DataStax

Developers

Realtime data pipelines with Apache Pulsar™ and Apache Cassandra

Data in motion with Pulsar Functions

Director of Developer Relations

clunven

clunven

clun

- Trainer
- Public Speaker
- Developers Support
- Developer Applications
- Developer Tooling

- Creator of ff4j (ff4j.org)
- Maintainer for 8 years+

- Happy developer for 14 years
- Spring Petclinic Reactive & Starters
- Implementing APIs for 8 years

Cédrick Lunven

Streaming Developer Advocate

PULSAR

ddieruf

ddieruf

@dierufdavid

.NET

- Developer/Architect
- Specialize in cloud-native
- Kubernetes
- Application modernization
- CI/CD

David Dieruf

Cedrick Lunven

David Dieruf

Rags Srinivas

Artem Chebotko

Stefano Lottini

Aleksandr Volochnev

Aaron Ploetz

Jack Fryer

Kirsten Hunter

Gary Harvey

Mary Grygleski

Ryan Welford

David Gilardi

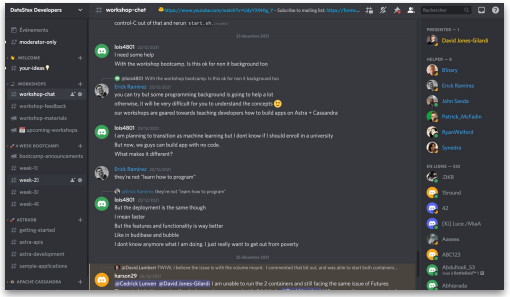
DataStax Developers Crew

Livestream: youtube.com/DataStaxDevs

YouTube

Twitch

Questions: https://dtsx.io/discord

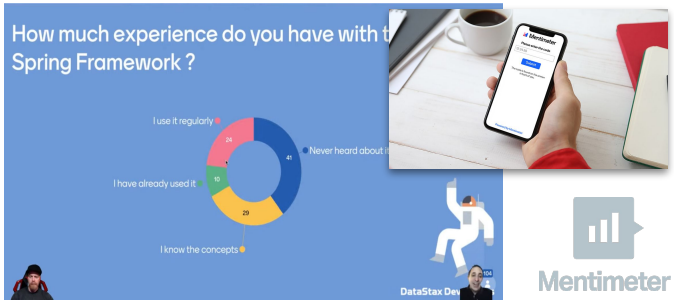


YouTube
(with nighbot)

Discord
(#workshop-chat)

!discord

Games and quizzes: menti.com



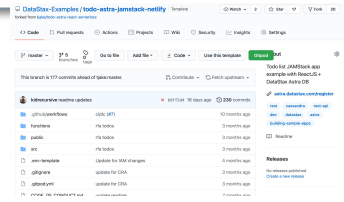
Mentimeter

!menti

Live Sessions

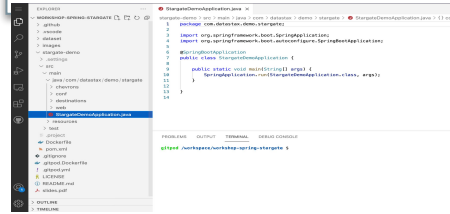
Nothing to install !

Source code + exercises + slides



GitHub

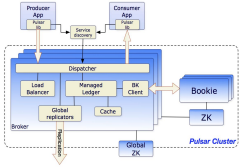
IDE



Gitpod

Messaging system (based on Pulsar)

PULSAR

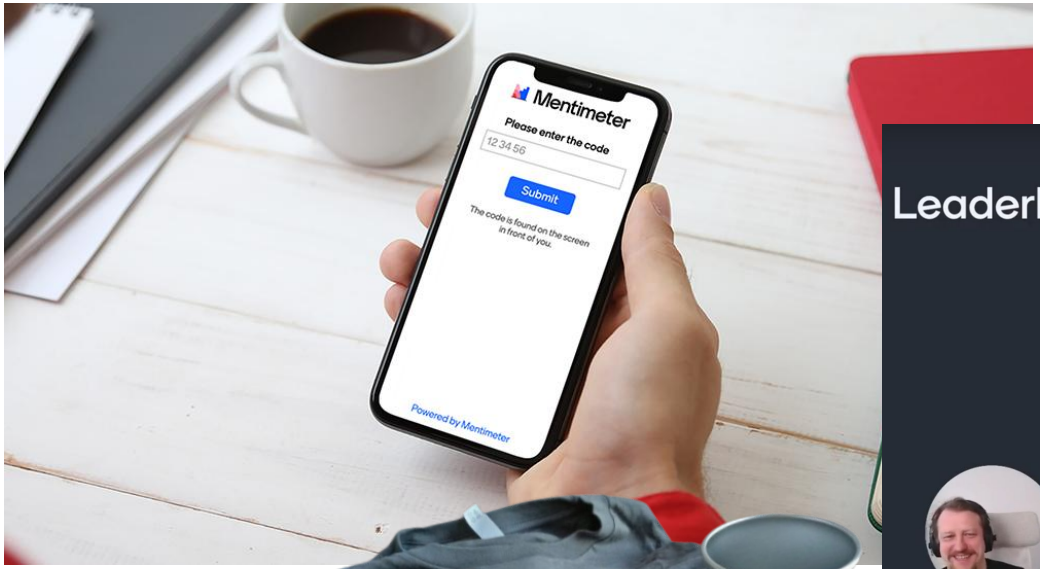


DataStax Astra STREAMING

Database + GraphQL + PlayGround

DataStax Astra DB

Hands-On Housekeeping



`menti.com` ⇒ enter code
Don't answer in YT chat
Look at phone (not at YT)

Quiz on "Menti" !

01 PULSAR

Introduction to Apache Pulsar™

02

DataScience in Event Streaming

03

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05

Pulsar Sinks

06

What's next?

Quiz, Homework, Next week

Agenda

01 PULSAR

Introduction to Apache Pulsar™

02

DataScience in Event Streaming

03

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05

Pulsar Sinks

06

What's next?

Quiz, Homework, Next week

Agenda

Event Streaming == Message Streaming

- Watch for events with “the system” or application
- Publish messages and receive events
- Make decisions on data in real time
- Ingest high frequency of messages with very low latency and consume at a different rate

Event Streaming == Message Streaming

Open source

Created by Yahoo
Contributed to the Apache Software Foundation 2016
Top-level project 2018

Cloud-native design

Cluster based
Multi-tenant
Simple client APIs (Java, C#, Python, Go, Node, …)
Separate compute and storage!

Guaranteed message delivery

If a message successfully reaches a Pulsar broker, it will be delivered to its intended target.

Light-weight serverless functions framework

Create complex processing logic within a Pulsar cluster (aka: data pipeline)

Tiered storage offloads

Offload data from hot/warm storage to cold/long-term storage when the data is aging out

