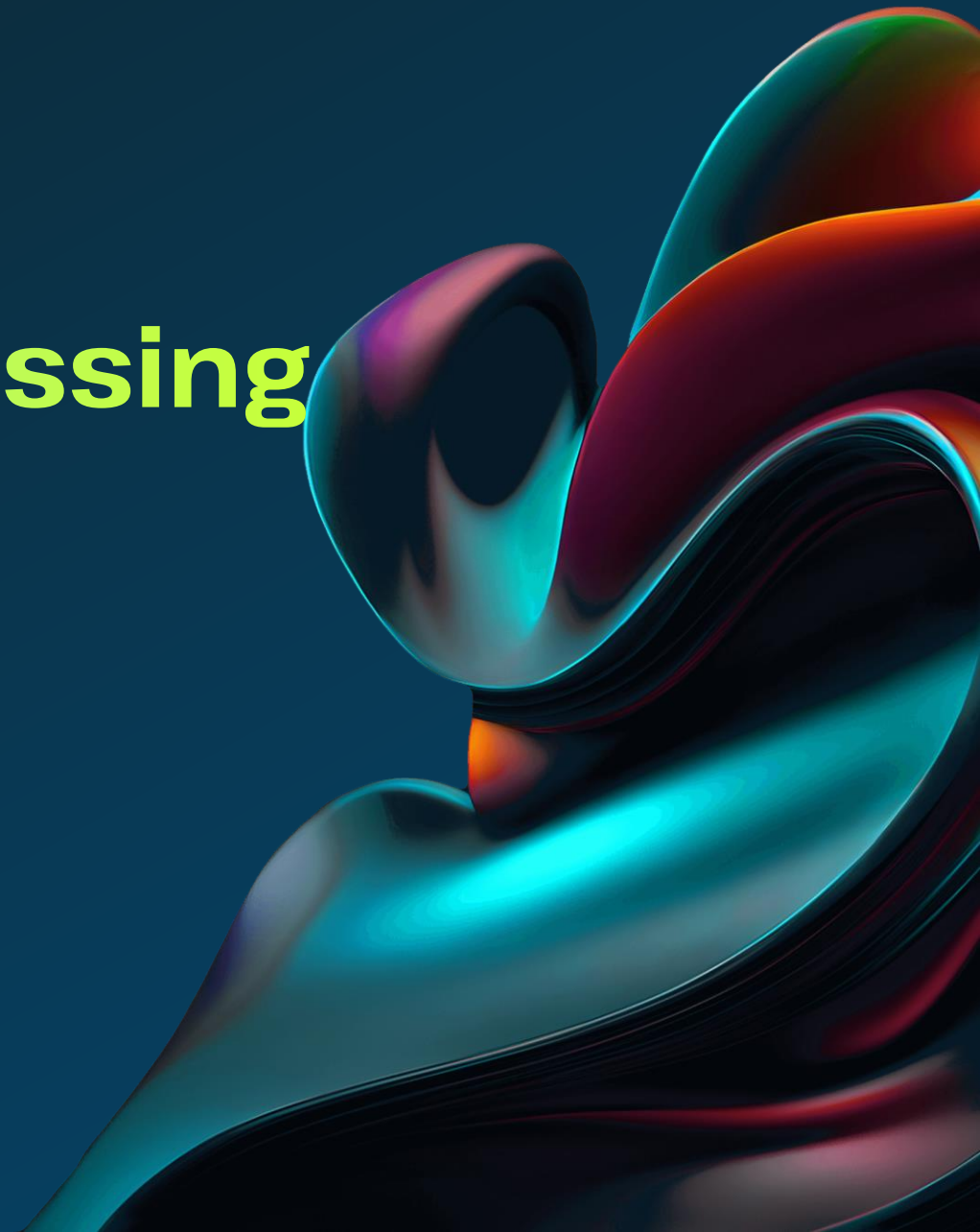
HAZELCAST

Real-time Stream Processing without Migraines



Fawaz Ghali, PhD
Principal Data Science Architect
Head of Developer Relations
@fawazghali

Scan me to win $100

Check In

Data != Context

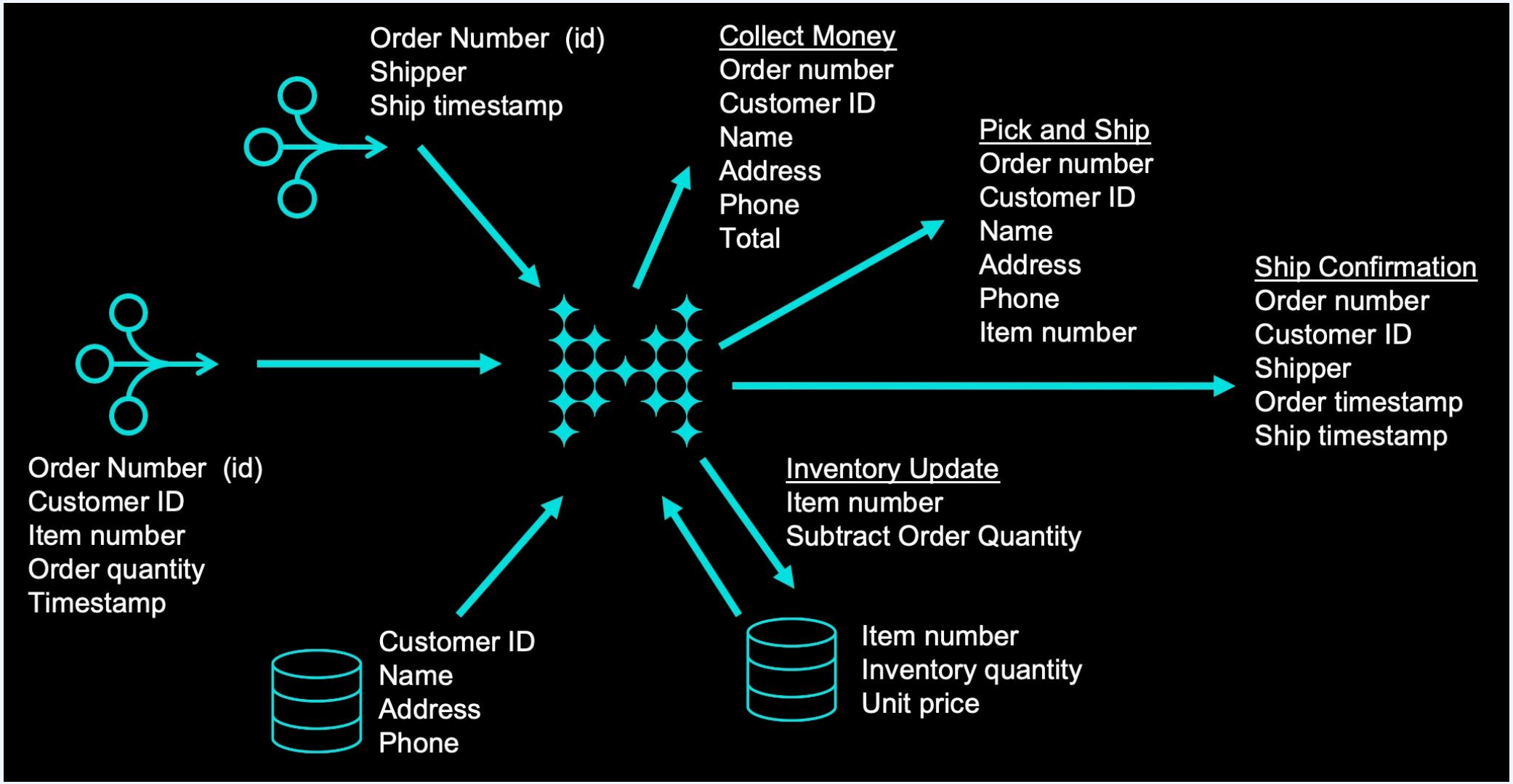
Time Decay

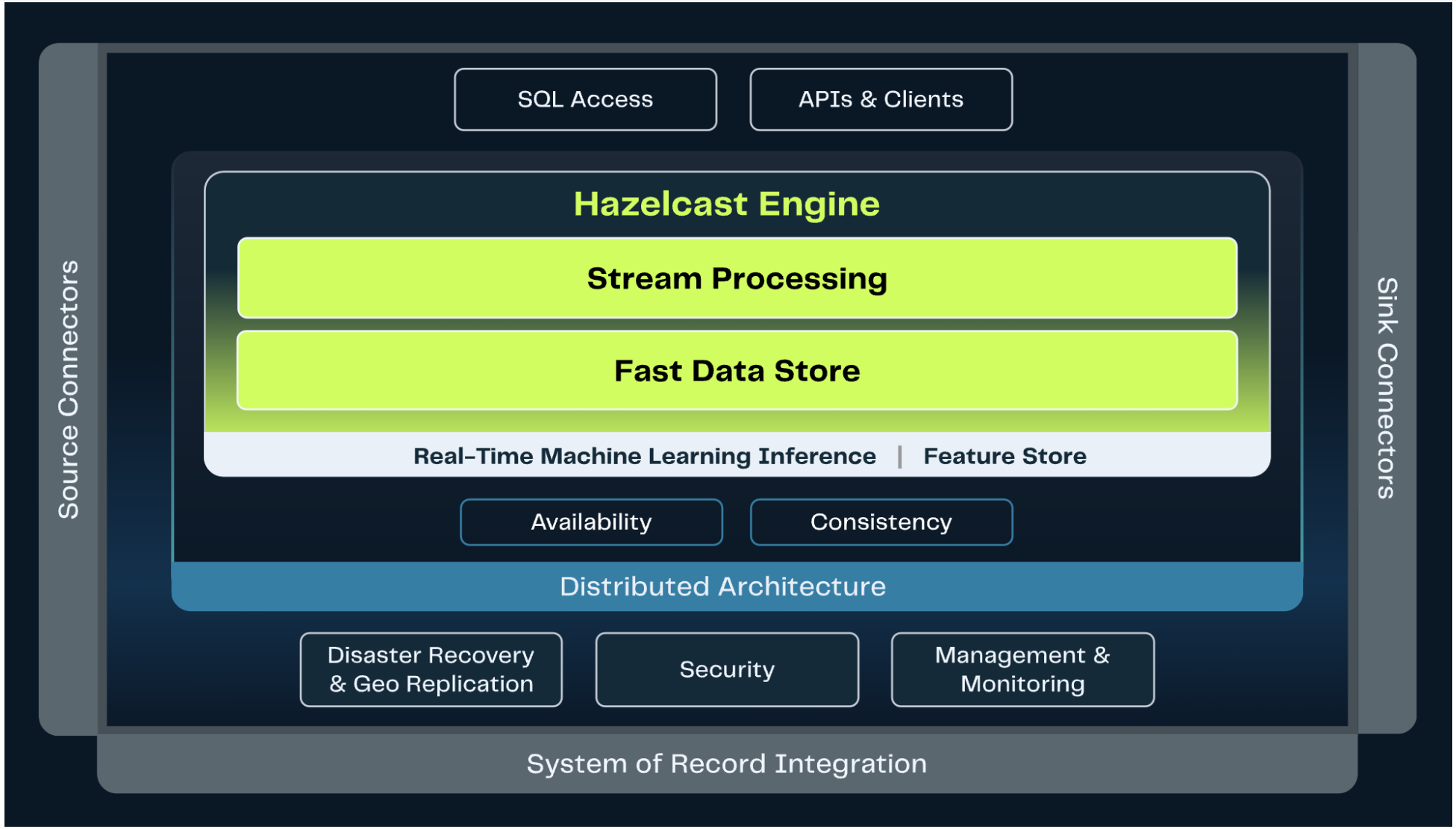
The longer you wait...

