



**Stream
Native**

Building Real-Time Pulsar Apps on K8

Tim Spann | Developer Advocate



Tim Spann
Developer Advocate

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, Flink, Spark, NiFi, Big Data, Cloud, MXNet, 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.



CLUDERA



Pivotal

BARNES
& NOBLE



Hewlett Packard
Enterprise

FLiP Stack Weekly



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



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



Stream Native

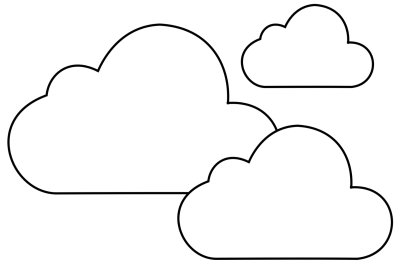
Founded by the original developers of Apache Pulsar.

Passionate and dedicated team.

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

streamnative.io

Apache Pulsar - Built for Containers / Modern Cloud

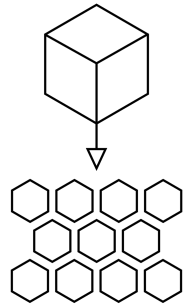


Hybrid & Multi-Cloud

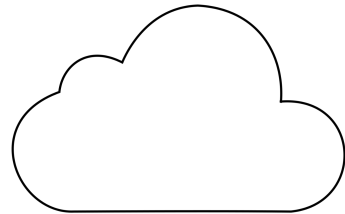


kubernetes

Containers



Microservices



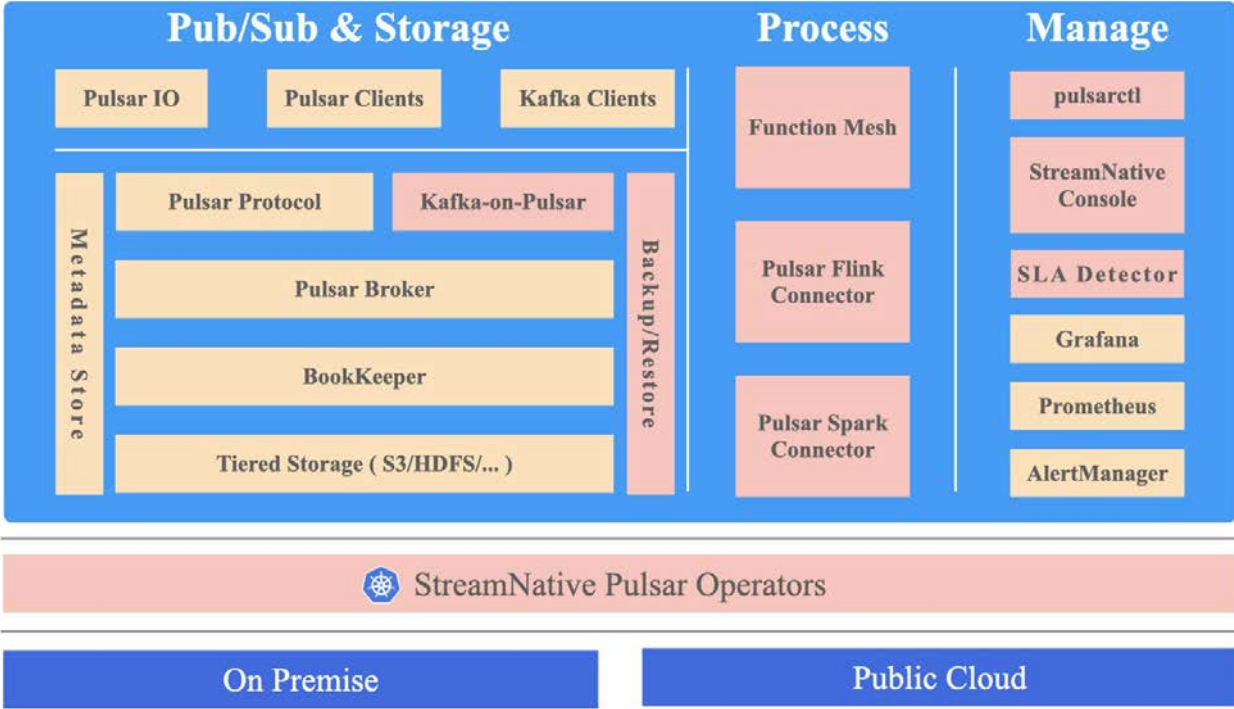
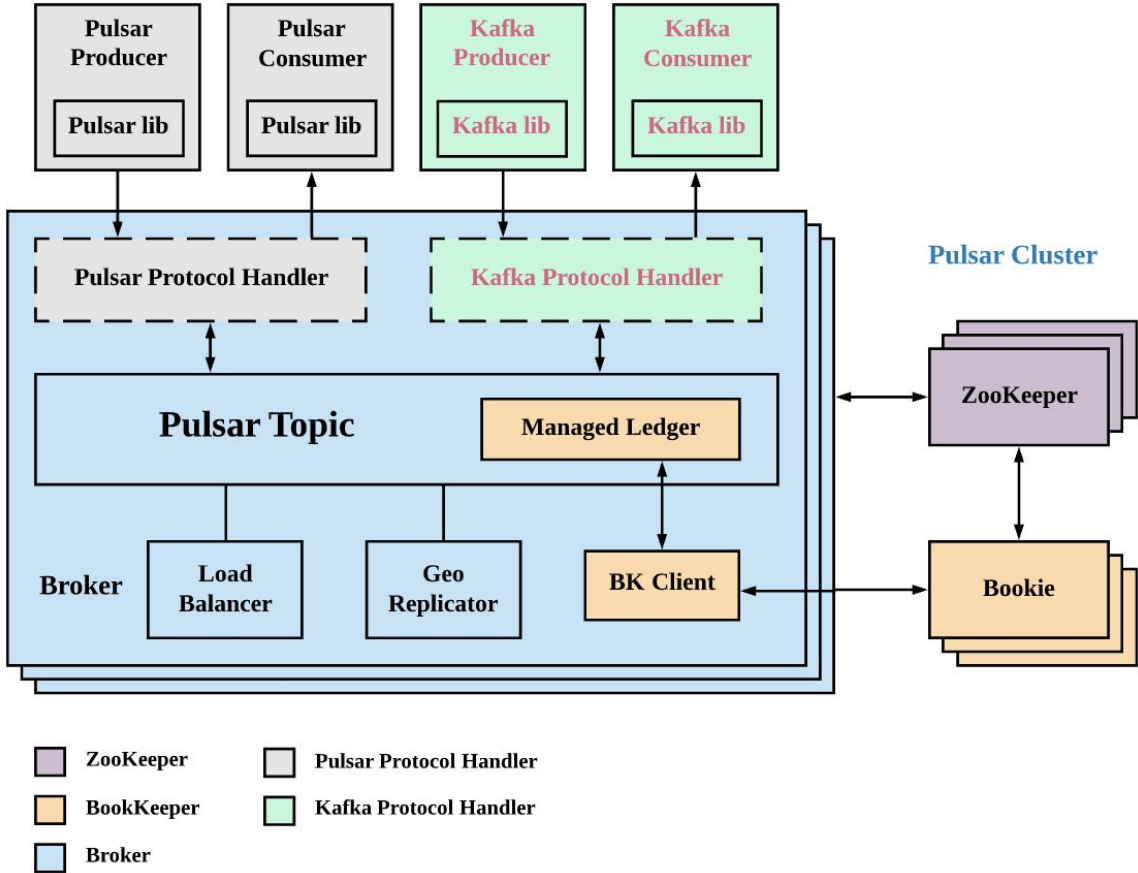
Cloud Native

Apache Pulsar adoption is being driven by organizations seeking cloud-native architectures and new uses cases.



