



Apache Pulsar with MQTT for Edge Computing

Timothy Spann | Developer Advocate



Tim Spann
Developer Advocate

DZone Zone Leader and Big Data
MVB Data DJay

- <https://www.datainmotion.dev/>
- <https://github.com/tspannhw/SpeakerProfile>
- <https://dev.to/tspannhw>
- <https://sessionize.com/tspann/>





Founded by the original developers of Apache Pulsar and Apache BookKeeper, StreamNative builds a cloud-native event streaming platform that enables enterprises to easily access data as real-time event streams.

Apache Pulsar



Apache  **PULSAR** is an open source, cloud-native distributed messaging and streaming platform.

What are the Benefits of Pulsar?



Multi-Tenancy

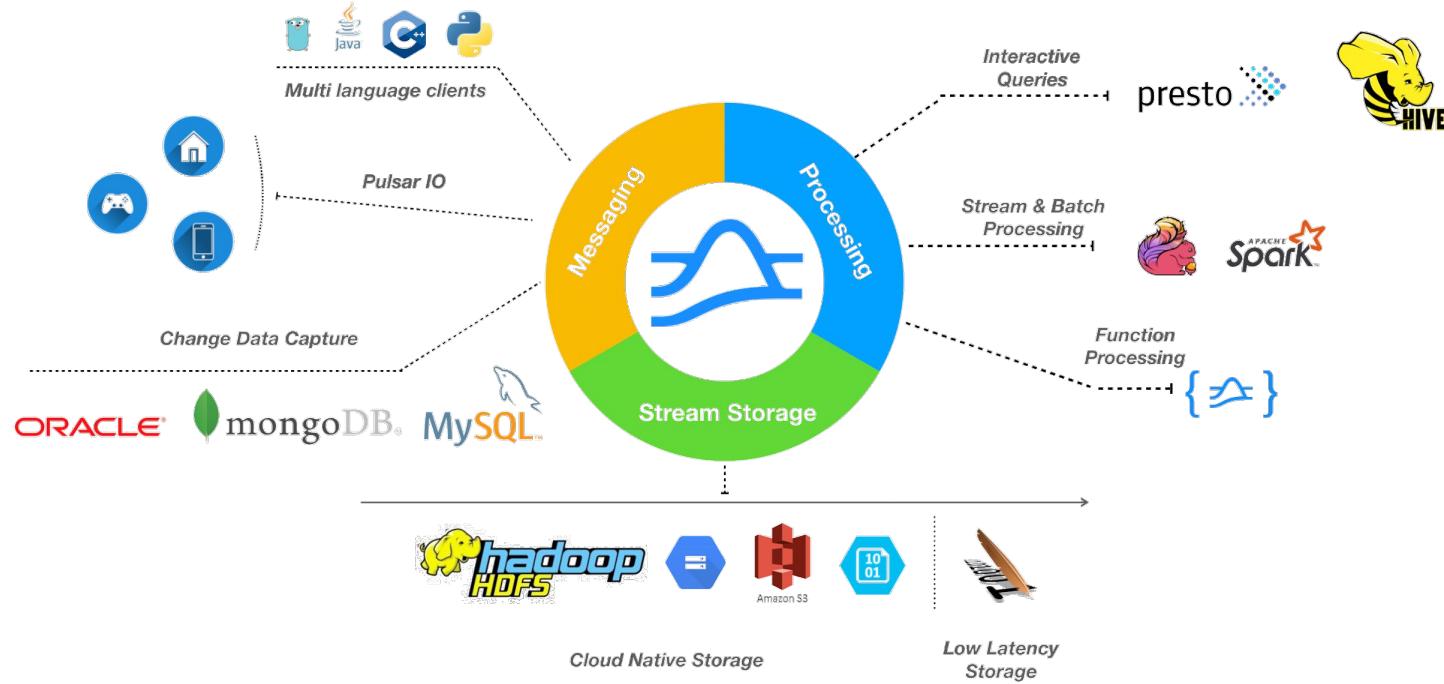
Scalability

Geo-Replication

Unified Messaging
Model

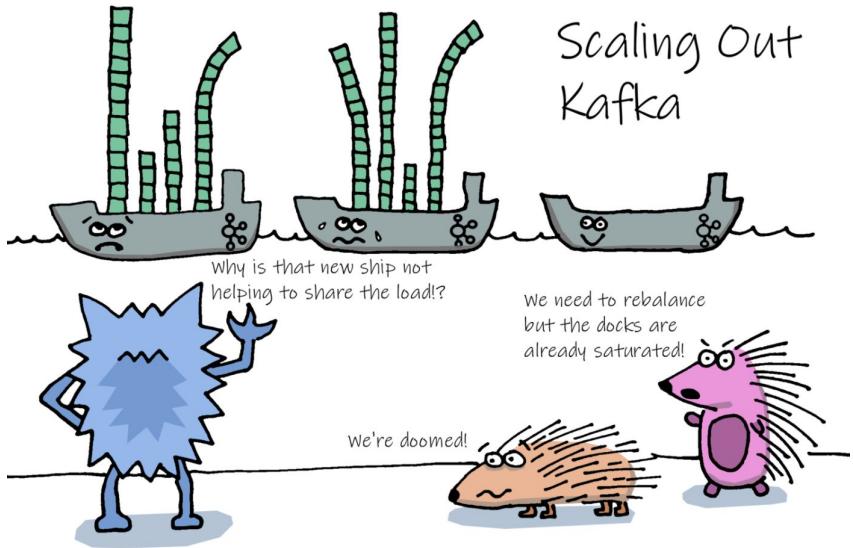
Data Durability

Apache Pulsar



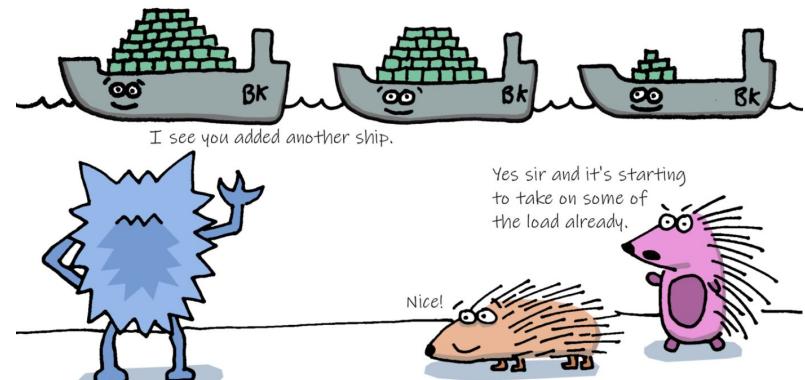
A Unified Messaging Platform





Scaling Out Kafka

Meanwhile, in a parallel universe...

