

# Best of Both Worlds - Apache Pulsar and Apache Kafka



 @KoTurk77

 PULSAR



 **kafka**®



 @mgrygles

**How to  
access this  
slide deck?**

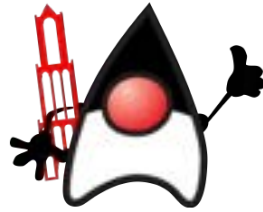
**Github**



**Google**



# Senior Developer and Community Lead



kafka

Rabobank 

blue4IT



**koturk77**



**ko-turk**



**koturk77**

- Conference speaker
  - Kafka (Streams)
  - Pulsar
  - Green coding
- Writer



# Senior Developer Advocate



Java Champion

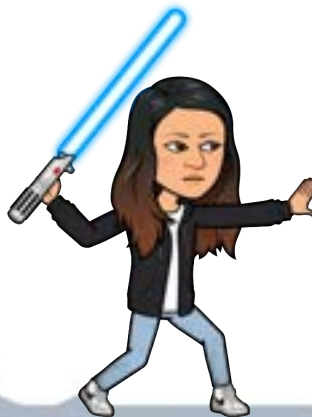


JAKARTA EE



Open Liberty

RED HAT OPENSIFT



- Streaming
- Distributed Systems
- Reactive Systems
- IoT/MQTT



mgrygles



mary-grygleski



mgrygles



mgrygles

## What's in it for me?

- > Choose which streaming framework suits your case
- > Create a basic Pulsar / Kafka app (spring boot)
- > And how to stream like a pro
  - With functions and streams



# Agenda

1. Welcome into a streaming world (Introductions)
2. Streaming architectures
3. Why should you consider both architectures?
4. Code in action (demo → datastax / code)
  - a. Pulsar Functions
  - b. Kafka Streaming API
5. Bridging Apache Pulsar and Apache Kafka together



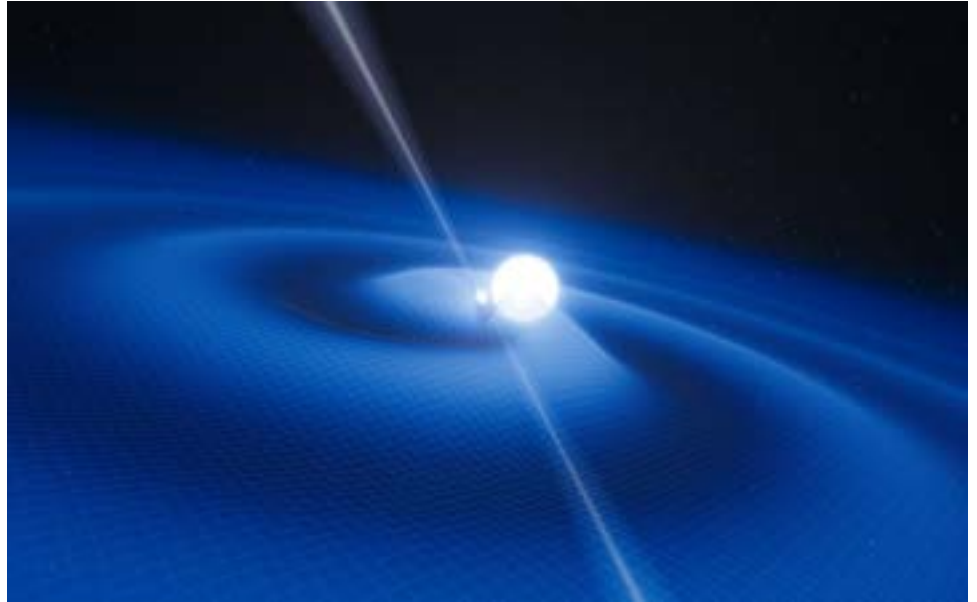
**Welcome into a streaming world**  
**The possibilities are endless**