The less value the information has

How fast is too fast?

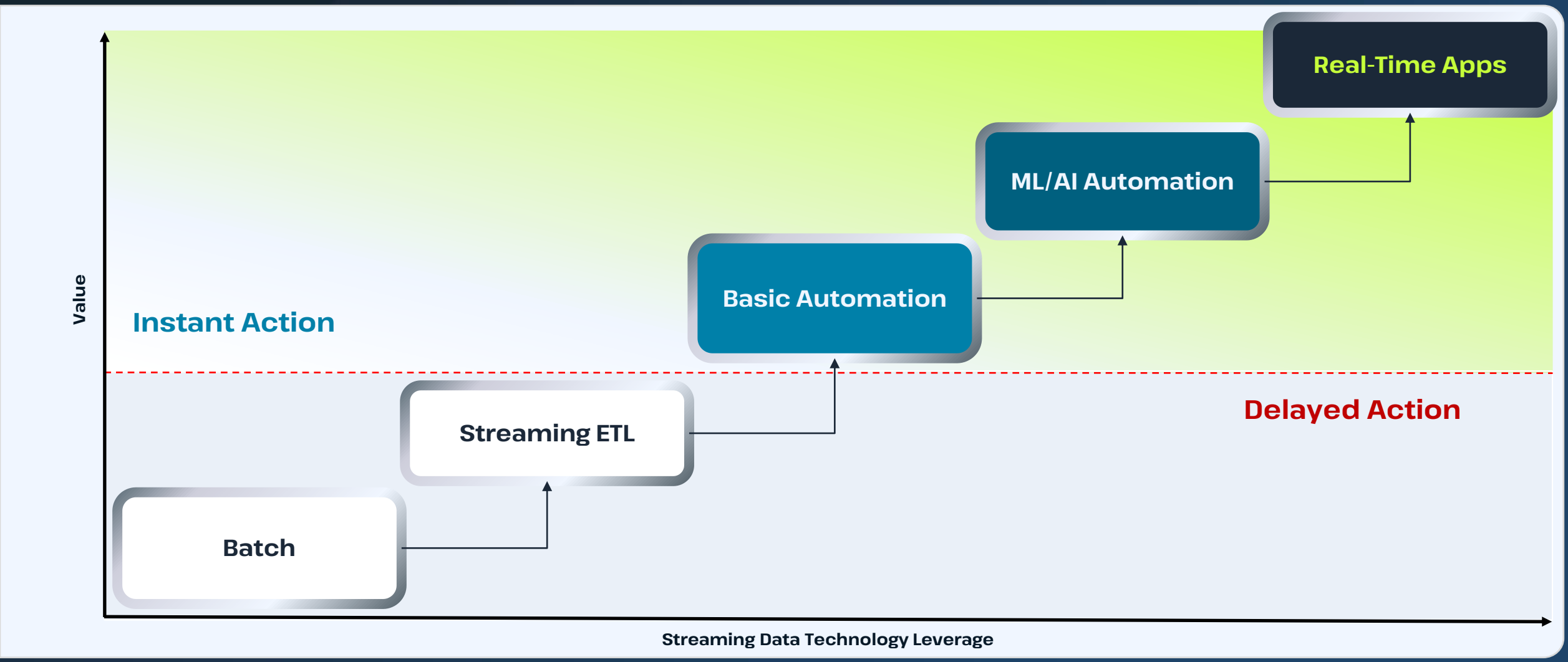
Let's build a stream-processing app!




SQL Access
APIs & Clients
Hazelcast Engine
Stream Processing
Fast Data Store
Real-Time Machine Learning Inference | Feature Store
Availability
Consistency
Distributed Architecture
Disaster Recovery & Geo Replication
Security
Management & Monitoring
Source Connectors
Sink Connectors
System of Record Integration

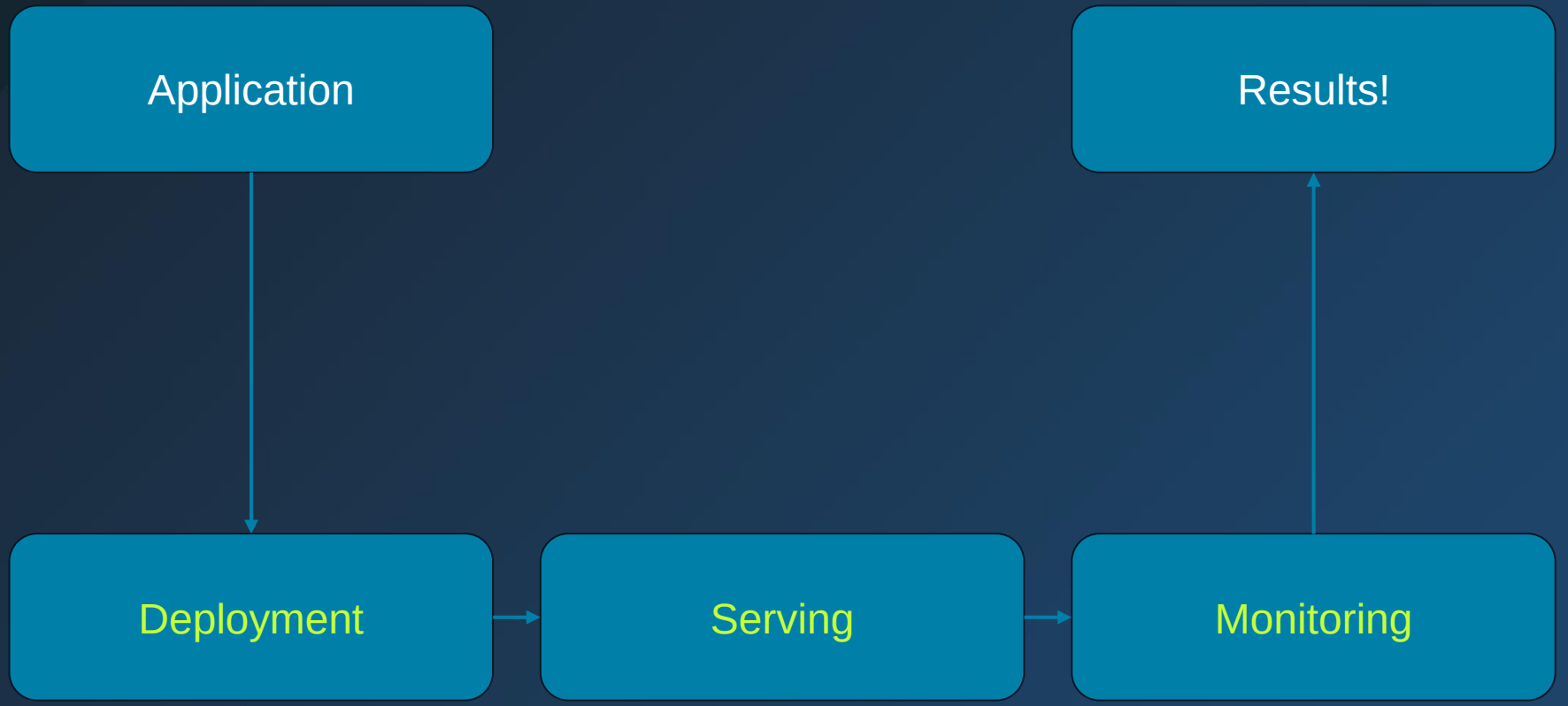
Real time is:

- **While** a customer is banking
- **When** a customer is shopping
- **During** a fraudulent event
- **Before** a process breaks
- **While** travelers are enroute


Real-Time Apps
ML/AI Automation
Basic Automation
Streaming ETL
Batch
Instant Action
Delayed Action
Value
Streaming Data Technology Leverage

I have the **perfect** application

But...


Application
Results!
Deployment
Serving
Monitoring

Where is my data?

- Data is deserialized/serialized in many places:
  - Data is written
  - Data is read
  - Data is replicated to a backup node
  - Data is saved to disk and reloaded
  - Data is rebalanced

How many serializations/ deserializations?

- V = map.put(K,V)
- Option A: 2 serializations and 1 deserialization.
- Option B: 3 serializations and 2 deserialization.

