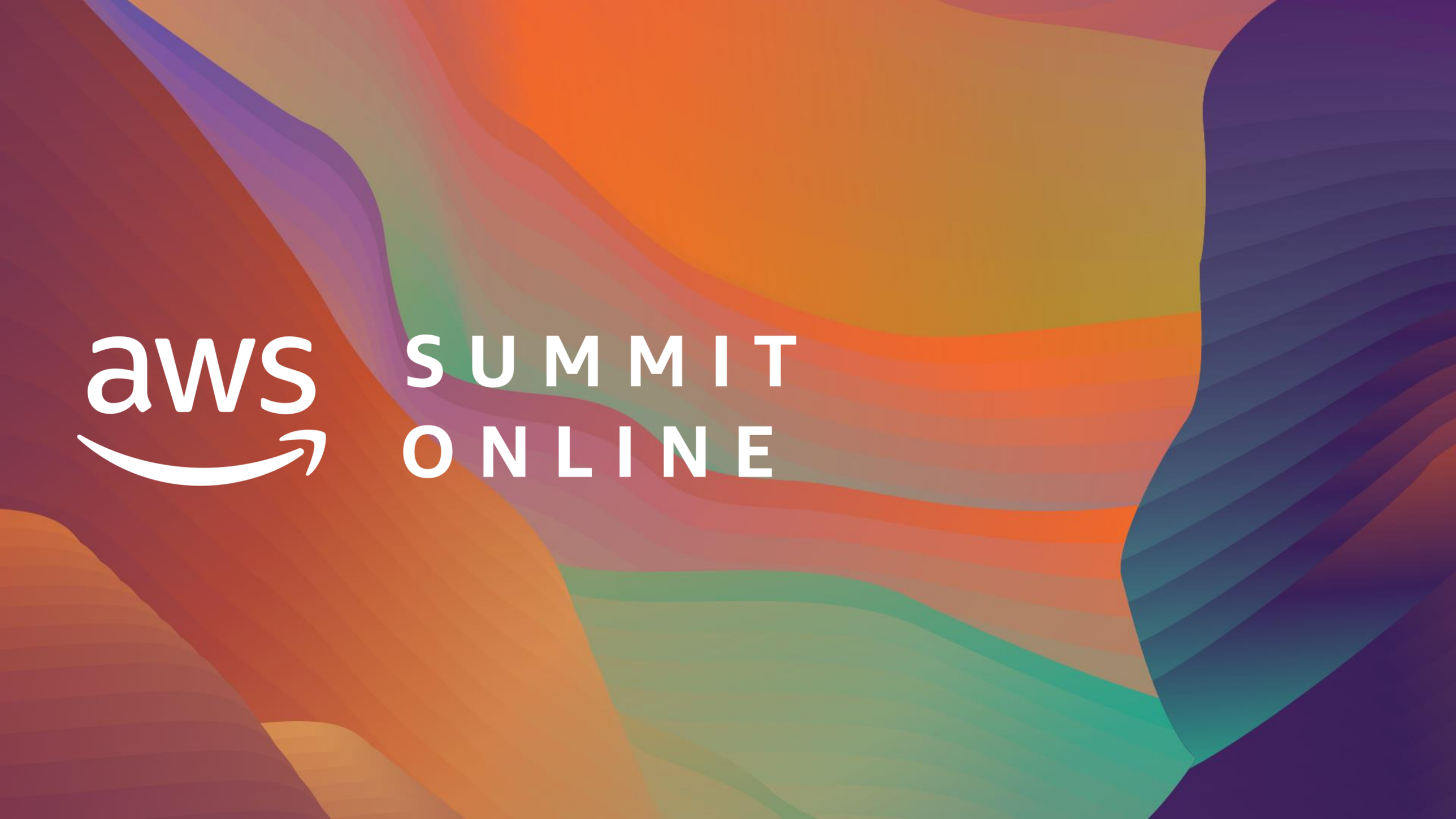

aws
SUMMIT
ONLINE

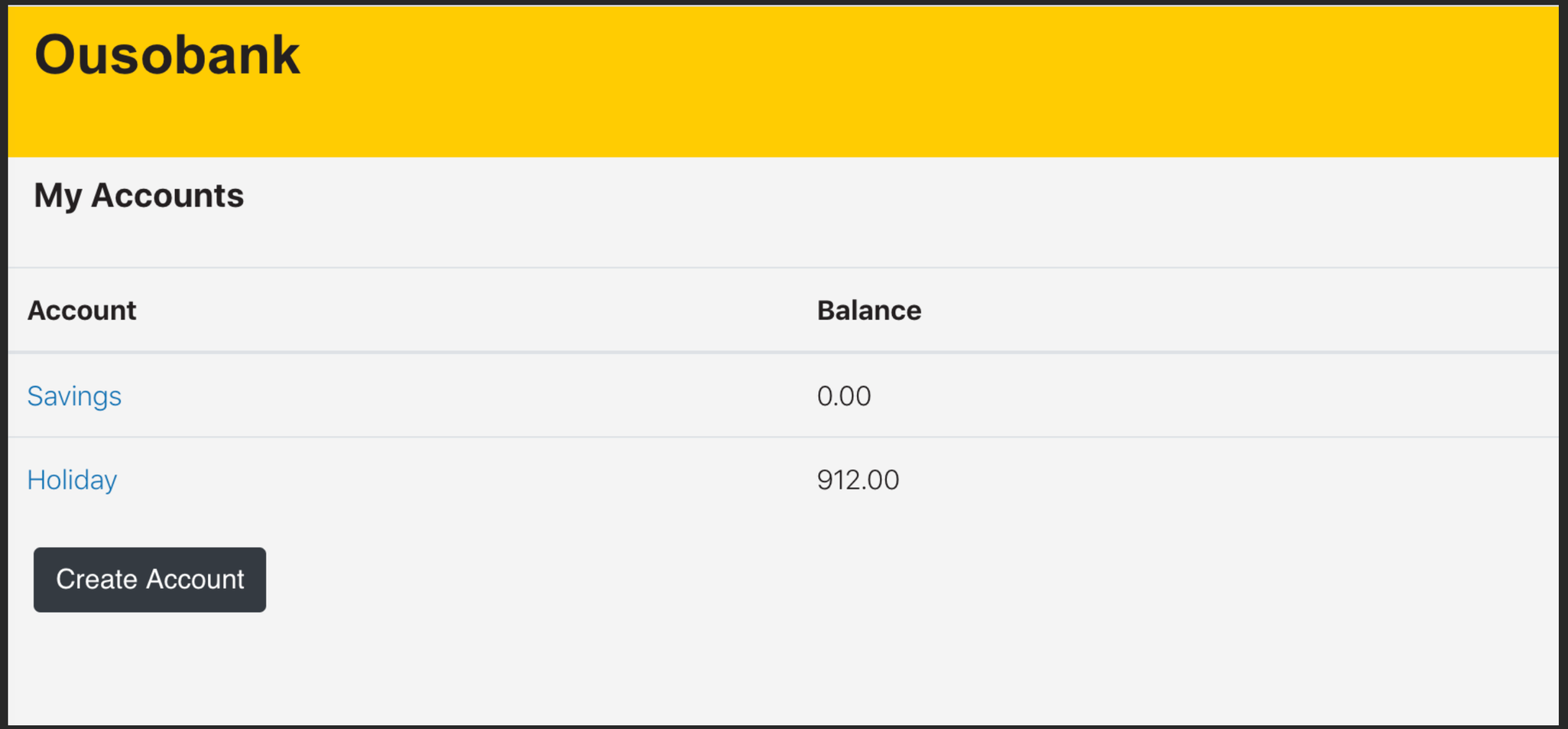
A path to Event Sourcing with Amazon MSK

James Ousby

Senior Solutions Architect
Amazon Web Services

aws SUMMIT ONLINE

Let’s build a bank



Learning objectives

- Understand the what and why of event sourcing and how Amazon Managed Streaming for Apache Kafka (MSK) can help
- Think differently about how you capture, store, and process your data
- Explore the design and build process behind an event sourcing demo built around MSK

What is event sourcing?