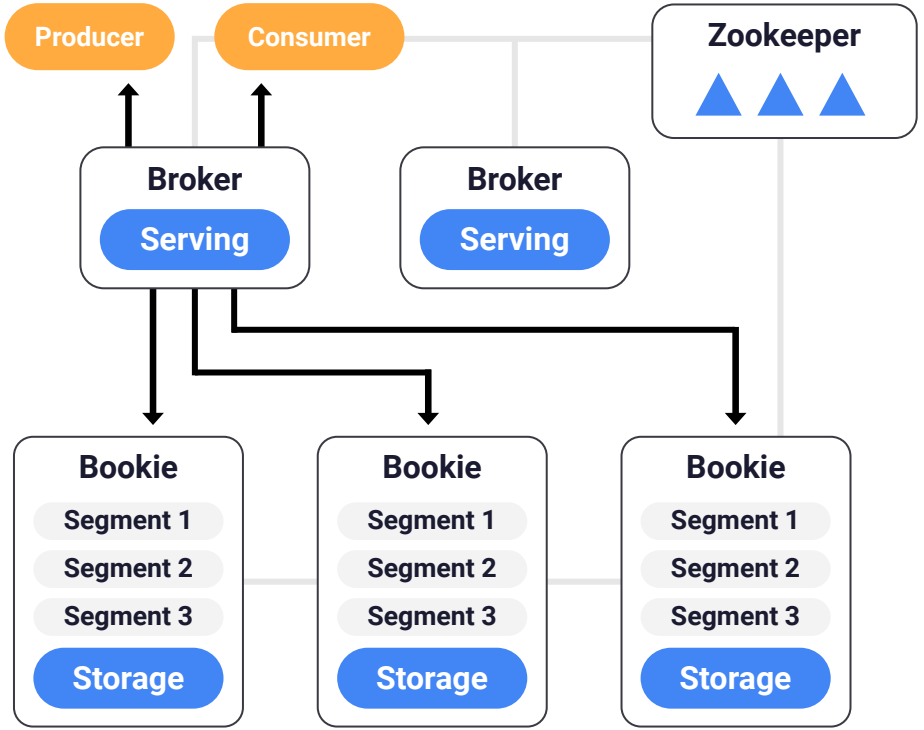
Distributed, tiered architecture

Separated compute from storage

Zookeeper holds metadata for the cluster

Stateless Broker handles producers and consumers

Storage is handled by Apache Bookkeeper



Producer

Client application sending messages to topic managed by Broker

Consumer

Client application reading messages from a topic managed by Broker

Broker

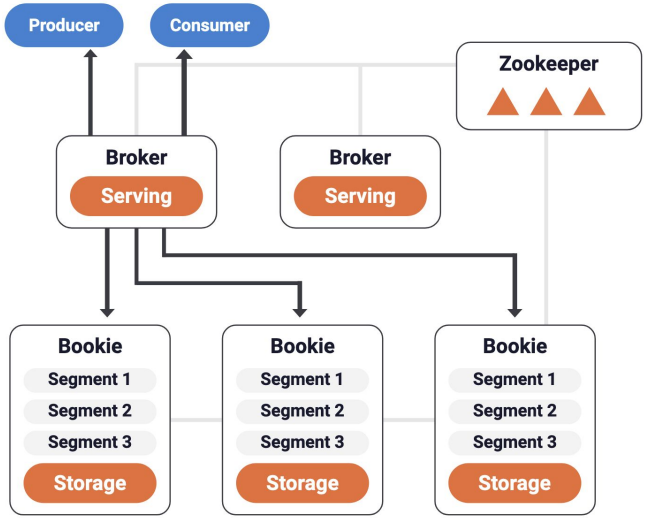
A stateless process that handles incoming message, message dispatching, communicates with the Pulsar configuration store, and stores messages in BookKeeper instances

BookKeeper

Persistent message store

ZooKeeper

Holds cluster metadata, handles coordination tasks between Pulsar clusters



Pulsar-as-a-Service

Streaming-as-a-Service built on Apache Pulsar

No Operations

Eliminate the overhead to install, operate, and scale Pulsar

Powerful Tools and APIs

Leverage the same tools used to interact with Pulsar on prem

Cloud Native

Built to run on any cloud

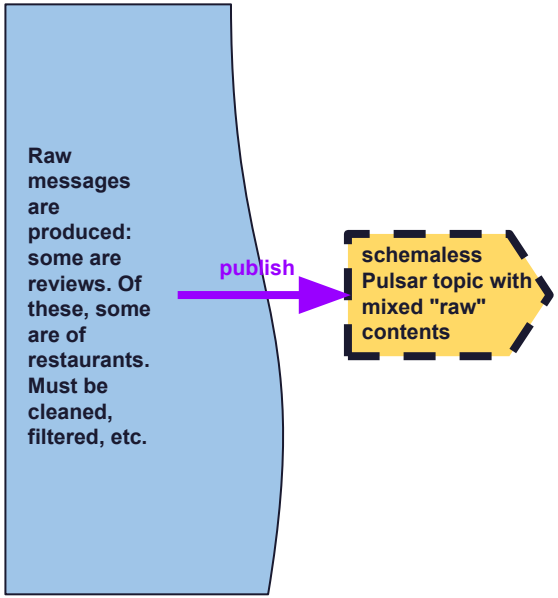
Zero Lock-in

Leverage Pulsar's built in integration with existing developer tools

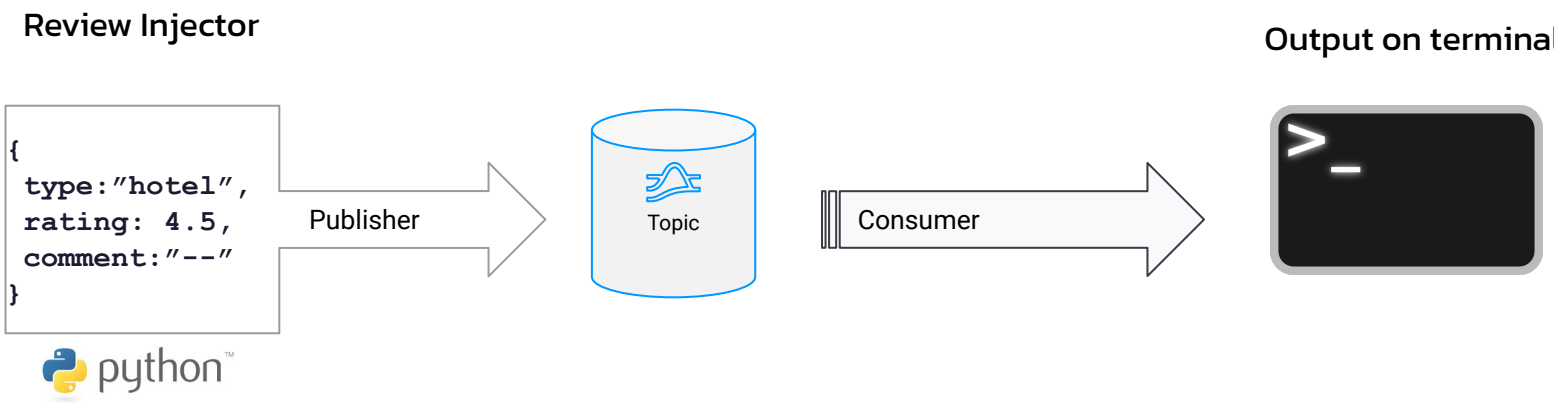
Start for Free

Free monthly credits to help you get started quickly

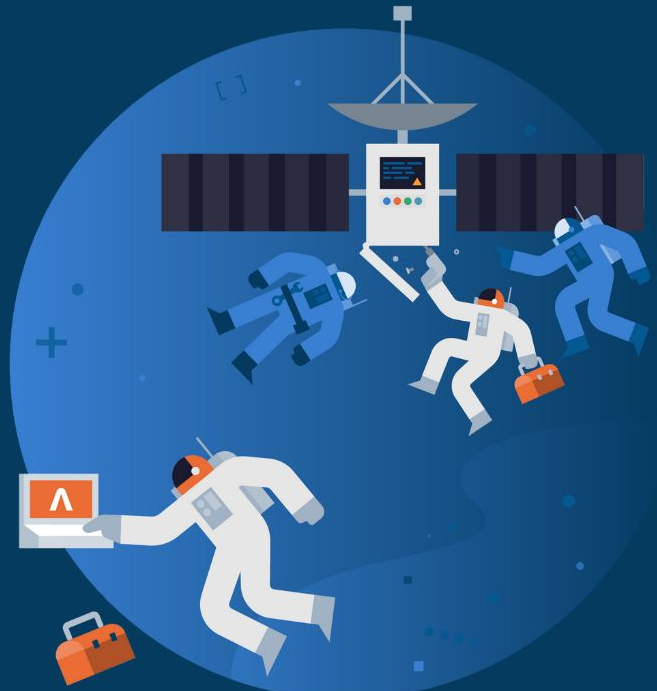
DataStax Astra: Streaming Made Easy in the Cloud



Business Architecture



Logical Architecture



Lab 1

Producer & Consumer

https://github.com/datastaxdevs/
workshop-pulsarfunctions-data-in-motion

01 PULSAR

Introduction to Apache Pulsar™

02

DataScience in Event Streaming

03

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05

Pulsar Sinks

06

What's next?

Quiz, Homework, Next week

Agenda

01

Fraud Detection

Needed to ingest high-speed writes of customer event traffic for real-time fraud detection and analysis. Geo-replication must have little to no latency.

