Build ML Enhanced Event Streaming Apps with Java

David Kjerrumgaard
Developer Advocate

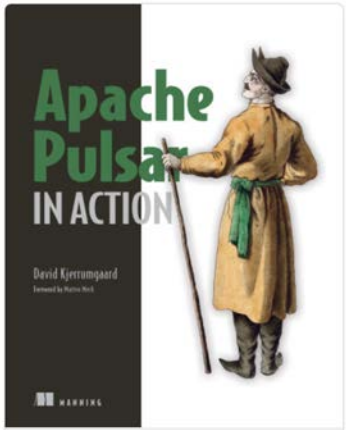
- Over two decades of experience in software development, big data, and event streaming.
- Committer on the Apache Pulsar project
- Published author

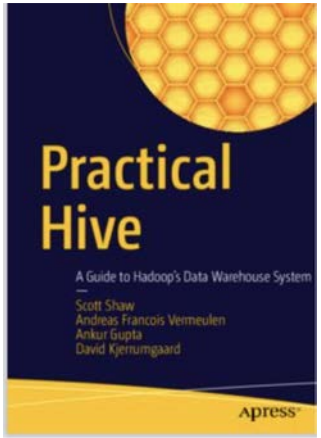
amazon FedEx splunk> HORTONWORKS LOCKHEED MARTIN Zappos.com

StreamNative

David Kjerrumgaard
Author

- Author of **Pulsar In Action.**
- Co-Author of ***Practical Hive***





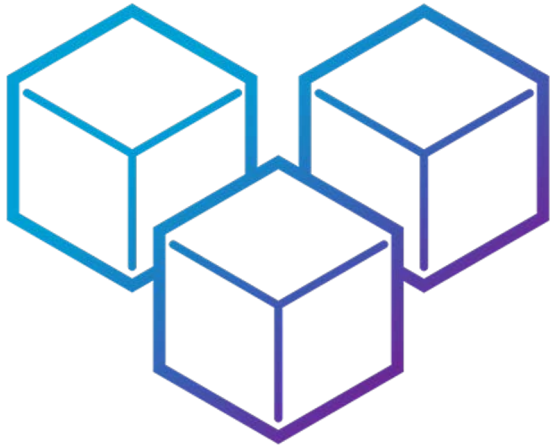
https://streamnative.io/download/manning-ebook-apache-pulsar-in-action

StreamNative

Agenda

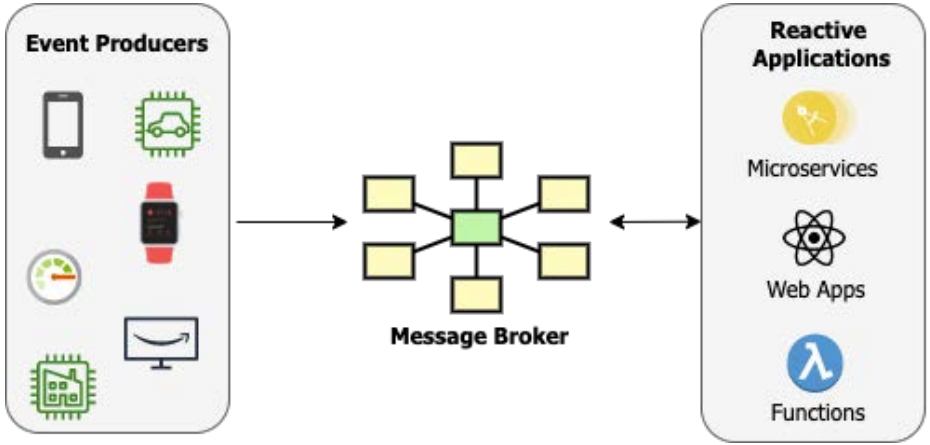
- Why Apache Pulsar for EDA Microservices?
- Building Event-Driven Microservices with Pulsar Functions
- Enhancing Event-Driven Microservice with ML

Event-Driven Microservices



What is an Event-Driven Architecture?

- An event-driven architecture (EDA) uses events to communicate between decoupled services.
- An event is a change in state, or an update, like an item being placed in a shopping cart on an e-commerce website.
- In an EDA services are loosely coupled and communicate asynchronously, typically via pub/sub.