What is event sourcing?

What it is not – persist entities by updating in place

invoice { id: 1, amt: 50, **status: 'issued'** }

What is event sourcing?

What it is not – persist entities by updating in place

invoice { id: 1, amt: 50, **status: ’overdue’** }

What is event sourcing?

What it is not – persist entities by updating in place

invoice { id: 1, amt: 50, **status: 'paid'** }

What is event sourcing?

Persist entities by storing a sequence of state-changing events

IssuedEvent { id: 1, amt: 50, **status: 'issued'** }

invoice { id: 1, amt: 50, **status: 'issued'** }

transient summary

What is event sourcing?

Persist entities by storing a sequence of state-changing events

IssuedEvent { id: 1, amt: 50, **status: 'issued'** }

OverdueEvent { id: 1, **status: 'overdue'** }

invoice { id: 1, amt: 50, **status: 'overdue'** }

transient summary

What is event sourcing?

Persist entities by storing a sequence of state-changing events

IssuedEvent { id: 1, amt: 50, **status: 'issued'** }

OverdueEvent { id: 1, **status: 'overdue'** }

PaidEvent { id: 1, **status: 'paid'** }

invoice { id: 1, amt: 50, **status: 'paid'** }

transient summary

Event sourcing’s close friends

- CQRS (Command Query Responsibility Segregation)
- Domain-driven design
- Distributed logs (often Apache Kafka)

Event sourcing challenges

It’s not a new concept. Why isn’t this the default way to think about data?

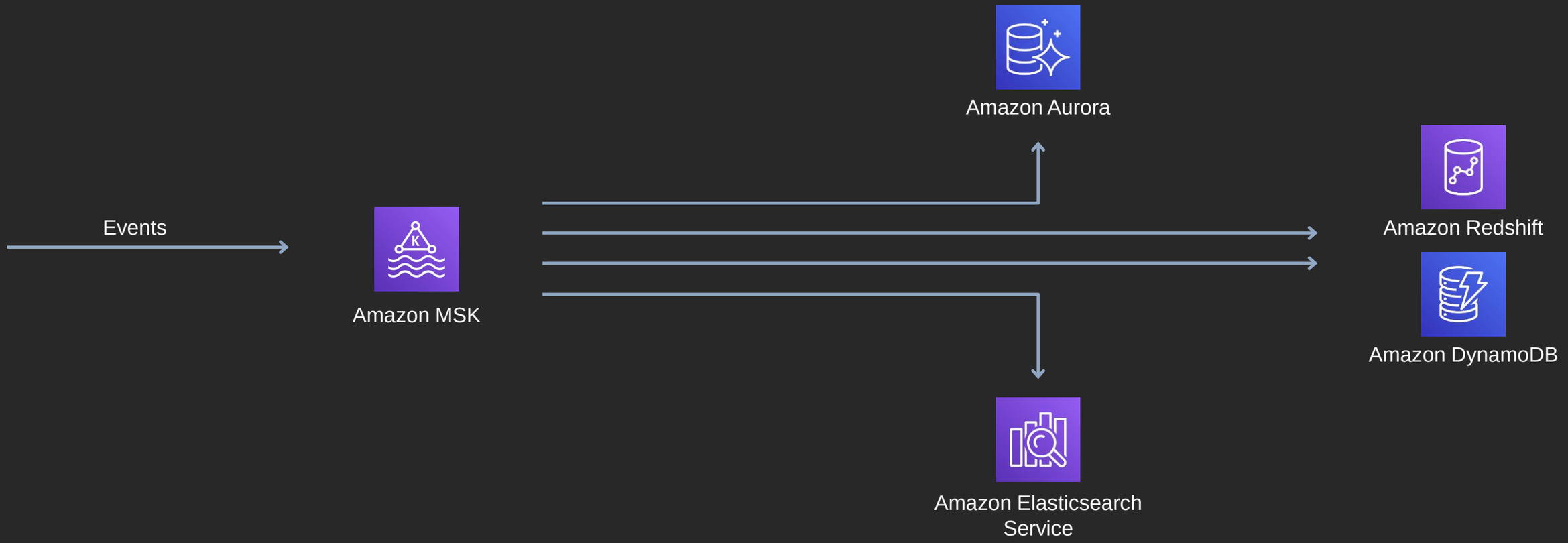
- Platform and tooling maturity
- Lack of framework support
- More complex/more moving parts

Why event sourcing?

Future-proofing your data architecture

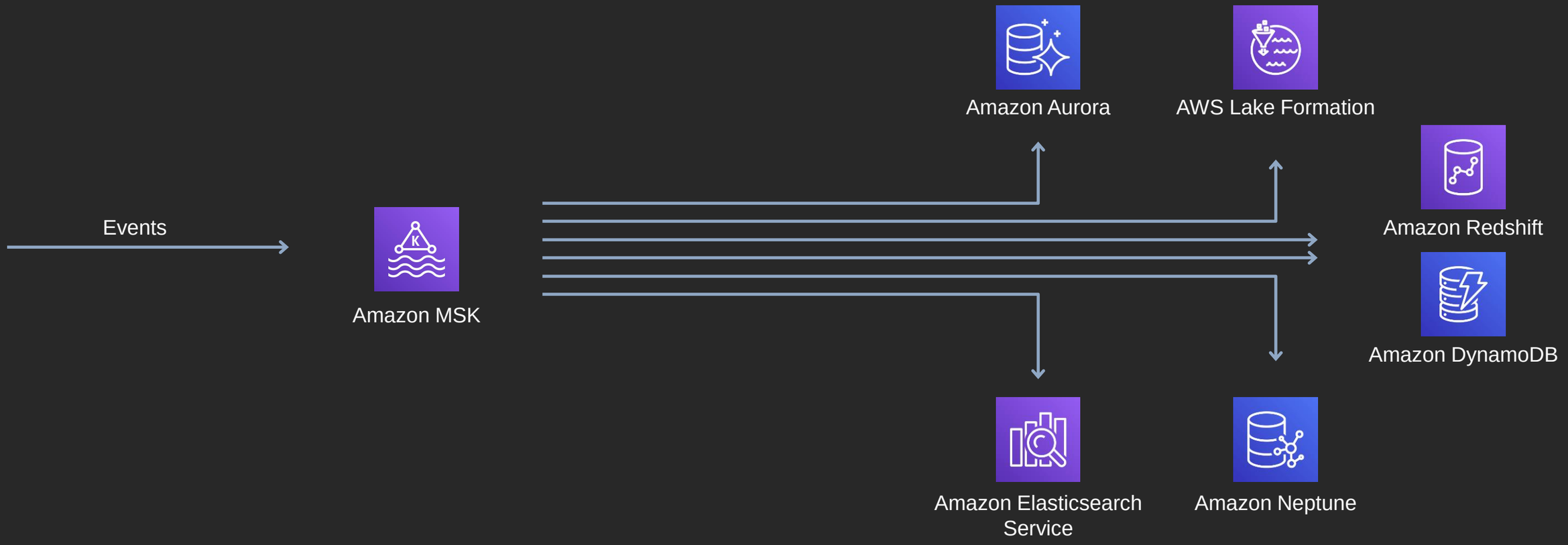
- Maintain complete history (one-way and two-way doors)
- It’s increasingly difficult to anticipate long-term data usage patterns at design/build time
- You would like your data architecture to support new patterns of consumption as technologies evolve over time