02

Secure Social Media, Protect Customer Privacy

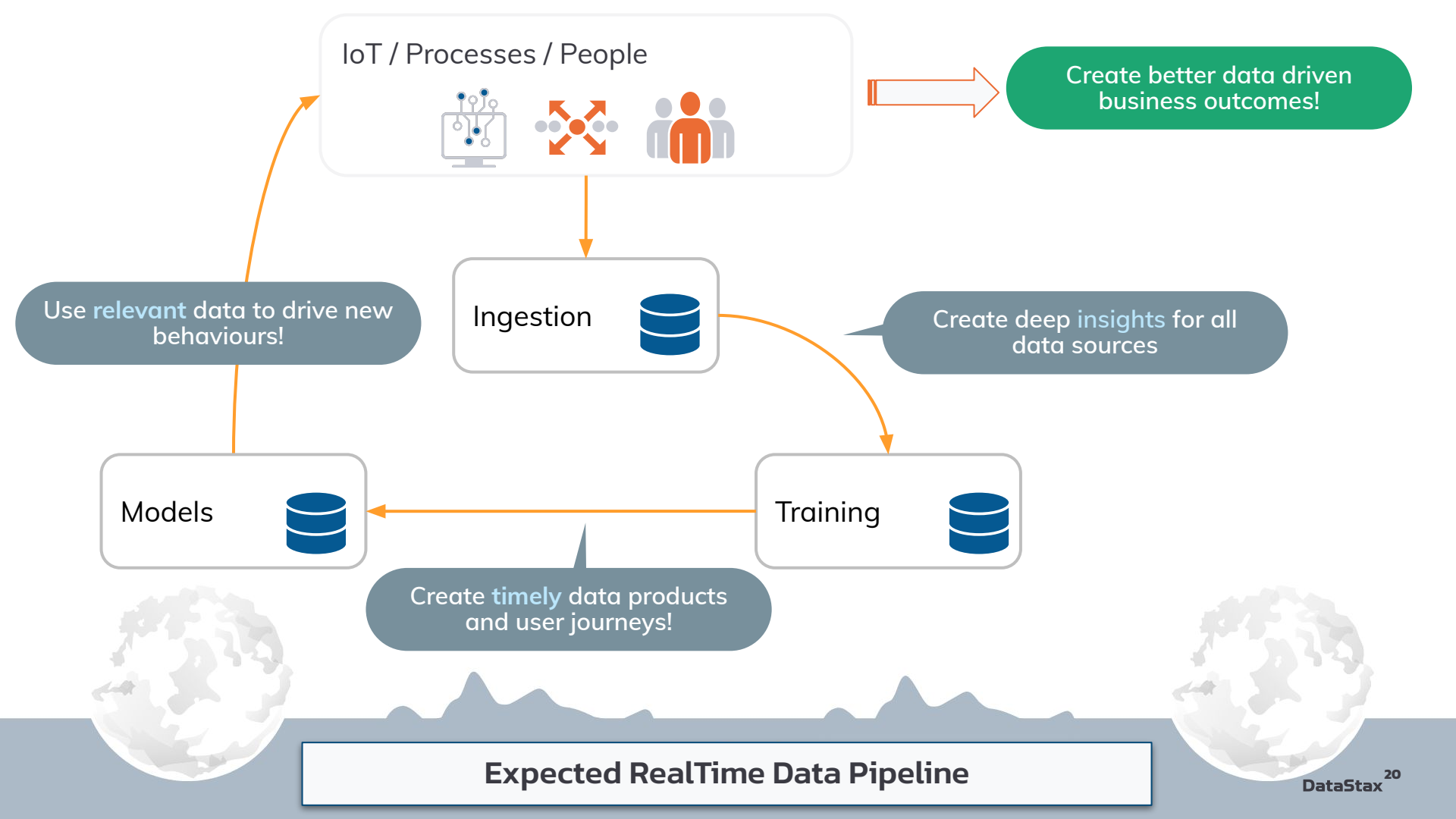
Identify out-of-the-ordinary patterns to prevent malicious attacks on digital and physical assets from unauthorized applications and individuals.