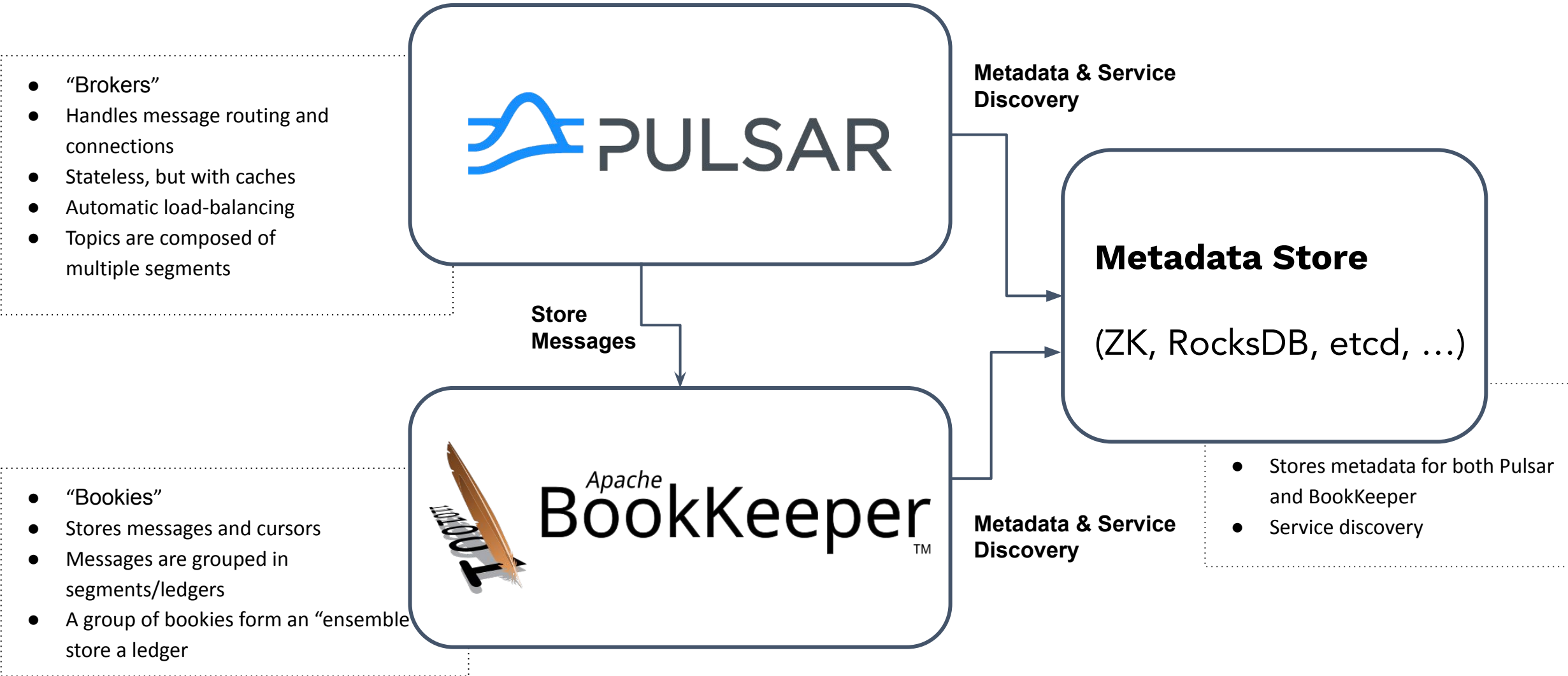
Apache Pulsar + Kafka K8

<https://docs.streamnative.io/platform/v1.3.0/quickstart>