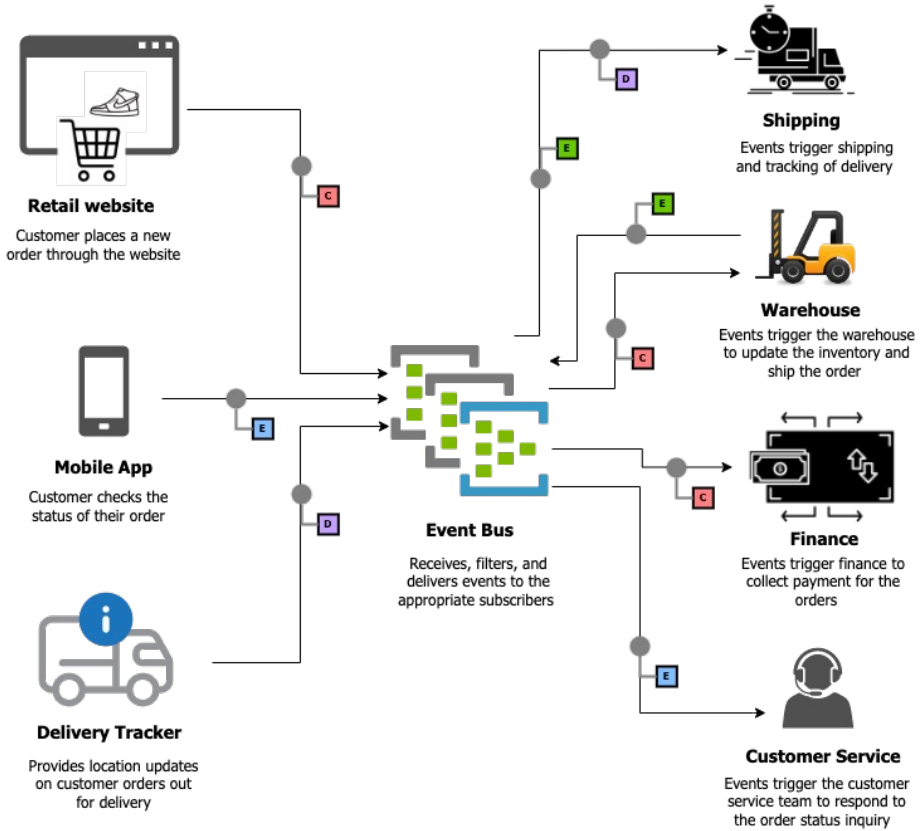


StreamNative

Event-Driven Microservices

- Microservices that are designed to communicate with one another over a message bus are considered event-driven.