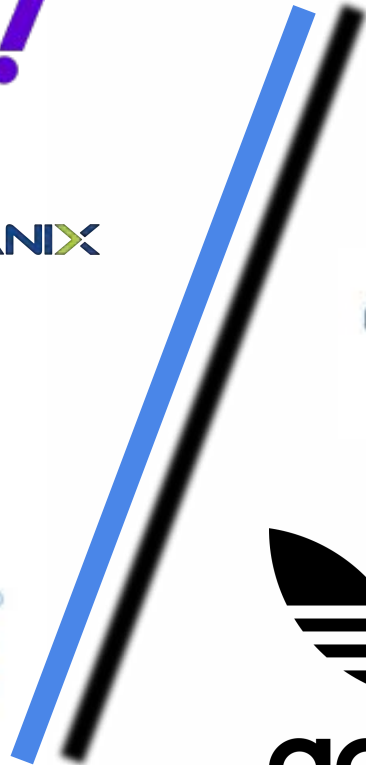




# Kafka or Pulsar ?



# Users of Apache Pulsar / Kafka



# Say 'hi' to Apache Pulsar

## a cloud-native, distributed messaging and streaming platform

Apache Pulsar is an Open Source project born in Yahoo! and then donated to the ASF

<https://github.com/apache/pulsar>

Key points:

- Scalable Storage - Apache BookKeeper
- Multi Tenancy
- Geo Replication
- Tiered-storage message offload
- Native Schema Registry support (Apache AVRO, JSON and Google Protobuf - Pulsar Schema)
- Connectors/Integrations (Pulsar IO framework)
- Message processing (Pulsar Functions framework)
- Multiple Client bindings: Java, C++, Python, Go, NodeJS, C#...

# Say 'hi' to Apache Pulsar



# Apache Pulsar - why ?

**Blazing performance**

**Horizontal scalability and object storage offloading**

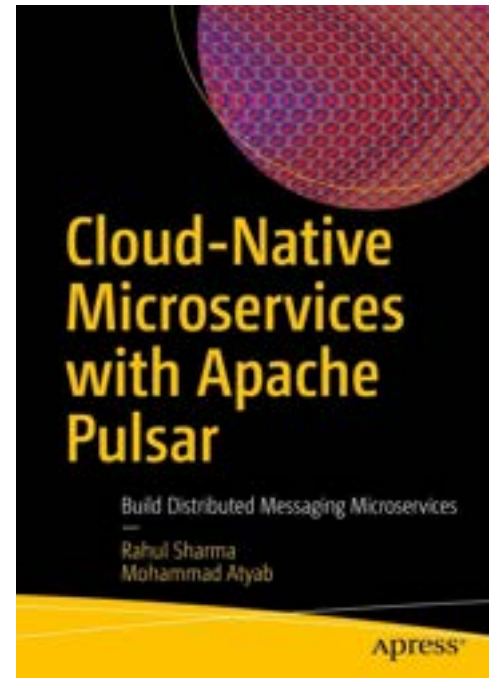
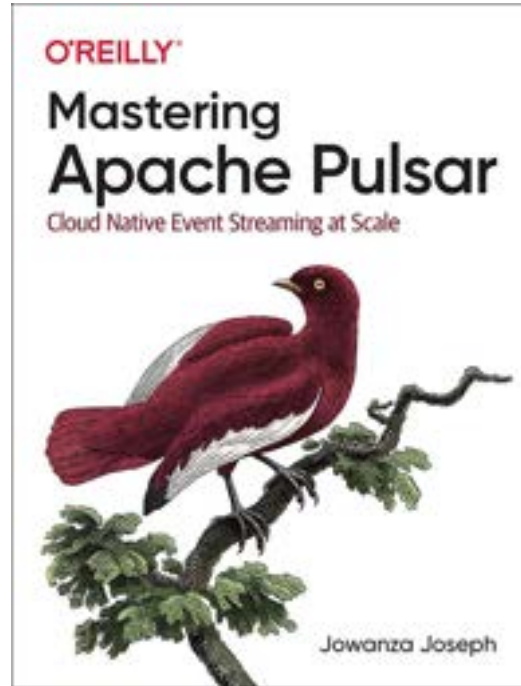
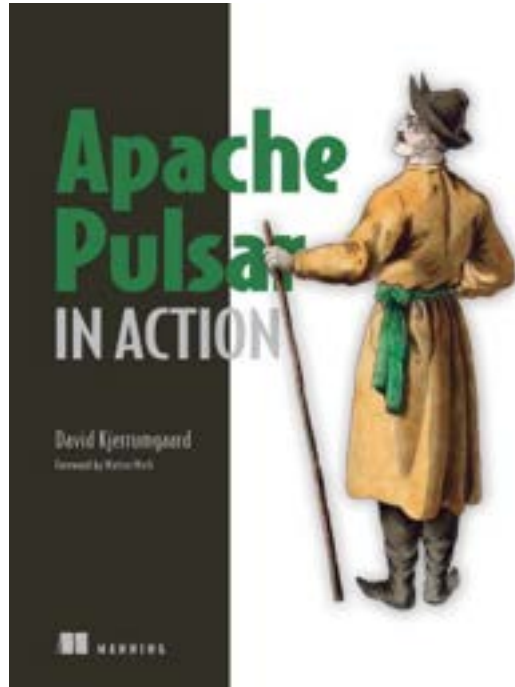
**Consolidation**