03

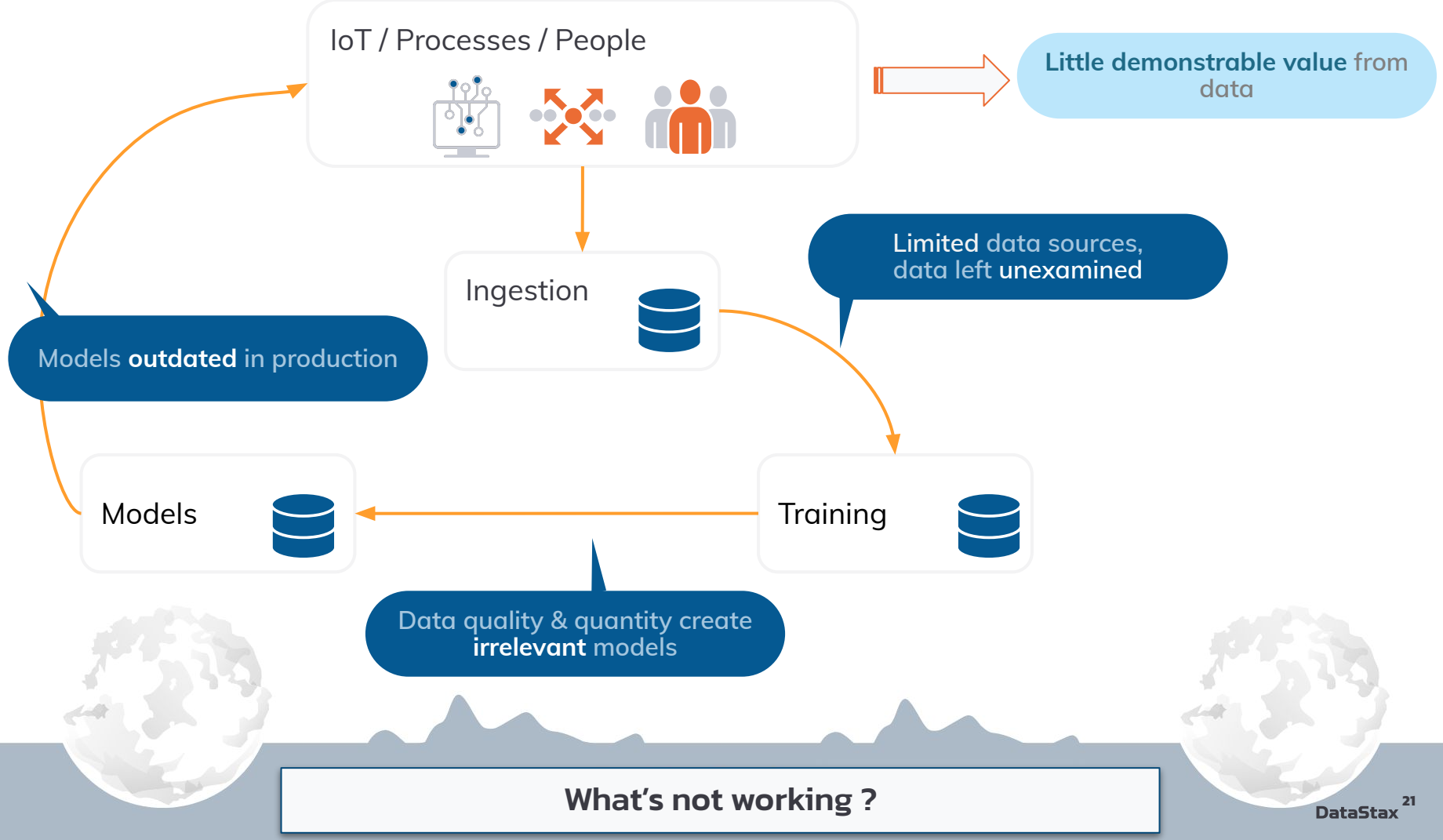
IoT Data Ingestion and Classification

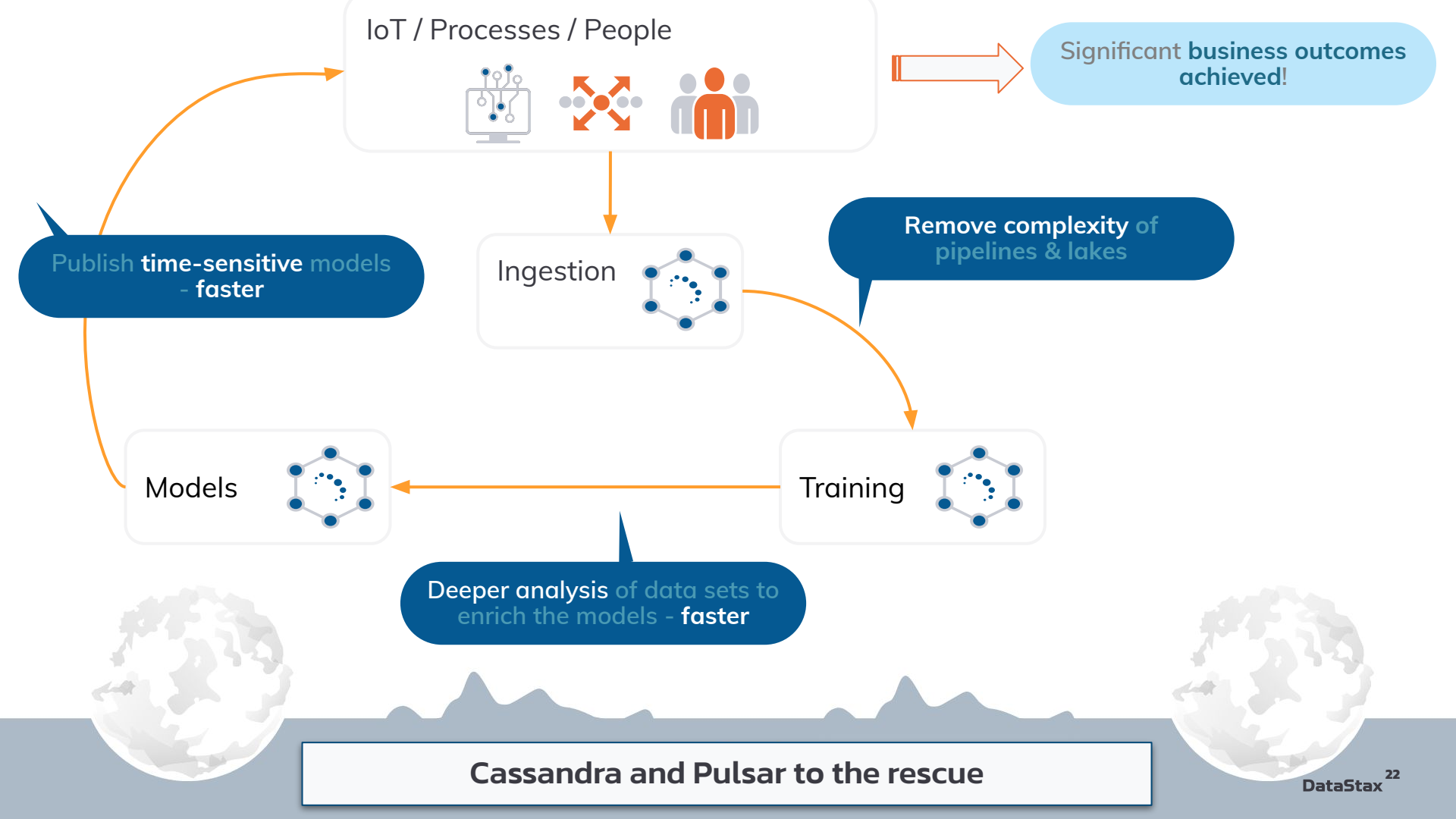
Take in high speed data with very little latency, while processing at a different [slower] speed internally.

DataScience with events



Expected RealTime Data Pipeline


IoT / Processes / People
Little demonstrable value from data
Limited data sources, data left unexamined
Ingestion
Models outdated in production
Models
Training
Data quality & quantity create irrelevant models
What's not working ?


IoT / Processes / People
Significant business outcomes achieved!
Publish time-sensitive models - faster
Ingestion
Remove complexity of pipelines & lakes
Models
Training
Deeper analysis of data sets to enrich the models - faster
Cassandra and Pulsar to the rescue

Our Data pipeline today

01 PULSAR

Introduction to Apache Pulsar™

02 

DataScience in Event Streaming

03 

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05 

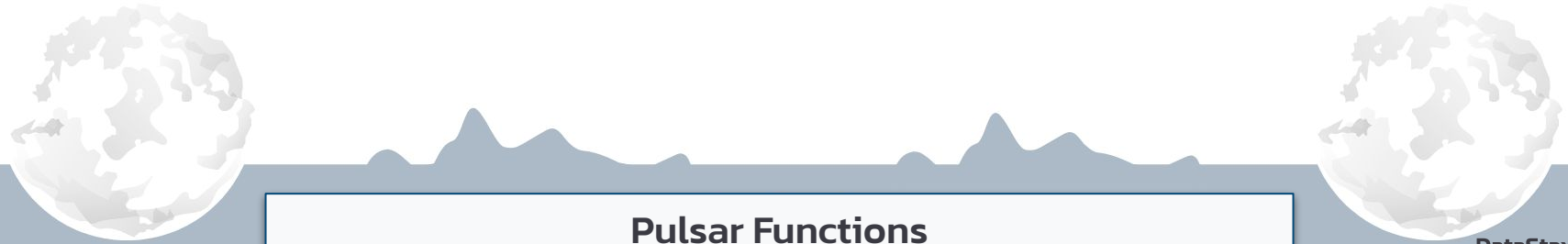
Pulsar Sinks

06 

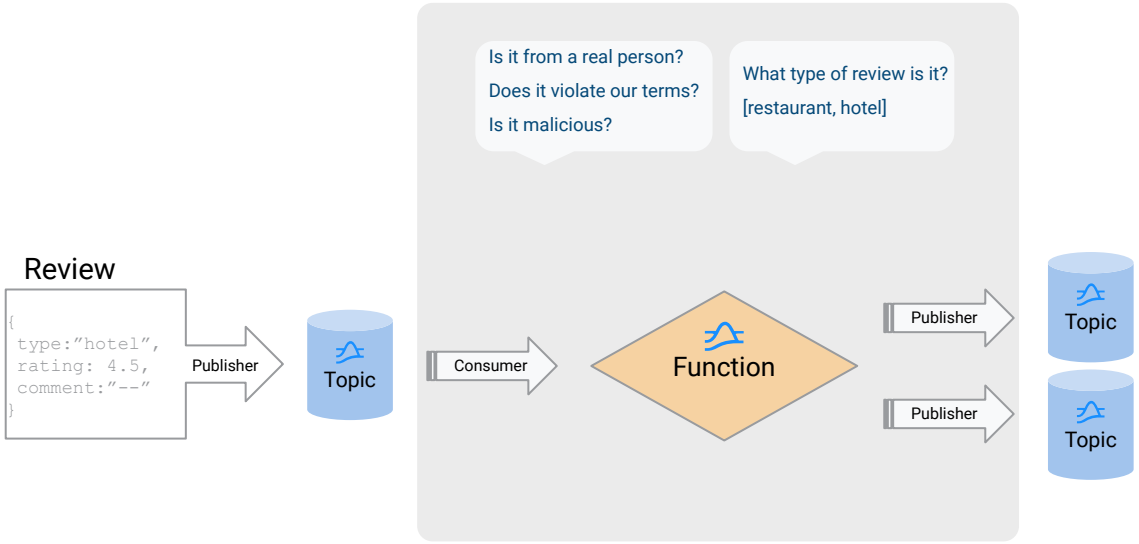
What's next?

Quiz, Homework, Next week

Agenda



Pulsar Functions



Architecture Overview

Lab 2

Pulsar Functions

https://github.com/datastaxdevs/workshop-pulsarfunctions-data-in-motion

01 PULSAR

Introduction to Apache Pulsar™

02

DataScience in Event Streaming

03

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05

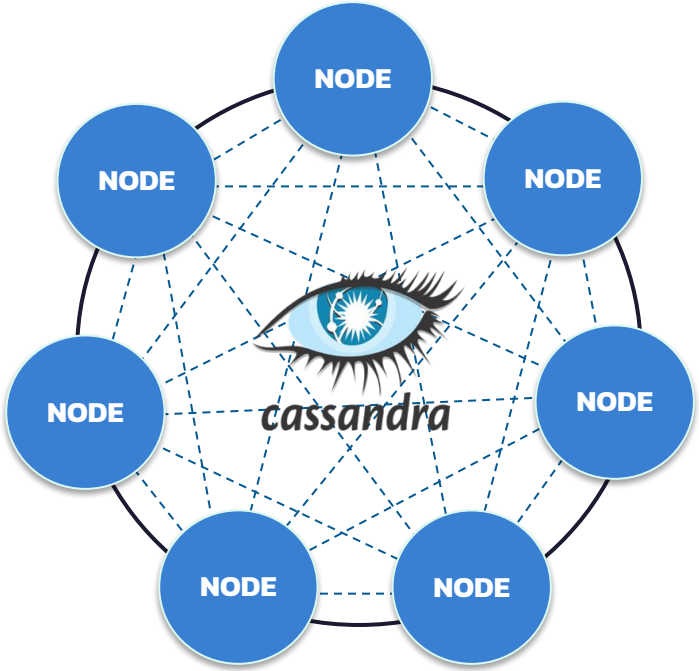
Pulsar Sinks

06

What's next?