High-Performance Real-time

=

Instant computation on both

new and **historical** data

Microservices:

- Are Easier to Build and Enhance

Microservices:

- Are Easier to Build and Enhance
- Are Easier to Deploy

Microservices:

- Are Easier to Build and Enhance
- Are Easier to Deploy
- Are Easier to Maintain, Troubleshoot, and Extend

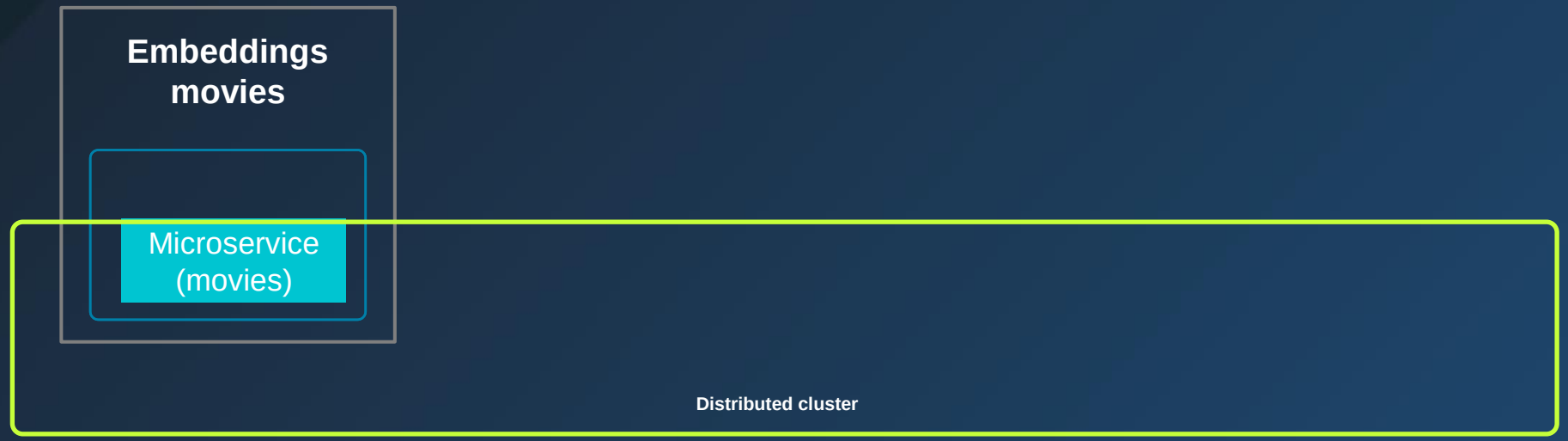
Microservices:

- Are Easier to Build and Enhance
- Are Easier to Deploy
- Are Easier to Maintain, Troubleshoot, and Extend
- Deliver Performance and Scale

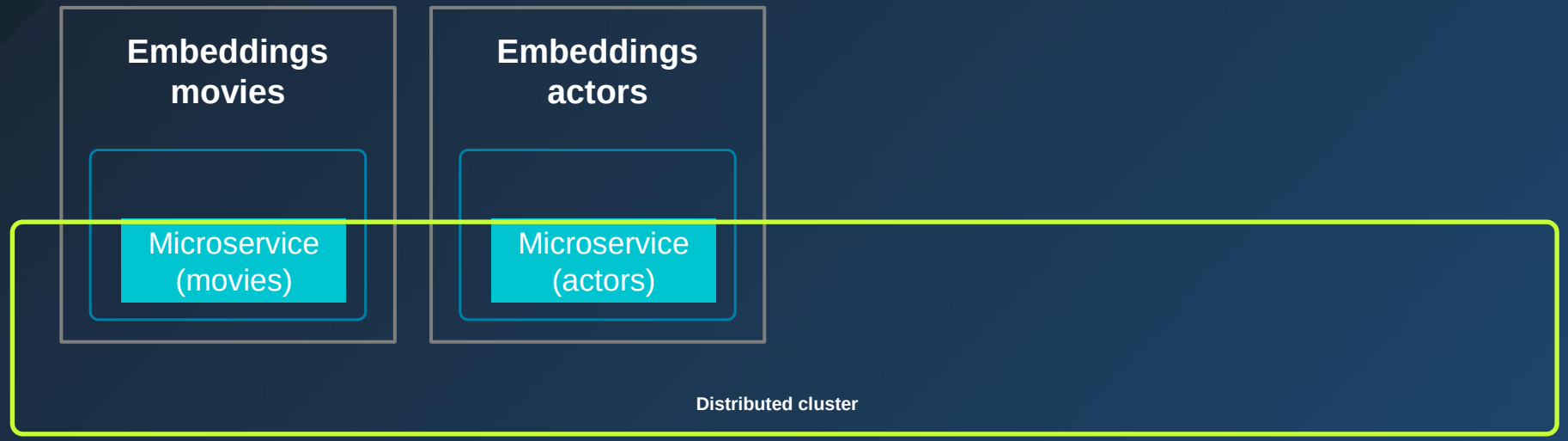
Microservices:

- Are Easier to Build and Enhance
- Are Easier to Deploy
- Are Easier to Maintain, Troubleshoot, and Extend
- Deliver Performance and Scale
- Simplify Real-Time Processing

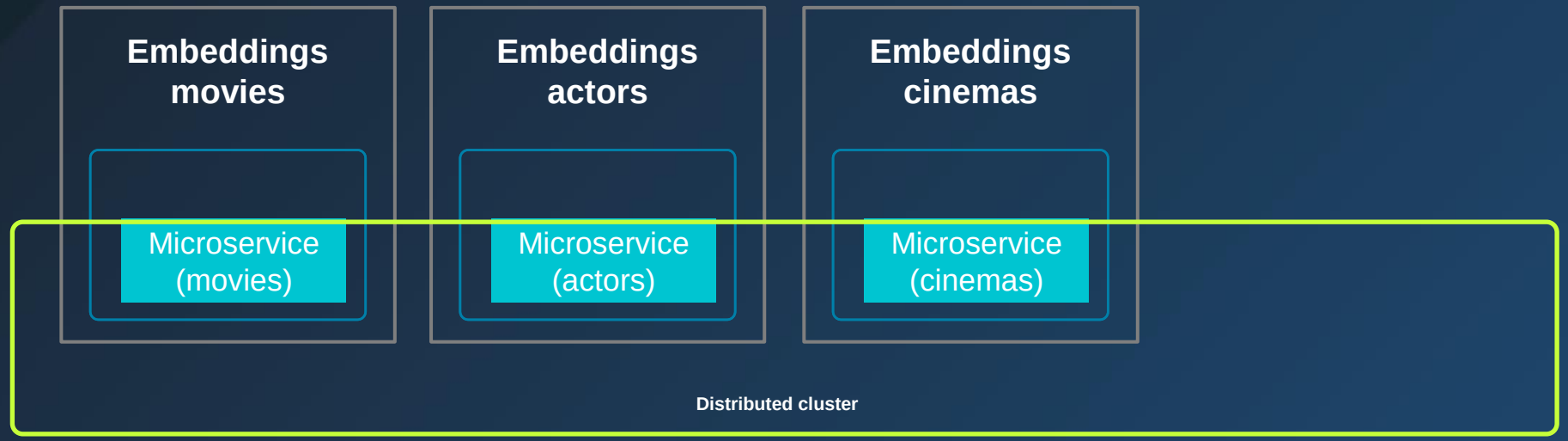
Microservices Pattern for Similarity Search and Stream Processing



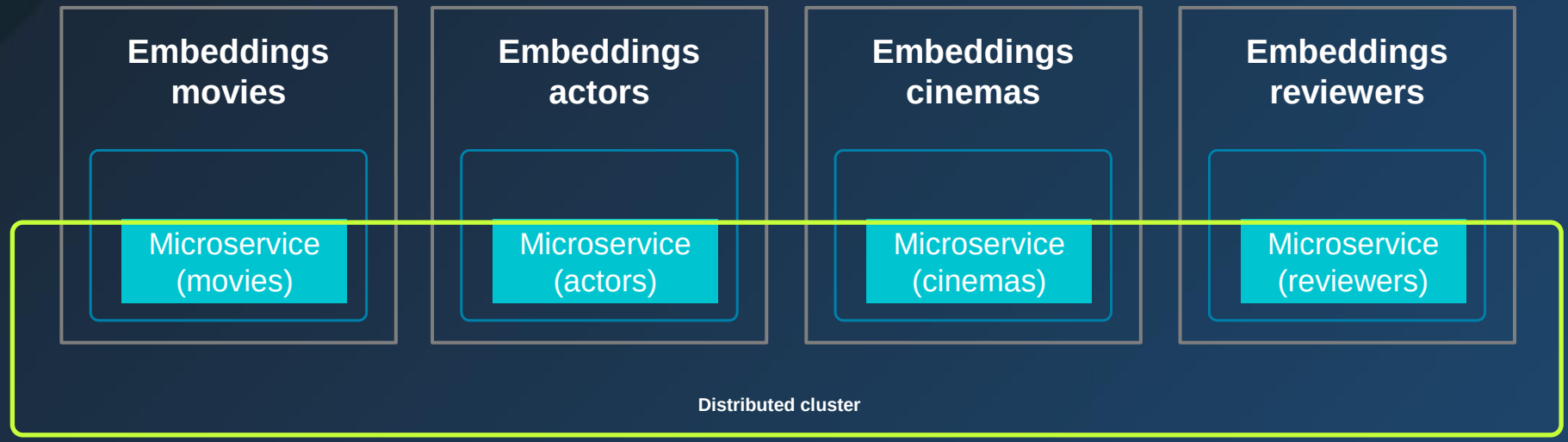
Microservices Pattern for Similarity Search and Stream Processing