**Message replay**

**Geo-replication**

**Future readiness**

# Books Apache Pulsar





# 'Oh hello' Apache Kafka!



A low-angle, upward-looking photograph of a building's exterior. The building has a classic architectural style with large, arched windows on the upper floor, some of which are reflecting the sky and other buildings. Below the windows, a large, vibrant red banner is stretched across the facade. The banner features the words "COMMUNITY IS STRENGTH." in a bold, white, sans-serif font, arranged in three lines. At the very bottom of the banner, in a smaller white font, is the phrase "BE STRONG LET'S LOOK OUT FOR ONE ANOTHER". The sky above is a pale, overcast blue.

**COMMUNITY  
IS  
STRENGTH.**

BE STRONG LET'S LOOK OUT FOR ONE ANOTHER

Tutorial



*Olympia*

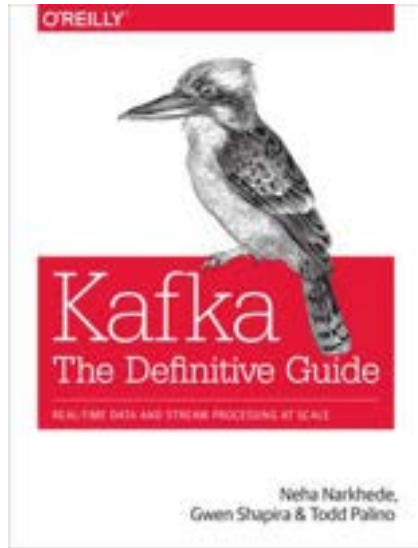
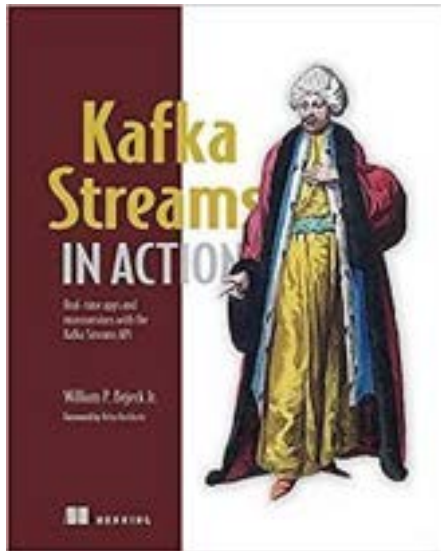


# Apache Kafka!

- **Simplicity** → Simple architecture, simple coding with Kafka Streams
- **Mature ecosystem** → bigger and older community, more integrations with other third party system (Kafka Connect).
- **Performance** → Designed for high throughput and low latency data processing
- **Replayability** → Messages can be replayed if something bad happened
- **Fault tolerant** → continue if something fails → different exception handlers (deserialization / production)



