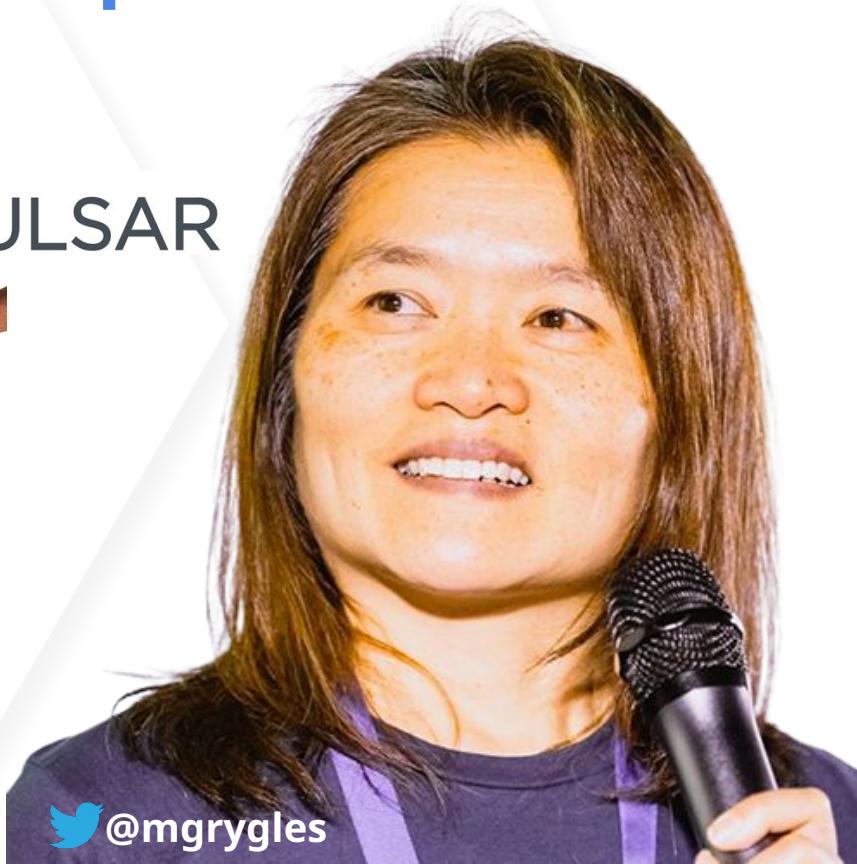


Best of Both Worlds - Apache Pulsar and Apache Kafka



@KoTurk77



@mgrygles

**How to
access this
slide deck?**

<http://bit.ly/3ltNtkK>





Ko Turk
Going all-in the code



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

Who is Ko?