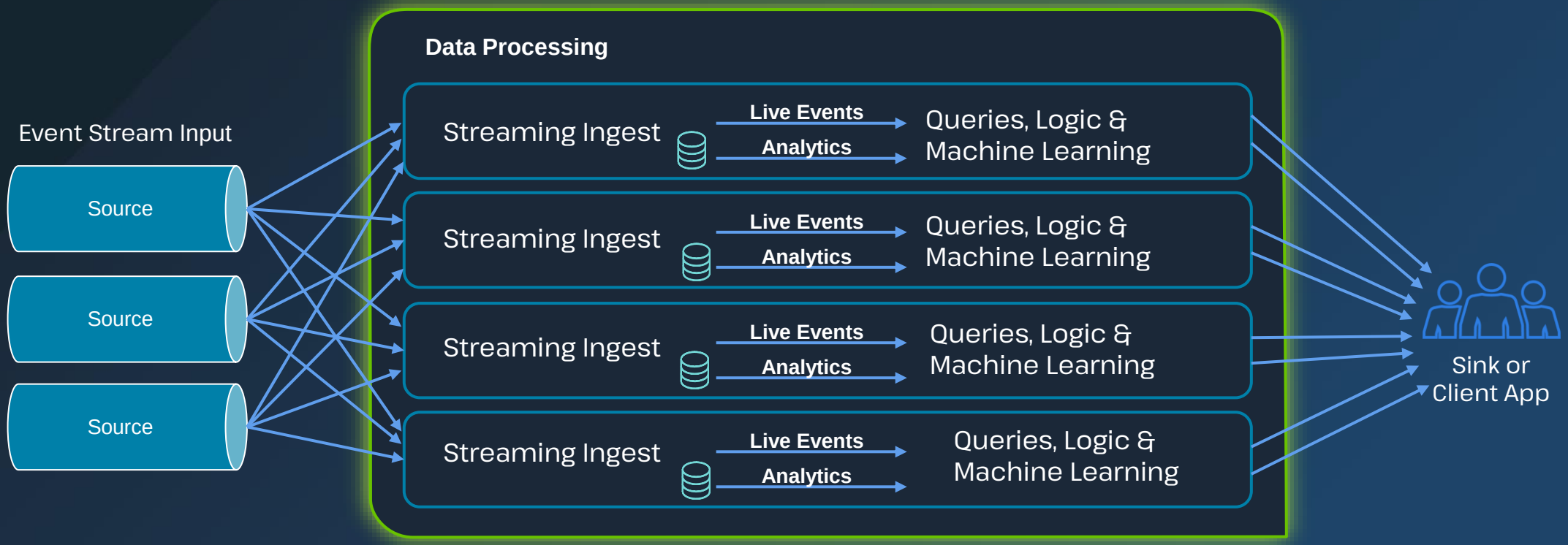


Microservices Pattern for Similarity Search and Stream Processing



Microservices Pattern for Similarity Search and Stream Processing




Event Stream Input
Source
Source
Source
Data Processing
Streaming Ingest
Live Events
Analytics
Queries, Logic &
Machine Learning
Streaming Ingest
Live Events
Analytics
Queries, Logic &
Machine Learning
Streaming Ingest
Live Events
Analytics
Queries, Logic &
Machine Learning
Streaming Ingest
Live Events
Analytics
Queries, Logic &
Machine Learning
Sink or
Client App

Kafka is great for:

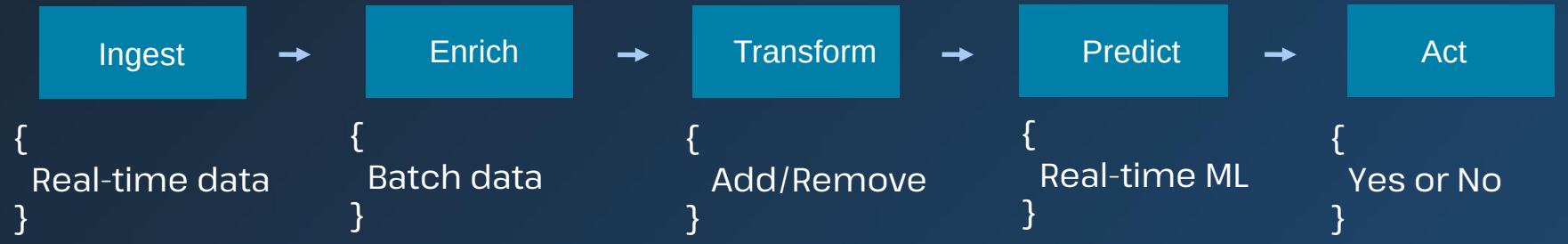
Messaging

Event sourcing

Data pipelines

Time-Based SLA

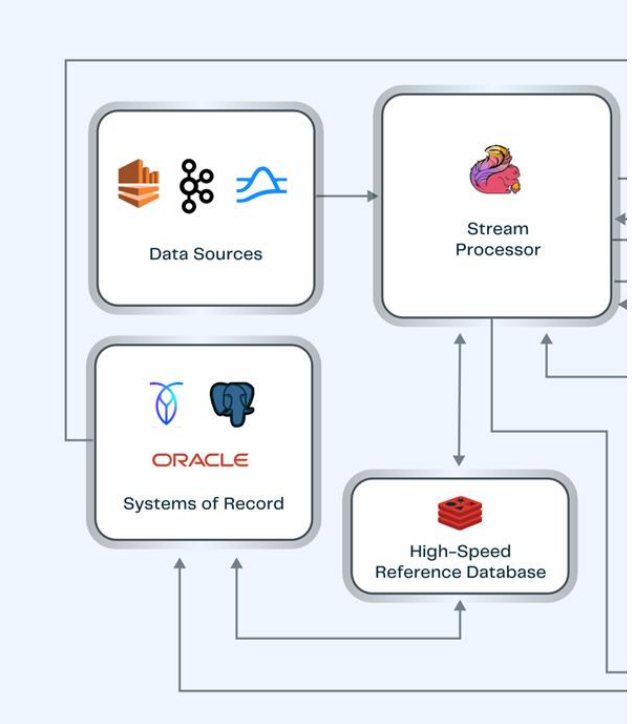
Action - milliseconds **Response**



DIY Approach

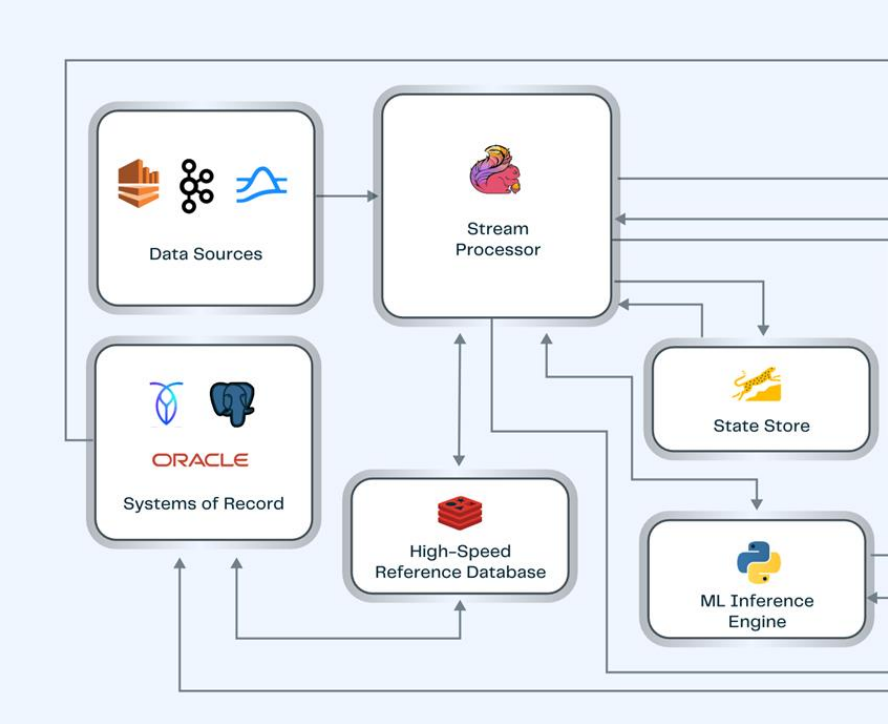
- More integration work

DIY Approach



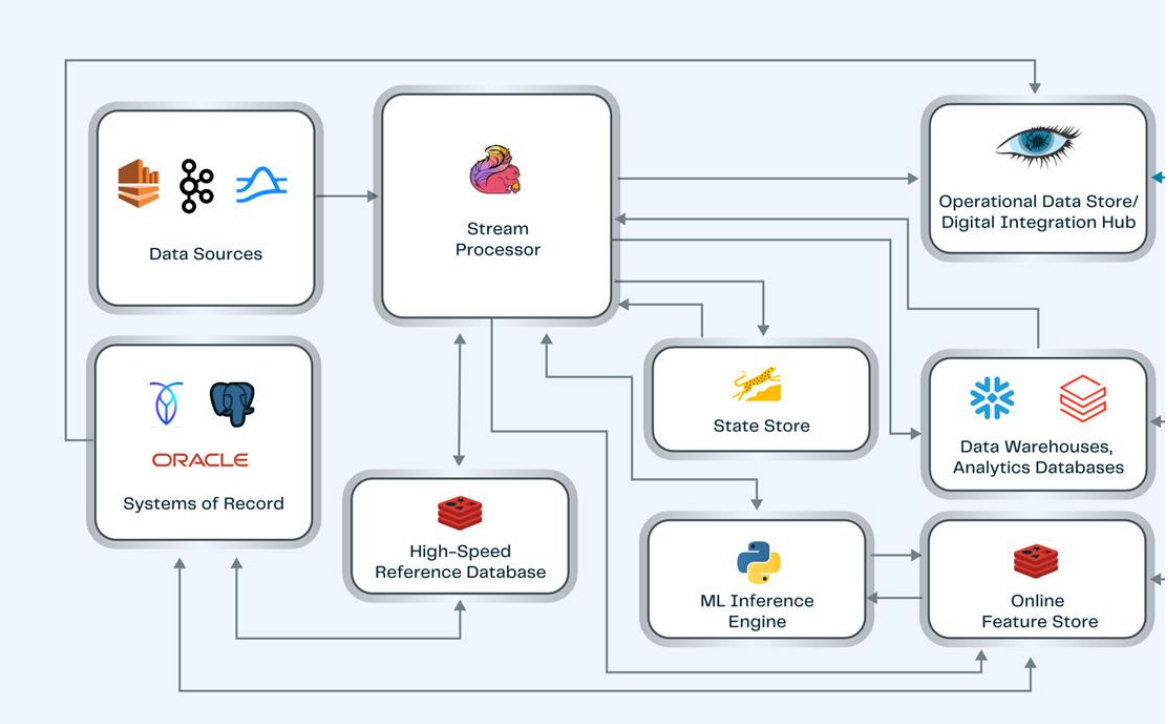
- More integration work
- Higher maintenance