Quiz, Homework, Next week

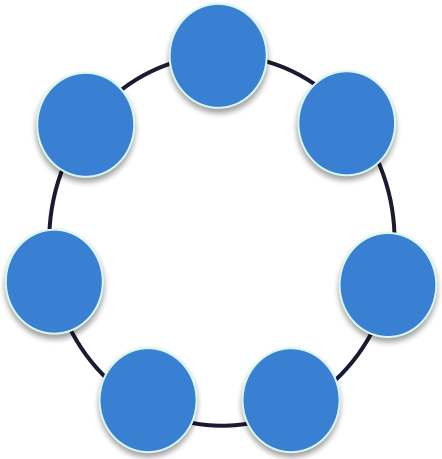
Agenda



- Big Data Ready
- Read / Write Performance
- Linear Scalability
- Highest Availability
- Self-Healing and Automation
- Geographical Distribution
- Platform Agnostic
- Vendor Independent

Apache Cassandra's Awesomeness

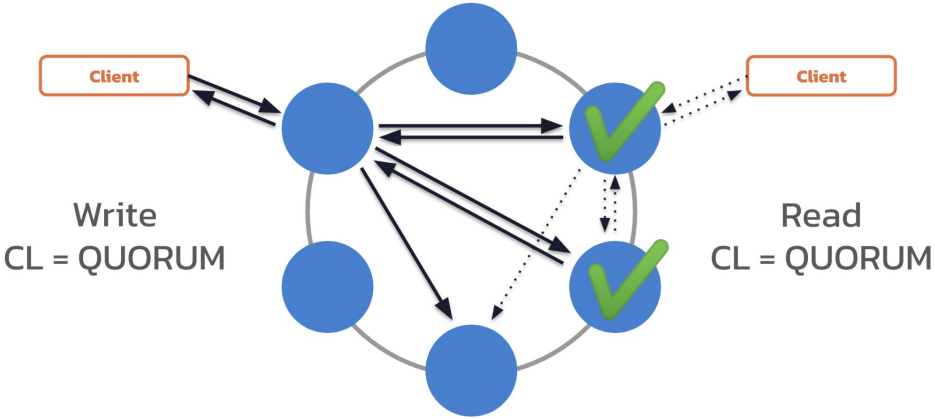
Partitioning over distributed architecture makes the database capable to handle data of any size: we mean petabytes scale. Need more volume? Add more nodes.



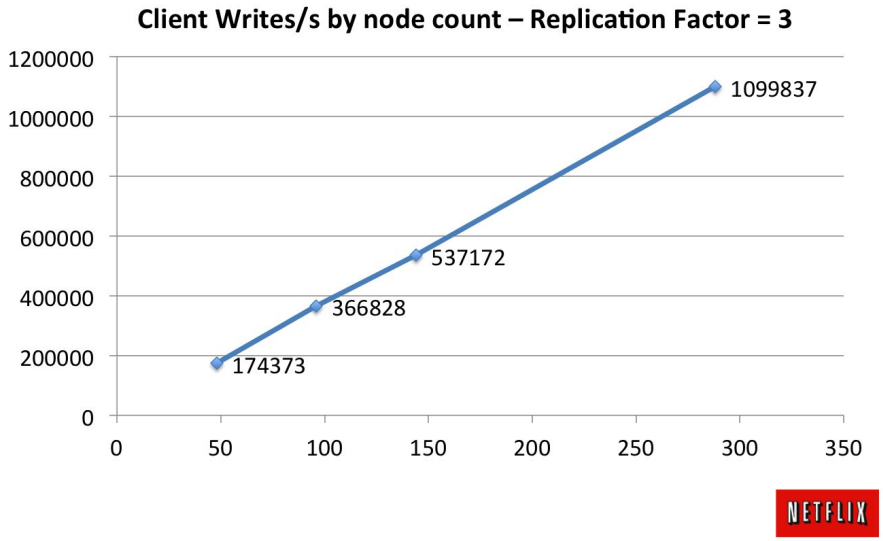
Big Data Ready

Even a single Cassandra node is very performant but a cluster consisting of multiple nodes and data centers brings throughput to the next level.

Decentralisation (**masterless architecture**) means that every node is able to deal with any request, read or write.



- For volume or velocity, there are no limitations
- **Linear** - No overhead on new nodes, scales with your needs*

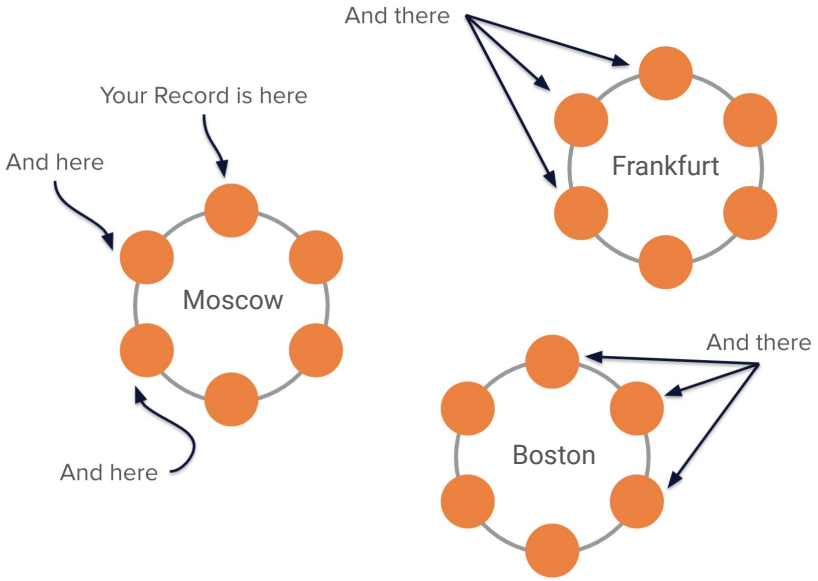


NETFLIX

Linear Scalability

Replication, Decentralisation, and Topology-Aware Placement Strategy take care of possible downtimes:

- Multiple Live Replicas
- No Single Point of Failure
- Network topology-aware data placement
- Client-side Smart Reconnection and Strong Retry Mechanism

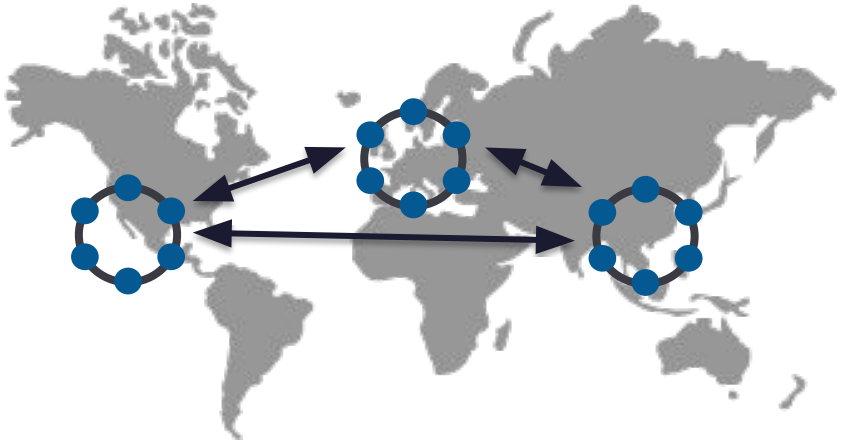


Highest Availability

Operations for a huge cluster can be exhausting so Apache Cassandra clusters are smart and able to scale, change data placement and recover automagically.

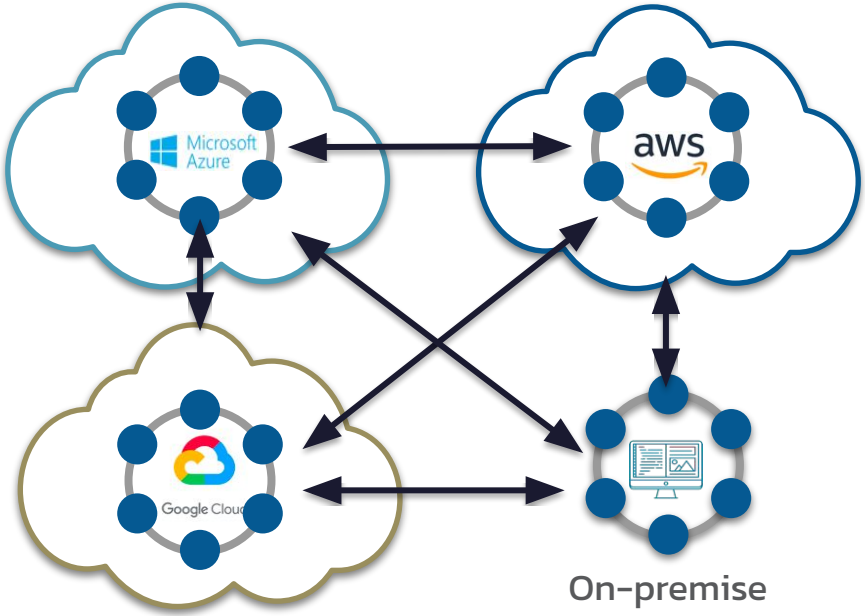
Cassandra's trademark is multi-datacenter deployments, granting you an exceptional capability for disaster tolerance while keeping your data close to your clients - worldwide.