# Books Apache Kafka





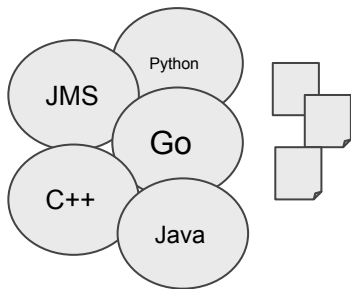


**How does it work?**

**Show me the candy (or architecture)**

# Apache Pulsar - Basic Architecture

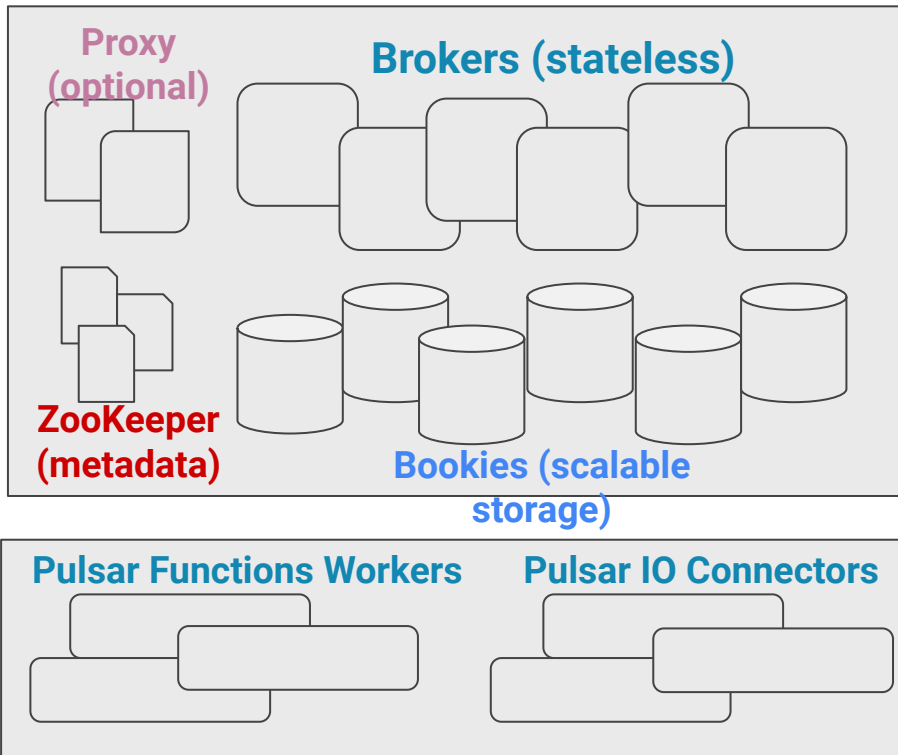
## Producers



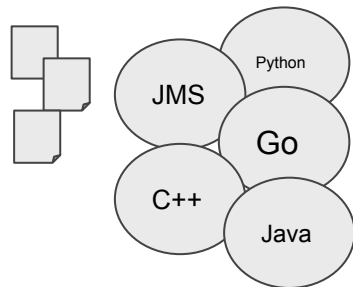
Every component is Horizontally Scalable