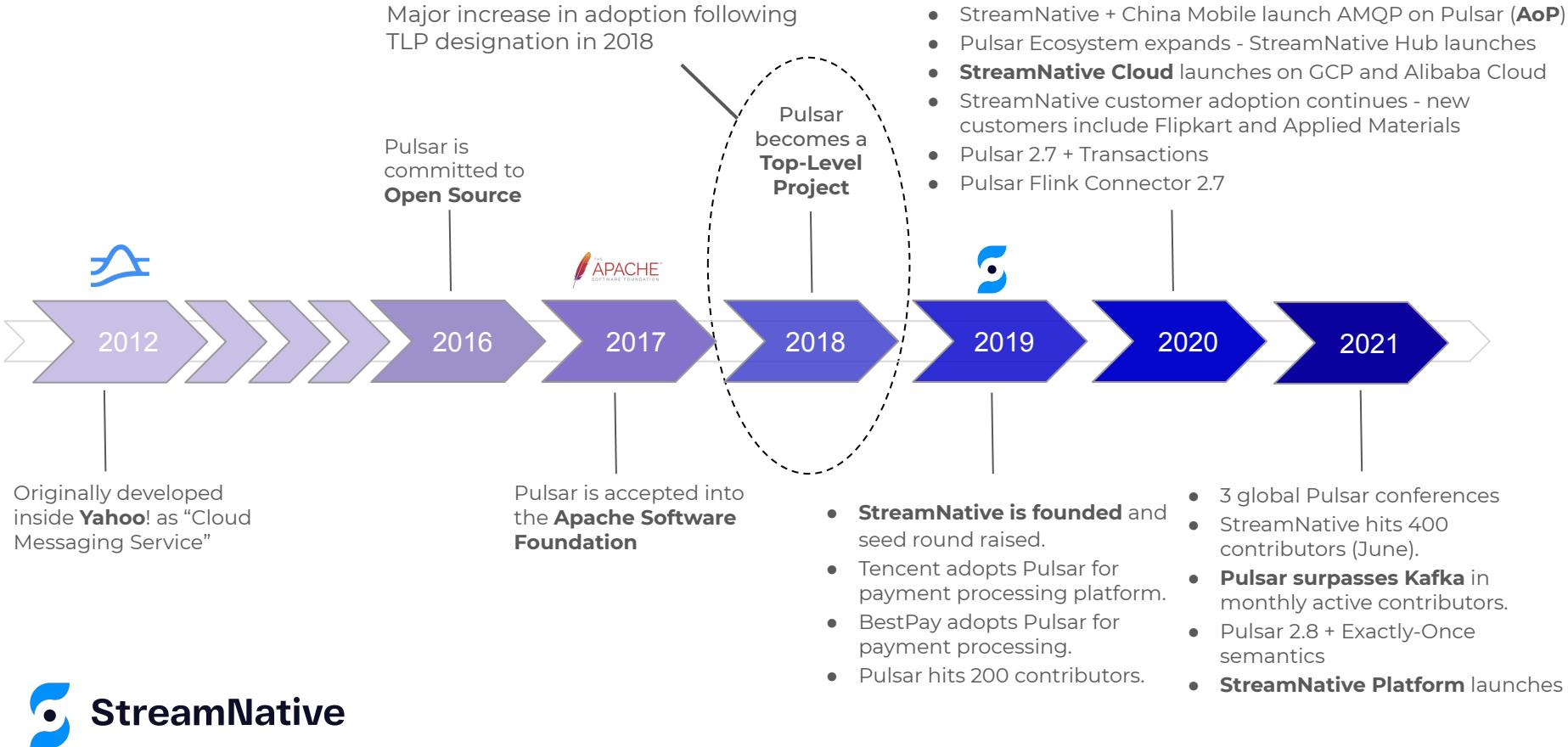


Scaling Out Pulsar

*Illustrations by Jack Vanlightly

Pulsar is built for easy scale-out.