Developer

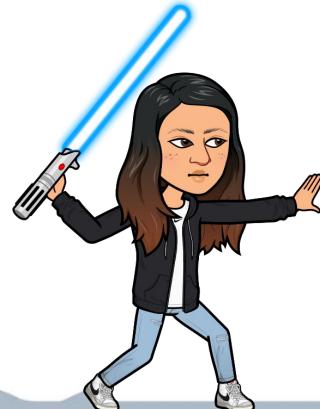


Community Lead

Magazine writer

Speaking at conferences

Senior Developer Advocate



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



mrgrygles



mary-grygleski



mrgrygles



mrgrygles

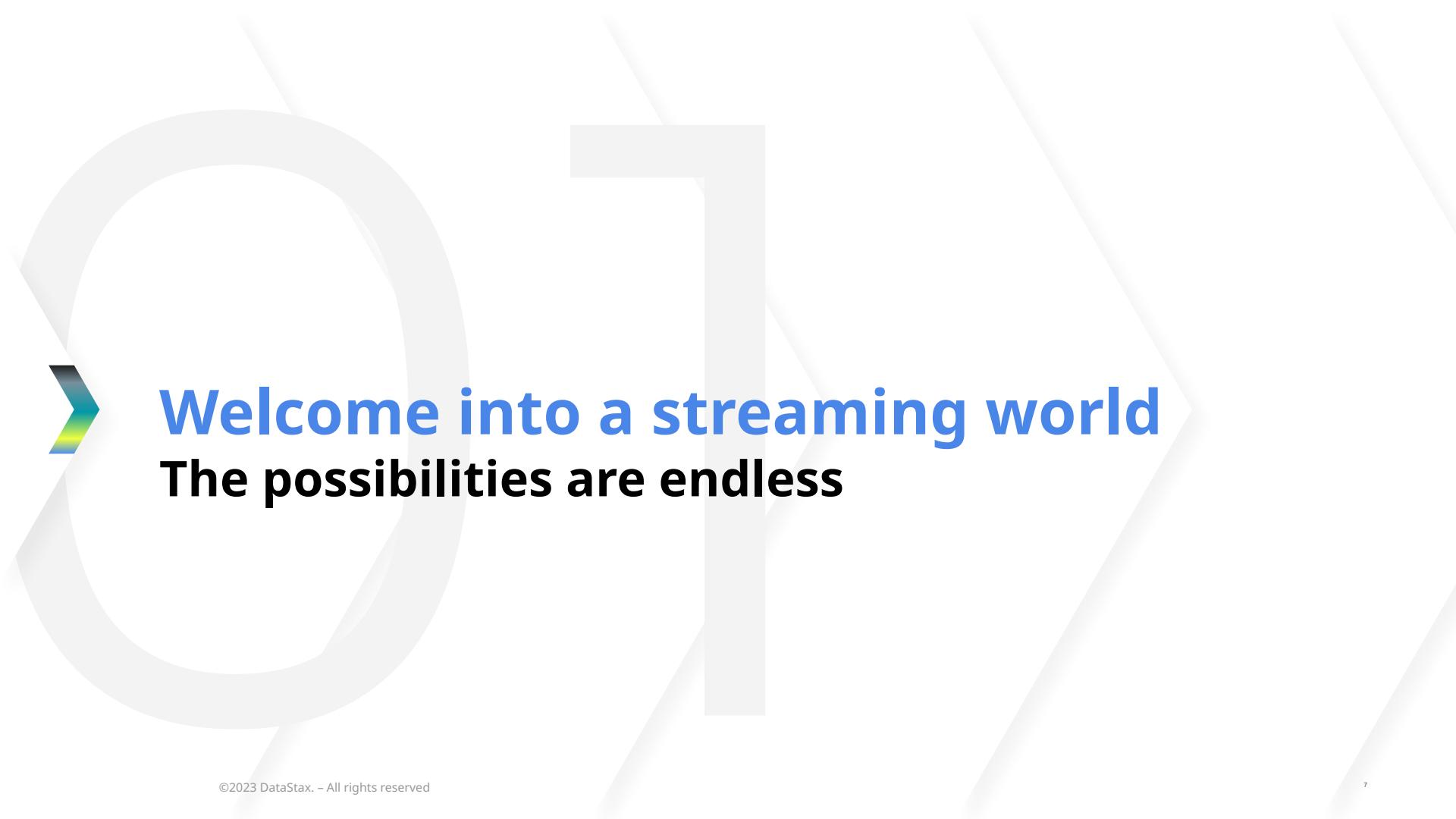
What's in it for me?

- > 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

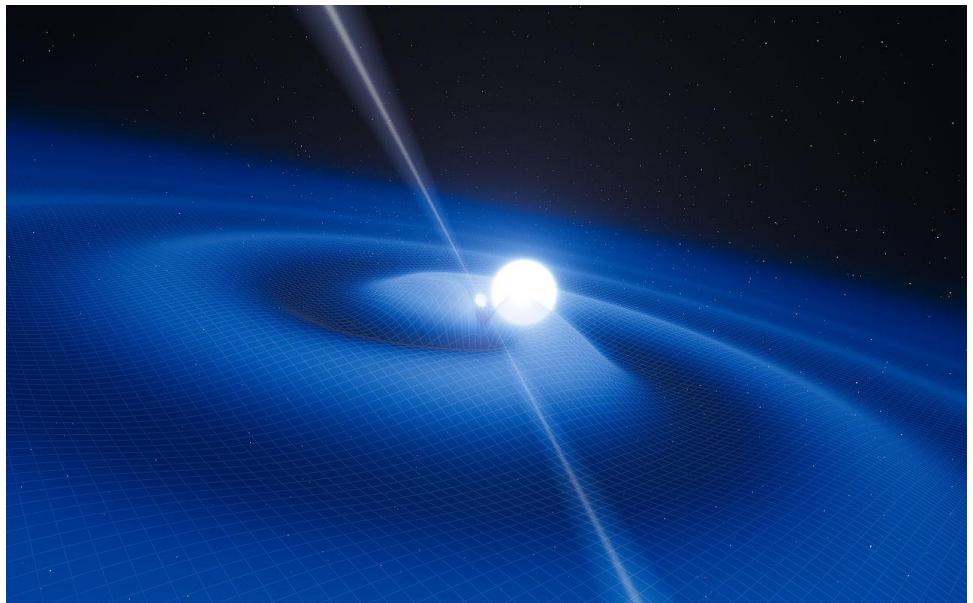
1. Welcome into a streaming world (Introductions)
2. Streaming architectures
3. Why should you consider both architectures?
4. Code in action (demo → datastax / code)
 - a. Pulsar Functions
 - b. Kafka Streaming API



Welcome into a streaming world
The possibilities are endless



Kafka or Pulsar ?



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
- Offloading old data to Object Storage
- Native Kubernetes support
- Native Schema Registry support (Apache AVRO, JSON and Google Protobuf)
- Connectors/Integrations (Pulsar IO framework)
- Message processing (Pulsar Functions framework)
- Multiple Client bindings: Java, C++, Python, Go, NodeJS, C#...

Apache Pulsar - why ?

