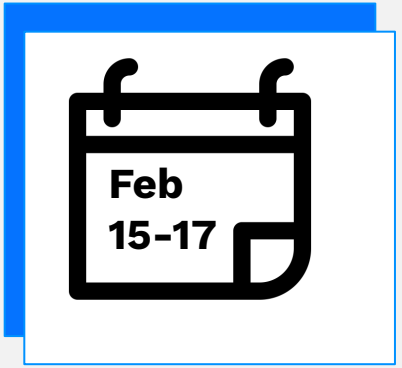
Start Survey



Now Available

On-Demand Pulsar Training

Academy.StreamNative.io



Live 3-day

Developers Training

Save Your Spot!

Times:

- Europe: 3:00 PM CET - 7:00 PM CET
- EasternTime: 9:00 AM - 1: 00 PM EST
- Pacific Time: 6:00 AM - 10 AM PST

StreamNative

Let’s Keep in Touch!

Tim Spann

Developer Advocate

@PaaSDev

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

https://github.com/tspannhw

StreamNative