

June 13-15, 2022 · Data platforms · Data engineering · Big data

BUDAPEST DATA FORUM



Unified Messaging and Data Streaming 101

Timothy Spann

Developer Advocate
StreamNative

Hotel Mercure Buda + Streaming

budapestdata.net



Tim Spann

Developer Advocate

- FLiP(N) Stack = Flink, Pulsar and NiFi Stack
- Streaming Systems/ Data Architect
- Experience:
 - 15+ years of experience with batch and streaming technologies including Pulsar, Flink, Spark, NiFi, Spring, Java, Big Data, Cloud, MXNet, Hadoop, Datalakes, IoT and more.



CLUSTERNA



Pivotal

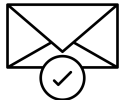
BARNES
& NOBLE



PULSAR 101



**Unified
Messaging
Platform**



**Guaranteed
Message
Delivery**

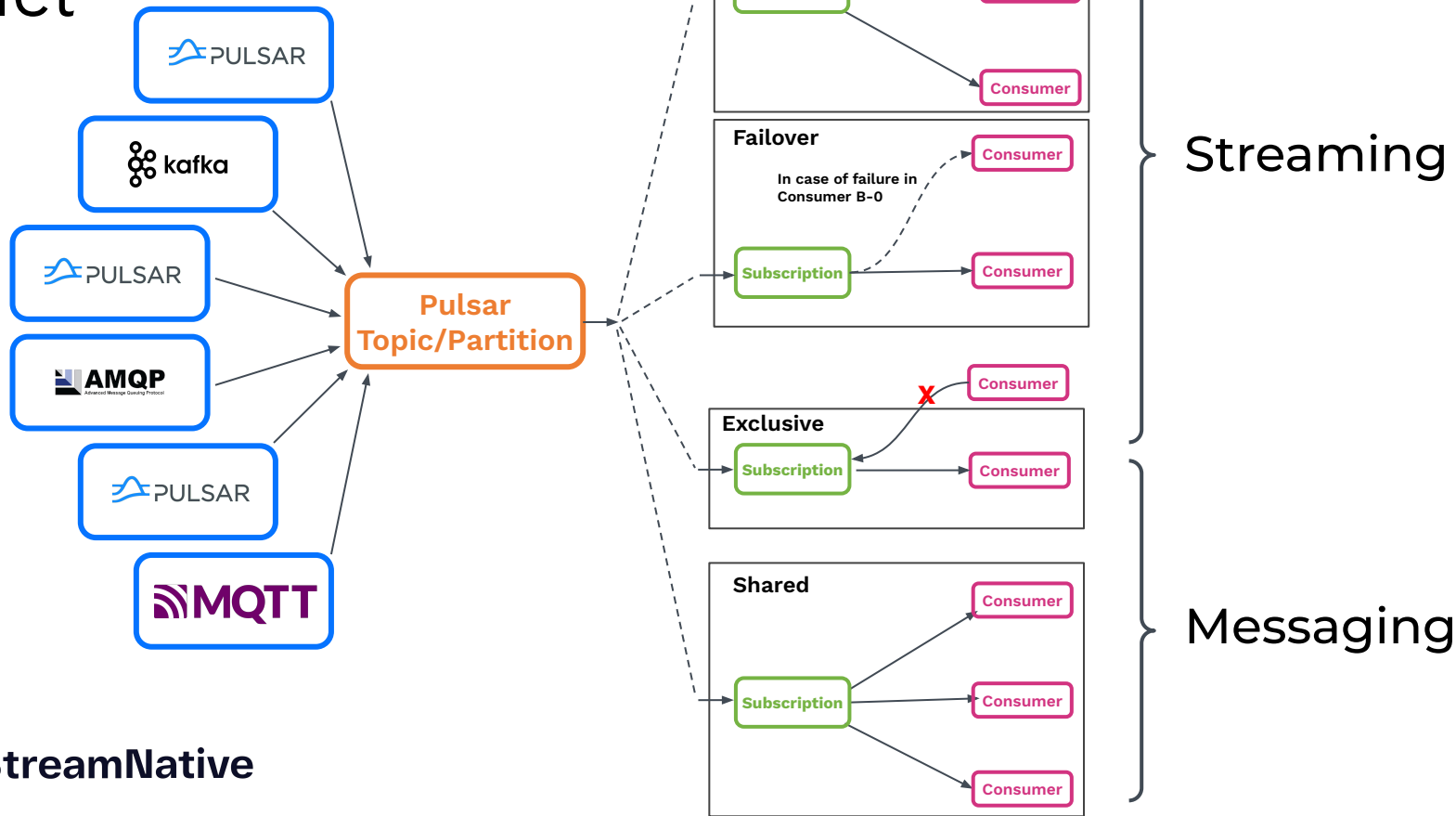


Resiliency



**Infinite
Scalability**

Unified Messaging Model



Tenants / Namespaces / 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)

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.