Blazing performance

Horizontal scalability and object storage offloading

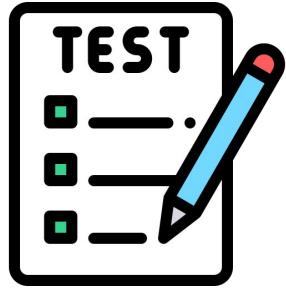
Consolidation

Message replay

Geo-replication

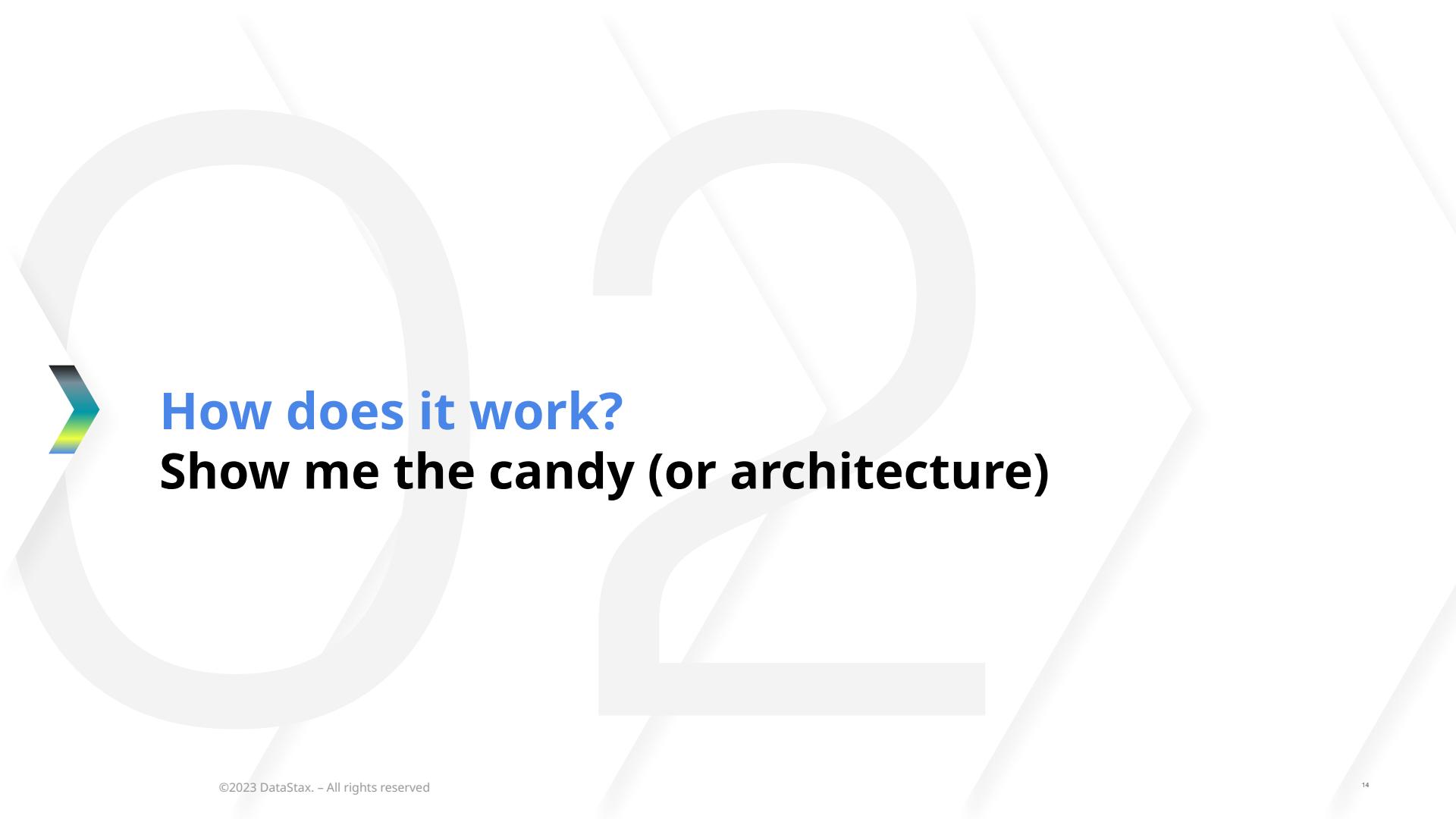
Future readiness

'Oh hello' Apache Kafka!



