confluent HIVEMQ

Best Practices for Streaming IoT Data with MQTT and Apache Kafka®

Kai Waehner, Technology Evangelist, Confluent
Dominik Obermaier, CTO, HiveMQ

Speakers

Kai Waehner
Technology Evangelist, Confluent
kai.waehner@confluent.io
@KaiWaehner

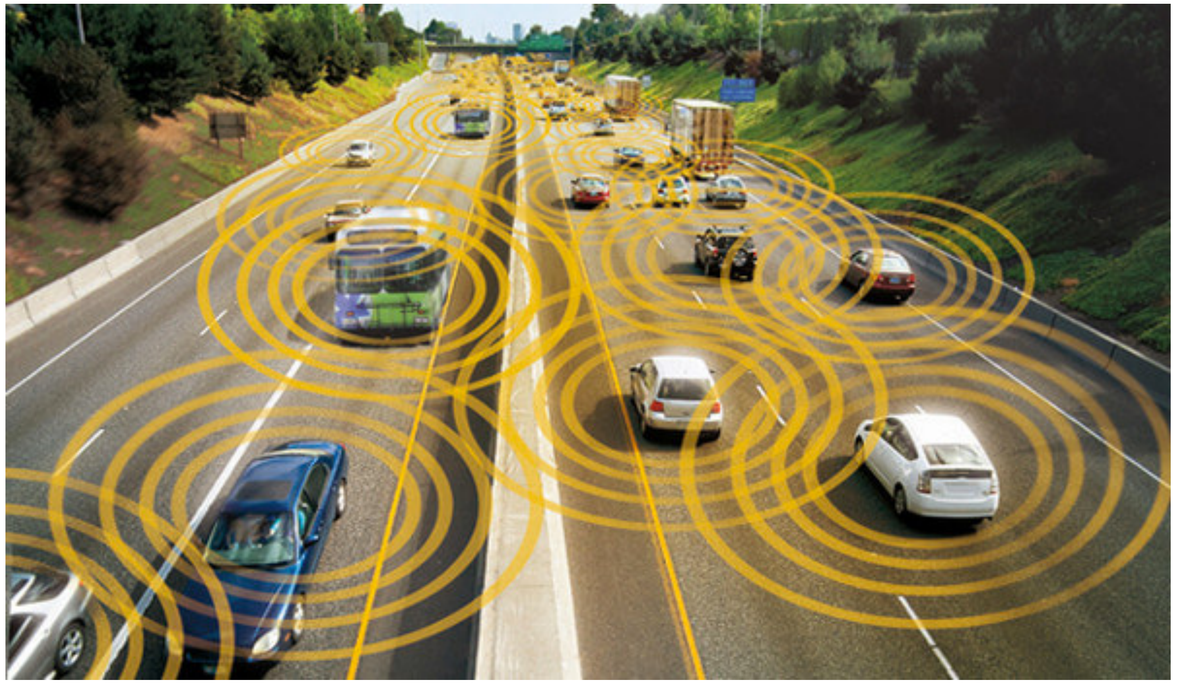
Dominik Obermaier
CTO, HiveMQ
dominik.obermaier@hivemq.com
@dobermai