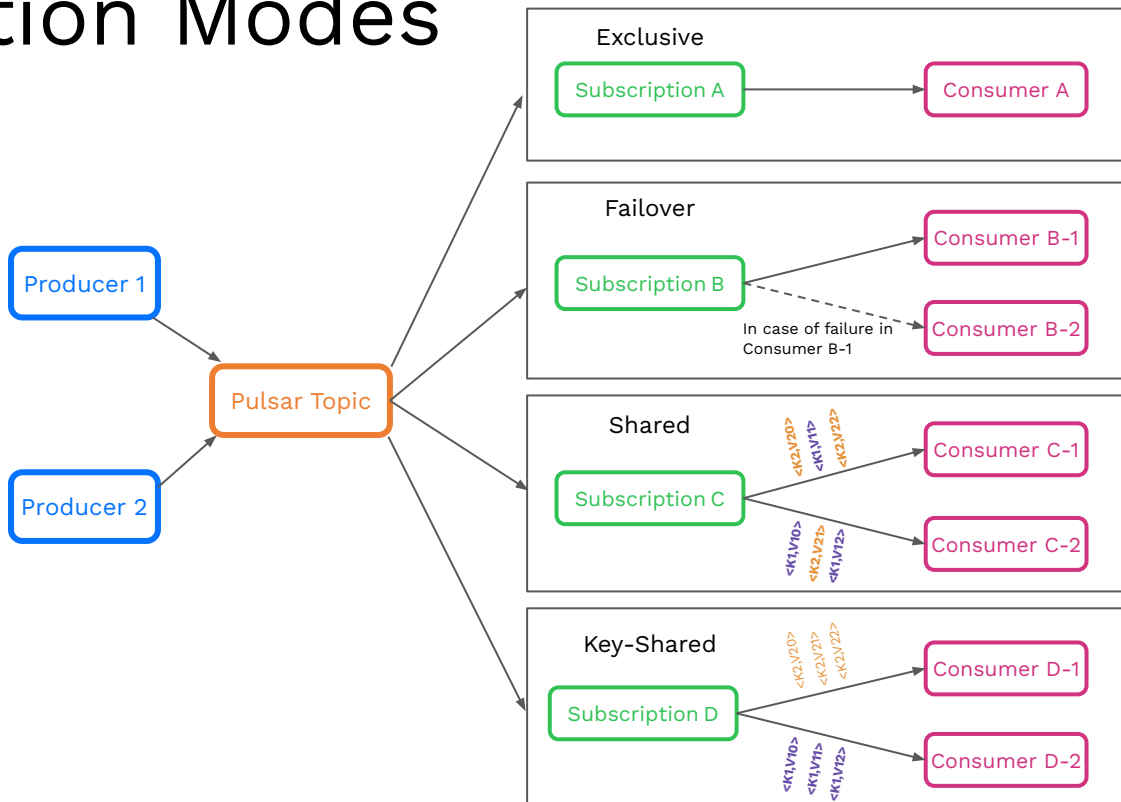
Pulsar Subscription Modes

Different subscription modes have different semantics:

Exclusive/Failover - guaranteed order, single active consumer

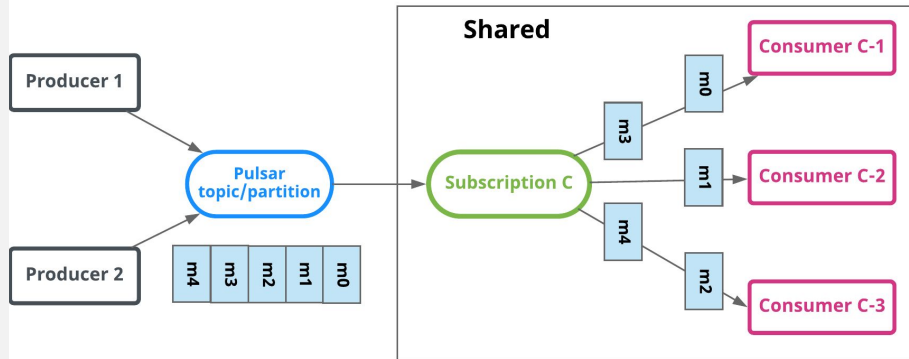
Shared - multiple active consumers, no order

Key_Shared - multiple active consumers, order for given key



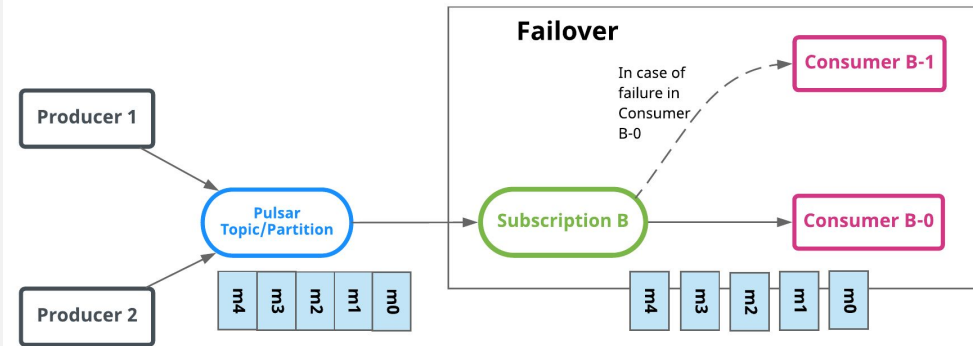
Flexible Pub/Sub API for Pulsar - Shared

```
Consumer consumer = client.newConsumer()  
    .topic("my-topic")  
    .subscriptionName("work-q-1")  
    .subscriptionType(SubType.Shared)  
    .subscribe();
```