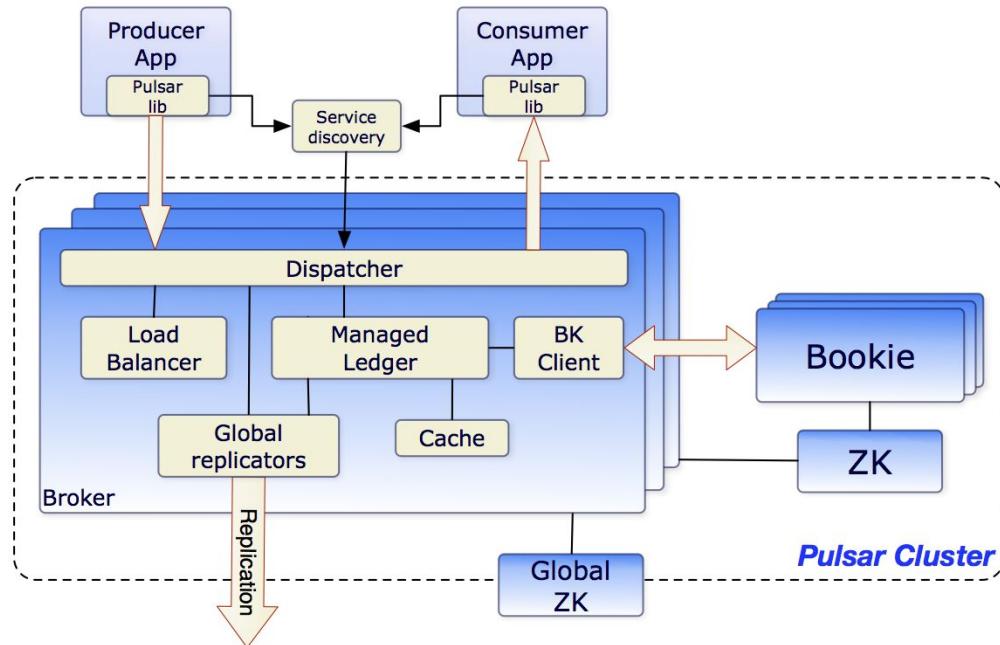
Key Milestones



Apache Pulsar Overview

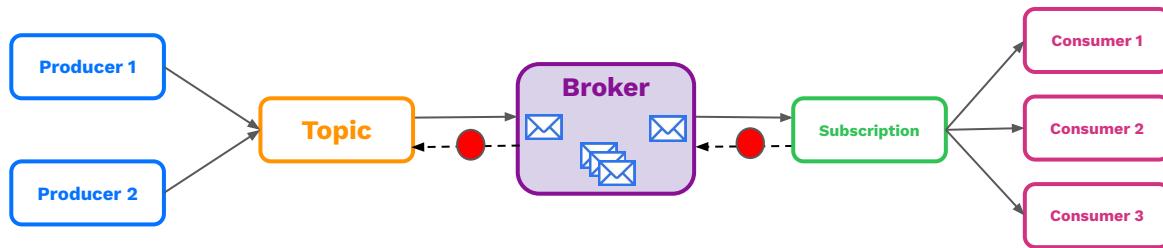
Enable Geo-Replicated Messaging

- Pub-Sub
- Geo-Replication
- Pulsar Functions
- Horizontal Scalability
- Multi-tenancy
- Tiered Persistent Storage
- Pulsar Connectors
- REST API
- CLI
- Many clients available
- Four Different Subscription Types
- Multi-Protocol Support
 - MQTT
 - AMQP
 - JMS
 - Kafka
 - ...



Pulsar's Publish-Subscribe model

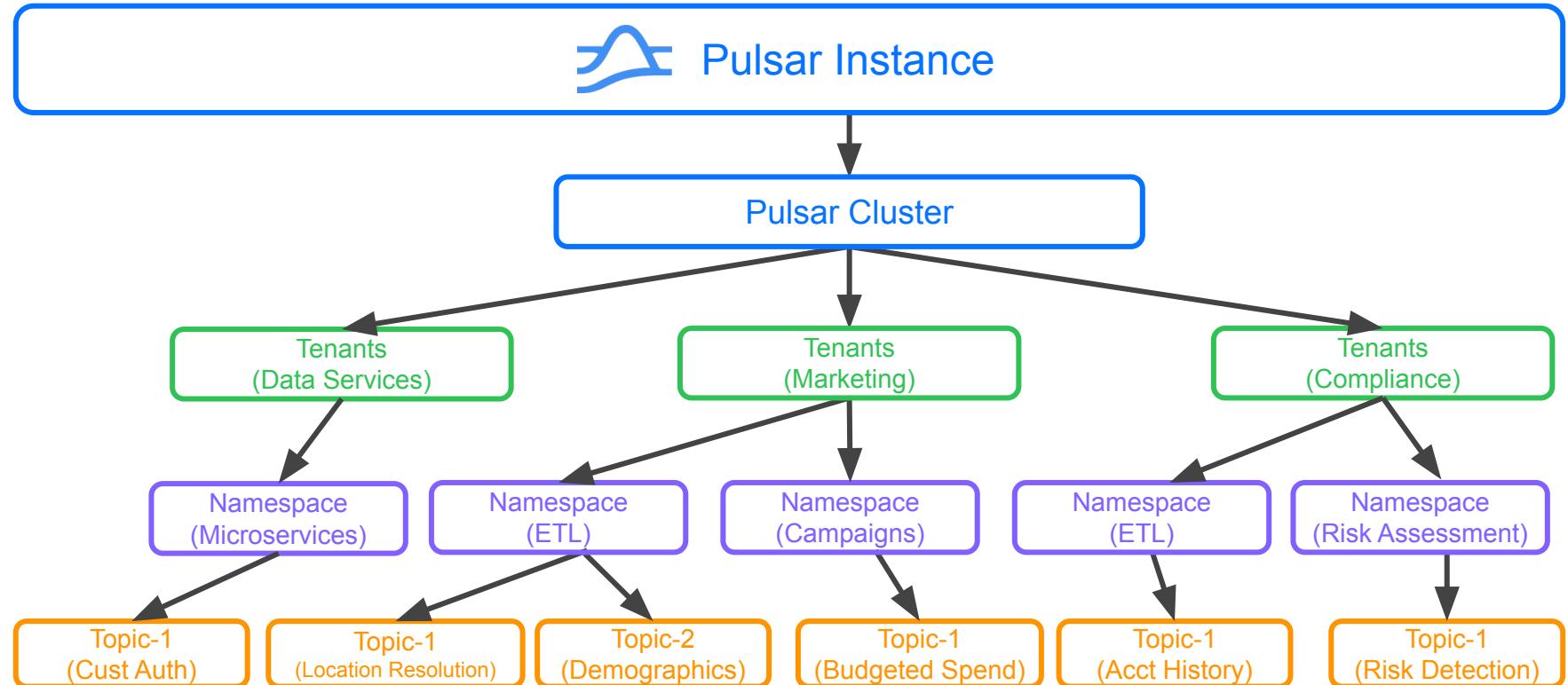
- **Producers** send messages.
- **Topics** are an ordered, named channel that producers use to transmit messages to subscribed consumers.
- **Messages** belong to a topic and contain an arbitrary payload.
- **Brokers** handle connections and routes messages between producers / consumers.
- **Subscriptions**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



What is the Pulsar Ecosystem?

- **Functions and Connectors**
 - Functions: Lightweight stream processing
 - Connectors: Part of “Pulsar IO”, includes “Source” and “Sink” APIs
 - Files, Databases, Data tools, Cloud Services, etc
- **Protocol Handlers**
 - Allows Pulsar to handle additional protocols by an extendable API running in the broker
 - AoP (AMQP), KoP (Kafka), MoP (MQTT)