confluent

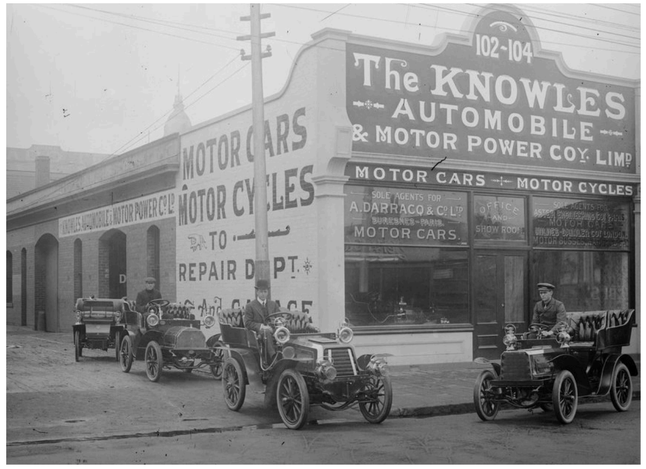
HIVEMQ

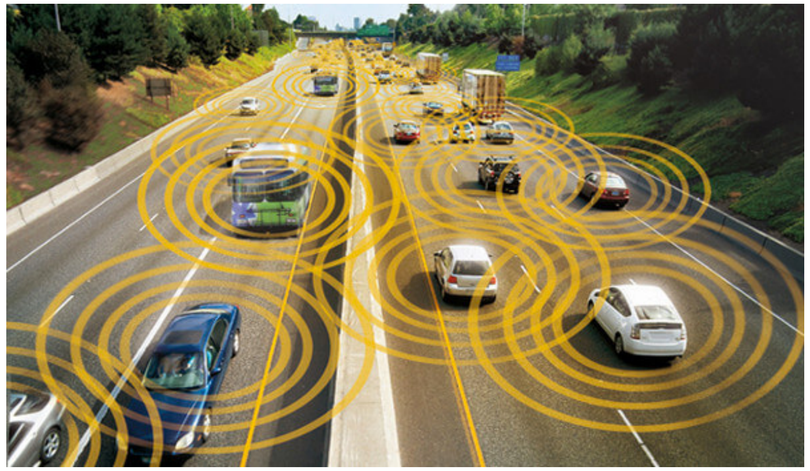
Agenda

- **Use Case**
- Architecture
- Live Demo
- Best Practices
- Next steps



Global Automotive Company
Builds Connected Car Infrastructure





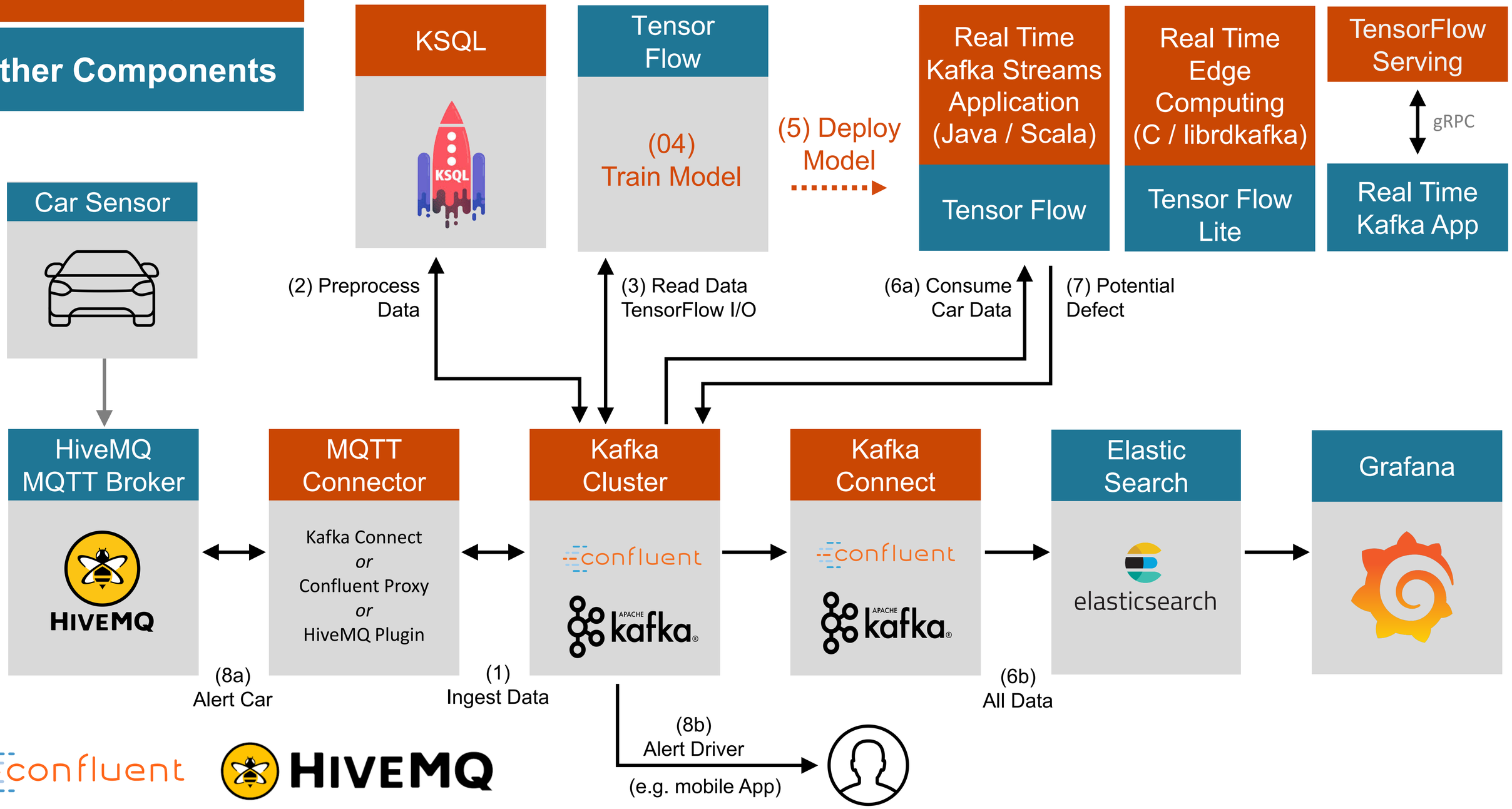
Use Cases:

- Connected Car Infrastructure (Cars, Traffic Lights, Cloud Services, etc.)
- Real Time Analytics (Predictive Maintenance, etc.)
- Continuous Services / Sales
- Partner Integration (Car workshop, gas station, food market, etc.)
- …

confluent HIVEMQ

Kafka Ecosystem

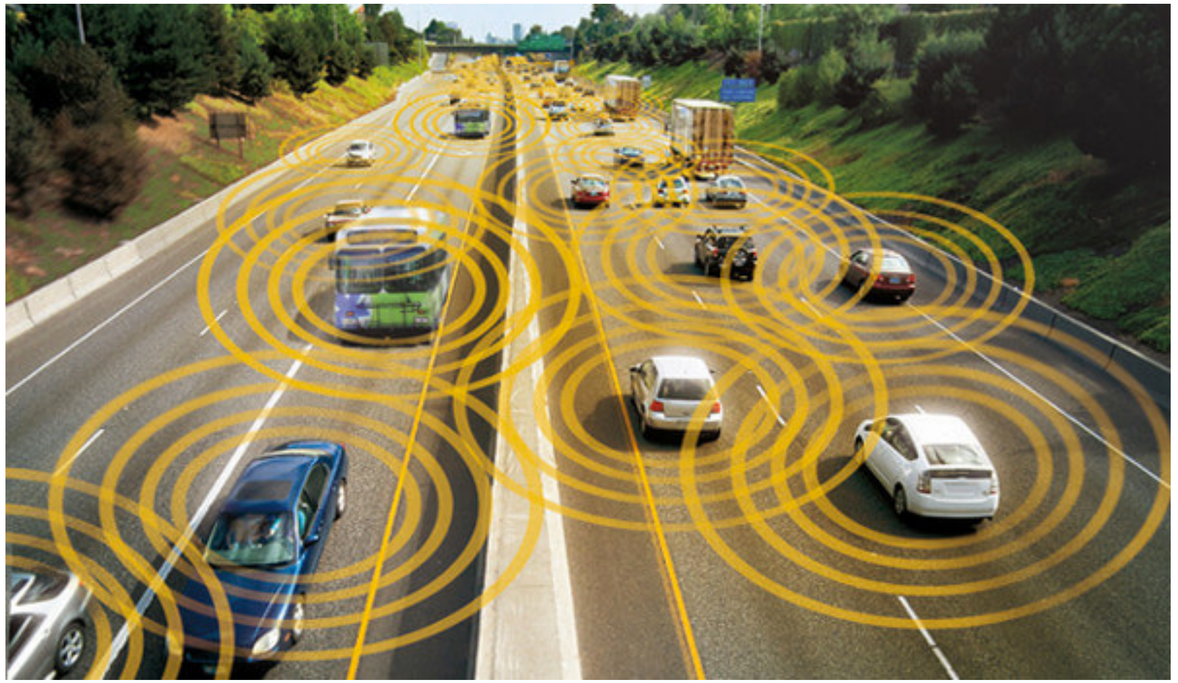
Other Components



confluent HIVEMQ

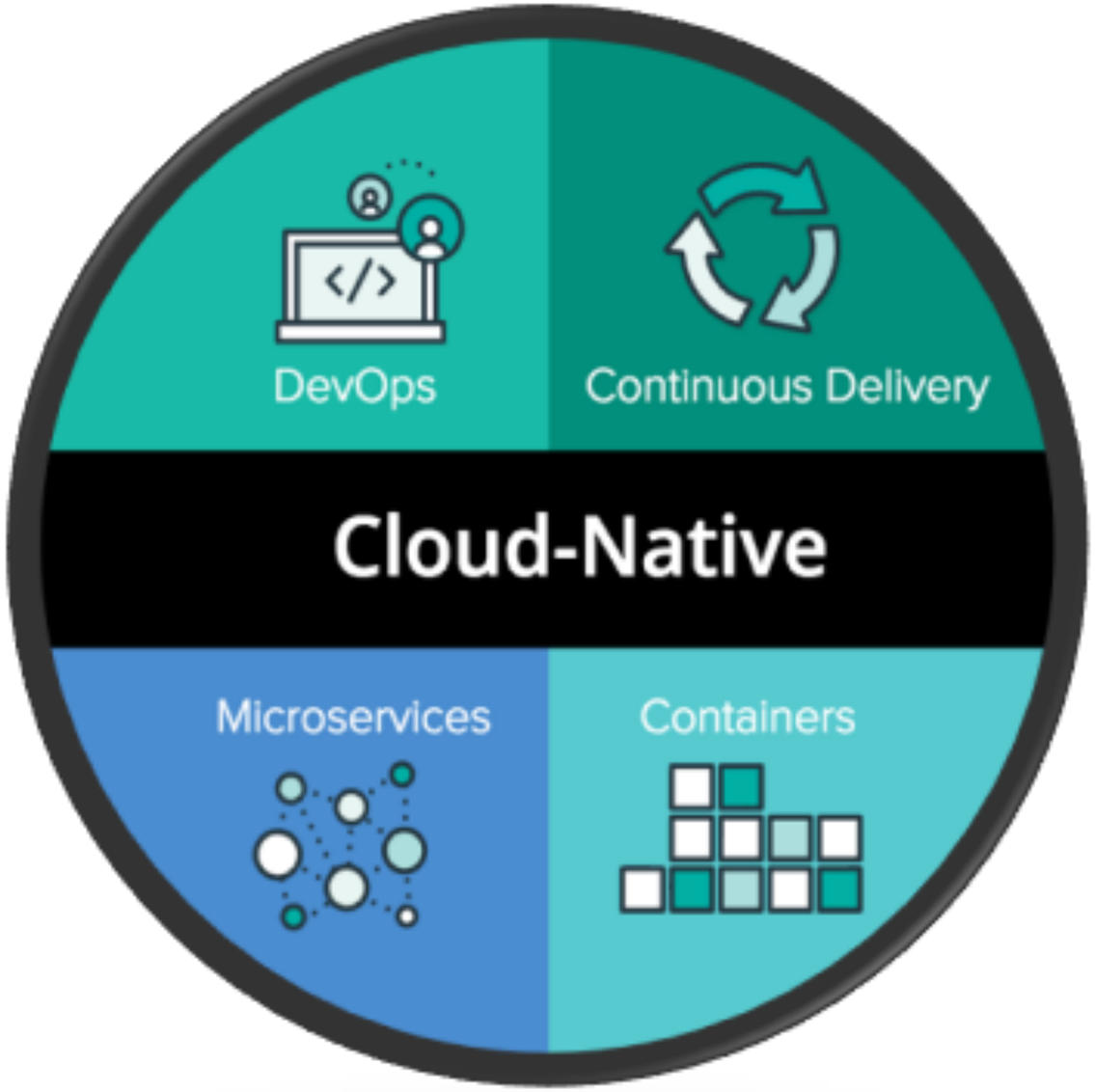
Agenda

- Use Case
- **Architecture**