Dynamic addition/removal of components without service interruption

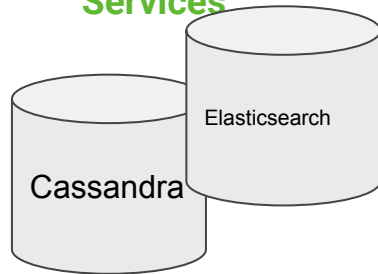
## Object Storage



## Consumers

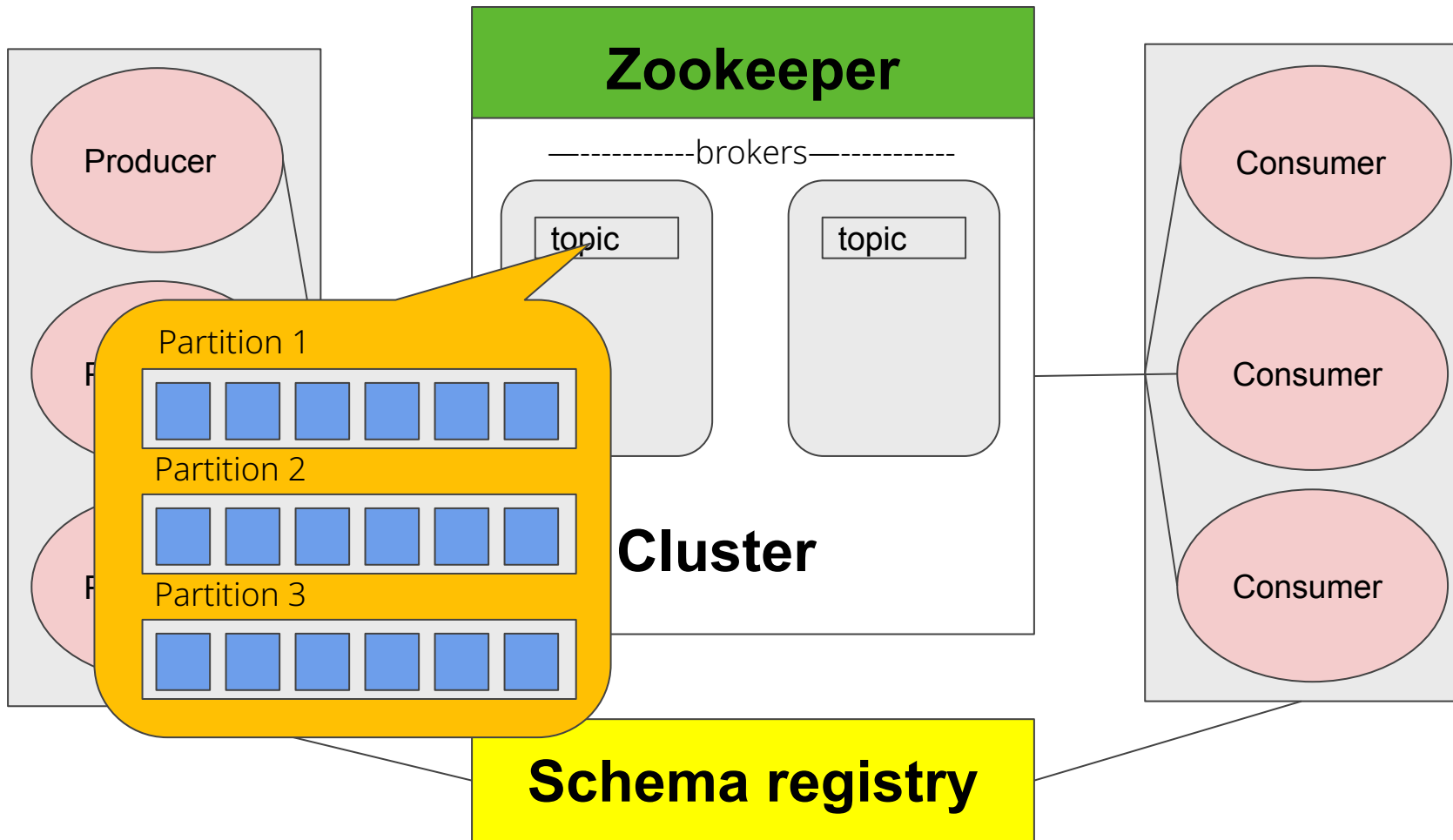


## Enterprise Services



# The architecture of Kafka

## an overview



## Compute Layer

Create, Process, and Query Event Streams and Tables



Kafka Streams API

Kafka  
Consumer API

Kafka  
Producer API

Import Data as Event Streams with Connectors



CLOUD SERVICE, DB,  
OR OTHER SYSTEM

Kafka Connect &  
Producer API

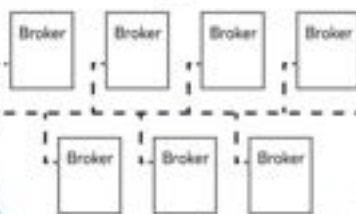
Export Data as Event Streams with Connectors

Kafka Connect &  
Consumer API

CLOUD SERVICE, DB,  
OR OTHER SYSTEM



Kafka Cluster



Elastic & Scalable  
Streaming Storage

## Storage Layer

From Confluent  
developers site





**Why should you consider  
both architectures.....**



# Why combining?

## A developers view



- Apache Pulsar Functions are great in saving memory and the world (only takes memory when run)
- Serverless computing framework