Topics



StreamNative

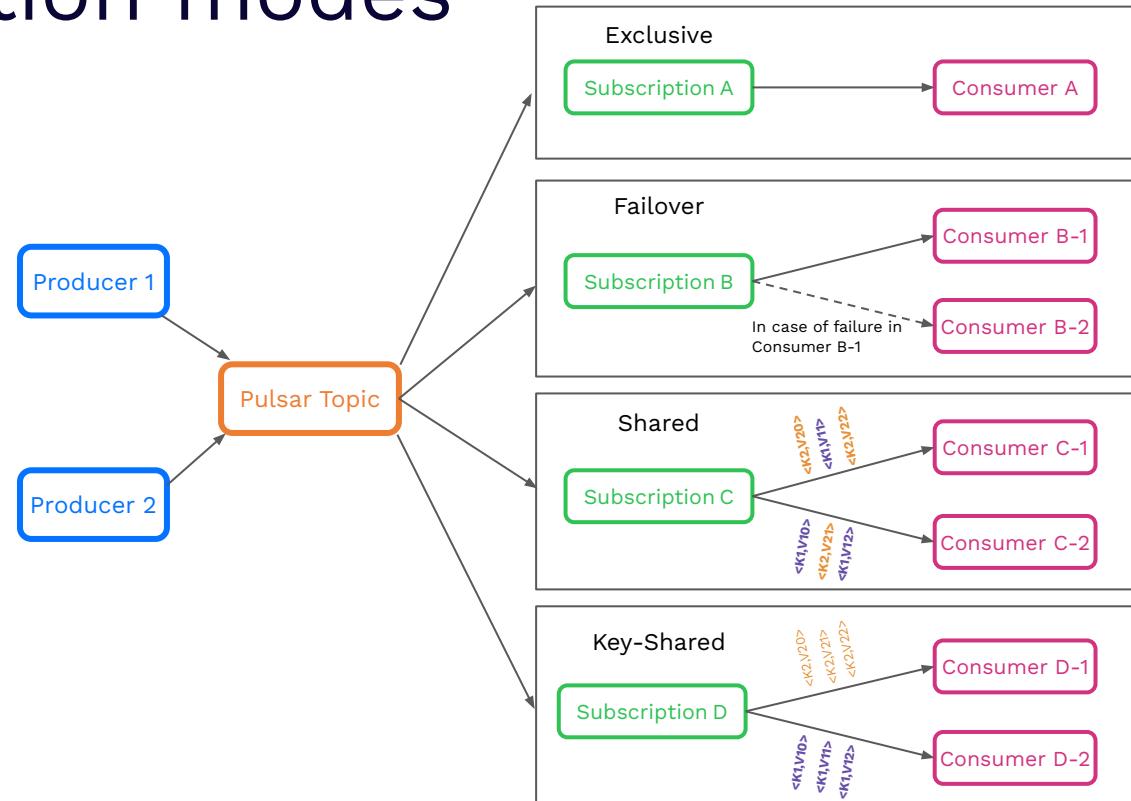
Pulsar subscription modes

Different subscription modes have different semantics:

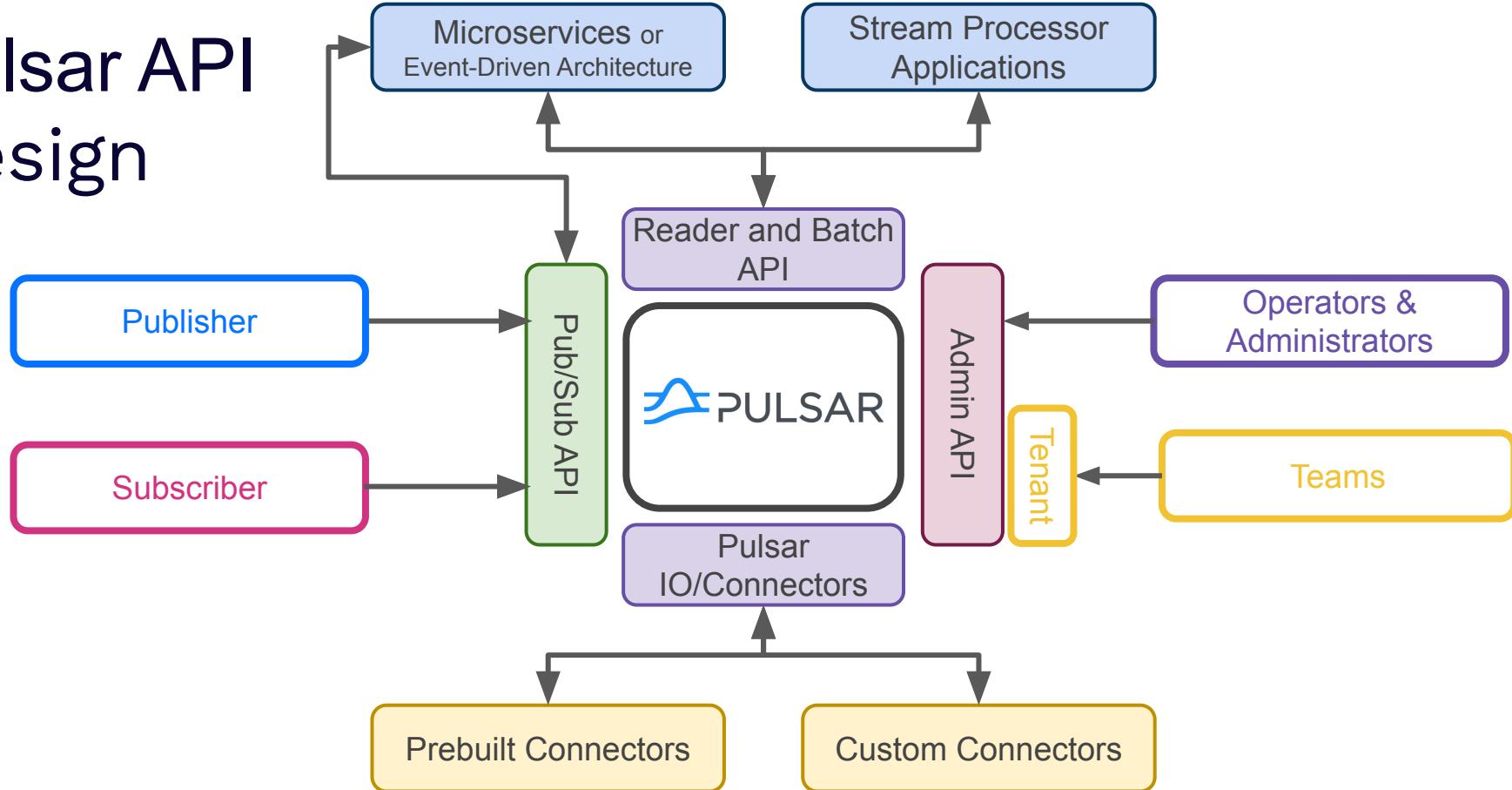
Exclusive/Failover - guaranteed order, single active consumer