Event-Driven Microservices ***NEED*** an Event Bus



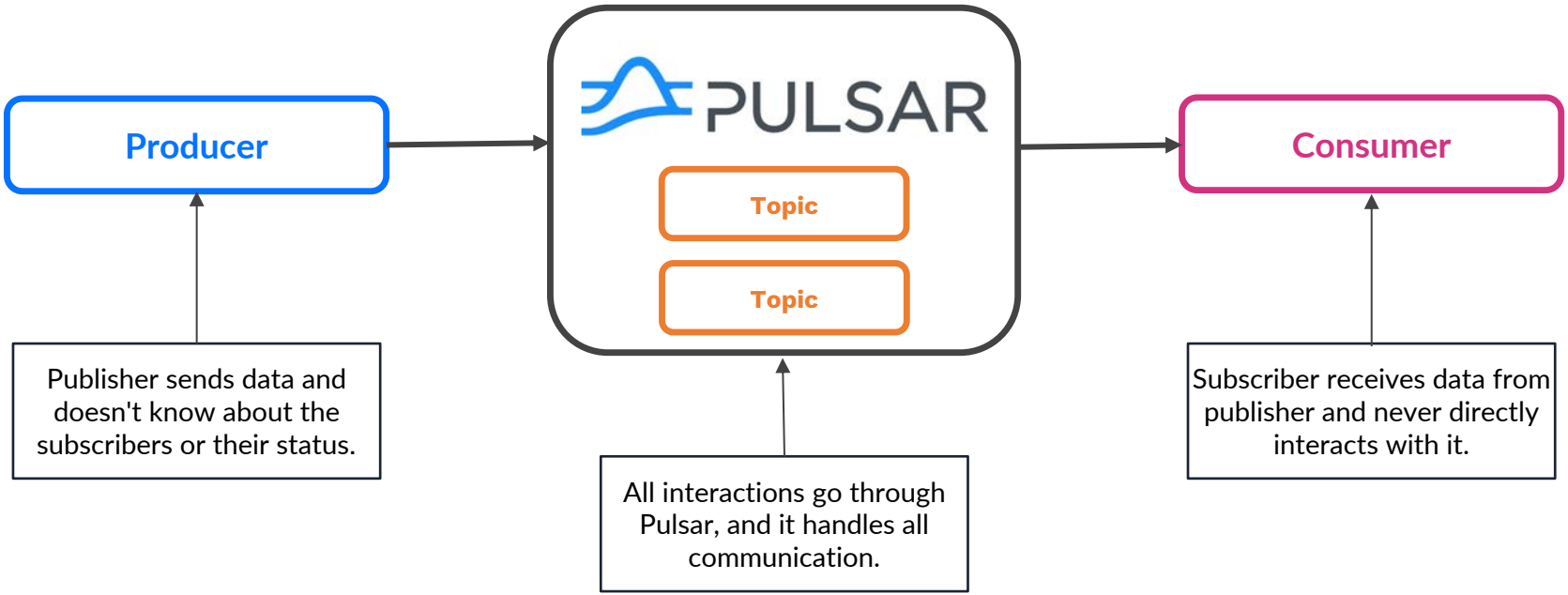
StreamNative

What is Apache Pulsar?

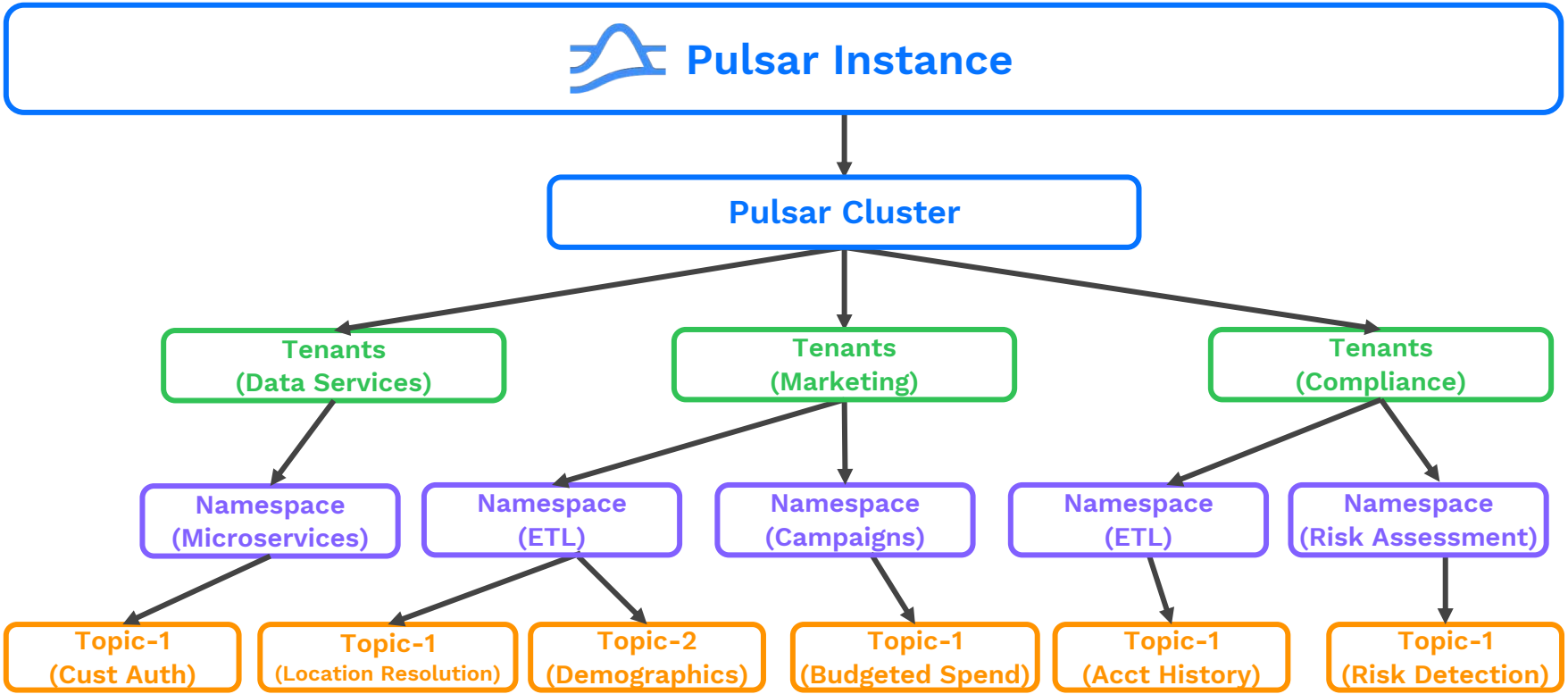
Cloud-Native Messaging and Event-Streaming Platform