Flexible Pub/Sub API for Pulsar - Failover

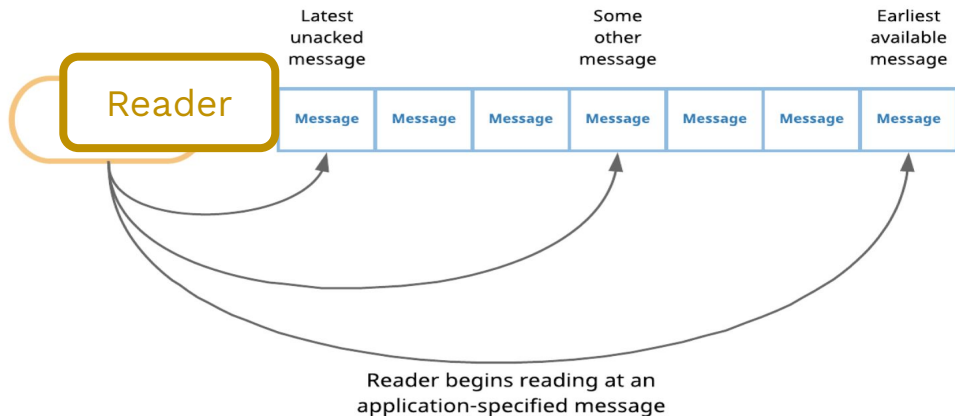
```
Consumer consumer = client.newConsumer()  
    .topic("my-topic")  
    .subscriptionName("stream-1")  
    .subscriptionType(SubType.Failover)  
    .subscribe();
```



Reader Interface

Create a reader that will read from some message between earliest and latest.

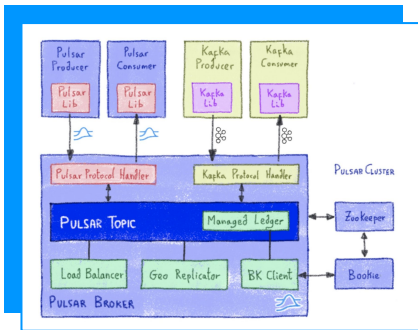
Applications manually control topic cursors



```
byte[] msgIdBytes = // Some byte array
```

```
MessageId id =  
MessageId.fromByteArray(msgIdBytes);
```

```
Reader<byte[]> reader =  
pulsarClient.newReader()  
    .topic(topic)  
    .startMessageId(id)  
    .create();
```