Future-proofing your data architecture



transient – rebuild from event stream

Future-proofing your data architecture

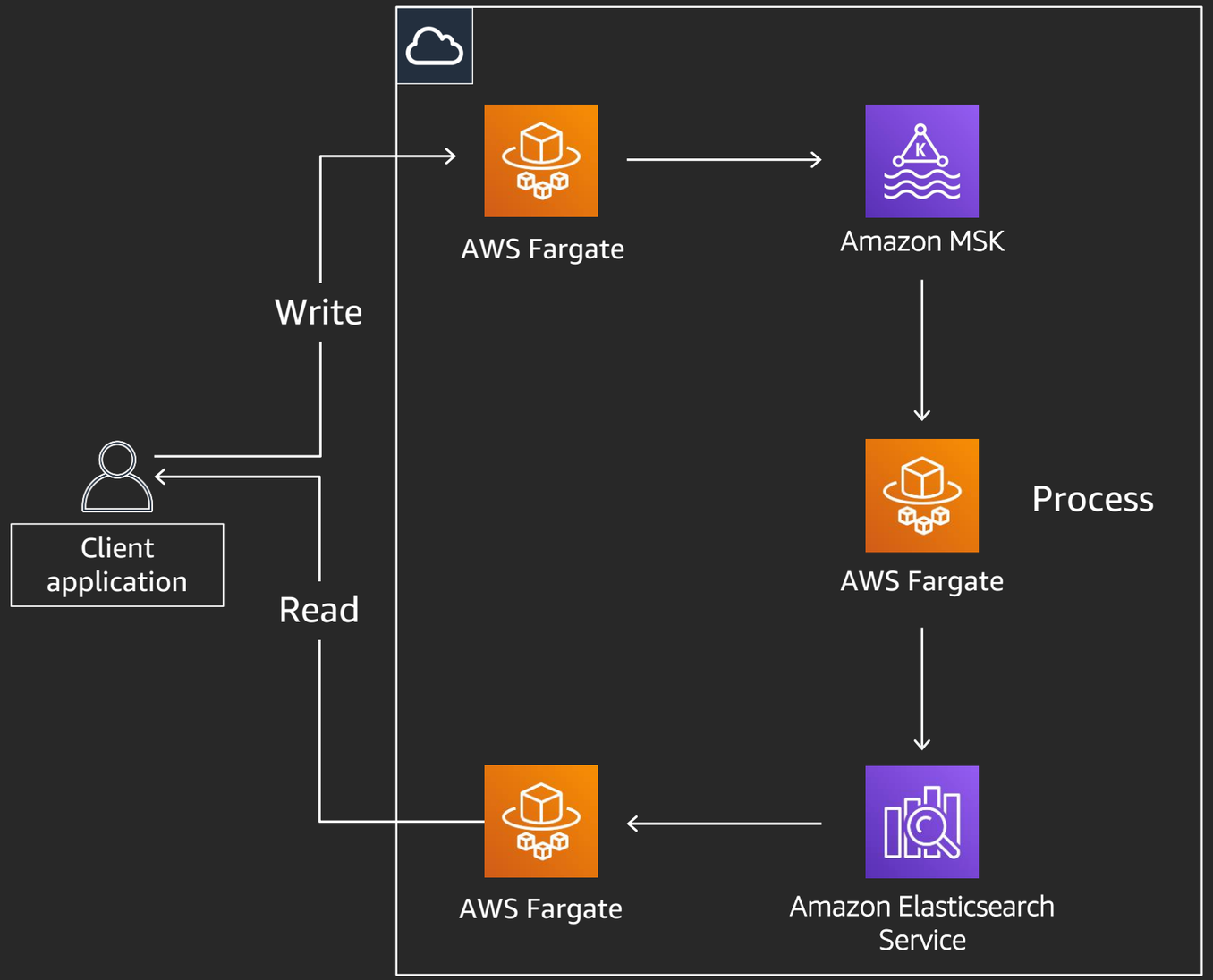


transient – rebuild from event stream

Capturing your raw data is great, but…

Event sourcing + CQRS benefits

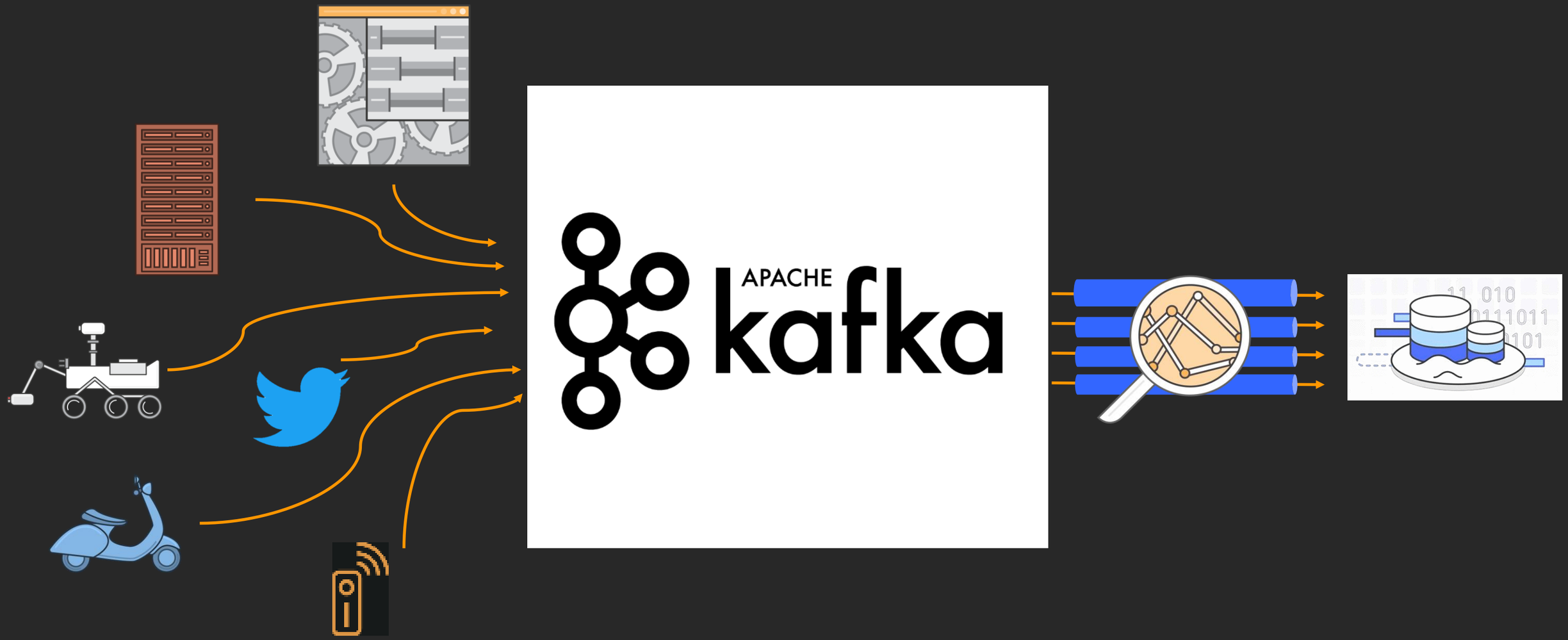
- Data architectures that can scale elastically as demand increases
- Data architectures that can prioritize read/write performance
- Compute components can be scaled independently