StreamNative

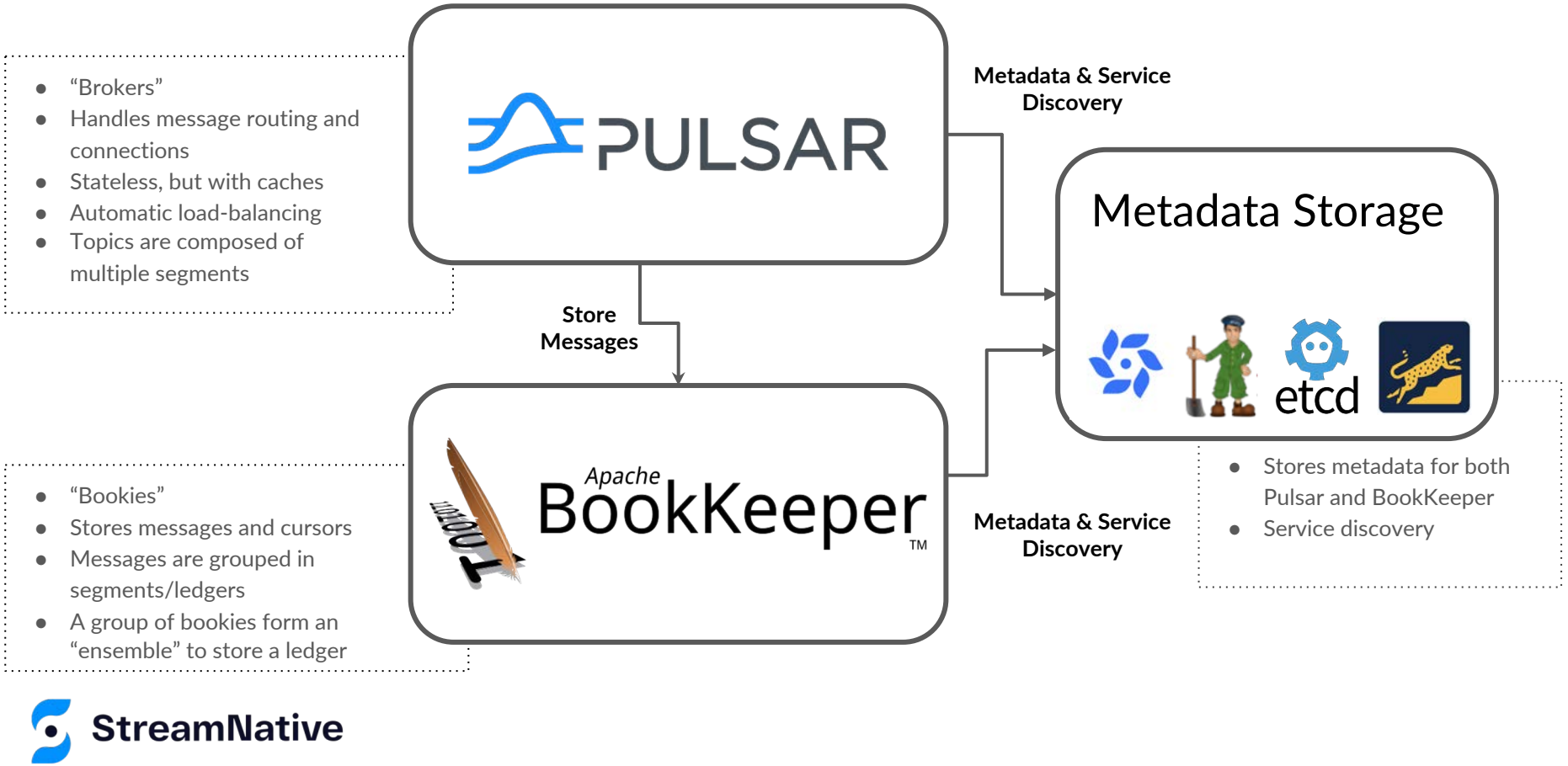
Pulsar Pub/Sub Model



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)

Physical Architecture of a Pulsar Cluster



StreamNative

What are Pulsar Functions?