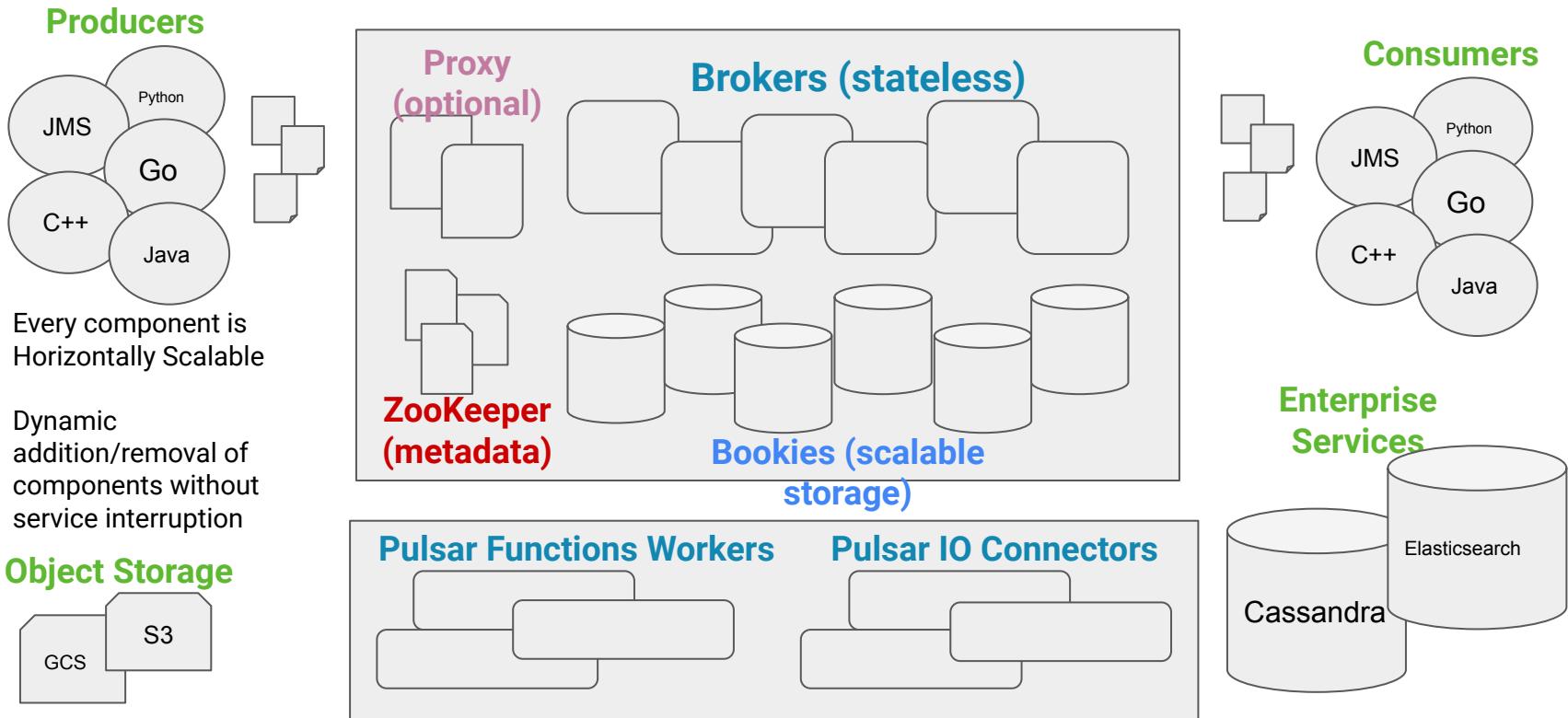
Apache Kafka!

- **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)



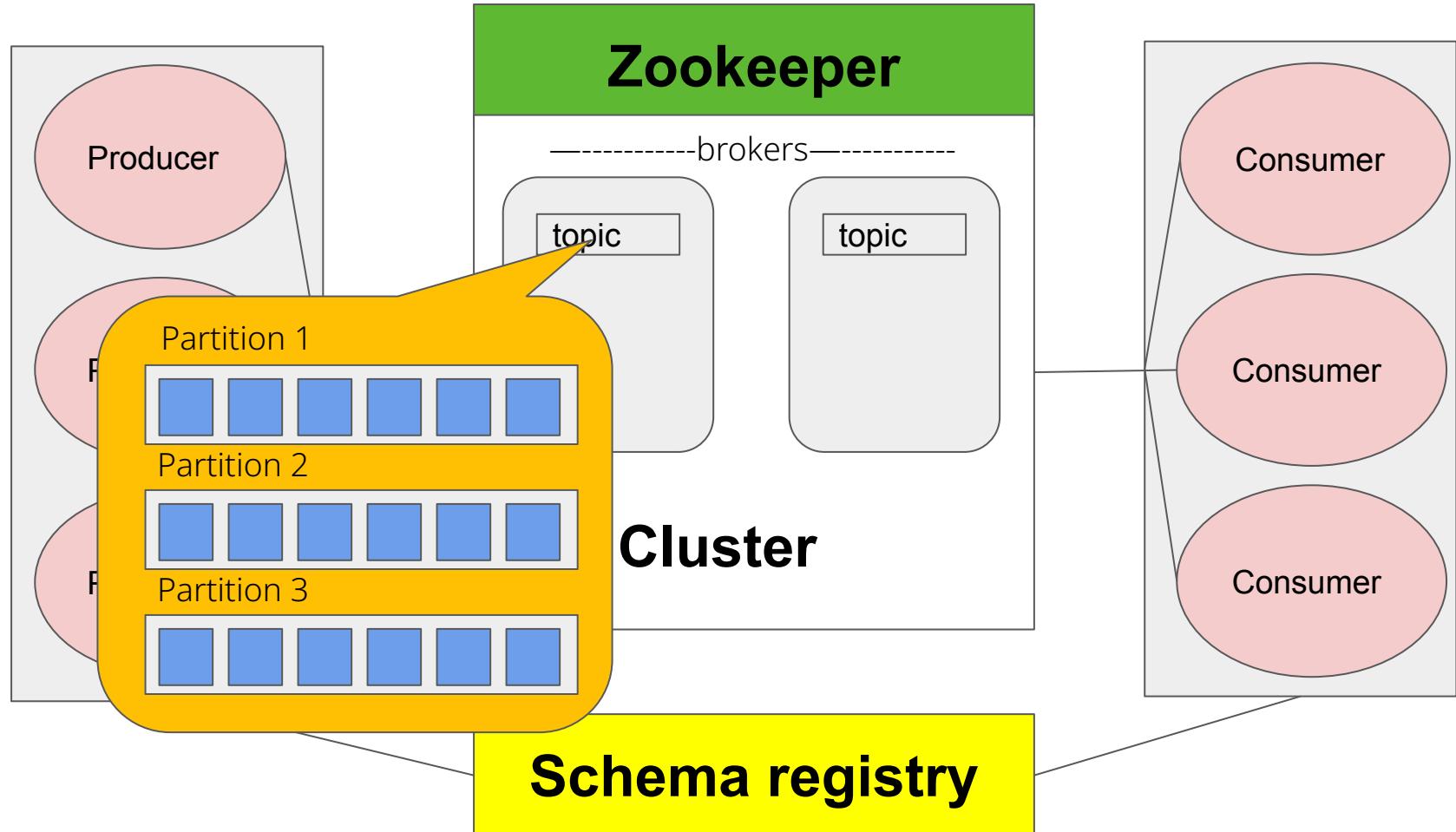
How does it work?
Show me the candy (or architecture)