Shared - multiple active consumers, no order

Key_Shared - multiple active consumers, order for given key



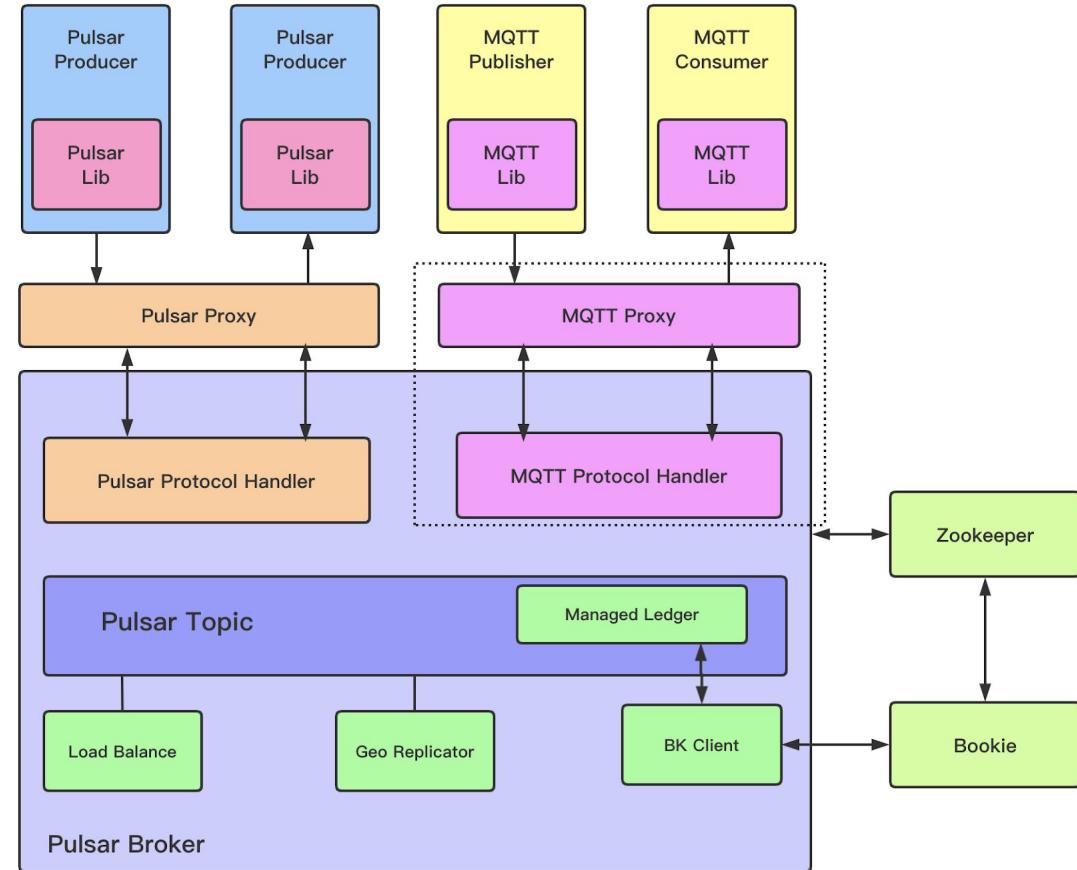
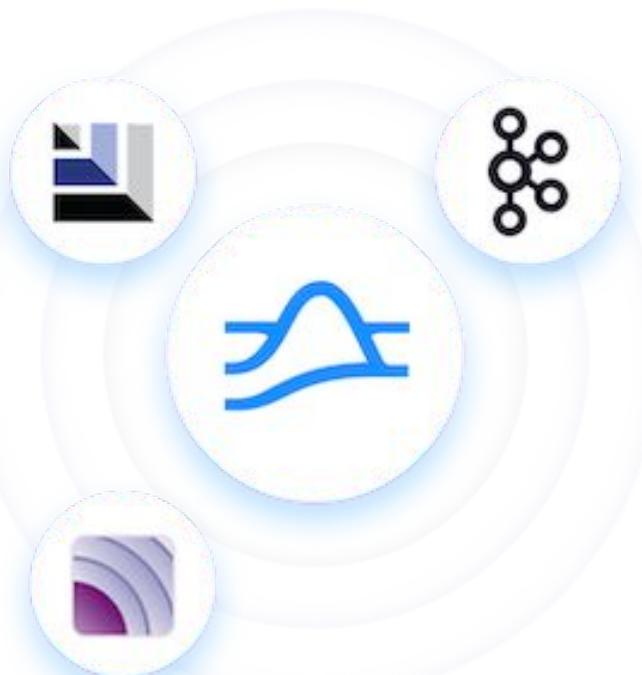
Pulsar API Design



What is the Pulsar Ecosystem? (cont'd)

- **Processing Engines**
 - Supports modern processing engines
 - Flink and Spark, as well as Pulsar SQL (Presto/Trino)
- **Offloaders**
 - Allows data to be offloaded to cloud storage and used with existing Pulsar APIs
 - S3, GCP Cloud Storage, HDFS, File (NFS), Azure Blob Storage (in Pulsar 2.7.0)

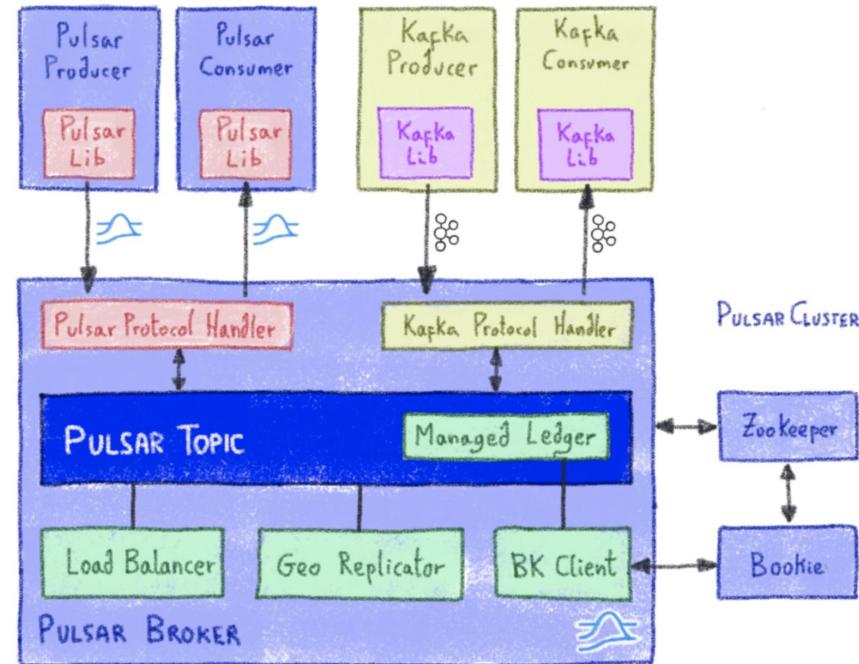
MQTT on Pulsar (MoP)



MQTT on Pulsar (MoP) Configuration

```
messagingProtocols= mqtt  
  
# directory  
protocolHandlerDirectory=./protocols  
  
#mqtt 3.1.1 - port / ip  
mqttListeners=mqtt://127.0.0.1:1883  
advertisedAddress=127.0.0.1
```

Kafka-on-Pulsar (Kop)



Pulsar Functions

Provides a simple API to:

- Receive a message (consume)
- Process the message using your own code
- Send a message (produce)

Takes care of the boilerplate code so there is no need to create producers and consumers.