Amazon Managed Streaming for Apache Kafka

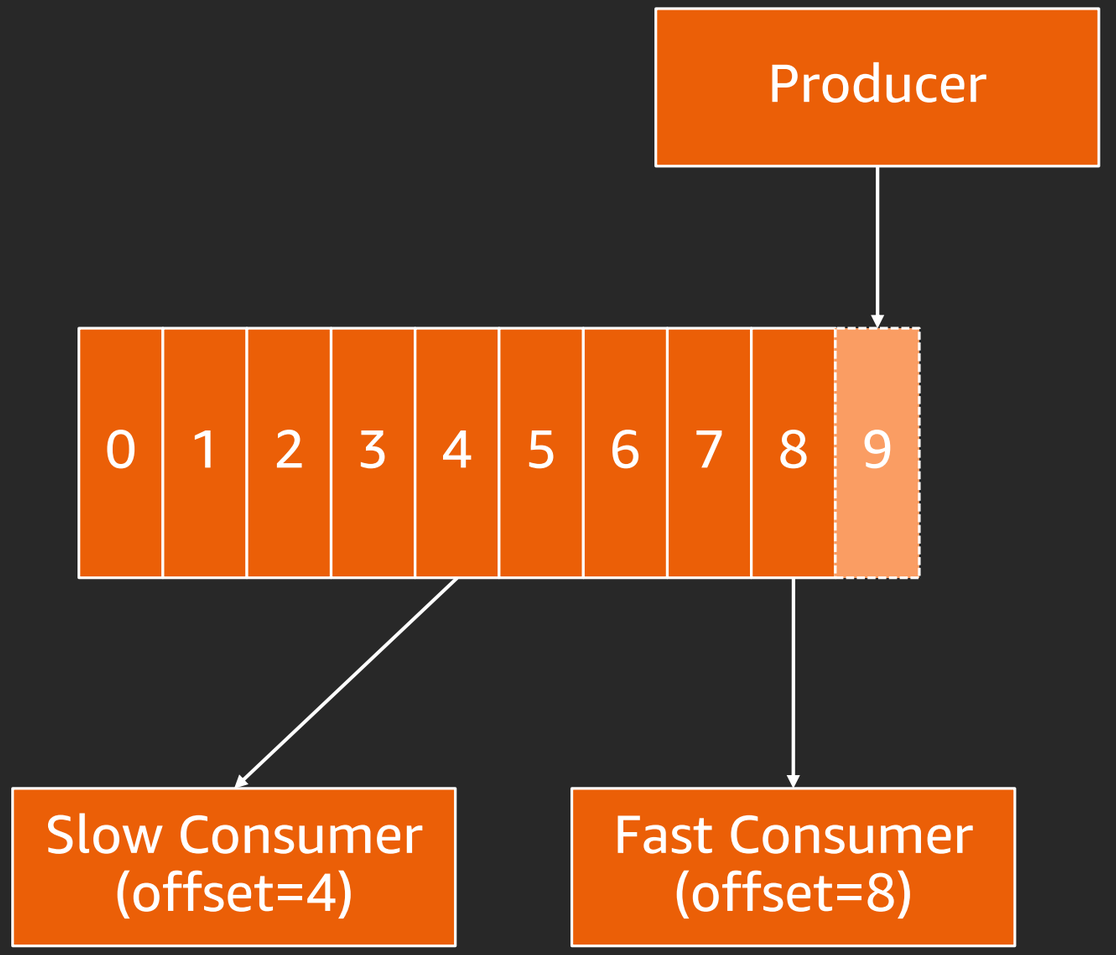
aws SUMMIT ONLINE

Apache Kafka

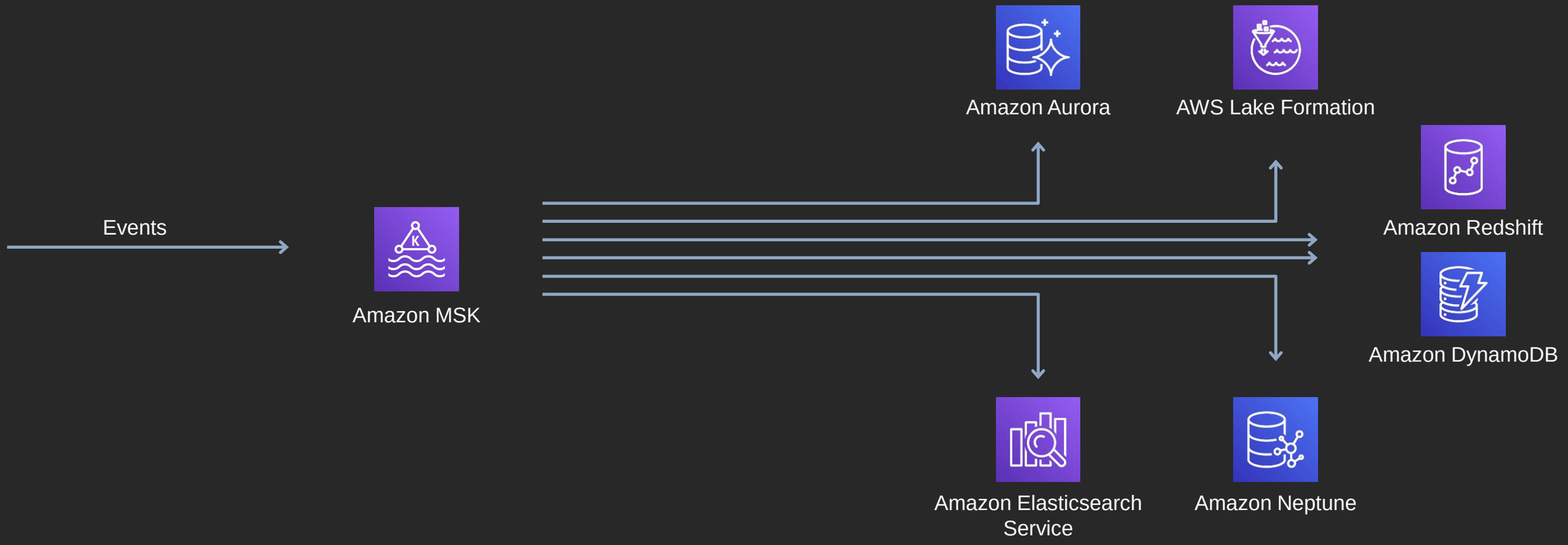


Why is Apache Kafka a good fit for event sourcing?