- Live Demo
- Best Practices
- Next steps



confluent HIVEMQ

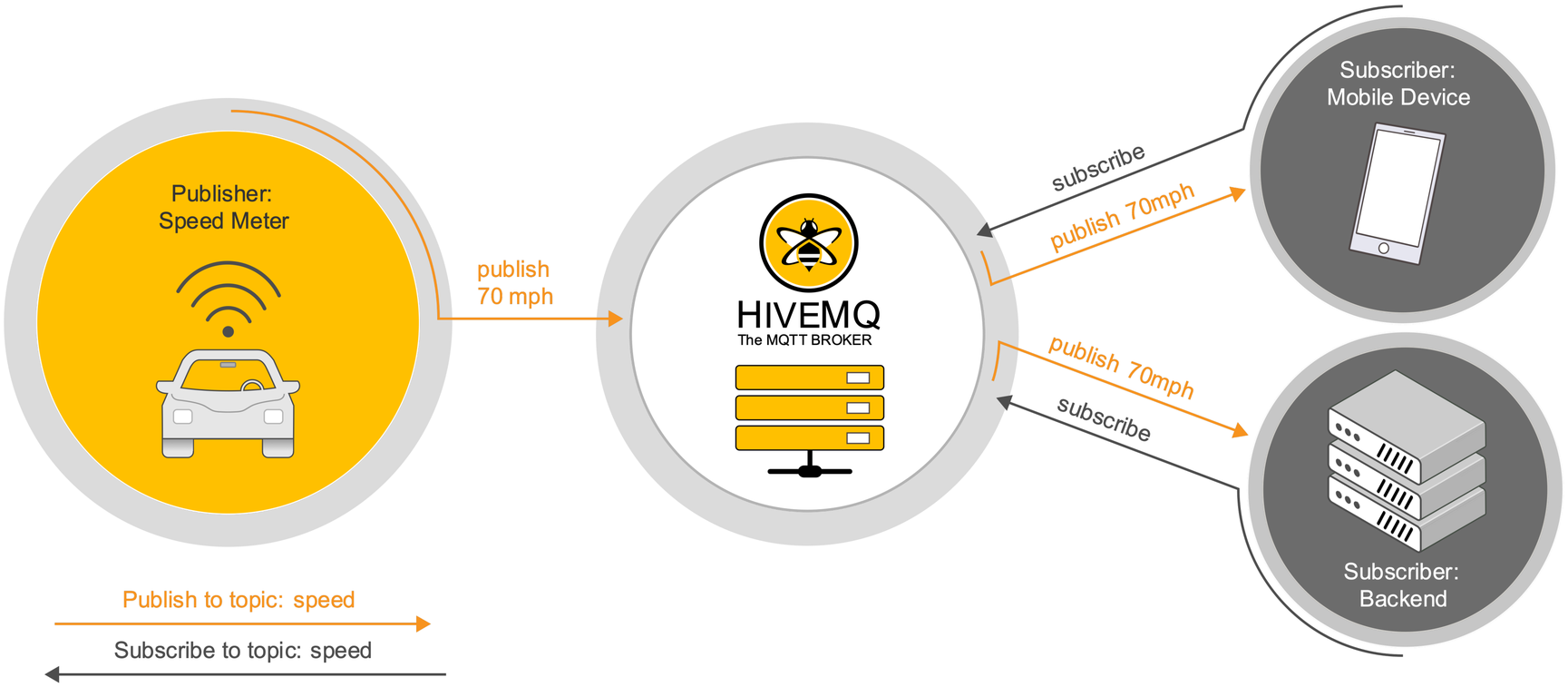
Cloud Native Infrastructure



Benefits

- Scalable
- Flexible
- Agile
- Elastic
- Automated
- Etc.