Moving Data In and Out of Pulsar

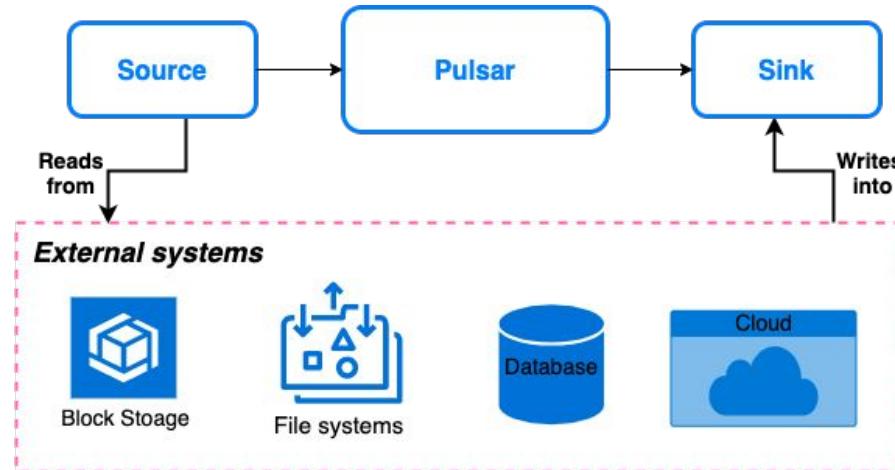
IO/Connectors are a simple way to integrate with external systems and move data in and out of Pulsar.

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



Use Azure BlobStore offloader with Pulsar

<https://pulsar.apache.org/docs/en/tiered-storage-azure/>





Apache Pulsar - Other Sinks



mongoDB



AWS Lambda



redis



AWS S3



GCS

Ingesting IoT Data via Java Pulsar

```
UUID uuidKey = UUID.randomUUID();
String pulsarKey = uuidKey.toString();
String OS = System.getProperty("os.name").toLowerCase();
String message = "" + jct.message;
IoTMessage iotMessage = parseMessage("") + jct.message);
String topic = DEFAULT_TOPIC;
if ( jct.topic != null && jct.topic.trim().length()>0) {
    topic = jct.topic.trim();
}
ProducerBuilder<IoTMessage> producerBuilder = client.newProducer(JSONSchema.of(IoTMessage.class))
    .topic(topic)
    .producerName("jetson")
    .sendTimeout(5, TimeUnit.SECONDS);

Producer<IoTMessage> producer = producerBuilder.create();

MessageId msgID = producer.newMessage()
    .key(iotMessage.getUuid())
    .value(iotMessage)
    .send();
```

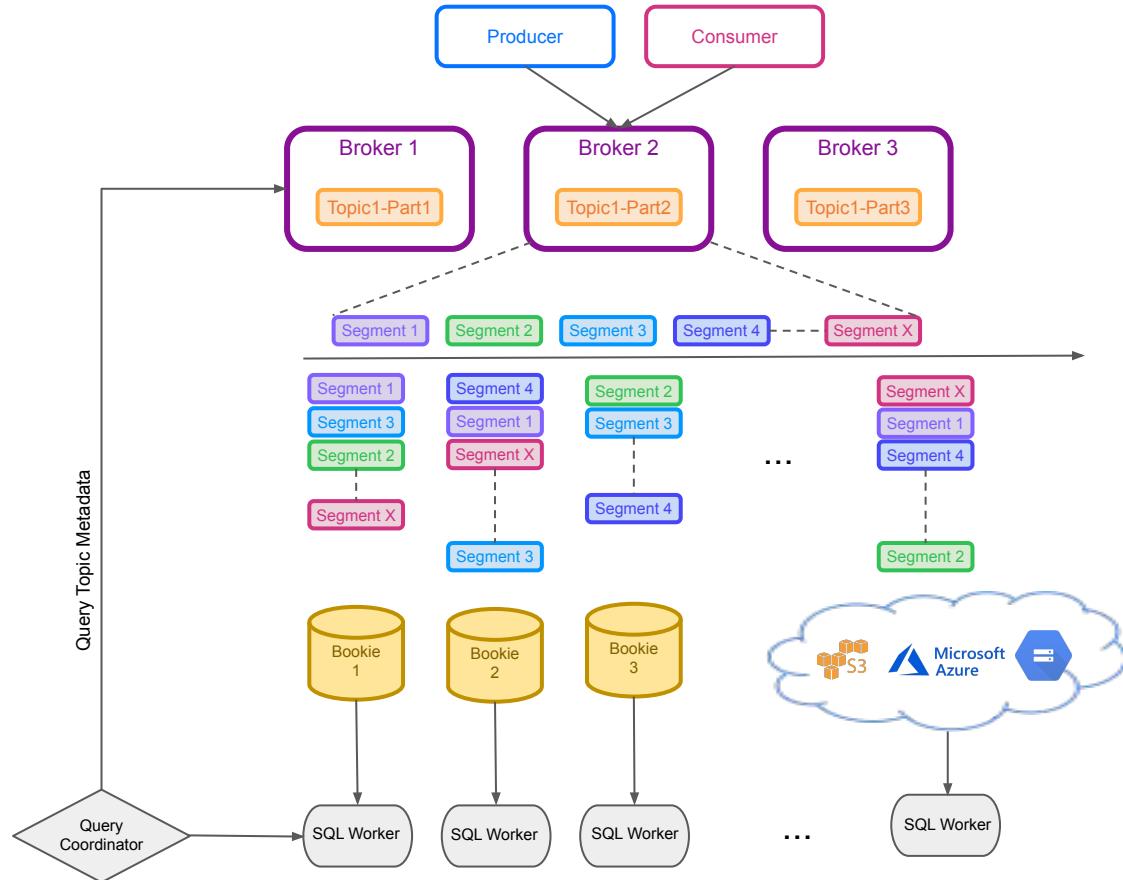
<https://github.com/tspannhw/StreamingAnalyticsUsingFlinkSQL/>