All DCs are active (available for both writes and reads)!



Geographical Distribution

Apache Cassandra is **not bound to any platform** or service provider, helping you build hybrid-cloud and multi-cloud solutions with ease.



Platform Agnostic

Cassandra doesn't belong to any of commercial vendors but controlled by a non-profit Open Source **Apache Software Foundation**, already familiar to you by *Hadoop*, *Spark*, *Kafka*, *Zookeeper, Maven* and many other projects.

Vendor Independent

01 PULSAR

Introduction to Apache Pulsar™

02

DataScience in Event Streaming

03

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05

Pulsar Sinks

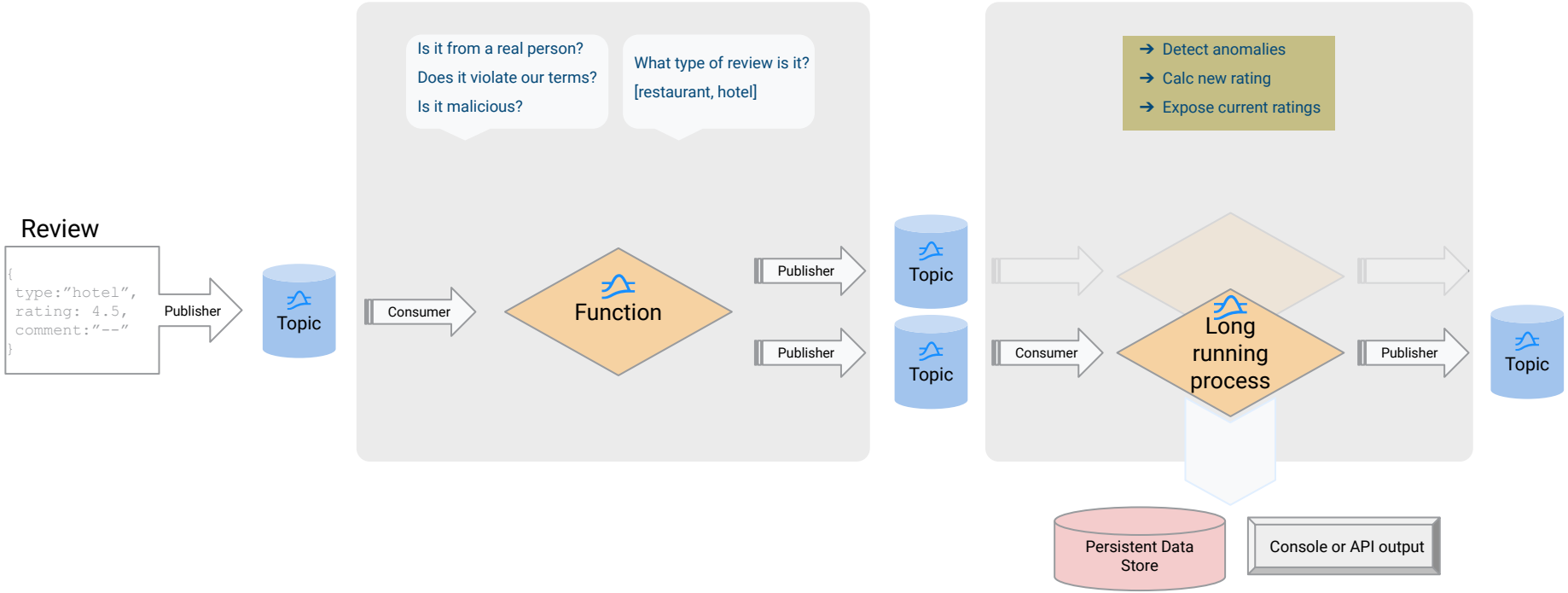
06

What's next?

Quiz, Homework, Next week

Agenda

Pulsar Sinks



Architecture Overview

Lab 3

Pulsar Sinks

https://github.com/datastaxdevs/
workshop-pulsarfunctions-data-in-motion

01 PULSAR

Introduction to Apache Pulsar™

02

DataScience in Event Streaming

03

Pulsar Functions

04 cassandra

Introduction to Apache Cassandra™

05

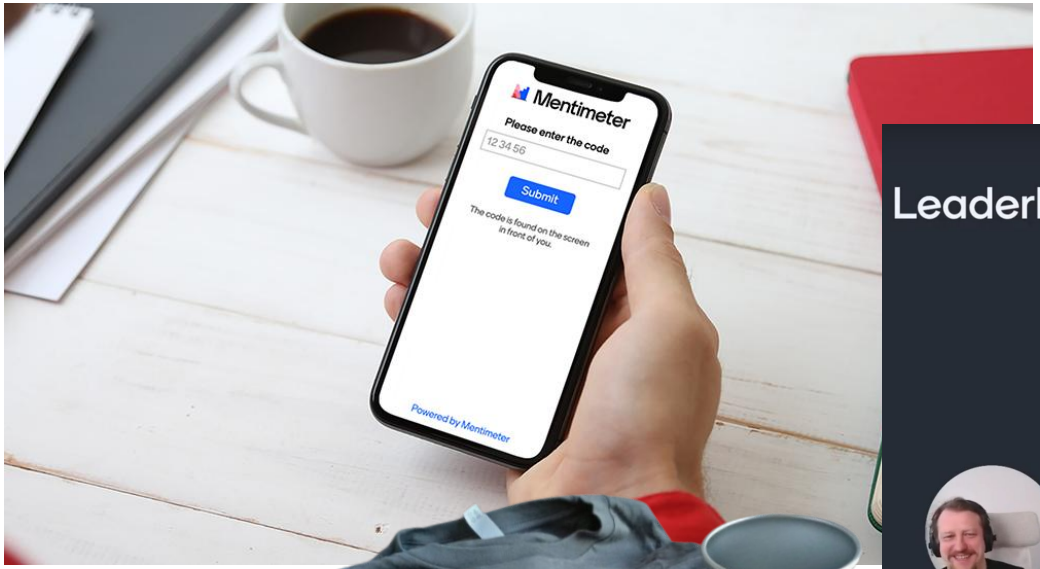
Pulsar Sinks

06

What's next?

Quiz, Homework, Next week

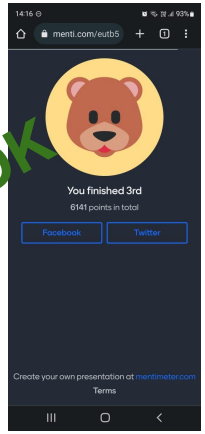
Agenda



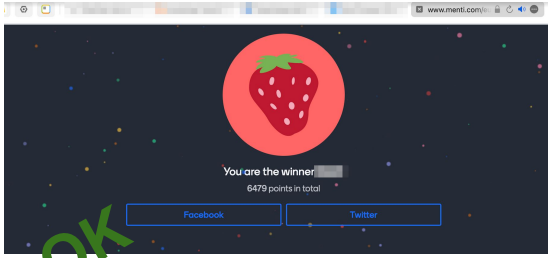
`menti.com` ⇒ enter code
Don't answer in YT chat
Look at phone (not at YT)

Quiz on "Menti" !

SWAG WINNERS 🎉



OK

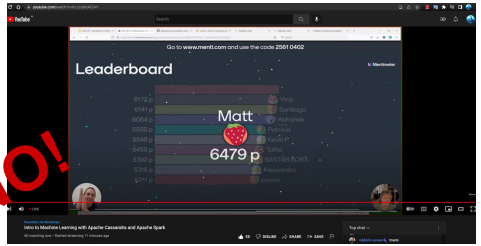


OK

Congratulations to 1st, 2nd and 3rd place on the Menti quiz!