- Immutable append-only log
- Guaranteed ordering



Multiple consumers all at different log offsets

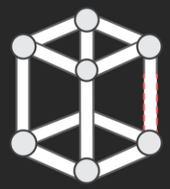


transient – rebuild from event stream

Challenges operating Apache Kafka

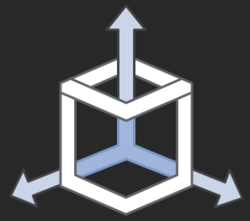
Setup

Difficult to set up



Availability

Hard to achieve



Scale

Hard to scale

Monitoring

Hard to capture metrics

Support

Labor-intensive

Amazon Managed Streaming for Apache Kafka

Fully managed and highly available Apache Kafka service

Fully compatible

Migrate and run your existing Apache Kafka applications on AWS without code changes

Fully managed

Manages provisioning, configuration, and maintenance of Apache Kafka clusters

Highly available

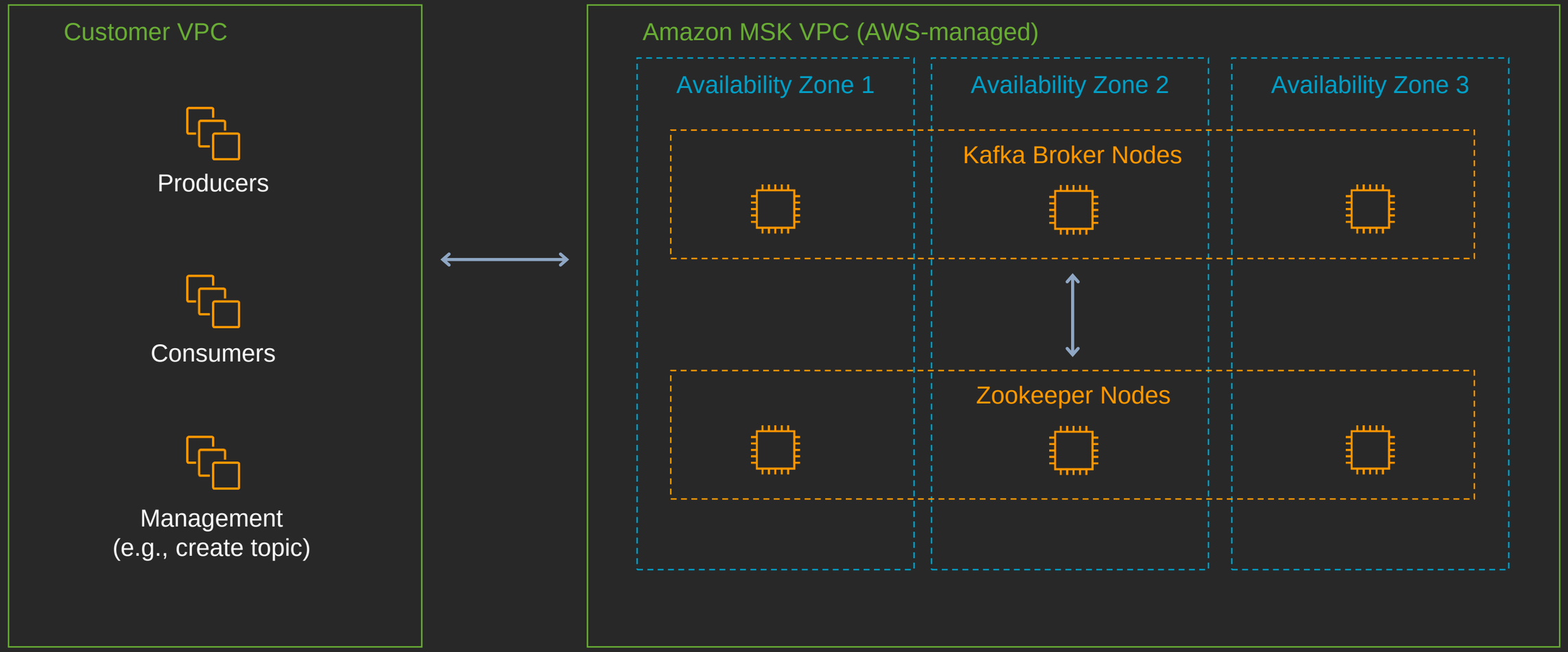
Multi-AZ replication within an AWS Region