DIY Approach



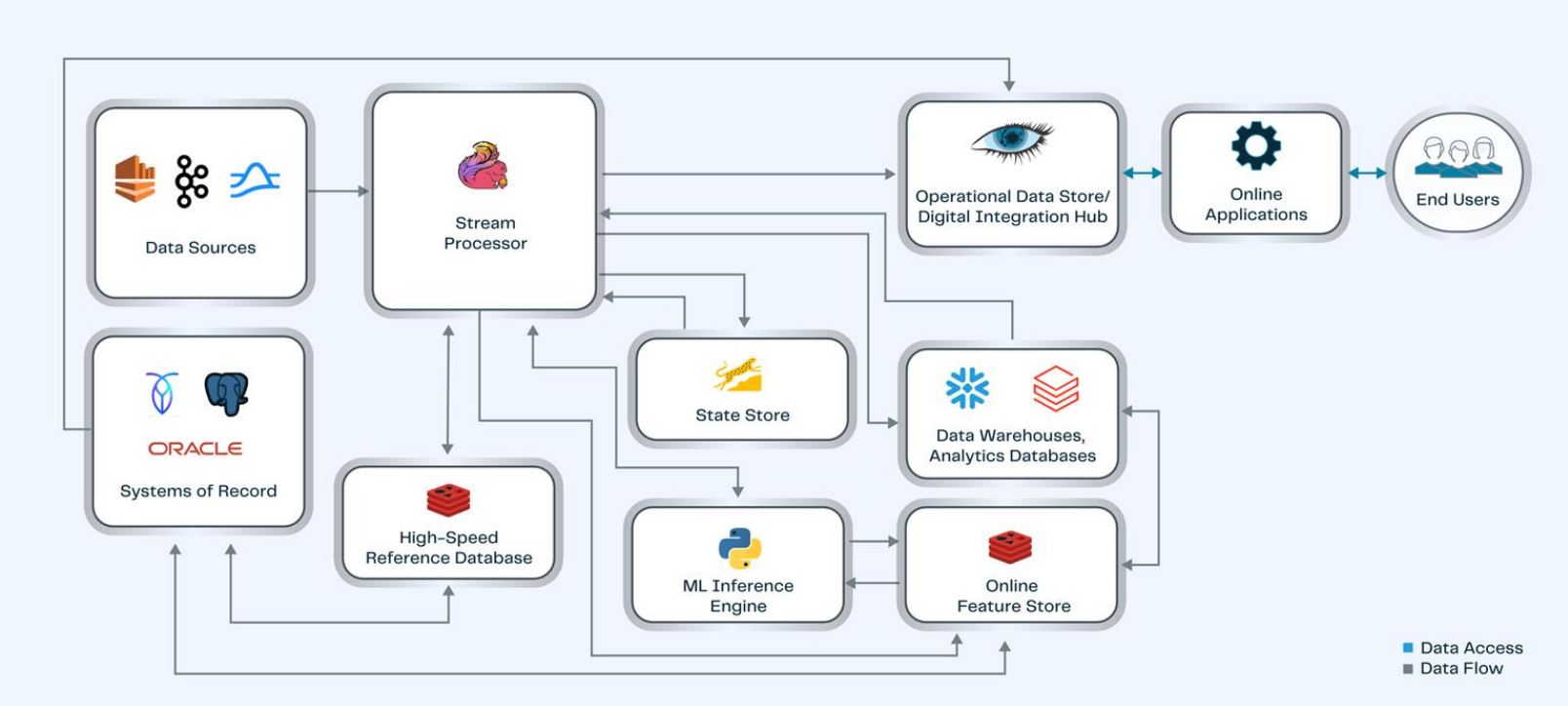
- More integration work
- Higher maintenance
- Higher operational cost

DIY Approach



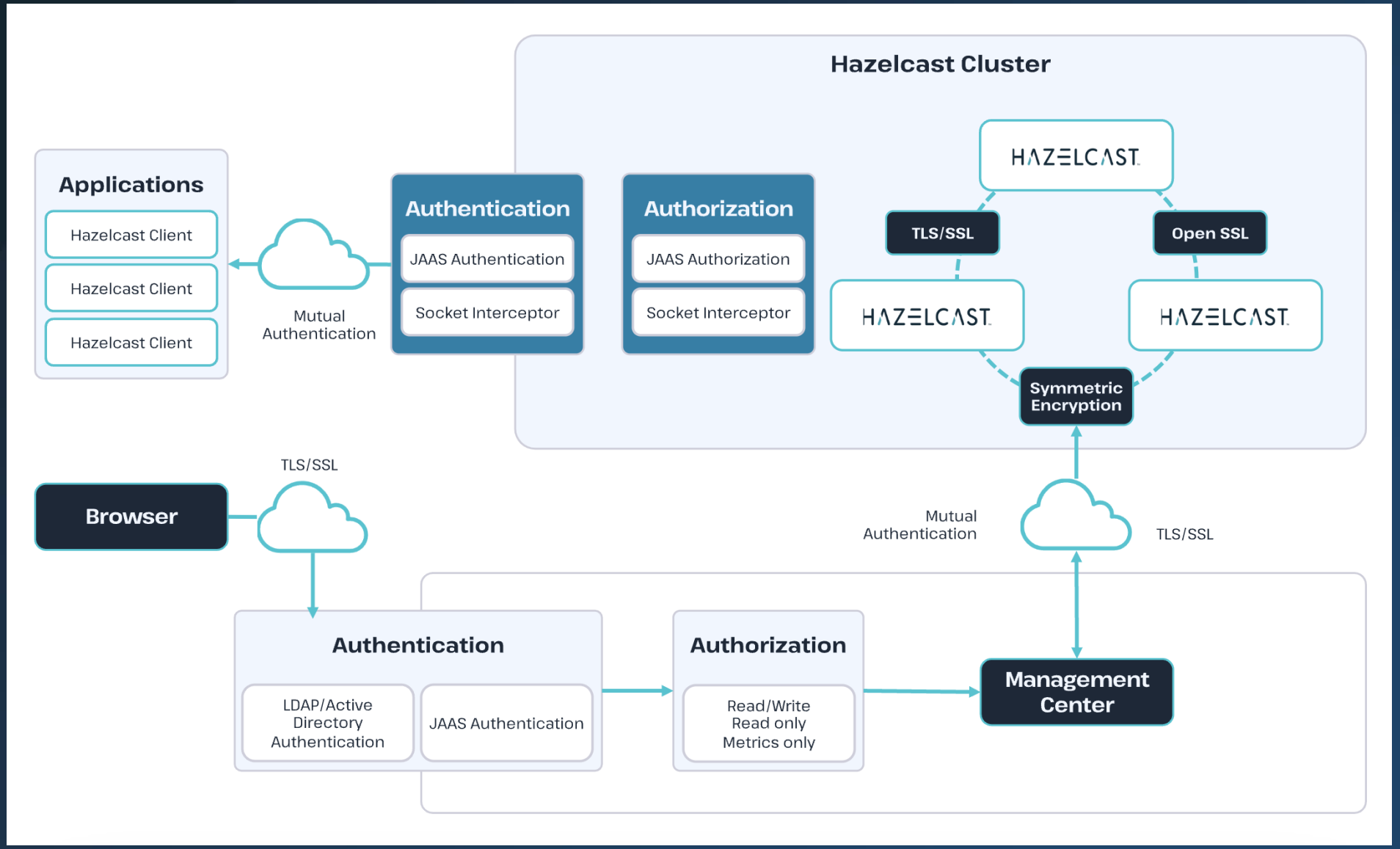
- More integration work
- Higher maintenance
- Higher operational cost
- **Higher Latency**