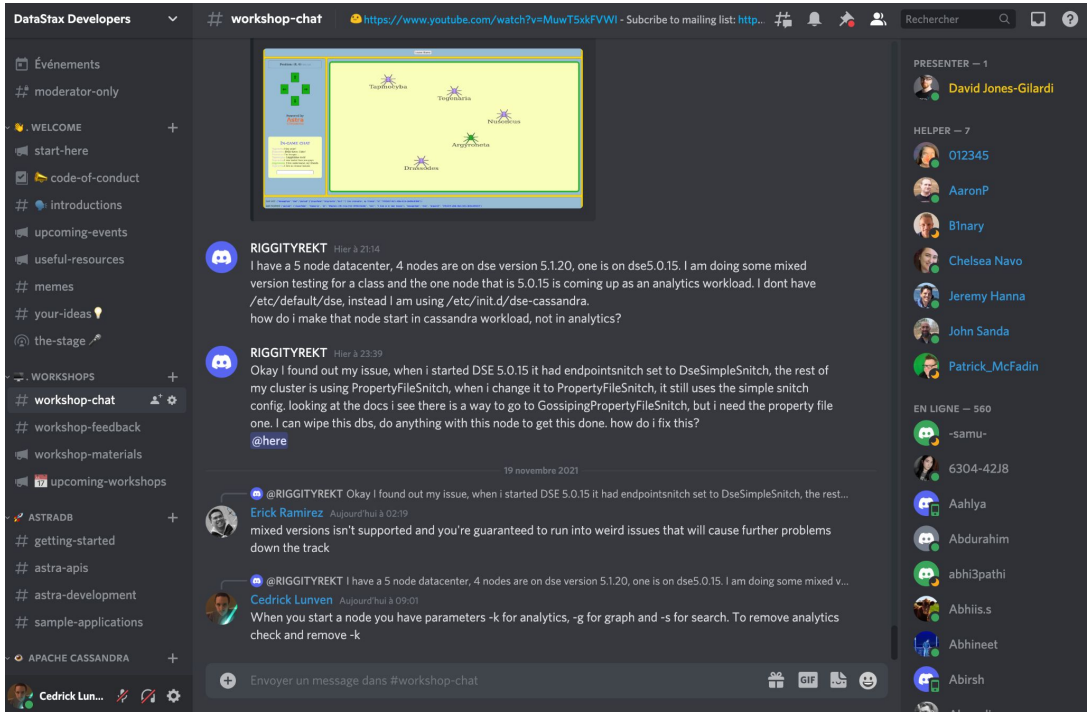
To claim your prize:

Take a screenshot of your Menti screen



NO!

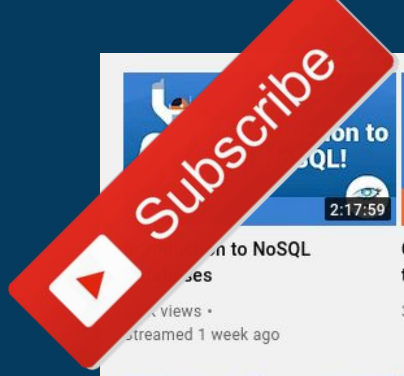
Fill the form at dtsx.io/workshop-swag



!discord

dtsx.io/discord

Datastax Developers Discord (18k+)



...n to NoSQL ...ses

...K views •
...Streamed 1 week ago

Crash Course | Introduction to Cassandra for Developers

331 views • 1 week ago

Introduction to NoSQL Databases

3.5K views •
Streamed 1 week ago

Introduction to NoSQL Databases

10K views •
Streamed 1 week ago

Build your own NETFLIX clone!

4K views •
Streamed 2 weeks ago

Build your own NETFL... clone!

7.4K views •
Streamed 2 weeks ago

Astra Streaming Demo

177 views • 2 weeks ago

Kubernetes Ingress Management with Traefik...

496 views •
Streamed 2 weeks ago

Build your own TikTok clone!

1.9K views •
Streamed 3 weeks ago

Build your own TikTok Clone!

4K views •
Streamed 3 weeks ago

How to use the Connect Driver in Astra DB

113 views • 4 weeks ago

How to use the CQL Console in Astra DB

39 views • 4 weeks ago

How to create an Authentication Token in...

37 views • 4 weeks ago

How to use the Data Loader in Astra DB

62 views • 4 weeks ago

Astra DB Sample App Gallery

36 views • 4 weeks ago

How to use Secure Connect in Astra DB

42 views • 4 weeks ago

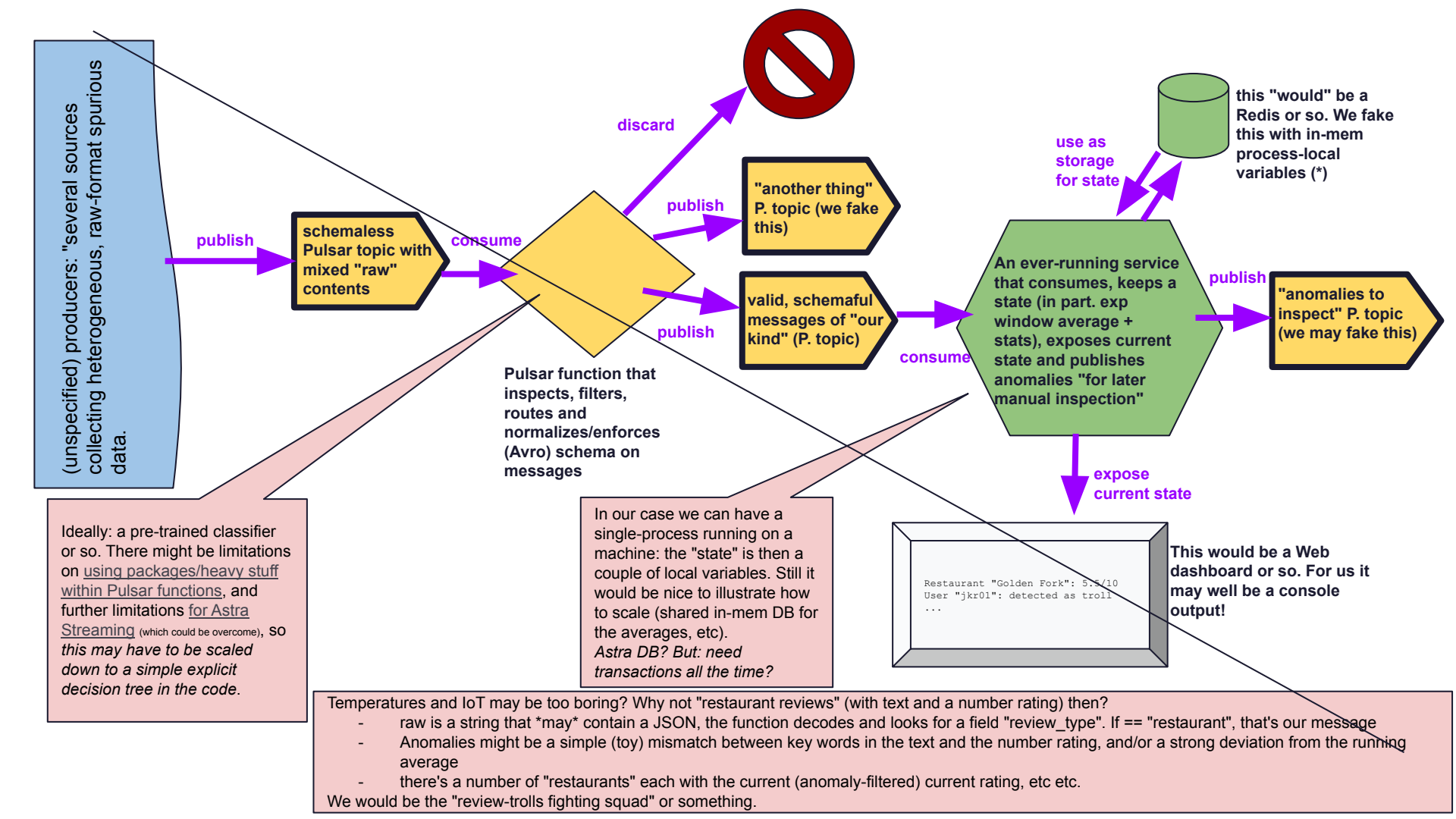
Cassandra Day India: CL Room (Workshops)