Apache Pulsar - Basic Architecture



The architecture of Kafka

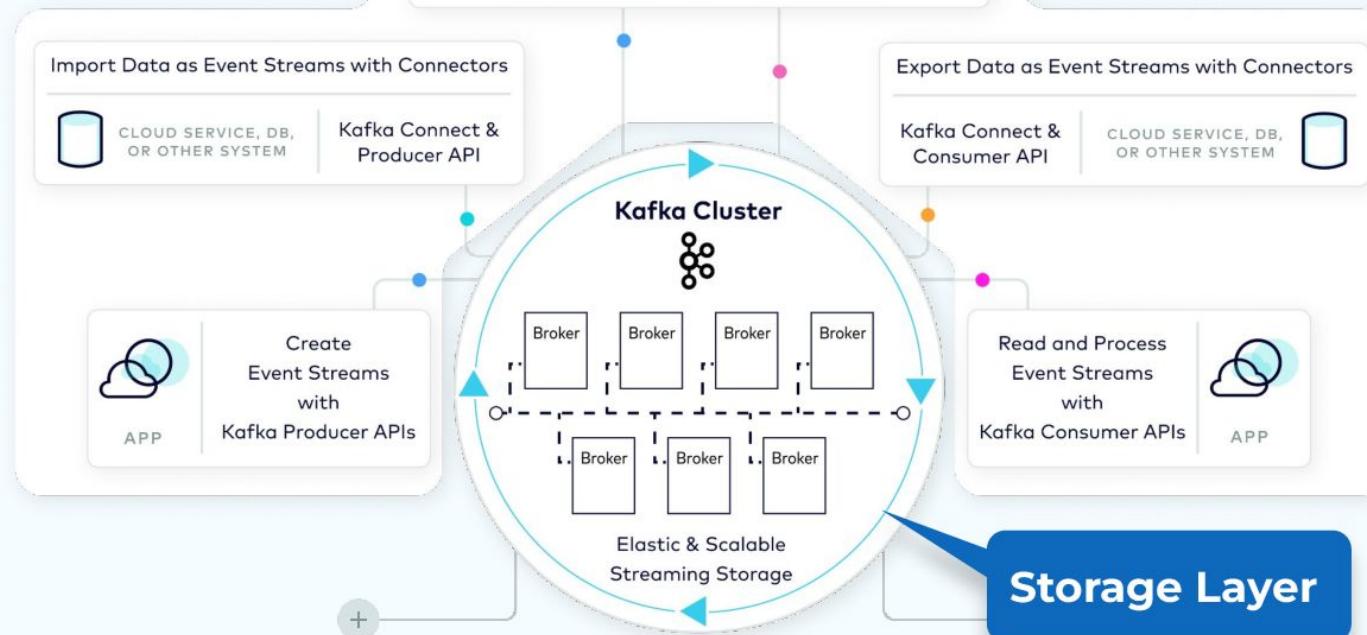
an overview



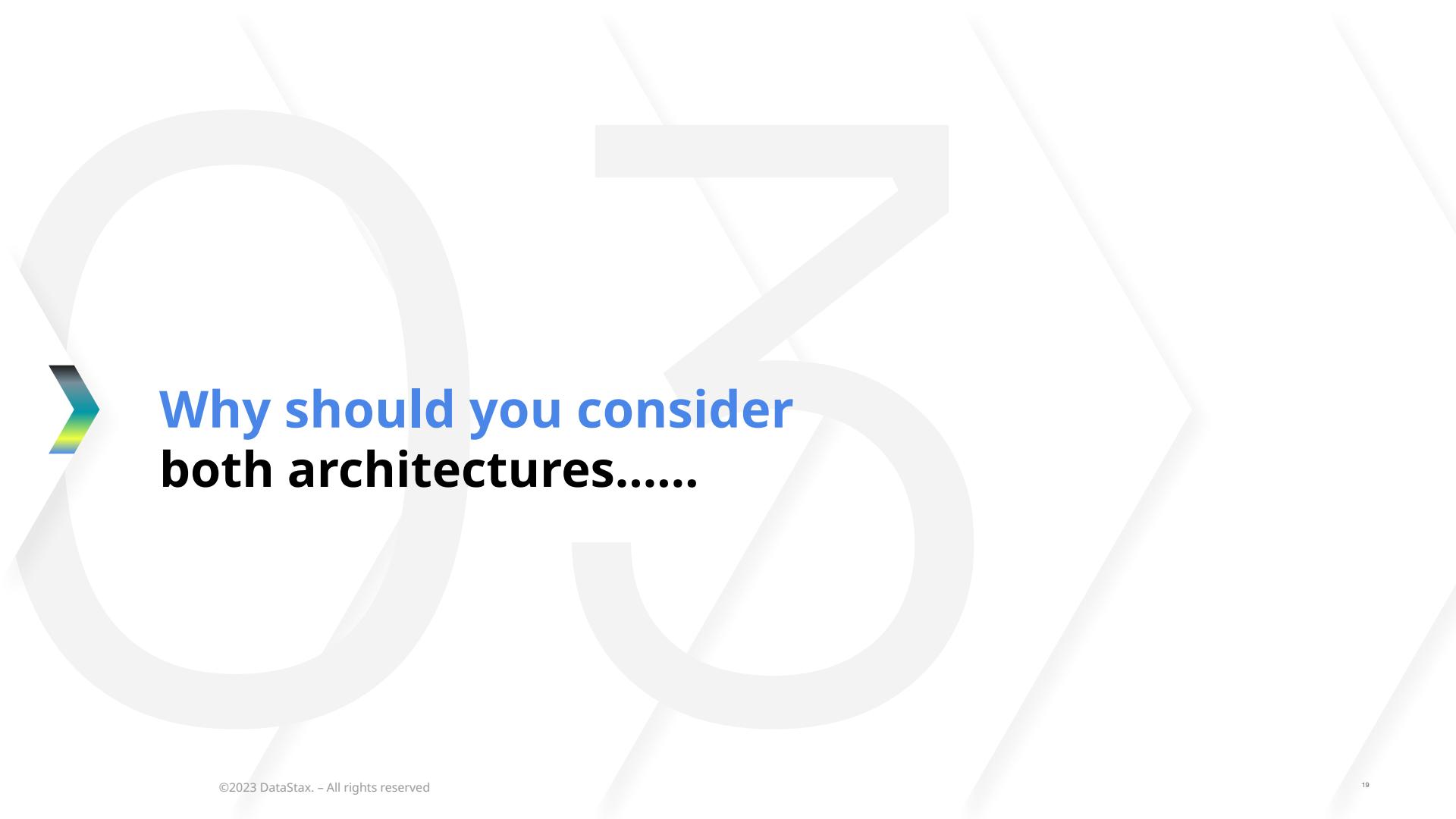
From Confluent
developers site



Compute Layer



Storage Layer