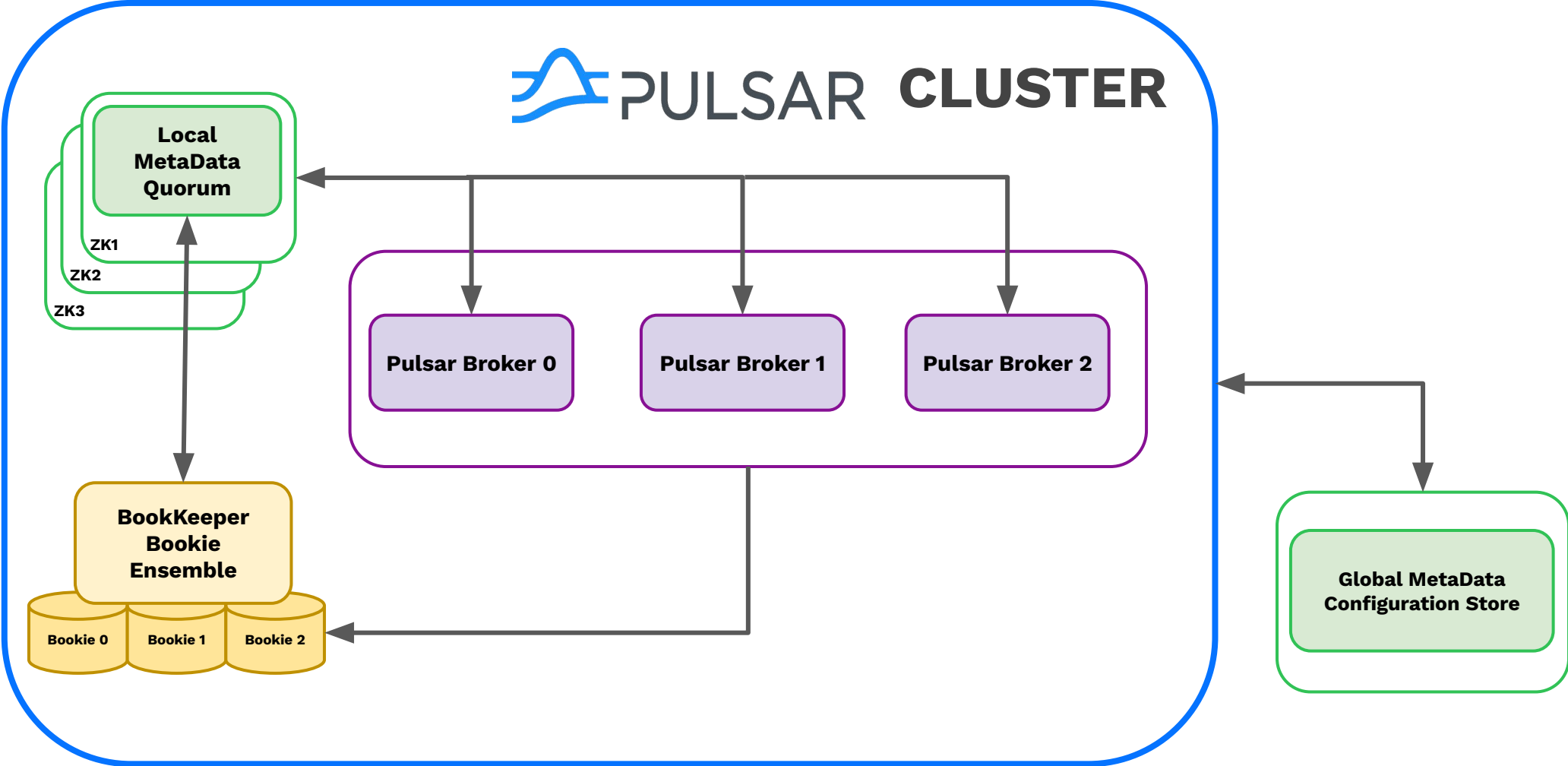


Legend: OSS (Orange), StreamNative Platform (Red)