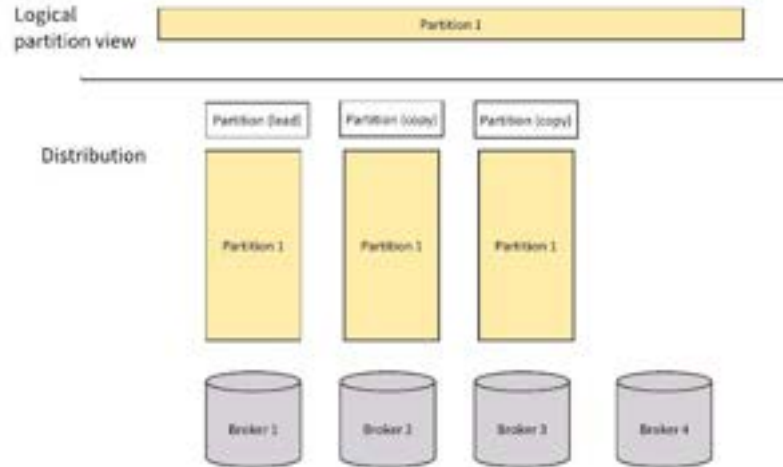


- Complex logic made simple with Kafka Stream API (like the Java Streams Api)
- More Developer Joy!!!!

$$1 + 1 = 2$$

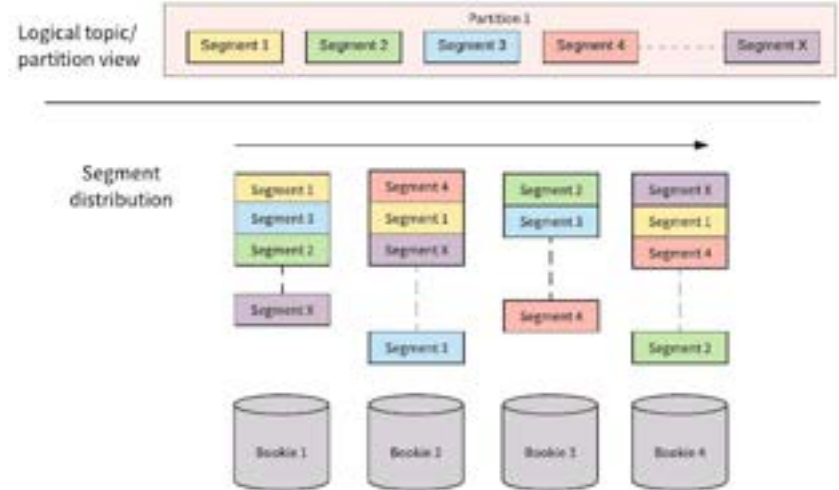
# Partition-Centric vs. Segment-Centric

## Apache Kafka



**Kafka Partitions** — All log segments are replicated in order across brokers (replication = 3 here).

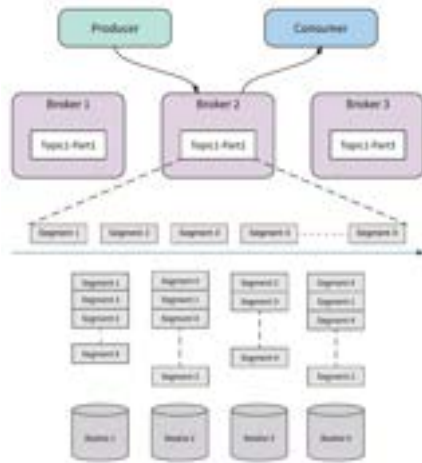
## Apache Pulsar/BookKeeper



**Pulsar/BookKeeper Stream** — All log segment are replicated to a configurable number of bookies (replication = 3 here) across N possible bookies (N = 4 here). Log segments are evenly distributed to achieve horizontal scalability with no rebalancing.