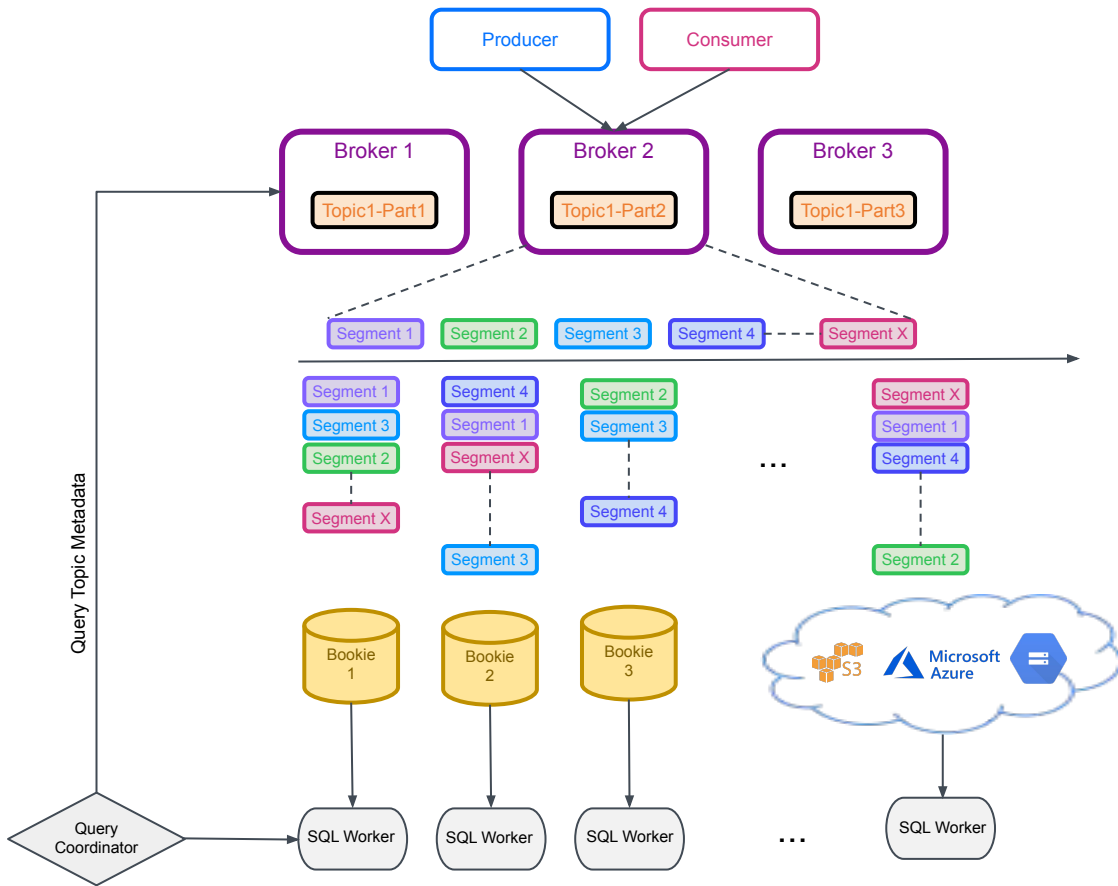
Connectivity

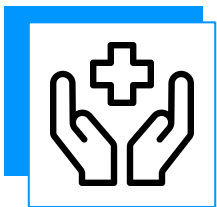
hub.streamnative.io

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