DIY Approach

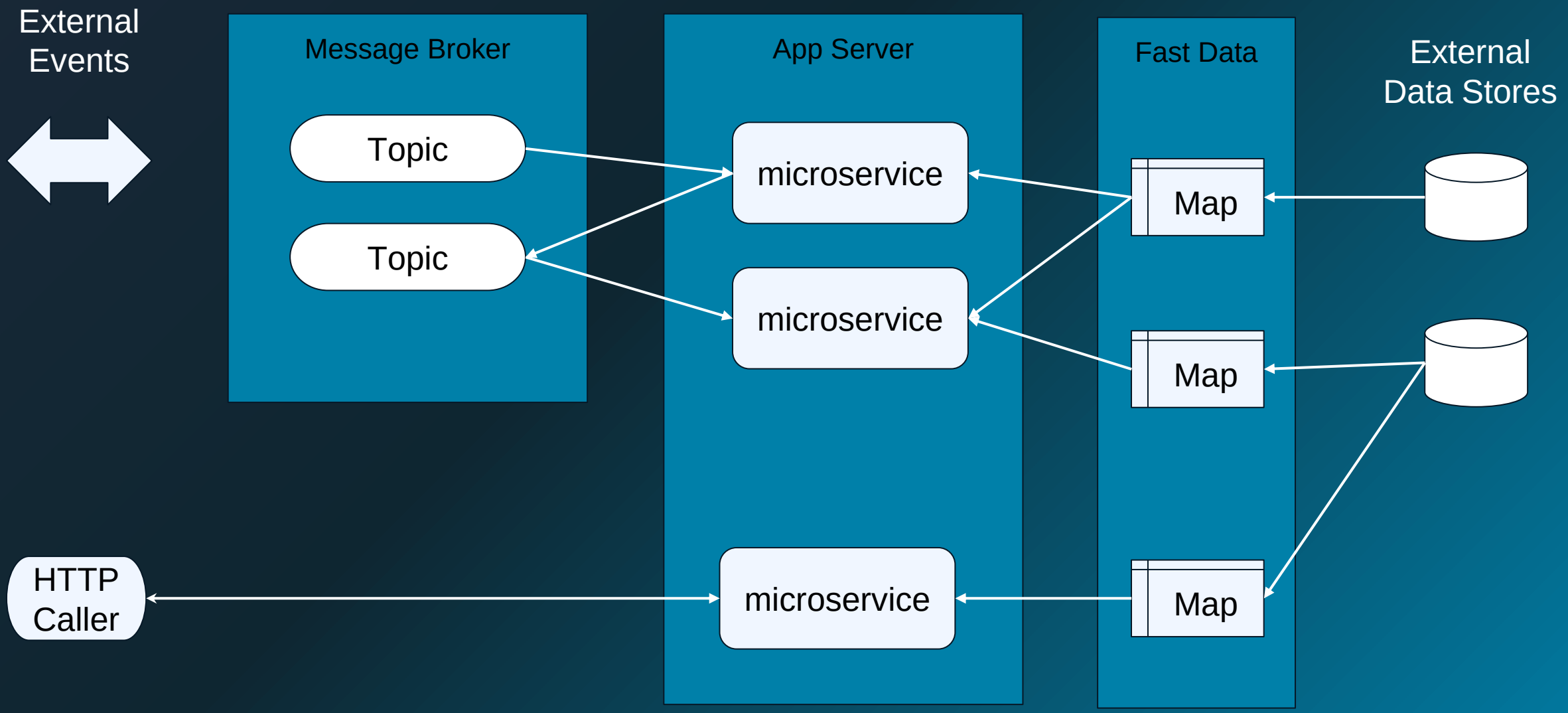


- More integration work
- Higher maintenance
- Higher operational cost
- **Higher Latency**

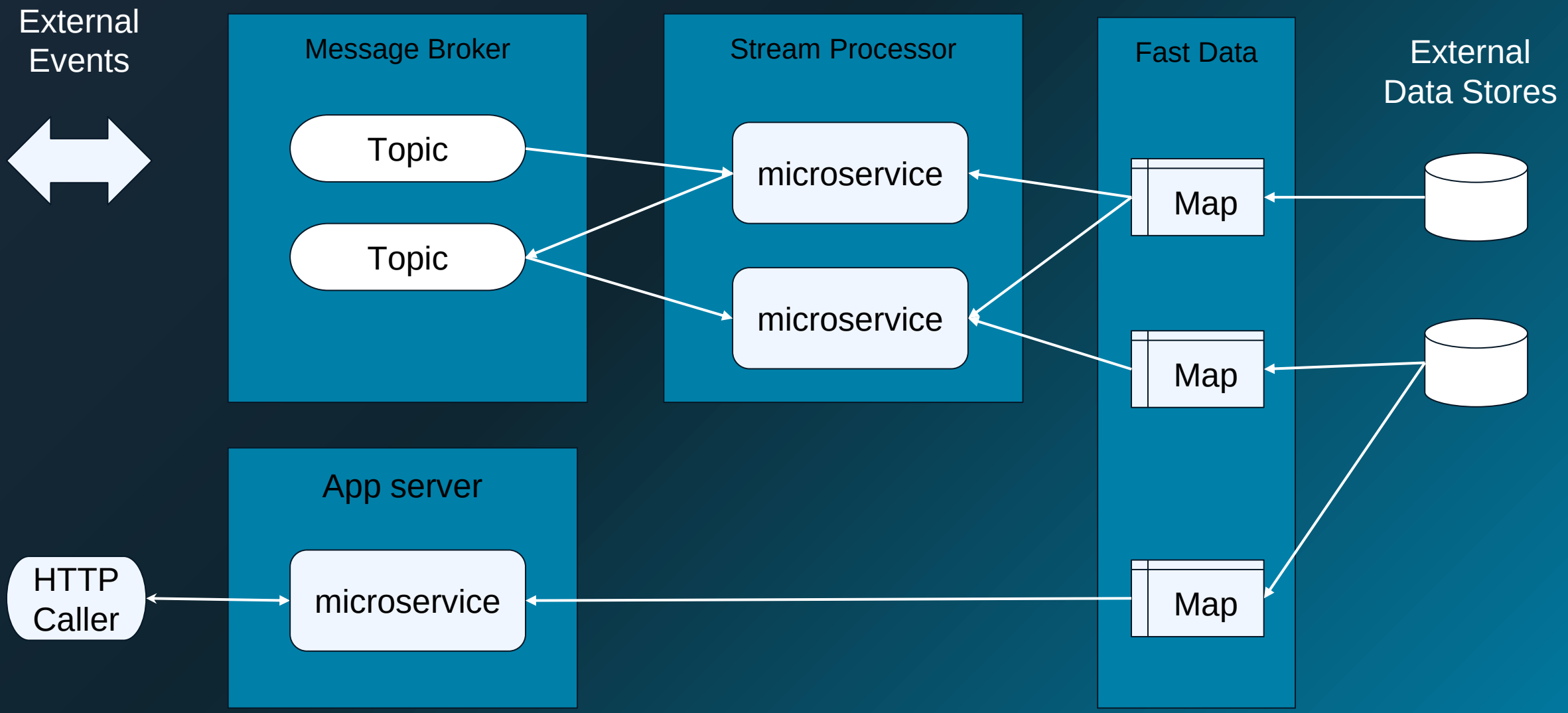
Simplified Architecture


Hazelcast Cluster
Applications
Hazelcast Client
Hazelcast Client
Hazelcast Client
Mutual Authentication
Authentication
JAAS Authentication
Socket Interceptor
Authorization
JAAS Authorization
Socket Interceptor
HAZELCAST
TLS/SSL
Open SSL
HAZELCAST
HAZELCAST
Symmetric Encryption
TLS/SSL
Browser
Mutual Authentication
TLS/SSL
Authentication
LDAP/Active Directory Authentication
JAAS Authentication
Authorization
Read/Write
Read only
Metrics only
Management Center

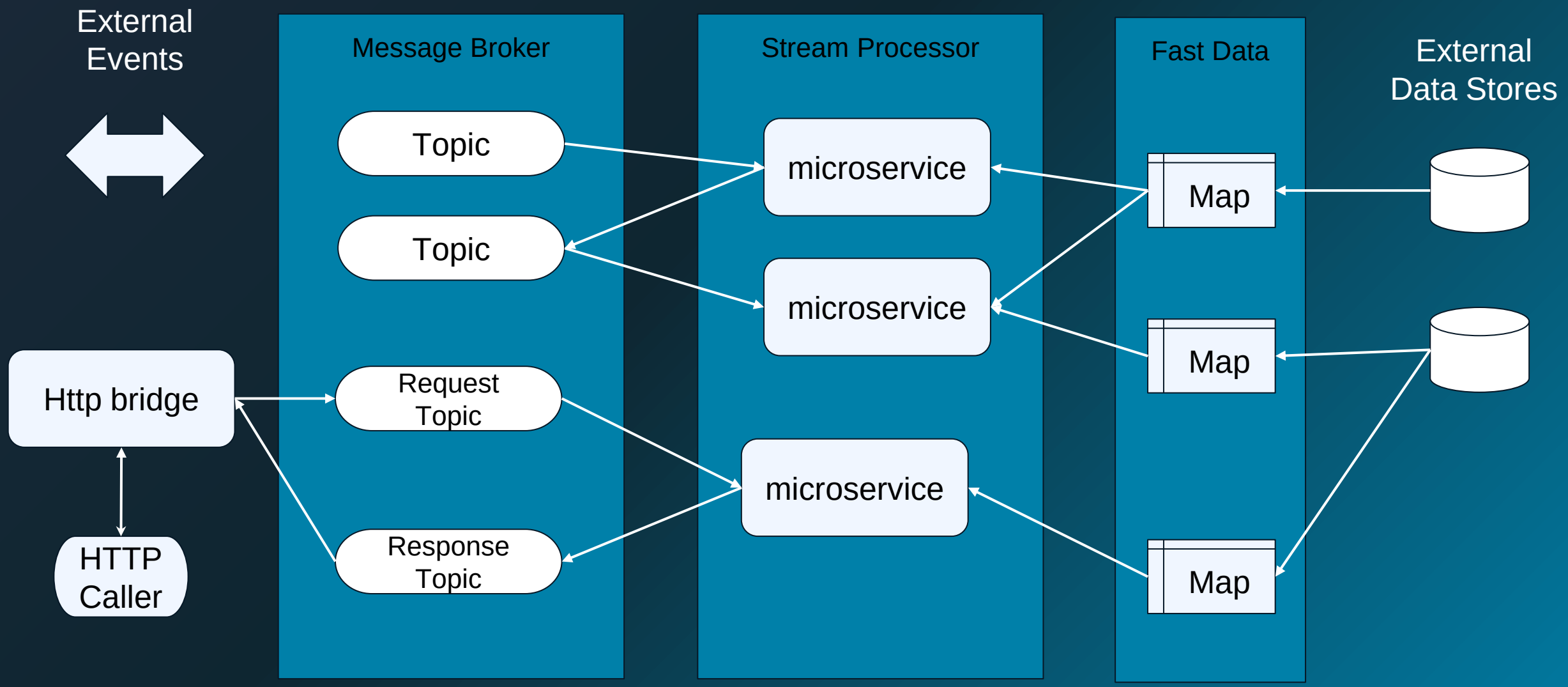
Tech Stack for Handling Events and Services: Option 1