2.4K views •
Streamed 4 weeks ago

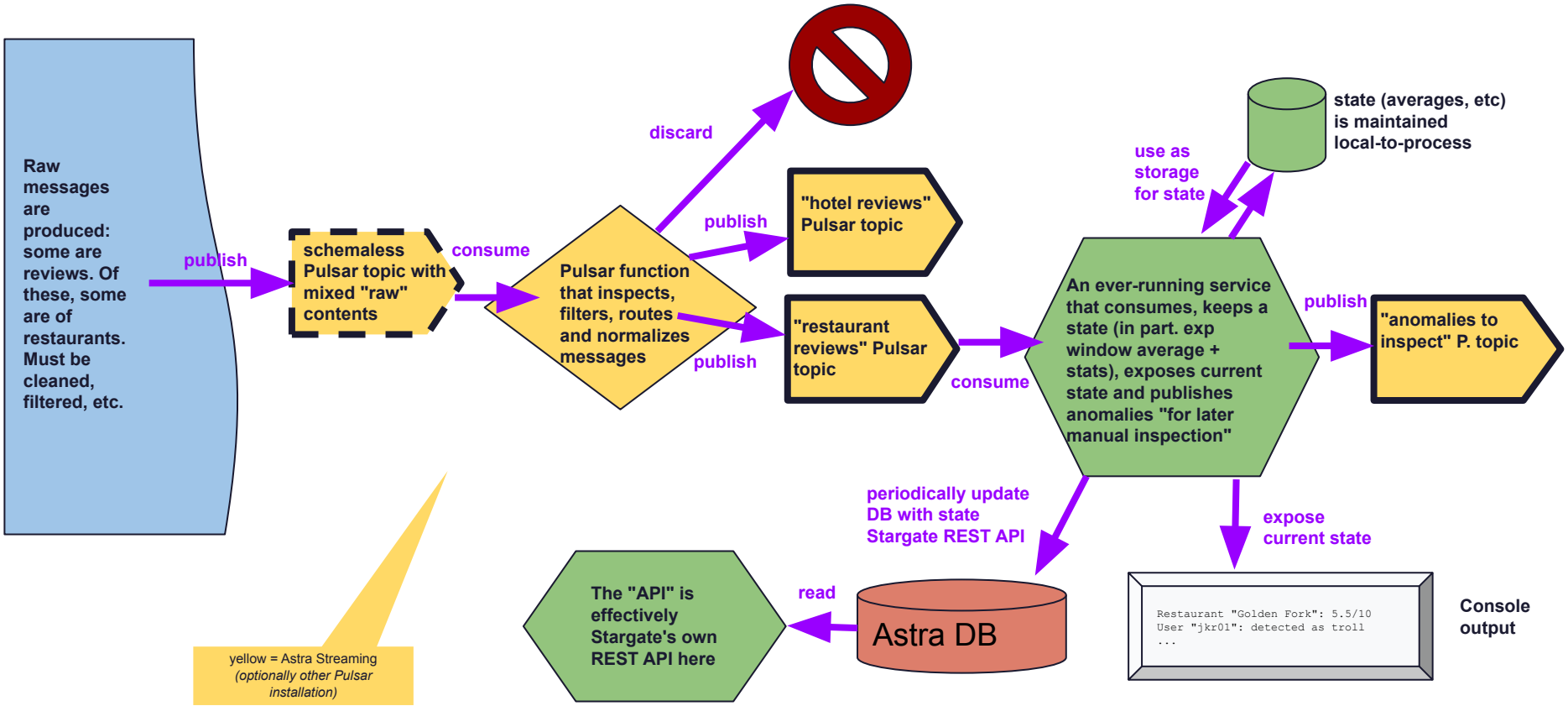
Cassandra Day India: RF Room (Talks)

1.3K views •
Streamed 1 month ago

Thank You!

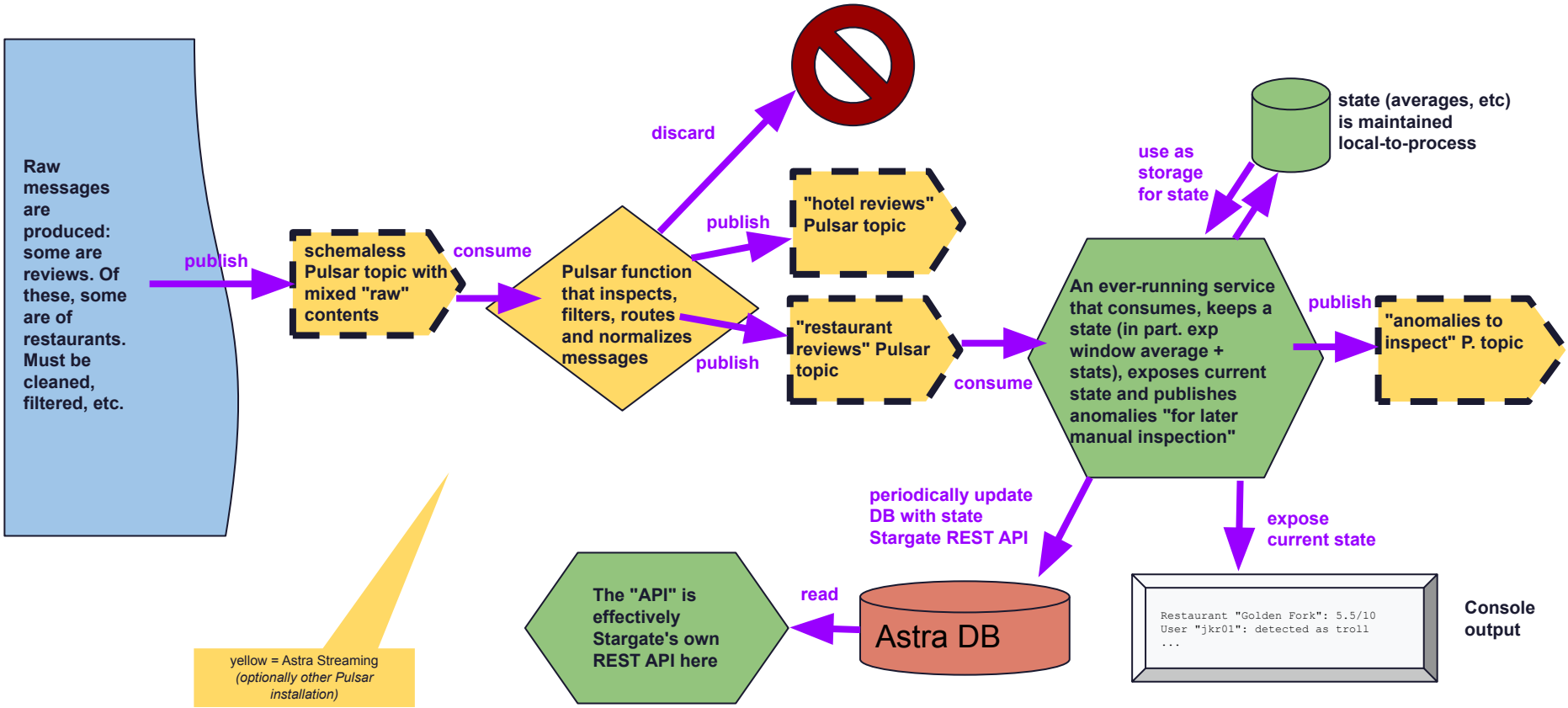

(unspecified) producers: "several sources collecting heterogeneous, raw-format spurious data.
publish
schemaless Pulsar topic with mixed "raw" contents
consume
discard
publish
"another thing" P. topic (we fake this)
publish
valid, schemaful messages of "our kind" (P. topic)
Pulsar function that inspects, filters, routes and normalizes/enforces (Avro) schema on messages
use as storage for state
this "would" be a Redis or so. We fake this with in-mem process-local variables (*)
consume
An ever-running service that consumes, keeps a state (in part. exp window average + stats), exposes current state and publishes anomalies "for later manual inspection"
publish
"anomalies to inspect" P. topic (we may fake this)
expose current state
Restaurant "Golden Fork": 5.5/10
User "jkr01": detected as troll
...
This would be a Web dashboard or so. For us it may well be a console output!
Ideally: a pre-trained classifier or so. There might be limitations on using packages/heavy stuff within Pulsar functions, and further limitations for Astra Streaming (which could be overcome), so this may have to be scaled down to a simple explicit decision tree in the code.
In our case we can have a single-process running on a machine: the "state" is then a couple of local variables. Still it would be nice to illustrate how to scale (shared in-mem DB for the averages, etc).
Astra DB? But: need transactions all the time?
Temperatures and IoT may be too boring? Why not "restaurant reviews" (with text and a number rating) then?
- raw is a string that *may* contain a JSON, the function decodes and looks for a field "review_type". If == "restaurant", that's our message
- Anomalies might be a simple (toy) mismatch between key words in the text and the number rating, and/or a strong deviation from the running average
- there's a number of "restaurants" each with the current (anomaly-filtered) current rating, etc etc.
We would be the "review-trolls fighting squad" or something.

Goal Architecture



DataStax

Current Architecture



DataStax

Data Layer: Architecture