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

Why combining? A developers view

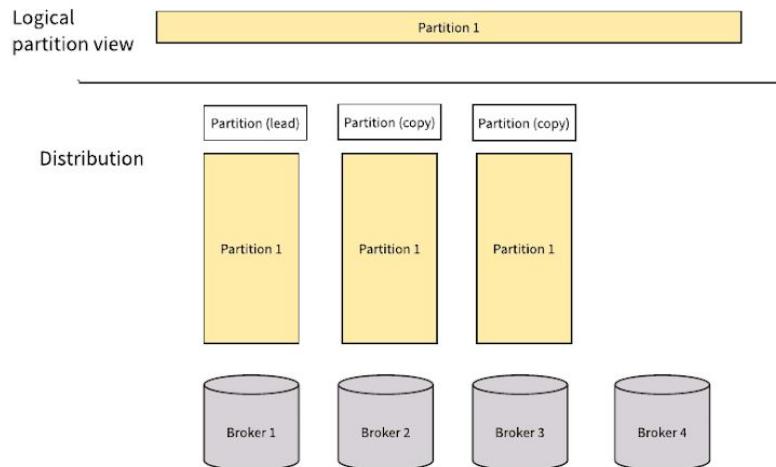
- Apache Pulsar Functions are great in saving memory and the world
- Complex logic made simple with Kafka Stream API
(like the Java Streams Api)