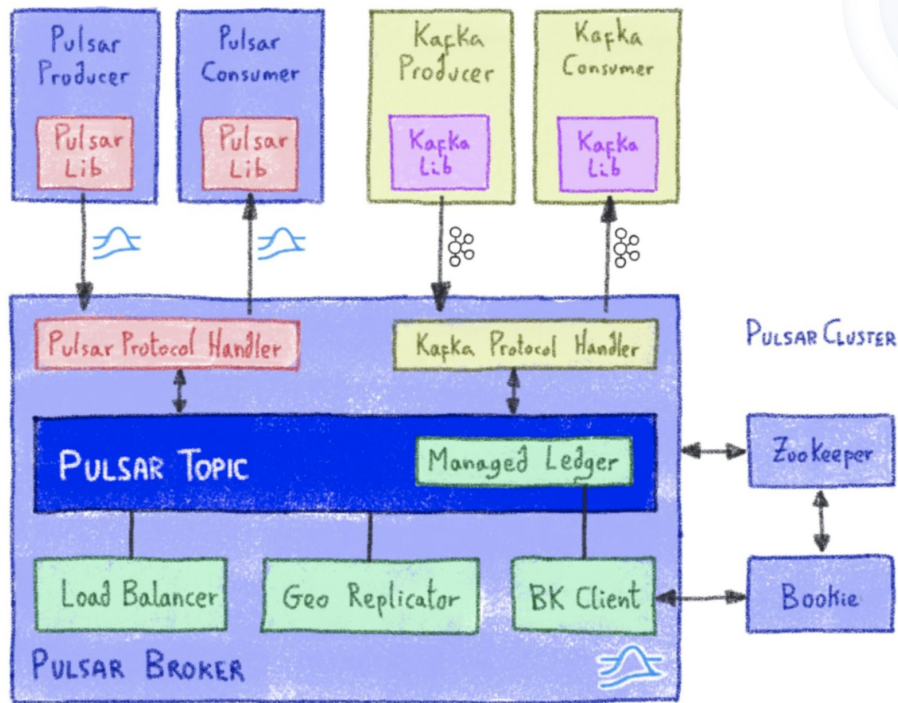
Pulsar SQL

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



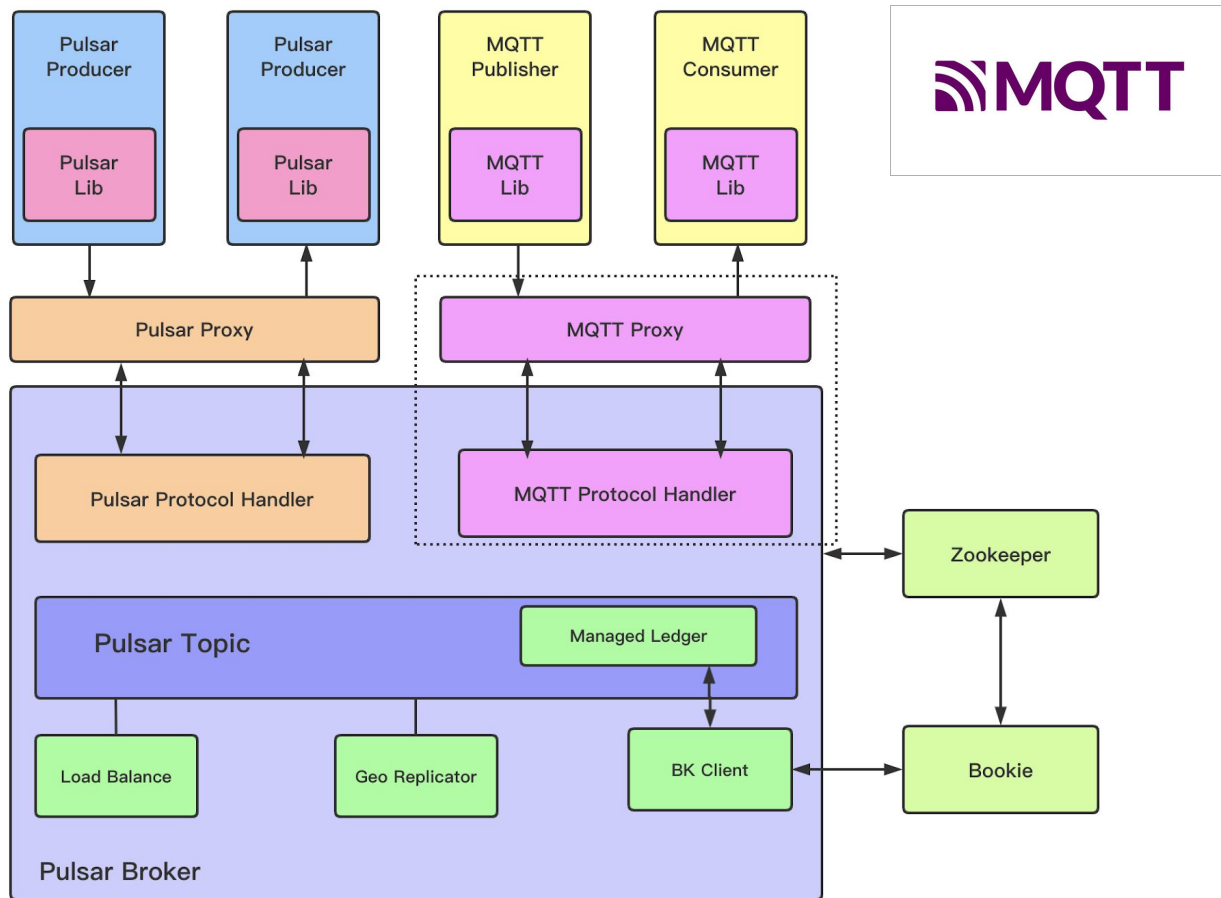


Kafka On Pulsar (KoP)





MQTT On Pulsar (MoP)