Health monitoring

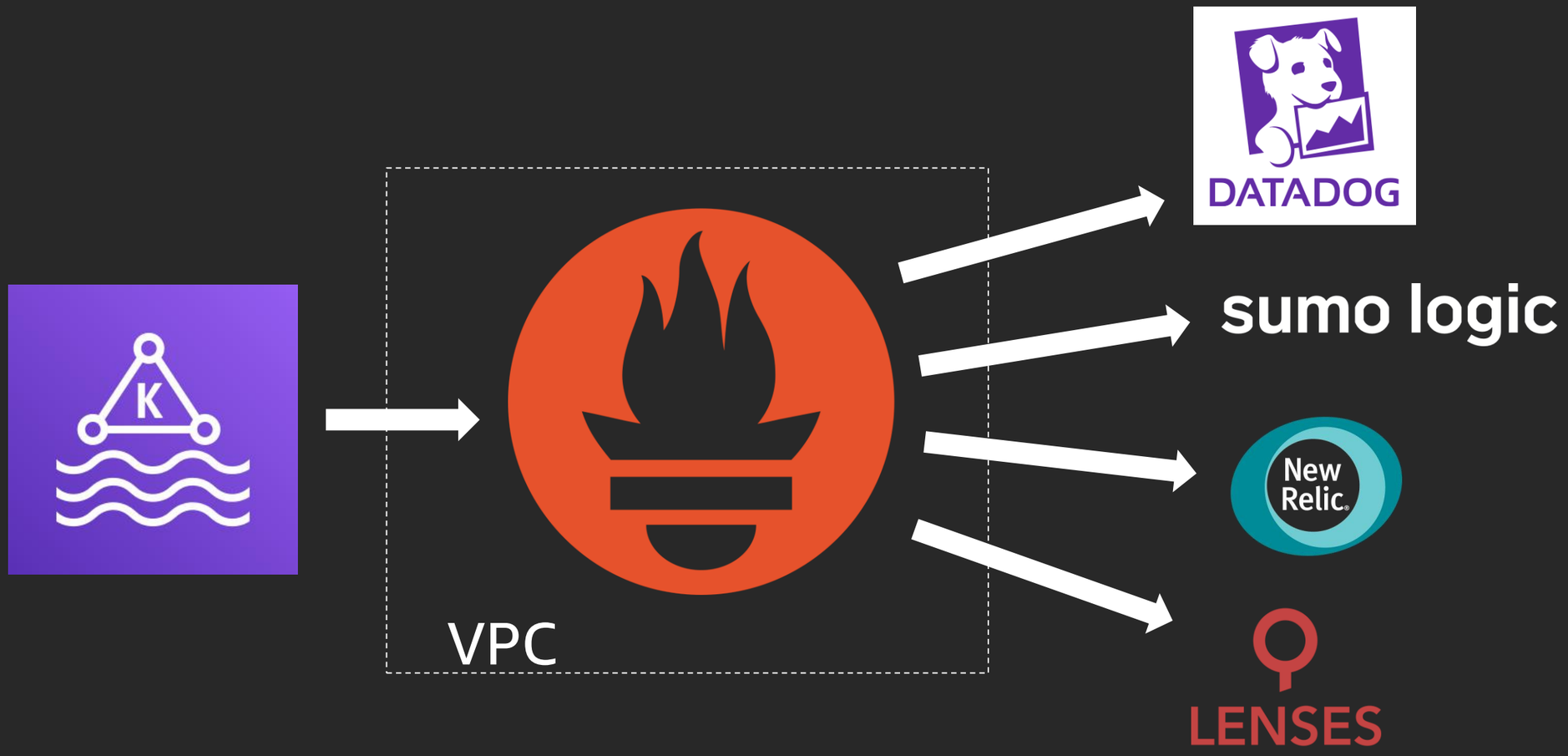
Highly secure

Multiple levels of security for your Apache Kafka clusters

Amazon MSK deployment architecture



Open monitoring with Prometheus



Getting started with Amazon MSK is easy

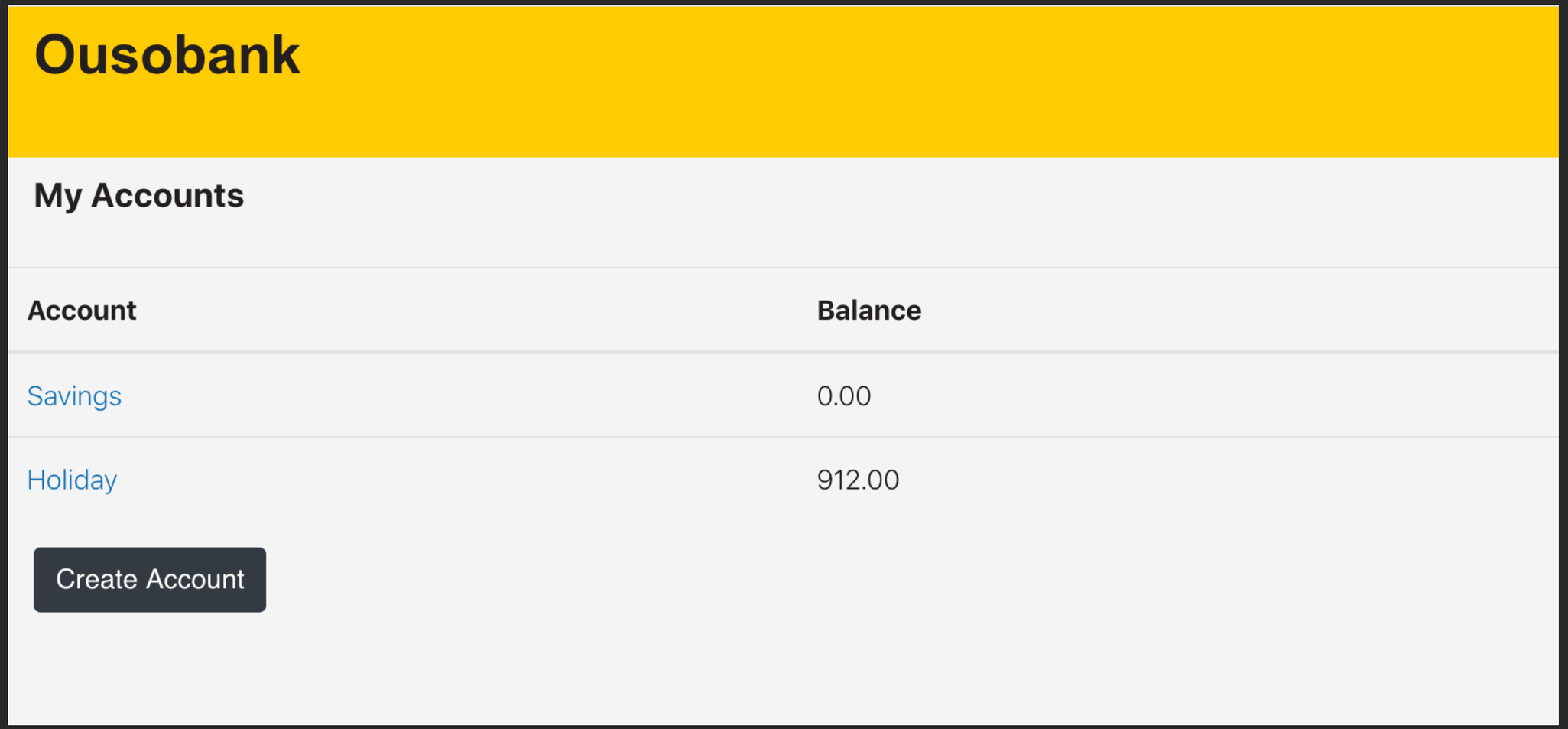
- Fully compatible with Apache Kafka 1.1.1, 2.2.1, and 2.3.1
- AWS Management Console and AWS API for provisioning
    - Clusters are automatically set up
    - Provision Kafka brokers and storage
    - Create and tear down clusters on demand
- Deeply integrated with AWS services

Amazon MSK roadmap (what I can share, TBC)

- In-place cluster upgrades
- Cluster autoscaling
- Identity and Access Management (IAM) support
- Schema registry
- AWS Lambda integration
- Storage tiering

Demo

Our banking app



Design process

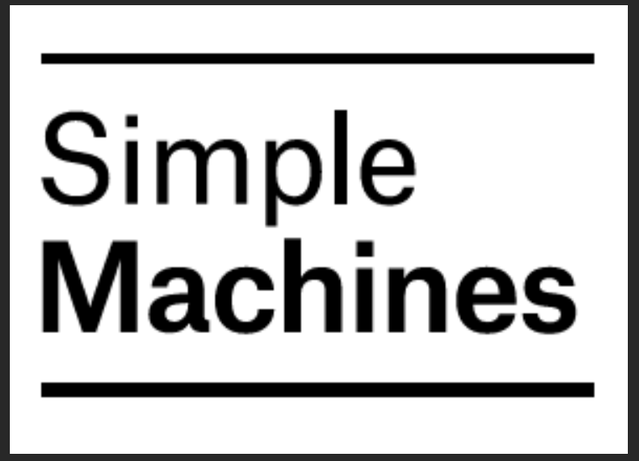
Terminology

- Command
- Event
- Aggregate
- Projection/ materialized view

1. Model your domain as commands, events, aggregates, and views
2. Design the command handlers
3. Design the event processor(s) that create your projection
4. Design the read side

Simple Machines

Simple Machines is a leading Australian technology consultancy at the intersection of data architecture, data engineering, and enterprise software delivery. The company specializes in engineering real-time, data-driven platforms and applications.