$$1+1=2$$

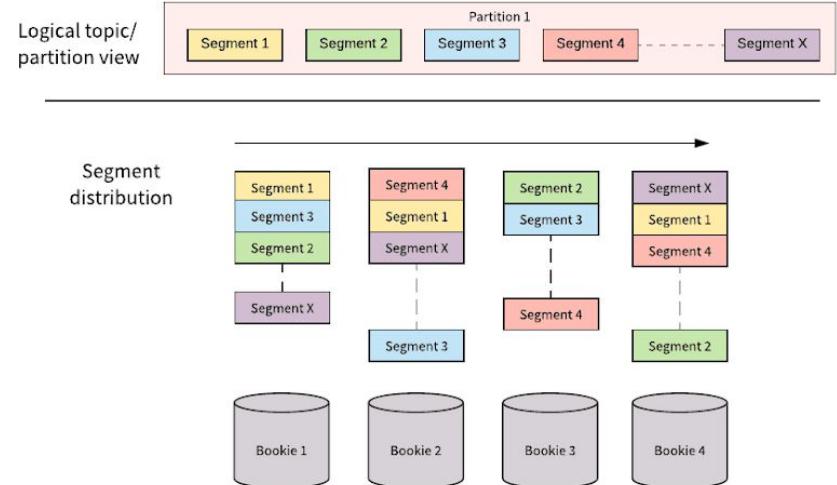
Partition-Centric vs. Segment-Centric

Apache Kafka



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

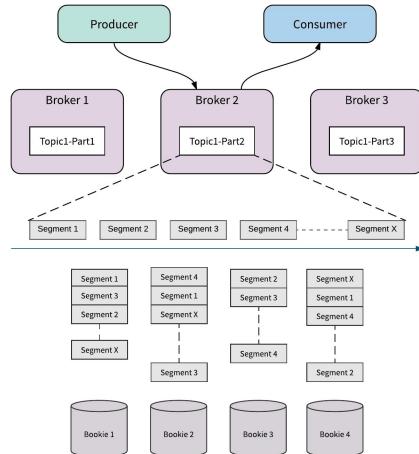
Apache Pulsar/BookKeeper



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.

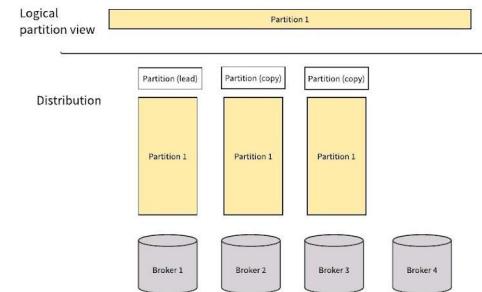
Architecture Advantage of Pulsar

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

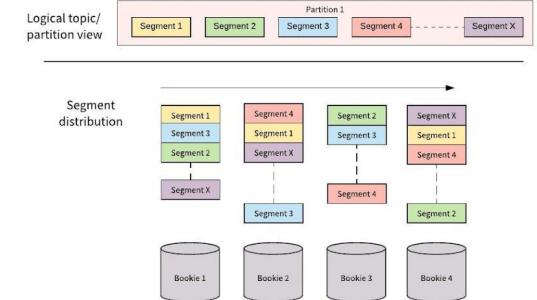


- Segment-Oriented Log Management
 - Segment (of a Partition) as the smallest replication unit
 - Efficient storage utilization; Unbounded partition storage
 - Truly horizontal scalability
 - Fast and low impact scaling and disaster recovery