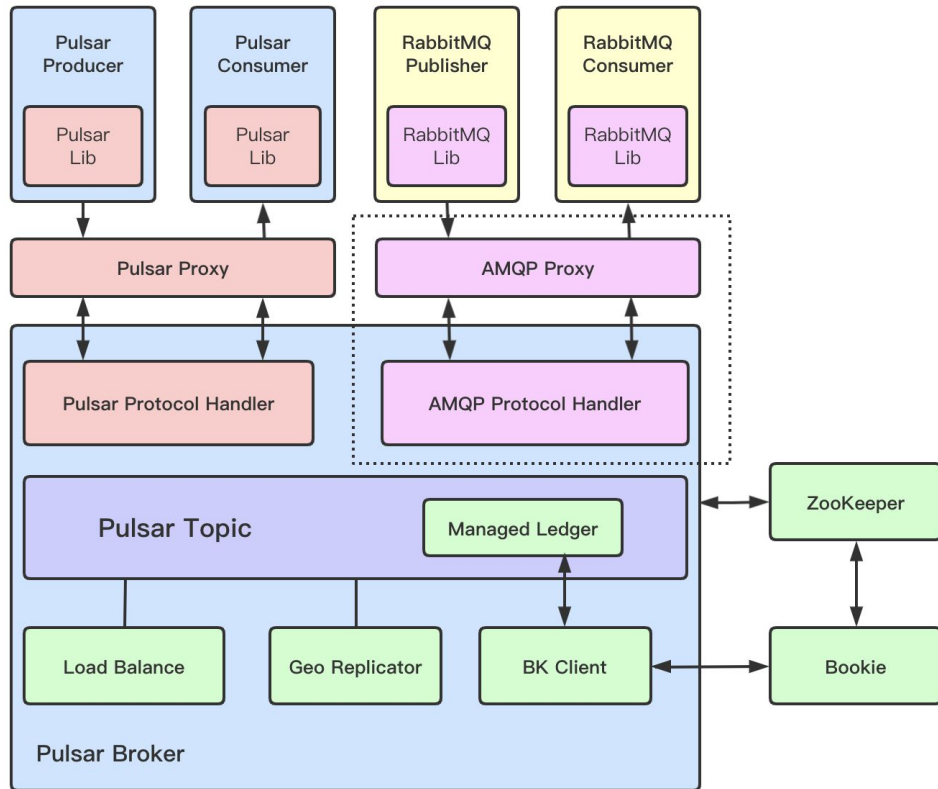




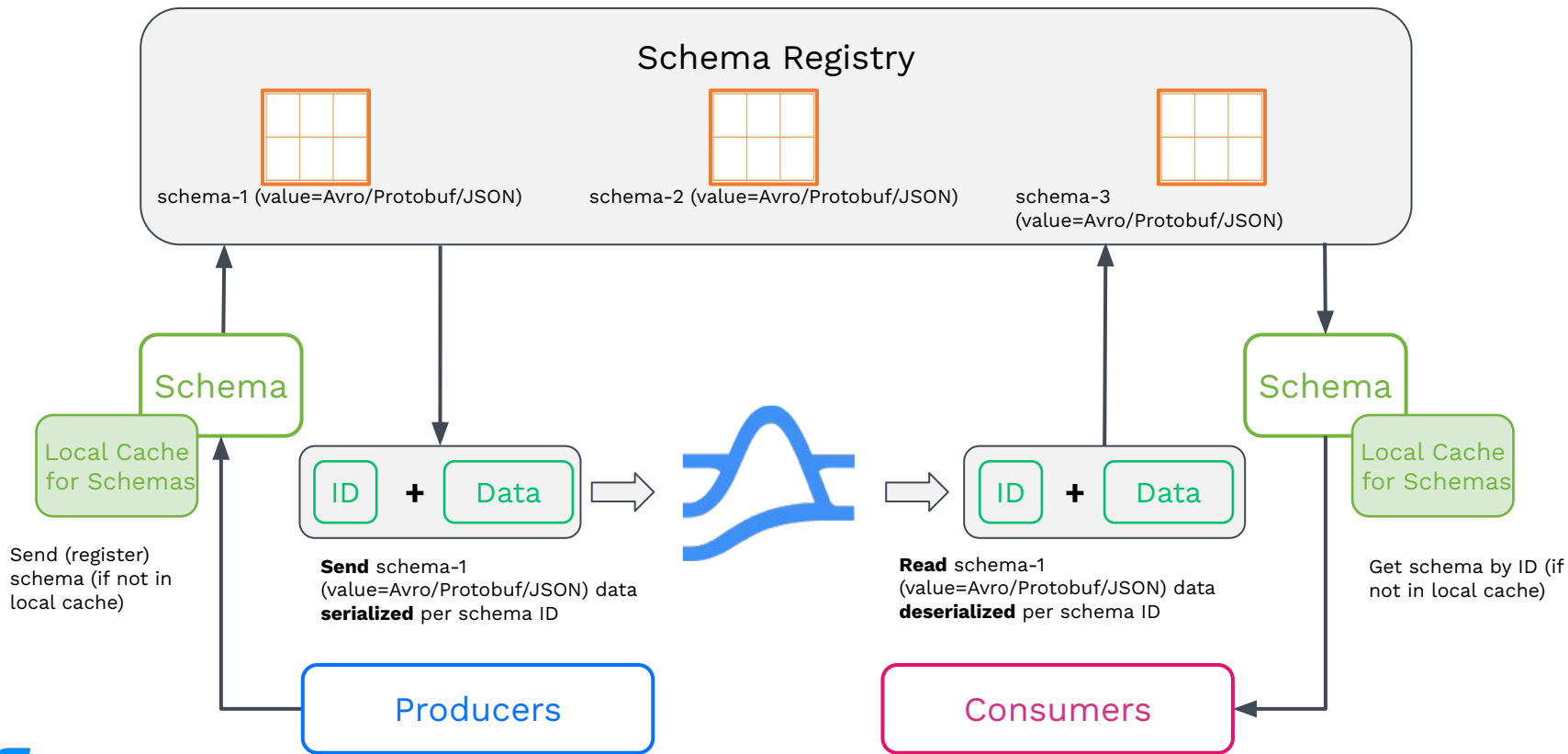
AMQP

On Pulsar

(AoP)