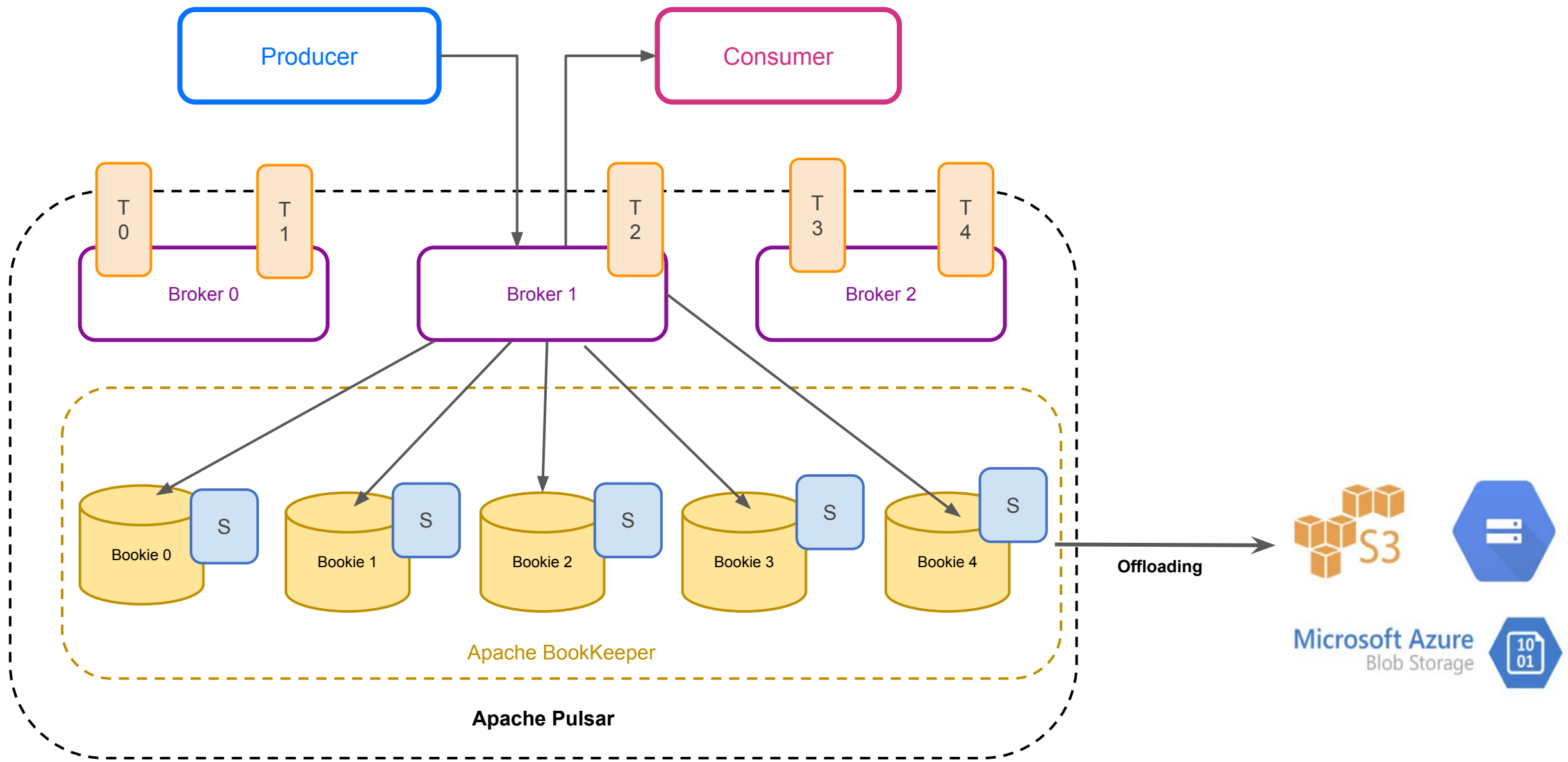
Pulsar Cluster



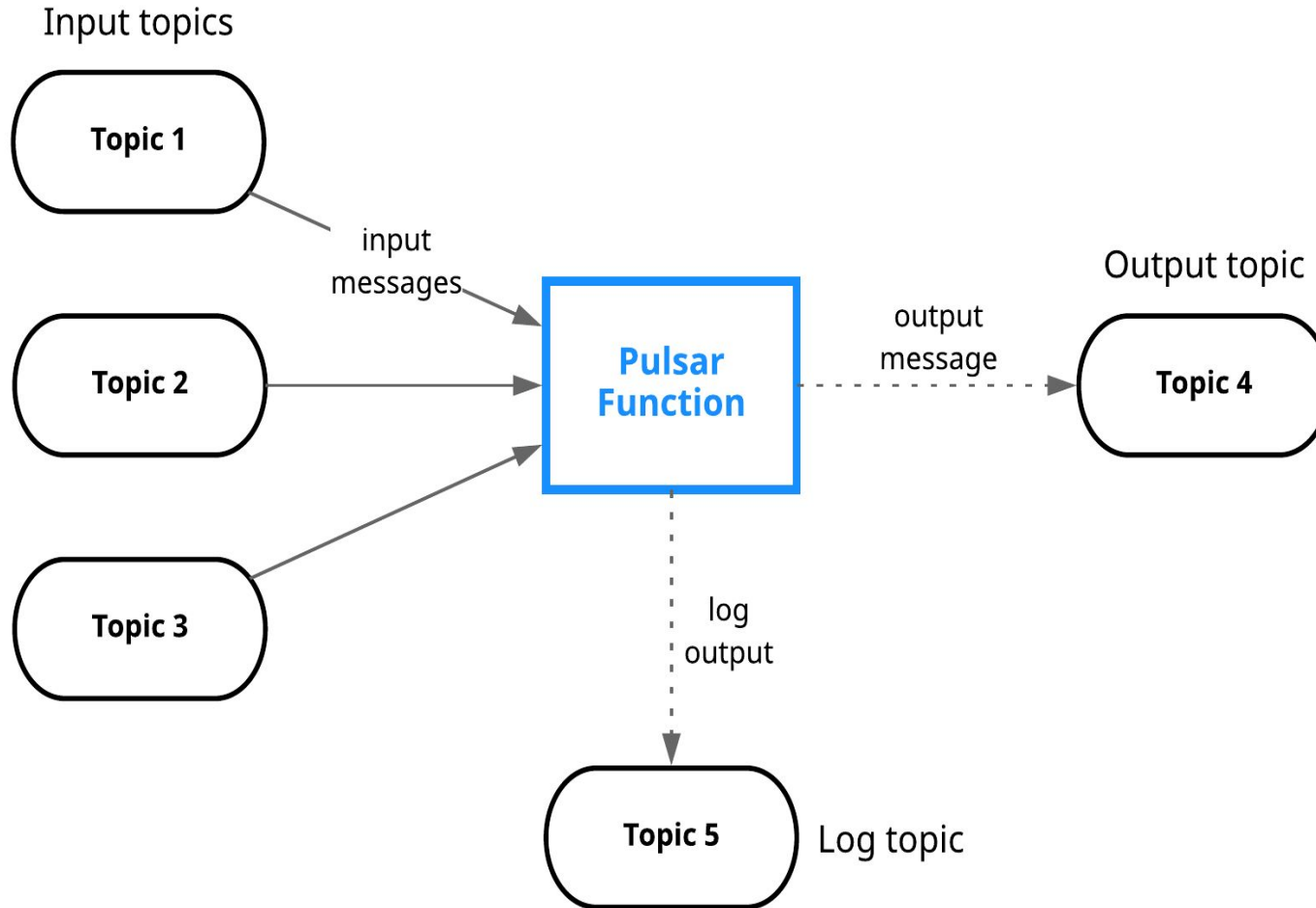
Pulsar Cluster



Offloader & Tiered Storage



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

Pulsar Python NLP Function

Entire Function



```
from pulsar import Function
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import json

class Chat(Function):
    def __init__(self):
        pass

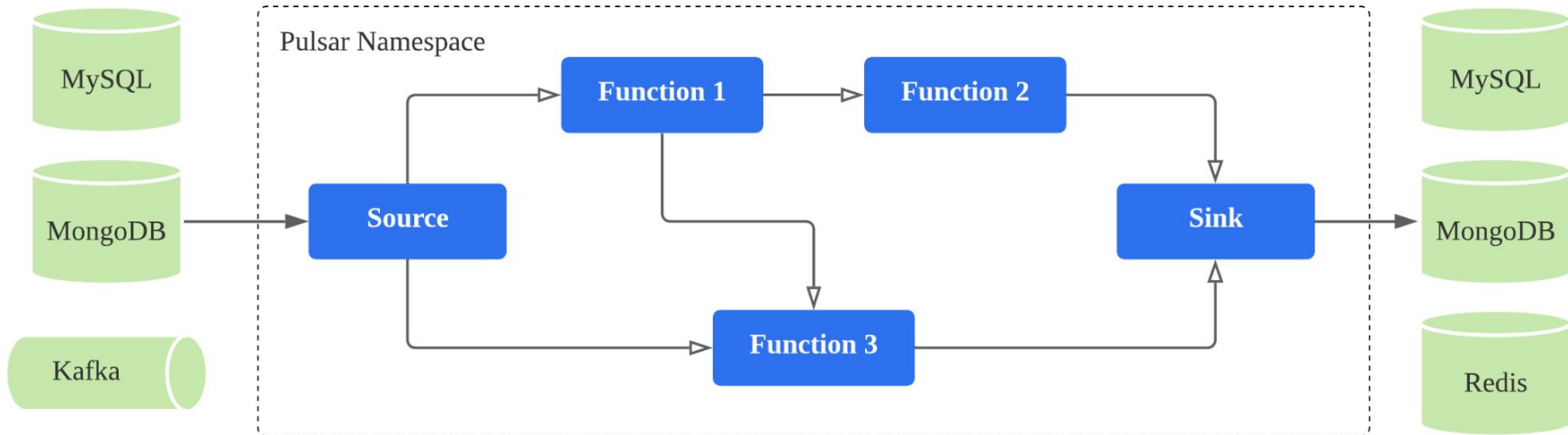
    def process(self, input, context):
        fields = json.loads(input)
        sid = SentimentIntensityAnalyzer()
        ss = sid.polarity_scores(fields["comment"])
        row = { }
        row['id'] = str(msg_id)
        if ss['compound'] < 0.00:
            row['sentiment'] = 'Negative'
        else:
            row['sentiment'] = 'Positive'
        row['comment'] = str(fields["comment"])
        json_string = json.dumps(row)
        return json_string
```