Apache Kafka



Apache Pulsar/BookKeeper



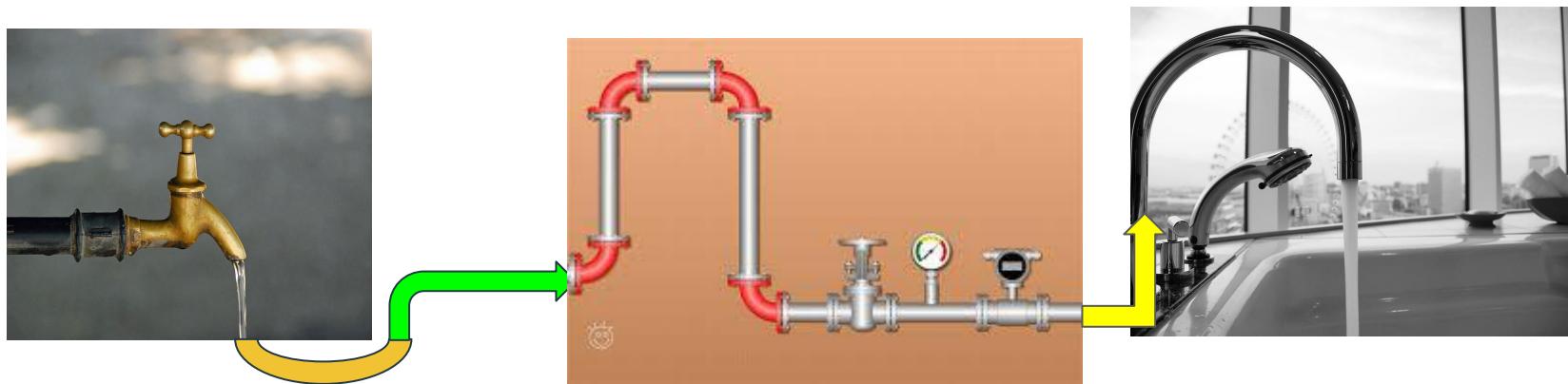




A. Use Apache Pulsar Functions and save the world together

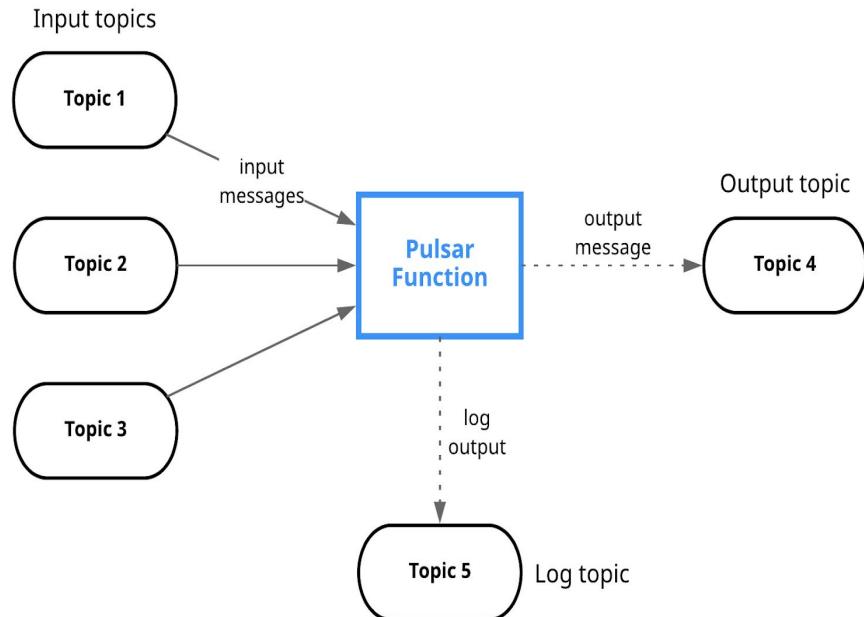
Data Pipeline Illustrated

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



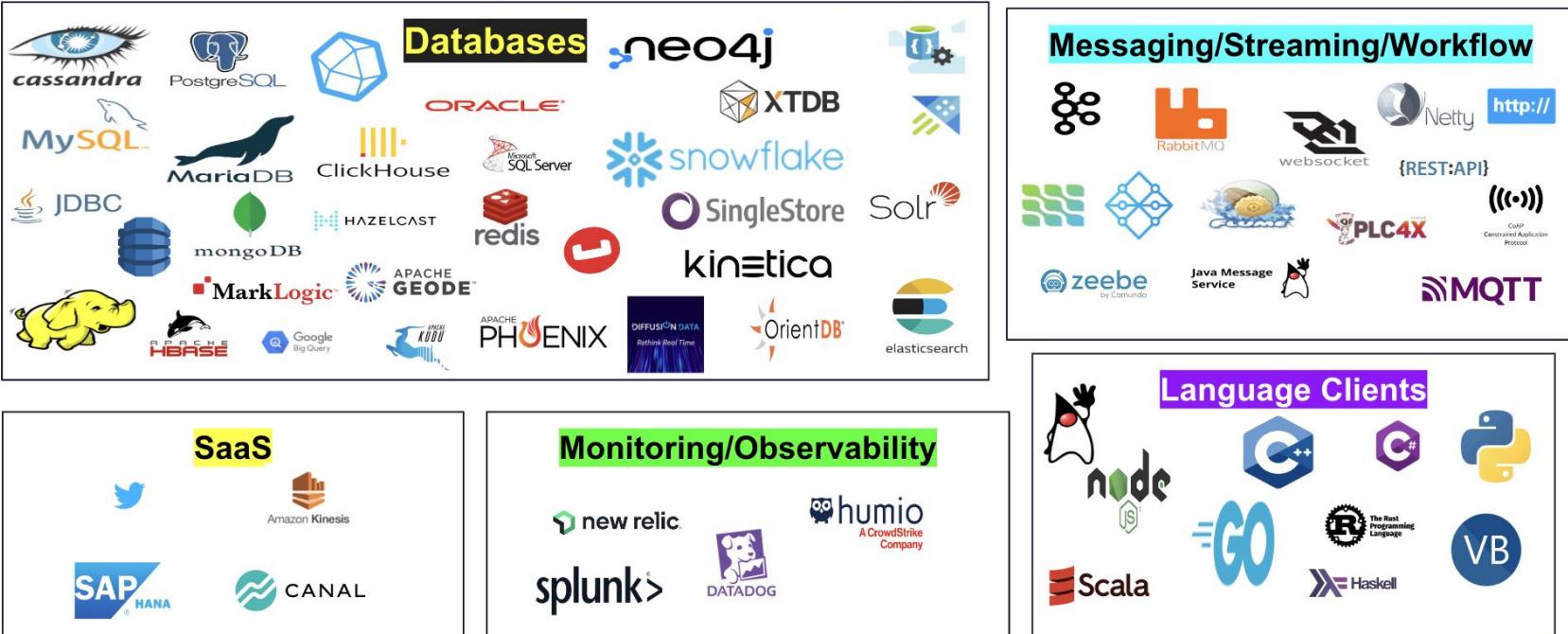
Pulsar Functions

- ❖ 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)





B. Apache Kafka Streams API

Complex logic made simple....

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



$$1+1=2$$

➤ 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>



CDC for Astra:

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

› Resources Apache Kafka



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



<https://zookeeper.apache.org>



Youtube video about Kafka





slides



Thank You



@KoTurk77



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



@mgrygles



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



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



<https://discord.gg/RMU4Juw>



<https://dev.to/mgrygles>



code