Tech Stack for Handling Events and Services: Option 2

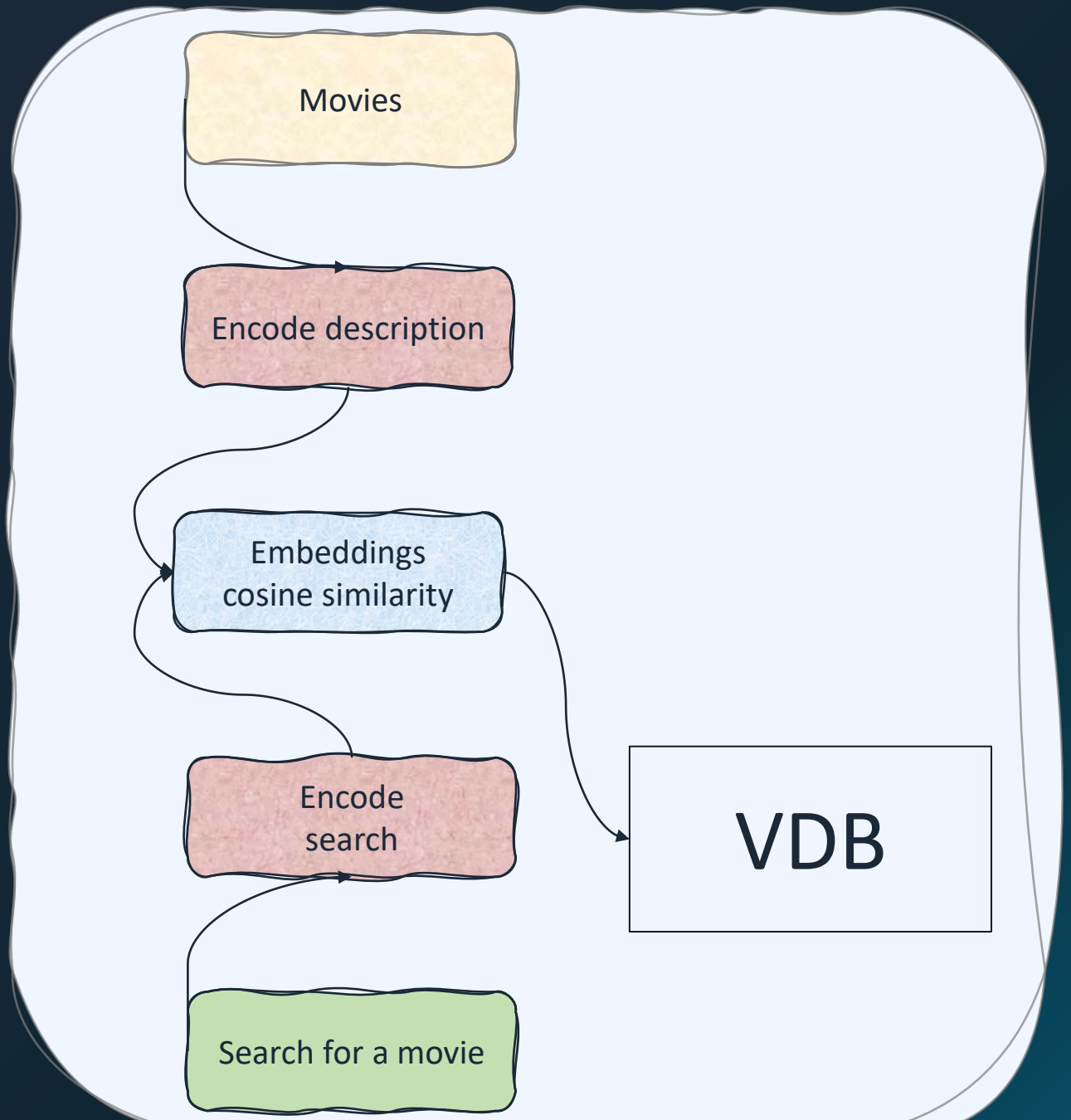


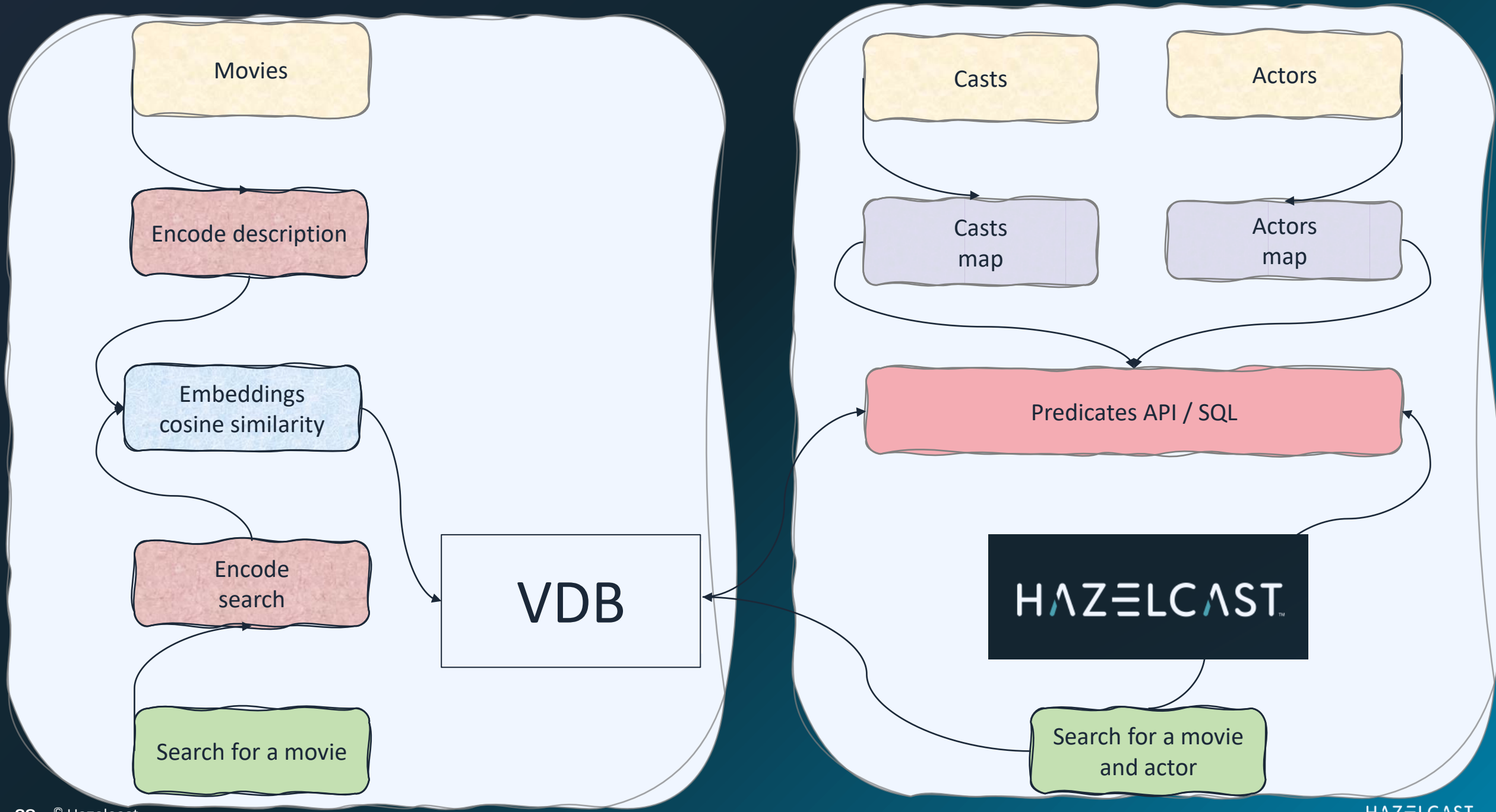
Tech Stack for Handling Events and Services: Option 3



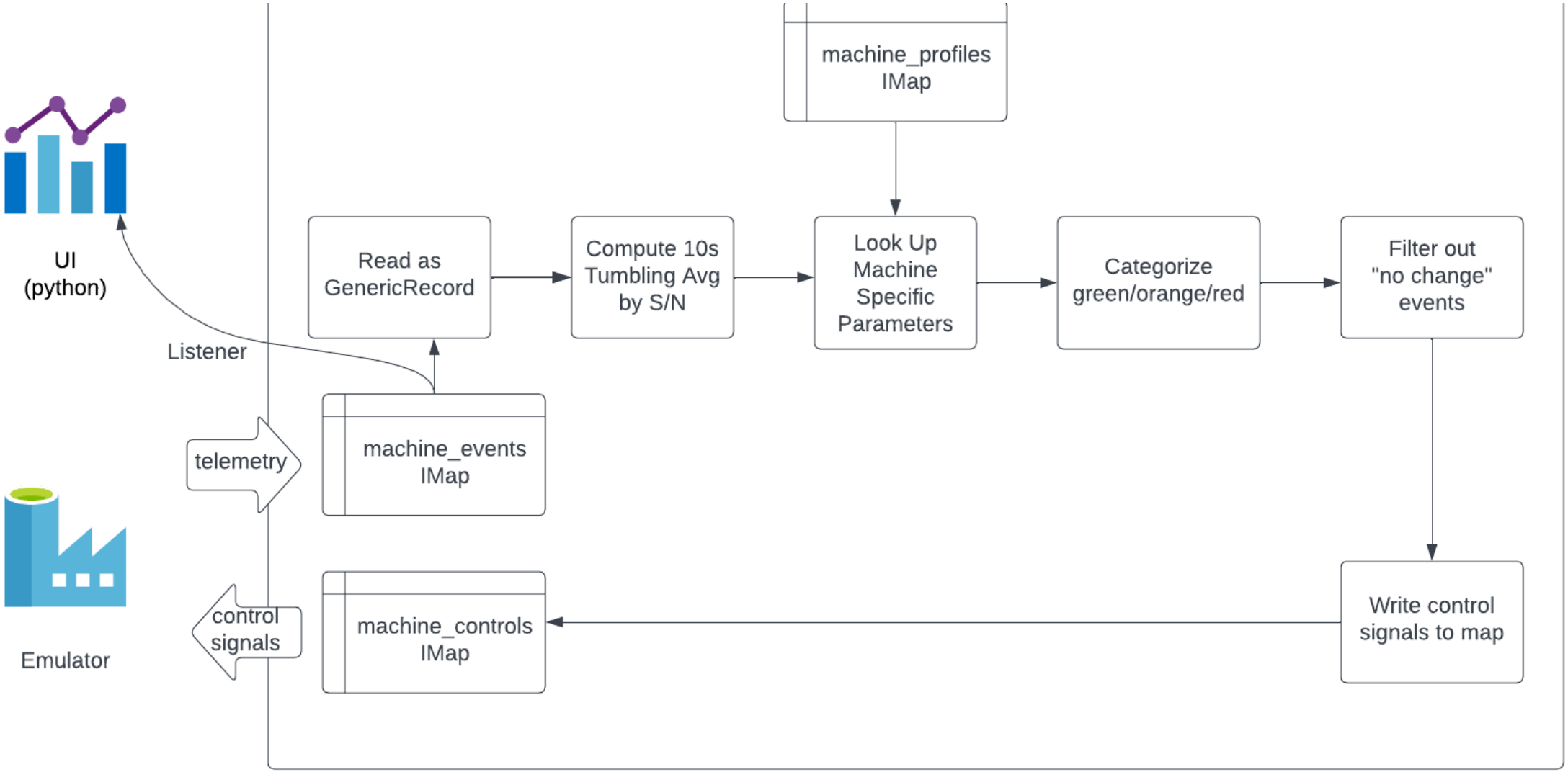
Vector Databases

VDBs lack context and many-to-many relationship representations


Movies
Encode description
Embeddings
cosine similarity
Encode
search
Search for a movie
VDB