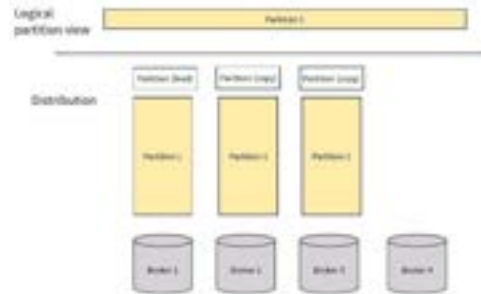
# Architecture Advantage of Pulsar

- Compute and Storage Separation
  - Stateless brokers
  - Independent scalability
  - Instantaneous broker scaling and disaster recovery

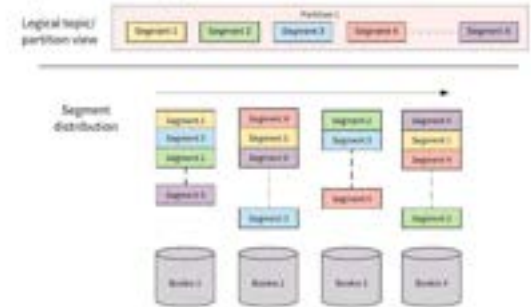


- Segment-Oriented Log Management
  - Segment (of a Partition) as the smallest replication unit
  - Efficient storage utilization; Unbounded partition storage
  - Truly horizontal scalability
  - Fast and low impact scaling and disaster recovery