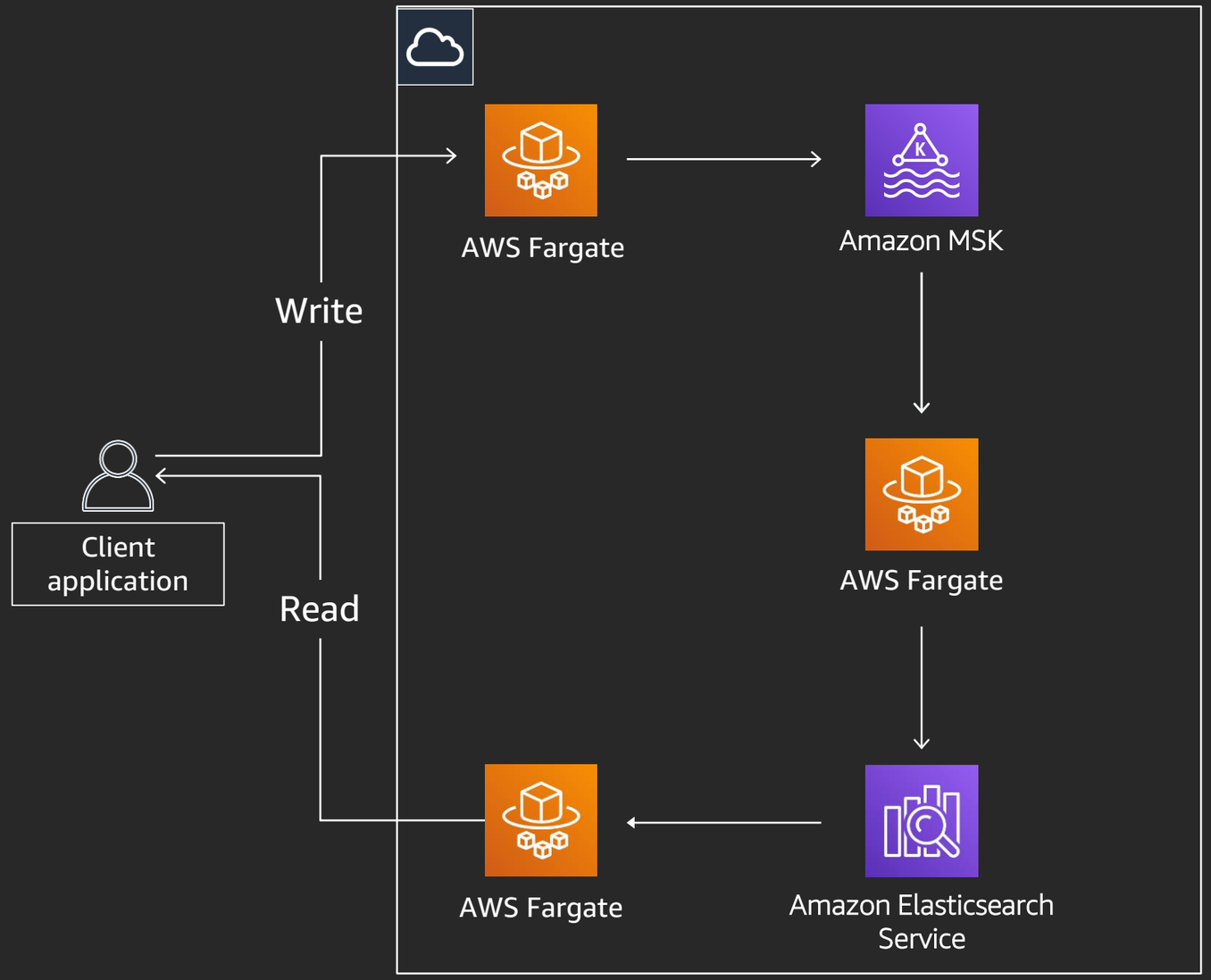
Simple Machines – experimenting

Simple Sourcing – event source data abstraction built to use Apache Kafka as primary data store

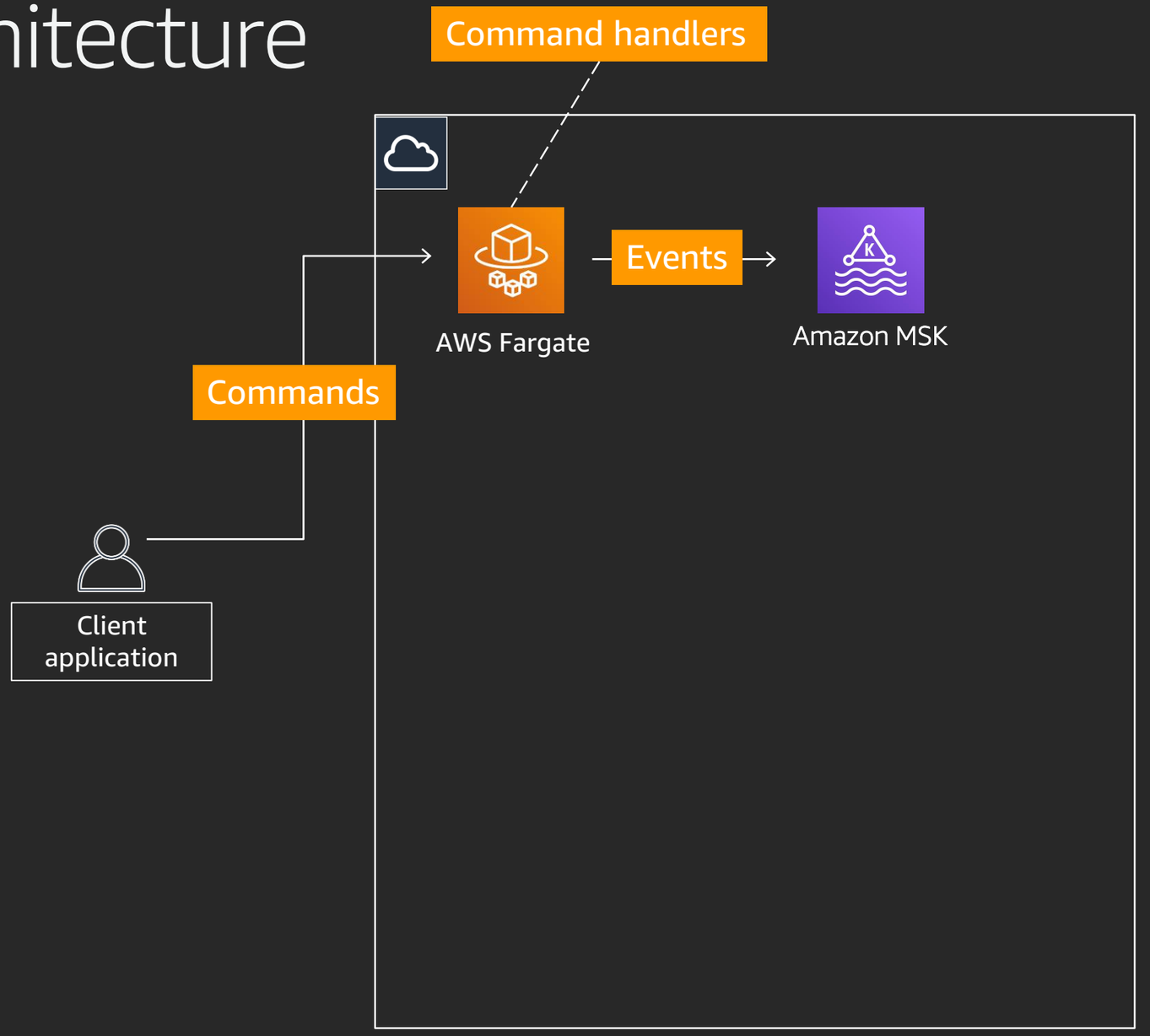
Simple Saga – Saga (coordination) layer built using Simple Sourcing to provide a robust execution engine with compensation semantics