Movies
Encode description
Embeddings
cosine similarity
Encode
search
Search for a movie
VDB
Casts
Actors
Casts
map
Actors
map
Predicates API / SQL
HAZELCAST
Search for a movie
and actor

Workflow

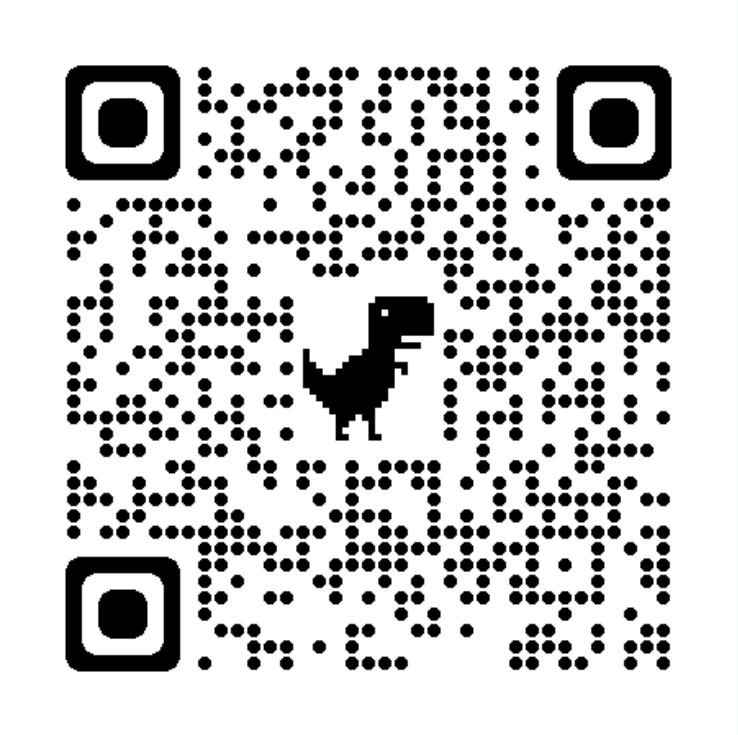

machine_profiles
IMap
UI
(python)
Read as
GenericRecord
Compute 10s
Tumbling Avg
by S/N
Look Up
Machine
Specific
Parameters
Categorize
green/orange/red
Filter out
"no change"
events
Listener
telemetry
machine_events
IMap
Emulator
control
signals
machine_controls
IMap
Write control
signals to map

- Access the UI at http://localhost:8050
- Open up the management center http://localhost:8080

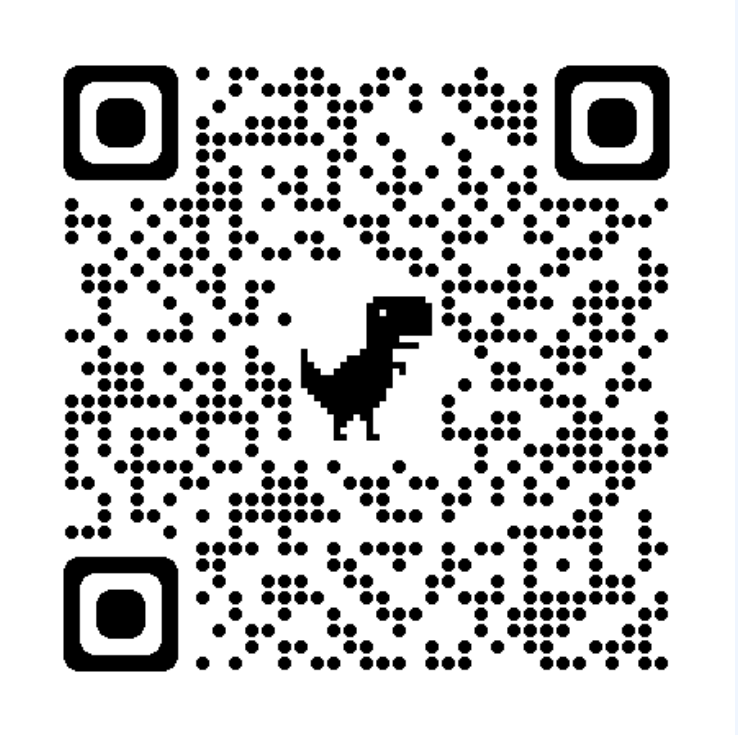
Scan me to win $100

Check In

Join the Community

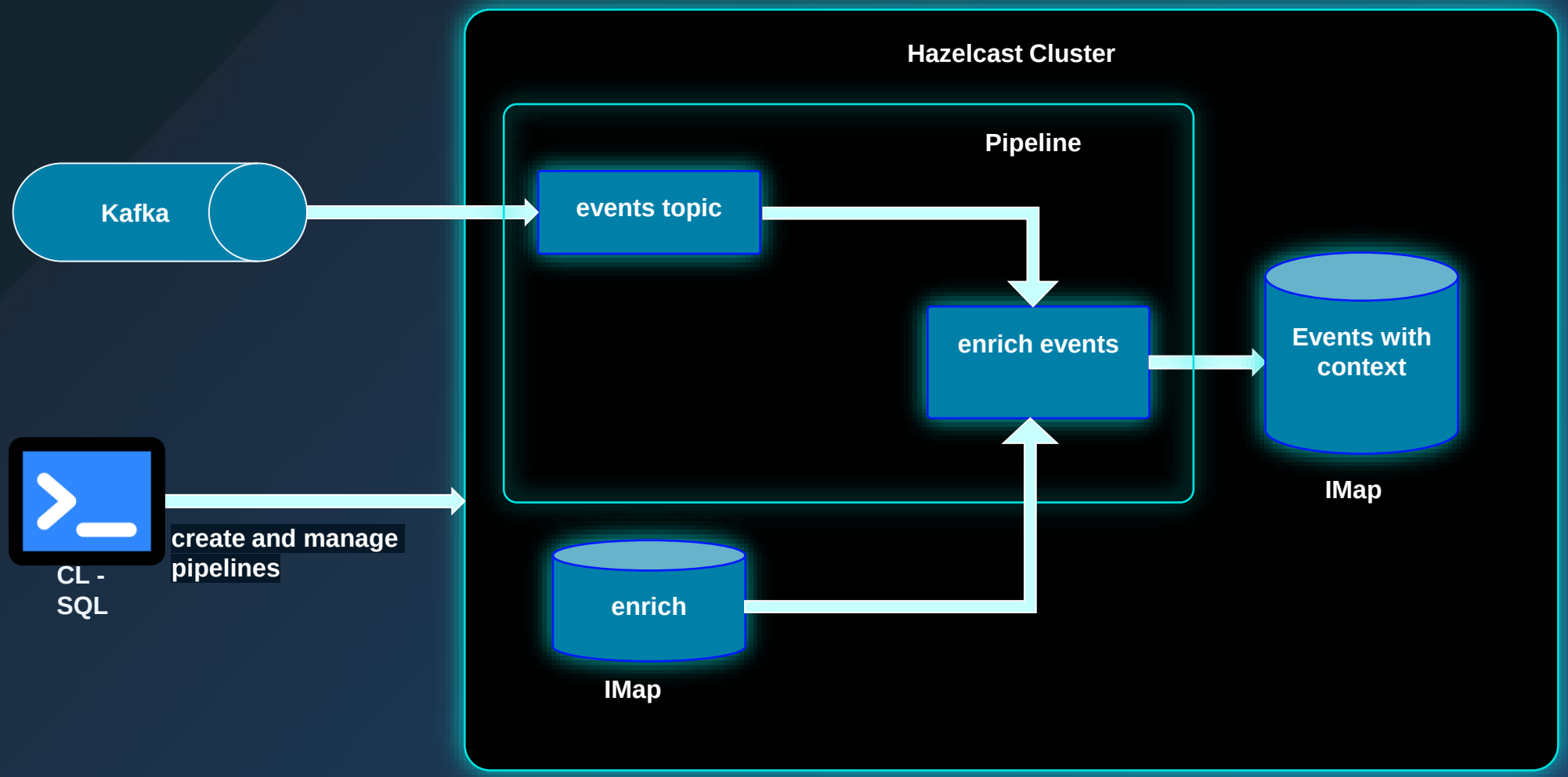


Twitch



Slack

Architecture Overview



Setup

- Install Hazelcast
  - **hz-start**
  - **hz-cli sql**
- Install Kafka
  - cd Documents/kafka_2.13-3.4.0
  - **bin/zookeeper-server-start.sh config/zookeeper.properties**
  - **bin/kafka-server-start.sh config/server.properties**
  - bin/kafka-server-stop.sh
  - bin/zookeeper-server-stop.sh