Ingesting IoT Data via Java Pulsar

```
private static IoTMessage parseMessage(String message) {  
    IoTMessage iotMessage = null;  
  
    try {  
        if ( message != null && message.trim().length() > 0 ) {  
            ObjectMapper mapper = new ObjectMapper();  
            iotMessage = mapper.readValue(message, IoTMessage.class);  
            mapper = null;  
        }  
    }  
    catch(Throwable t) {  
        t.printStackTrace();  
    }  
  
    if (iotMessage == null) {  
        iotMessage = new IoTMessage();  
    }  
    return iotMessage;  
}
```

Pulsar SQL

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



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__ | __key__
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
/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]
```

StreamNative Cloud



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



kubernetes

Built for Containers



Flink

Flink SQL



StreamNative

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

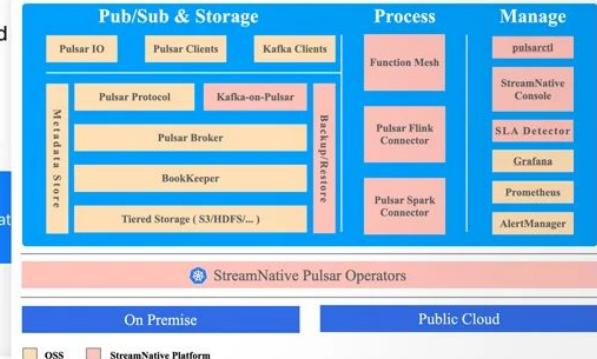
Built for Kubernetes. Made for the cloud. Enables multi-cloud and hybrid

Take A Tour

Contact Sales

Product Update

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



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

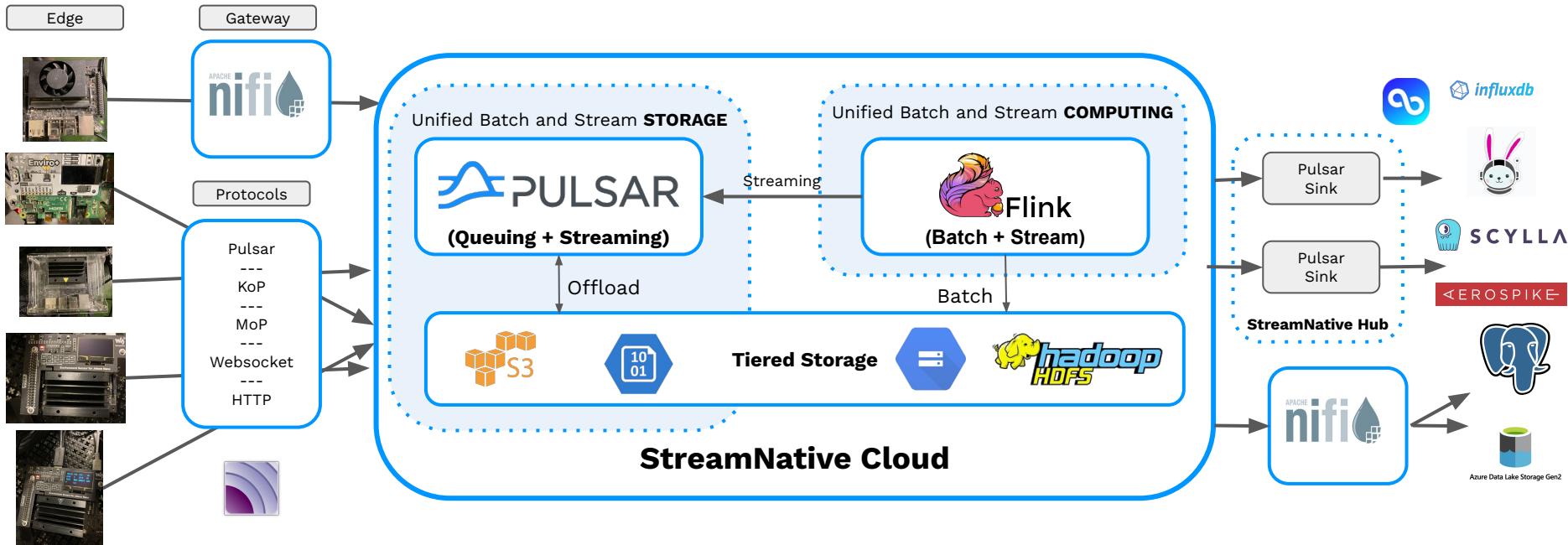


StreamNative

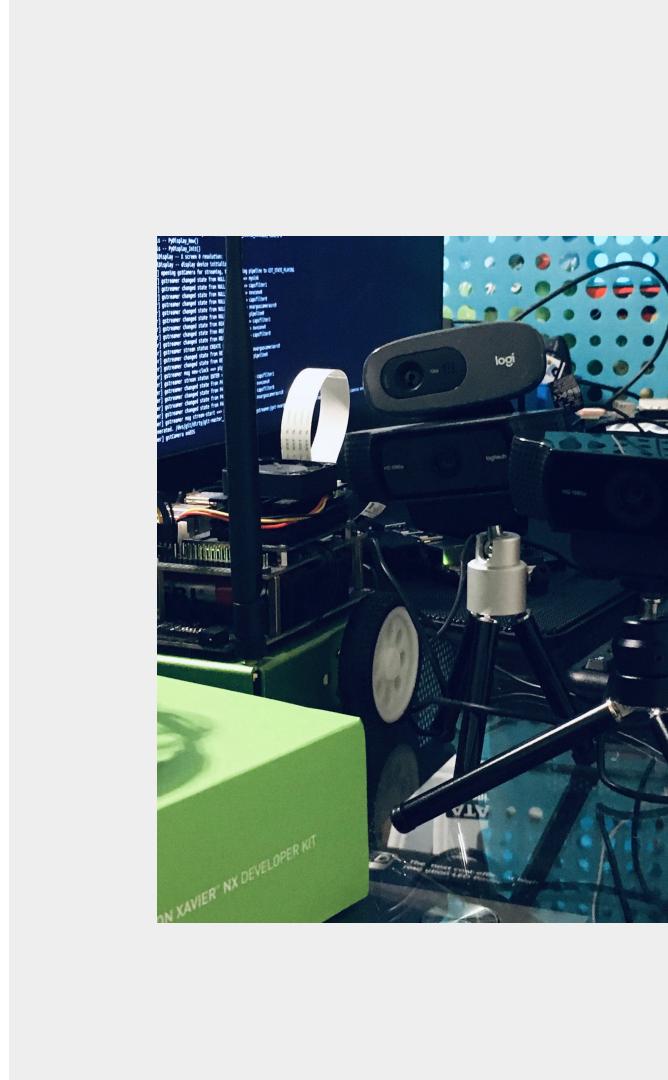
Best Practice Architectures

End-to-End Streaming FLiP(N) Apps

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



Demo



NVIDIA Device



MQTT from Python