https://simplesource.io/

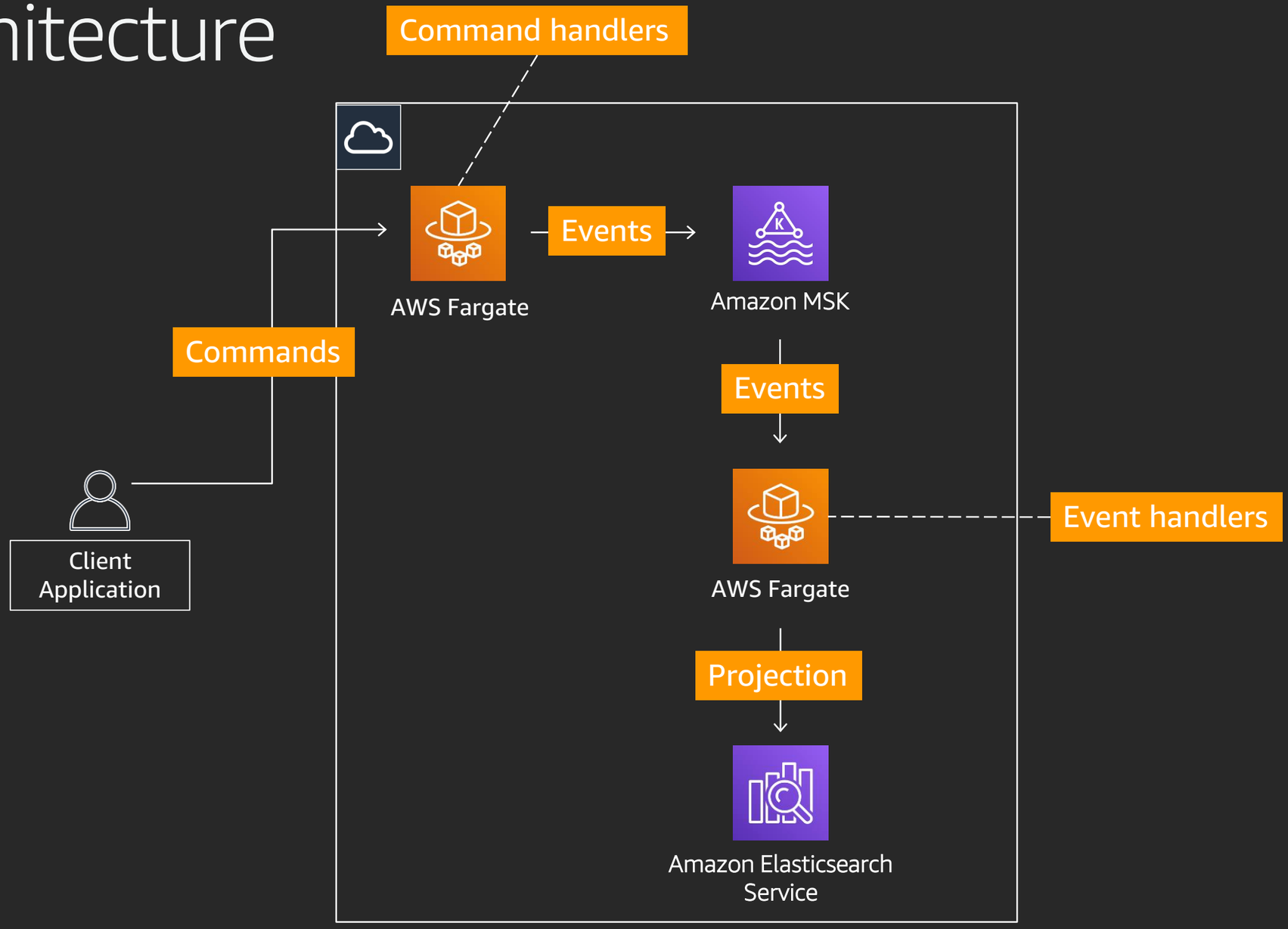
Demo architecture



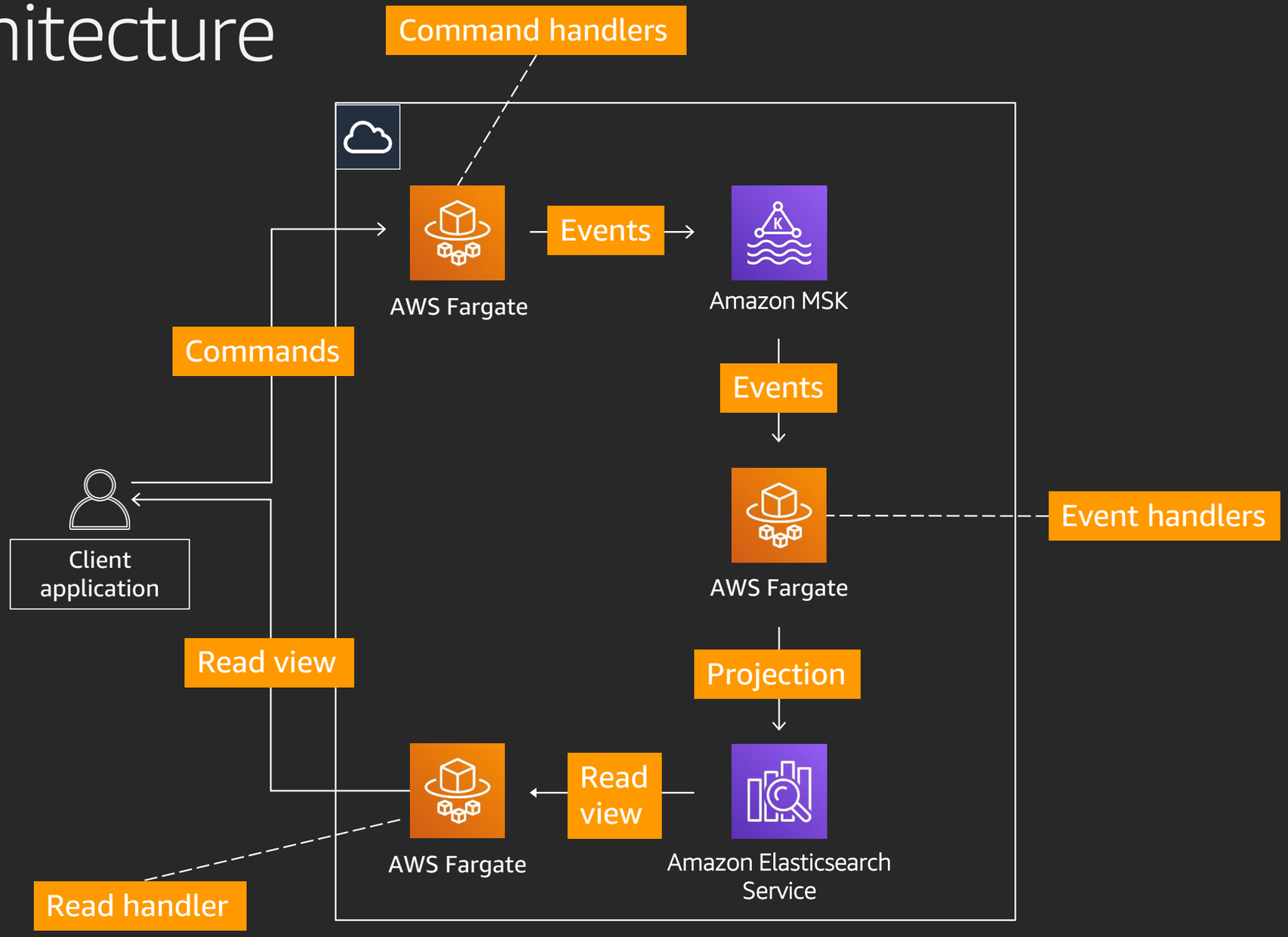
Demo architecture



Demo architecture



Demo architecture



< time for some code />

Closing

aws SUMMIT ONLINE

Learning objectives

- Understand the what and why of event sourcing and how Amazon Managed Streaming for Apache Kafka (Amazon MSK) can help
- Think differently about how you capture, store, and process your data
- Explore the design and build process behind an event sourcing demo built around MSK

Thank you!

James Ousby

joousby@amazon.com

aws SUMMIT ONLINE