## Apache Kafka



## Apache Pulsar/BookKeeper



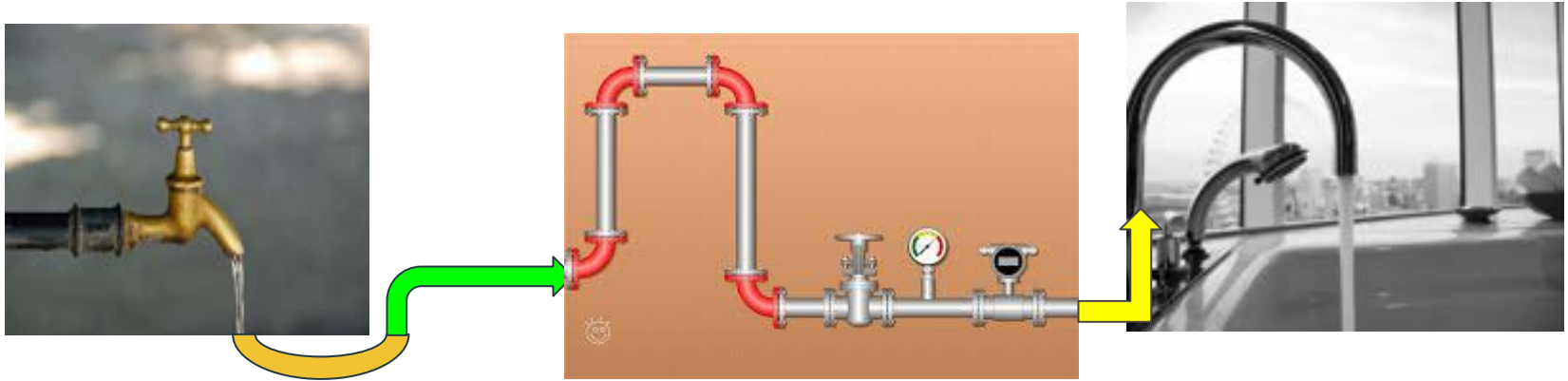
# DΛSTΛ><



## **A. Use Apache Pulsar Functions and save the world together**

# Data Pipeline Illustrated

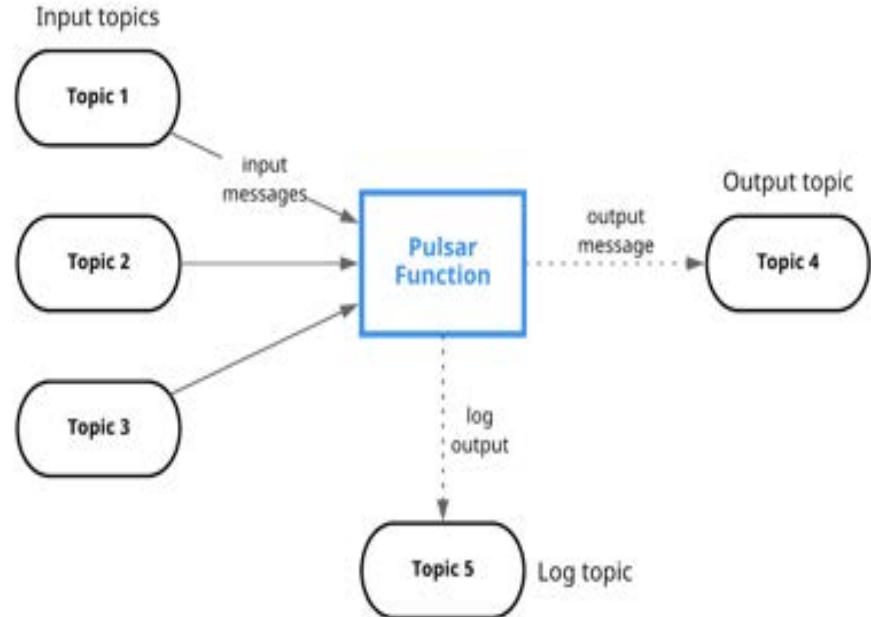
“Function” is there to transform the data in the most efficient way!





# Pulsar Functions

- ❖ Allows complex streaming processing
- ❖ Light-weight
- ❖ Function-as-a-service (“inspired by” AWS Lambda, Google Function, ...)
- ❖ Main languages:
  - Java
  - Python
  - Go



# Rich Ecosystem of Connectors and Clients (as of Jan 2023)



# Examples Utilizing Pulsar Functions

**Filtering messages**

**Counting the number of words flowing through the pipeline**

**Large enterprise applications such as fraud detection that needs to be running continuously**

**IoT applications in an agricultural farm identifying lost cows** 



## **B. Apache Kafka Streams API**

**Complex logic made simple....**



# **Bridging the Two Together**

## **Integrating Apache Pulsar and Apache Kafka**

# DataStax Starlight:

## Protocol Level Compatibility for Pulsar

Drop in replacement for existing messaging and streaming platforms.

Existing skill sets remain applicable

Eliminates interoperability challenges between various messaging platforms

Extensive testing to ensure full compatibility at a specification and feature set level.



*\*\*MQTT, ActiveMQ, RocketMQ coming soon*

# Wrapping up

## Apache Pulsar:

- Pulsar is a Cloud Native Messaging Platform with built-in Multi-Tenancy and GeoReplication
- Pulsar components are horizontally scalable, with cold data offloading
- Pulsar is open source, with a vibrant community - no vendor lock-in

## Apache Kafka:

- Larger and mature community
- More support and examples to find about Kafka
- Kafka has a lot of flavours to use and connect to other systems



## Key takeaway!

- Apache Pulsar Functions are great in saving memory and the world (only takes memory when run)
- Serverless computing framework
- “Write once, deploy, framework manages”



- Complex logic made simple with Kafka Stream API (like the Java Streams Api)
- More Developer Joy!!!!!!

$$1 + 1 = 2$$

# › Resources Apache Pulsar and Astra from DataStax



<https://pulsar.apache.org/>



<https://bookkeeper.apache.org/>

<https://zookeeper.apache.org>

<https://astra.datastax.com>

<https://www.datastax.com/products/astra-streaming>

<https://www.datastax.com/products/luna-streaming>

<https://docs.datastax.com/en/streaming/starlight-for-kafka/2.10.1.x/index.html>

CDC for Astra: <https://docs.datastax.com/en/astra/docs/astream-cdc.html>

DATASTAX

ASTRA DB

DATASTAX

ASTRA STREAMING

DATASTAX

LUNA STREAMING

DATASTAX

Starlight for Kafka

DATASTAX

ASTRA

## › Resources Apache Kafka



<https://kafka.apache.org/>



<https://zookeeper.apache.org>



More info about Kafka?  
Check the link



slides



code

# Thank You



[@KoTurk77](https://twitter.com/KoTurk77)



<https://www.linkedin.com/in/ko-turk-b271b929/>



[@mgrygles](https://twitter.com/mgrygles)



<https://www.linkedin.com/in/mary-grygleski/>



<https://www.twitch.tv/mgrygles>



<https://discord.gg/RMU4Juw>



<https://dev.to/mgrygles>