```
pip3 install paho-mqtt
```

```
import paho.mqtt.client as mqtt
client = mqtt.Client("rpi4-iot")

row = { }
row['gasKO'] = str(readings)
json_string = json.dumps(row)
json_string = json_string.strip()
client.connect("pulsar-server.com", 1883, 180)
client.publish("persistent://public/default/mqtt-2", payload=json_string,
               qos=0, retain=True)
```



Using NVIDIA Jetson Devices With Pulsar

<https://dev.to/tspannhw/unboxing-the-most-amazing-edge-ai-device-part-1-of-3-nvidia-jetson-xavier-nx-595k>

<https://github.com/tspannhw/minifi-xaviernx/>

<https://github.com/tspannhw/minifi-jetson-nano>

<https://github.com/tspannhw/Flip-iot>

<https://www.datainmotion.dev/2020/10/flank-streaming-edgeai-on-new-nvidia.html>

<https://github.com/tspannhw/FLiP-Mobile/blob/30bcc1ec98fc31e039b51a06180d98545c1e0542/python3/enviro.py>



Demo Walkthrough



```
Flink SQL> use catalog pulsarcatalog;
[INFO] Execute statement succeed.

Flink SQL> show tables;
+-----+
| table name |
+-----+
| # |
| click_events |
| hello_world |
| kafka-1 |
| kafka-2 |
| kafka-3 |
| kafka-4 |
| kafka-5 |
| mqtt-1 |
| mqtt-2 |
| mqtt-3 |
| mqtt-4 |
| mqtt-5 |
| mqtt-go |
| mqtt-mac |
| mqtt-nifi |
| mqtt-nvidia |
| mqtt-python |
| mqtt-rp4 |
| my-topic |
| nvidia-kafka-1 |
| rp4-kafka-1 |
| rwar |
+-----+
23 rows in set

Flink SQL> |
```

```
{"entriesAddedCounter":1,"numberOfEntries":1,"totalSize":651,"currentLedgerEntries":1,"currentLedgerSize":651,"lastLedgerCreatedTimestamp":"2021-09-13T16:13:06.6-04:00","waiting CursorsCount":0,"pendingAddEntriesCount":0,"lastConfirmedEntry":"7076:0","state":"LedgerOpened","ledgers":[{"ledgerId":7076,"entries":0,"size":0,"offloaded":false,"underReplicated":false}],"cursors":{},"schema Ledgers":[],"compactedLedger":{"ledgerId":-1,"entries":-1,"size":-1,"offloaded":false,"underReplicated":false}}
```

Wrap-Up

Connect with the Community & Stay Up-To-Date

- Join the Pulsar Slack channel - Apache-Pulsar.slack.com
- Follow [@streamnativeio](https://twitter.com/streamnativeio) and [@apache_pulsar](https://twitter.com/apache_pulsar) on Twitter
- [Subscribe](#) to Monthly Pulsar Newsletter for major news, events, project updates, and resources in the Pulsar community

Deeper Content

- <https://github.com/tspannhw/FLiP-Energy/>
- <https://github.com/tspannhw/Flip-iot>
- <https://github.com/tspannhw/Flip-jetson>
- <https://github.com/streamnative/pulsar-flink>
- <https://github.com/streamnative/mop>
- <https://www.linkedin.com/pulse/2021-schedule-tim-spann/>
- <https://github.com/tspannhw/FLiP-InfluxDB/blob/main/README.md>
- <https://streamnative.io/en/blog/release/2021-04-20-flink-sql-on-streamnative-cloud>
- <https://docs.streamnative.io/cloud/stable/compute/flink-sql>



@PaasDev



timothyspann

<https://www.pulsardeveloper.com/>

Interested In Learning More?



Resources

[Flink SQL Cookbook](#)

[The Github Source for Flink SQL Demo](#)

[The GitHub Source for Demo](#)



Free eBooks

[Manning's Apache Pulsar in Action](#)

[O'Reilly Book](#)



Upcoming Events

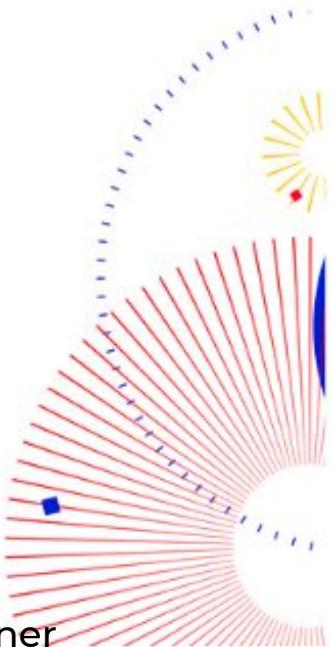
[\[10/21\] Trino Summit](#)



Pulsar Summit Asia

November 20-21, 2021

Contact us at partners@pulsar-summit.org to become a sponsor or partner



Wednesday, October 27, 2021

9:00 AM - 9:25 AM PDT

AI DEVWORLD

AI OPEN TALKS

AI FOR THE ENTERPRISE

Add

OPEN TALK (AI): Utilizing Apache  Pulsar, Apache NiFi and MiNiFi for EdgeAI IoT at Scale



Timothy Spann

StreamNative, Developer Advocate

Thursday, October 28, 2021

9:00 AM - 9:25 AM PDT

API INNOVATION

EMERGING APIs

Add

PRO TALK (API): Apache NiFi 101: Introduction and Best Practices



Timothy Spann

StreamNative, Developer Advocate

Let's Keep in Touch!



Speaker Name

Speaker title



@PassDev



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



<https://github.com/tspannhw>

Questions



Thank you to our Sponsors

Event Partners

Goldman
Sachs

NC NEUMONT.[®]

College of Computer Science

Premier Sponsors

AURISTOR

THE GLOBAL NAMESPACE FILE SYSTEM

stg

stgconsulting.com

dwelô