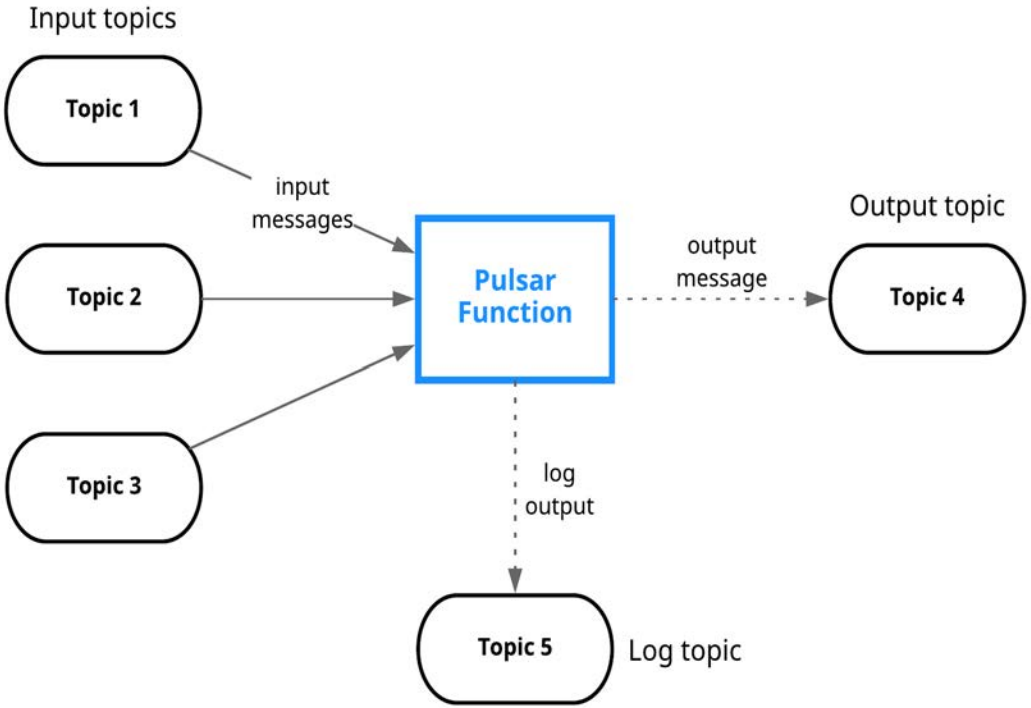
What are Pulsar Functions?

- Pulsar Functions are a serverless computing framework that runs on top of Pulsar. You can think of them as like FaaS (Functions as a Service) offerings on cloud providers, e.g., AWS Lambdas.

- Pulsar functions take care of the boilerplate code, so you don't have to create producers and consumers. You can focus on the business problem instead of the code.

- Pulsar Functions provide a way to run individual units of code that respond to the publication of messages on one or more topics.

Pulsar Functions Programming Model



- 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***.

StreamNative

Why Pulsar Functions?

Functions are the computing infrastructure of the Pulsar messaging system and can help ease stream processing complexity by providing:

- Simplified deployment and operations - you can create a data pipeline without deploying a separate Stream Processing Engine (SPE), such as Apache Storm, Apache Heron, or Apache Flink.
- Serverless computing (when you use Kubernetes runtime)
- Maximized developer productivity (both language-native interfaces and SDKs for **Java**/Python/Go).

When to Use Pulsar Functions

Pulsar Functions are designed to perform “lightweight” stream processing. They excel at **basic** use cases that do not require the complexity of a full stream processing engine.

- ***Event Driven Microservices***
- Simple per-message transformations for normalization, cleanup, or enrichment.
- “Chained” sequences of transformations on data in a single Topic.

Developing Pulsar Functions

Pulsar Functions SDK

Apache Pulsar also provides a software development kit (SDK) that you can use to write Pulsar Functions.

- The Pulsar Functions SDK supports Java, Python, and Go.
- The Pulsar Functions SDK provides a richer API for more complex use cases:
  - One to many output
  - Stateful functions
  - Producing to many different topics

Packaging Functions

In order to run a Pulsar Function, you must first bundle up the function code along with all the necessary third-party dependencies into a single deployable artifact.

- For Java-based functions, this is either a JAR or NAR file
- For Python-based functions, you can use a single Python file (.py), a ZIP file, or PIP install (Kubernetes runtime **only**).
- For Go-based functions, a compiled and packaged ***.go*** file

Demo

Sentiment Analysis

git@github.com:david-streamlio/sentiment-analysis.git

Summary

- Event-driven microservices use a message bus to communicate among loosely-coupled, collaborating services.
- Pulsar is a cloud-native, distributed messaging and event-streaming platform that provides pub/sub semantics required by EDA
- Pulsar includes native, lightweight compute capabilities known as Pulsar Functions that allow you to build microservices with a few lines of code.
- You can add third-party ML to your Functions to enhance your microservices with machine learning capabilities.

Let’s Keep in Touch!

David Kjerrumgaard

Developer Advocate

@Dkjerrumg1

https://www.linkedin.com/davidkj

https://github.com/david-streamlio

StreamNative