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

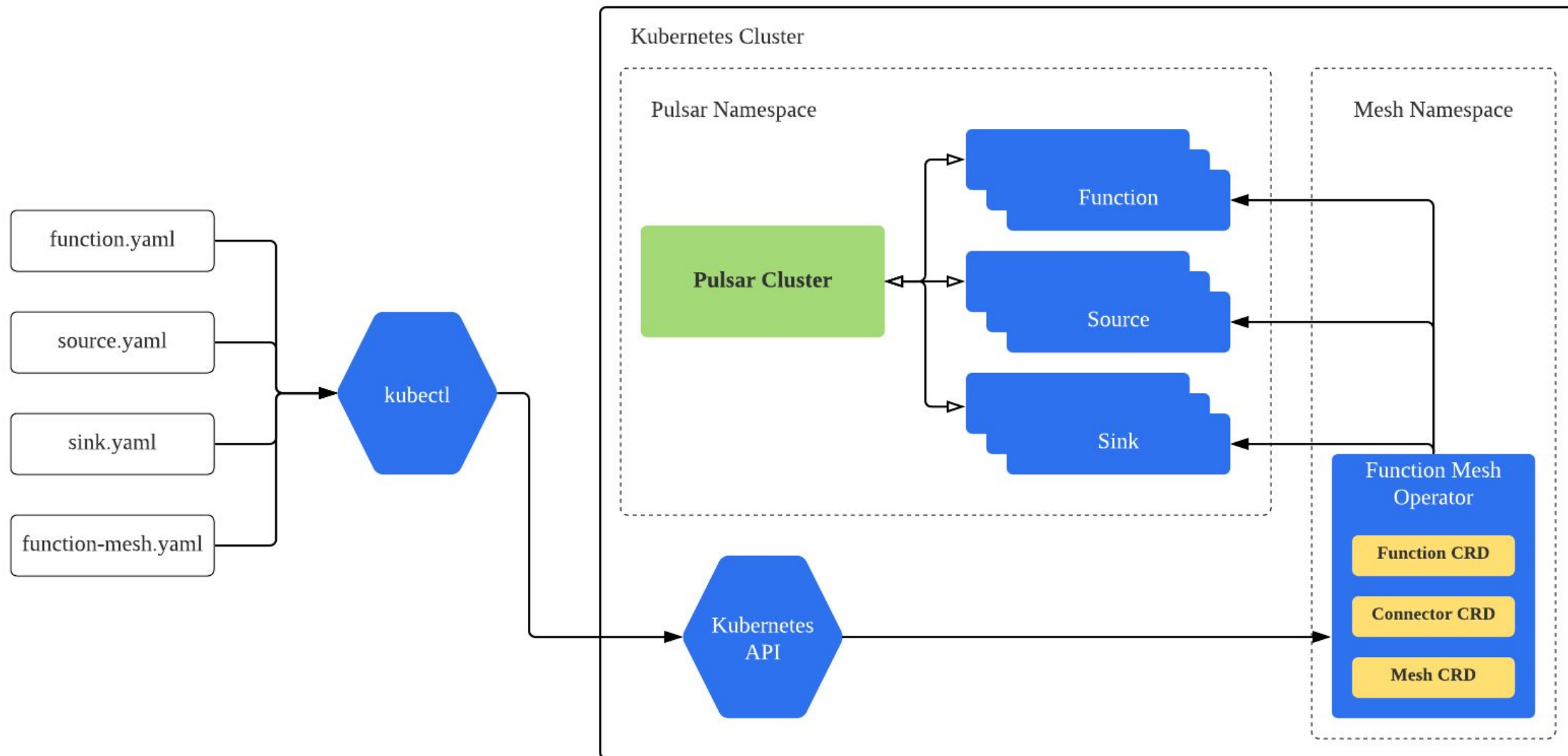
Function Mesh

for Pulsar Functions

OPEN SOURCED BY  Stream Native
BUILT FOR **kubernetes**



K8 Deploy



Apache Pulsar Resources



<https://github.com/tspannhw/FLiPN-Conf42-KubeNative-2022>