Schema Registry





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 Function SDK

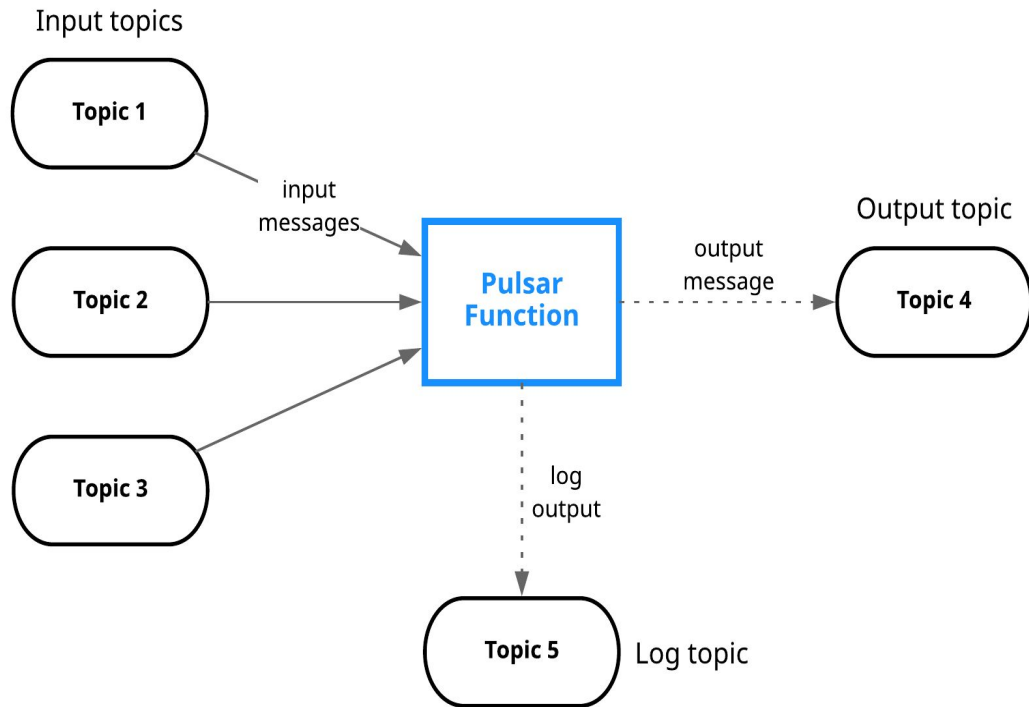
Your Code Here



```
import org.apache.pulsar.client.impl.schema.JSONSchema;
import org.apache.pulsar.functions.api.*;

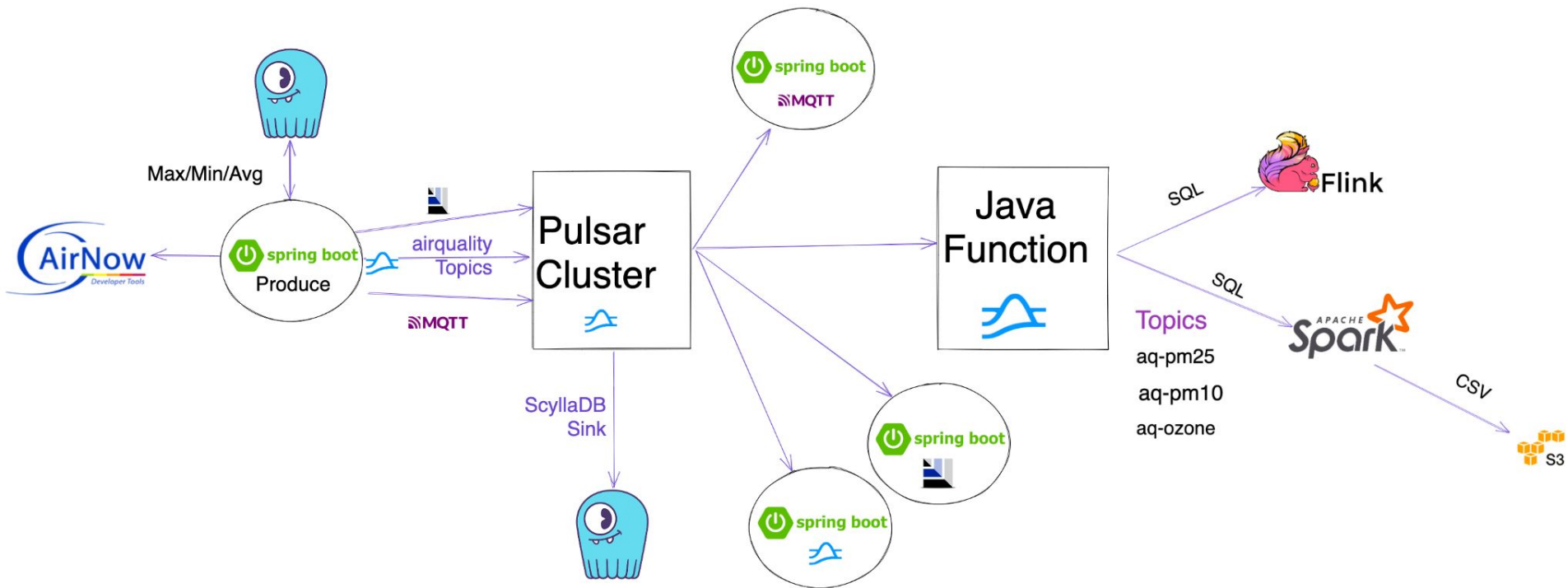
public class AirQualityFunction implements Function<byte[], Void> {
    @Override
    public Void process(byte[] input, Context context) {
        context.getLogger().debug("DBG:" + new String(input));
        context.newOutputMessage("topicname",
                                JSONSchema.of(Observation.class))
                .key(UUID.randomUUID().toString())
                .property("prop1", "value1")
                .value(observation)
                .send();
    }
}
```

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.***

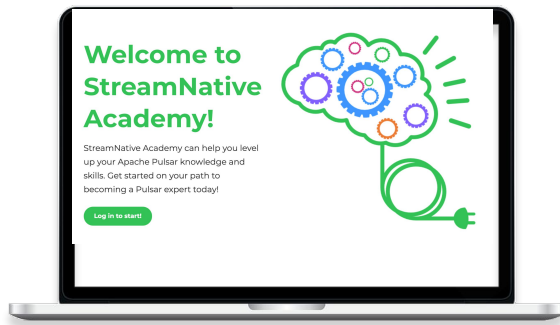
Demo



StreamNative Academy

Apache Pulsar Training

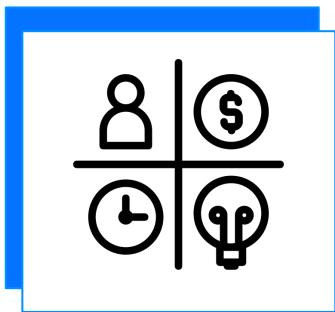
- Instructor-led courses
 - Pulsar Fundamentals
 - Pulsar Developers
 - Pulsar Operations
- On-demand learning with labs
- 300+ engineers, admins and architects trained!



Now Available

**On-Demand
Pulsar Training**

Academy.StreamNative.io



Pulsar Resources Page

[Learn More](#)

<https://spring.io/blog/2020/08/03/creating-a-function-for-consuming-data-and-generating-spring-cloud-stream-sink-applications>

<https://streamnative.io/blog/engineering/2022-04-14-what-the-flip-is-the-flip-stack/>

<https://streamnative.io/cloudforkafka/>

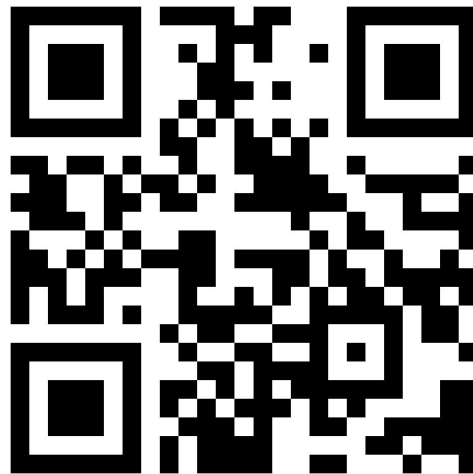
<https://github.com/tspannhw/airquality>

<https://github.com/tspannhw/pulsar-airquality-function>

FLiP Stack Weekly



<https://bit.ly/32dAJft>



This week in Apache Flink, Apache Pulsar, Apache NiFi, Apache Spark, Java and Open Source friends.



Stream Native

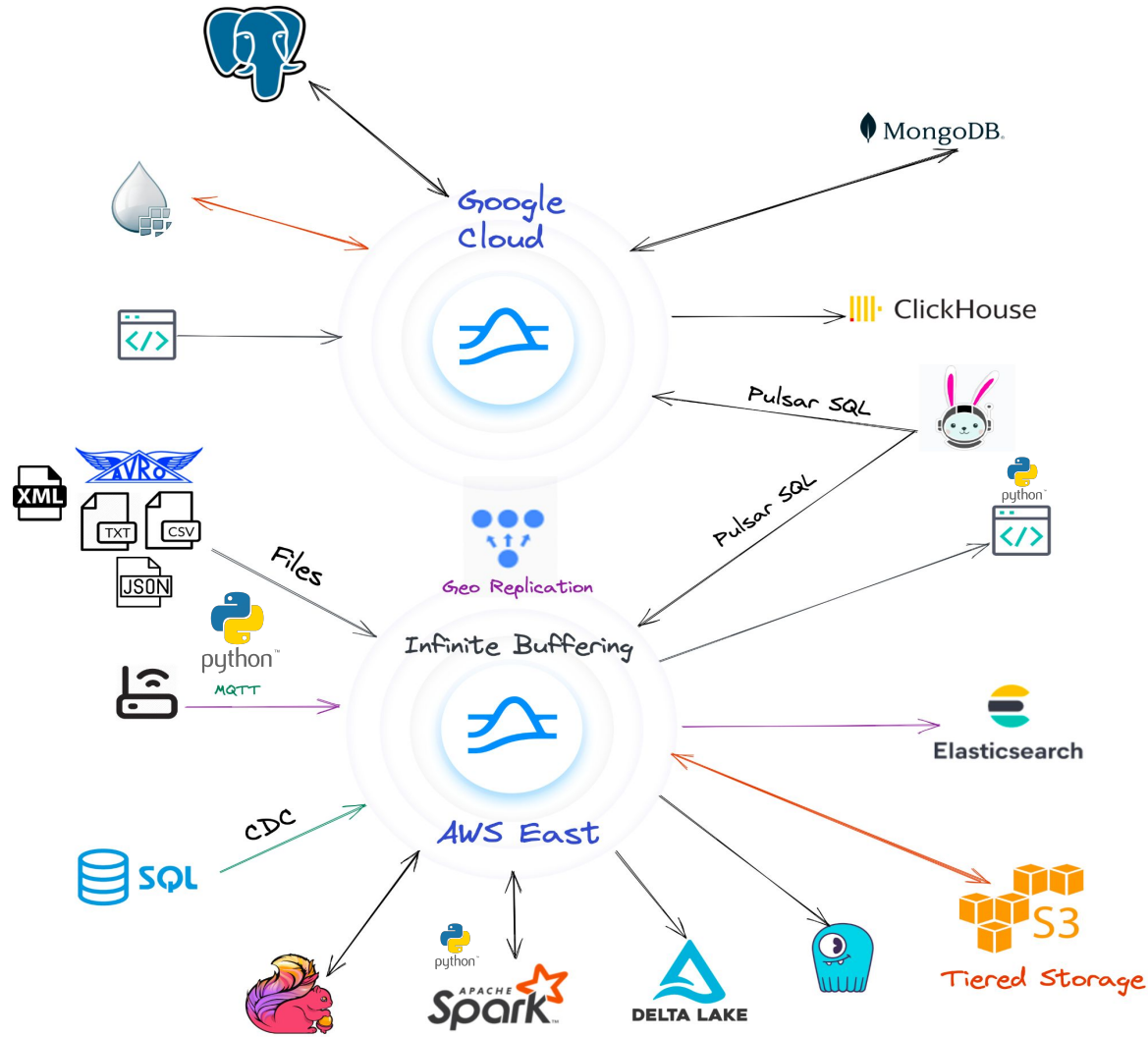
Founded by the original developers of Apache Pulsar.

Passionate and dedicated team.

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

streamnative.io

- Buffer
- Batch
- Route
- Filter
- Aggregate
- Enrich
- Replicate
- Dedupe
- Decouple
- Distribute



Let's Keep in Touch!



Tim Spann

Developer Advocate



[PaaSDev](#)



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



<https://github.com/tspannhw>

Notices

Apache Pulsar™

Apache®, Apache Pulsar™, Pulsar™, Apache Flink®, Flink®, Apache Spark®, Spark®, Apache NiFi®, NiFi® and the logo are either registered trademarks or trademarks of the [Apache Software Foundation](#) in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks.

Copyright © 2021-2022 The Apache Software Foundation. All Rights Reserved. Apache, Apache Pulsar and the Apache feather logo are trademarks of The Apache Software Foundation.