MQTT - Publish / subscribe messaging protocol



- Built on top of TCP/IP for constrained devices and unreliable networks
- Many (open source) broker implementations
- Many (open source) client libraries

- IoT-specific features for bad network / connectivity
- Widely used (mostly IoT, but also web and mobile apps via MQTT over WebSocket)

MQTT Trade-Offs

Pros

- Lightweight
- All programming languages supported
- Built for poor connectivity / high latency scenarios (e.g. mobile networks!)
- High scalability and availability *
- ISO Standard
- Most popular IoT protocol

Cons

- Only pub/sub, not stream processing
- Asynchronous processing (clients can be offline for long time)
- No reprocessing of events

confluent HIVEMQ

** Depending on Broker Implementation*

A Streaming Platform is the Underpinning of an Event-driven Architecture

Ubiquitous connectivity
Globally scalable platform for all event producers and consumers

Immediate data access
Data accessible to all consumers in real time

Single system of record
Persistent storage to enable reprocessing of past events

Continuous queries
Stream processing capabilities for in-line data transformation

Kafka Trade-Offs (from IoT perspective)

Pros

- **Stream processing**, not just pub/sub
- High throughput
- Large scale
- High availability
- Long term storage and buffering
- Reprocessing of events
- **Good integration to rest of the enterprise**

Cons

- Not built for tens of thousands connections
- Requires stable network and good infrastructure
- No IoT-specific features like keep alive, last will or testament

confluent HIVEMQ

(De facto) Standards for Processing IoT Data

MQTT.ORG

+

kafka

=

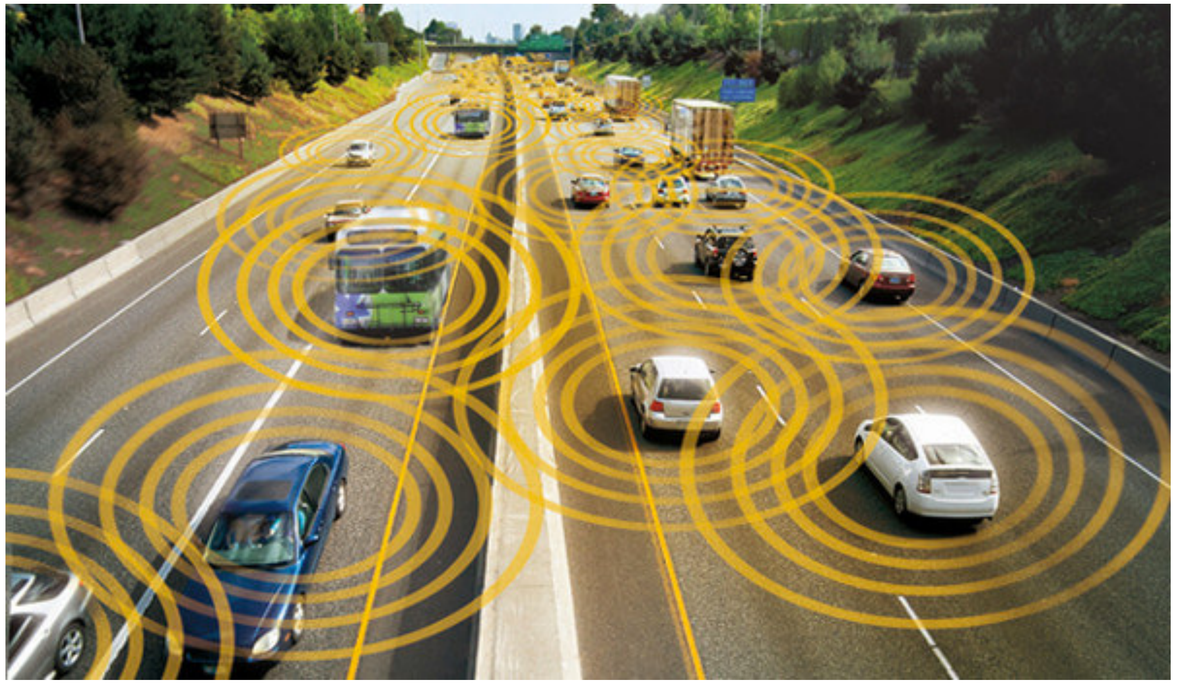
A Match Made In Heaven

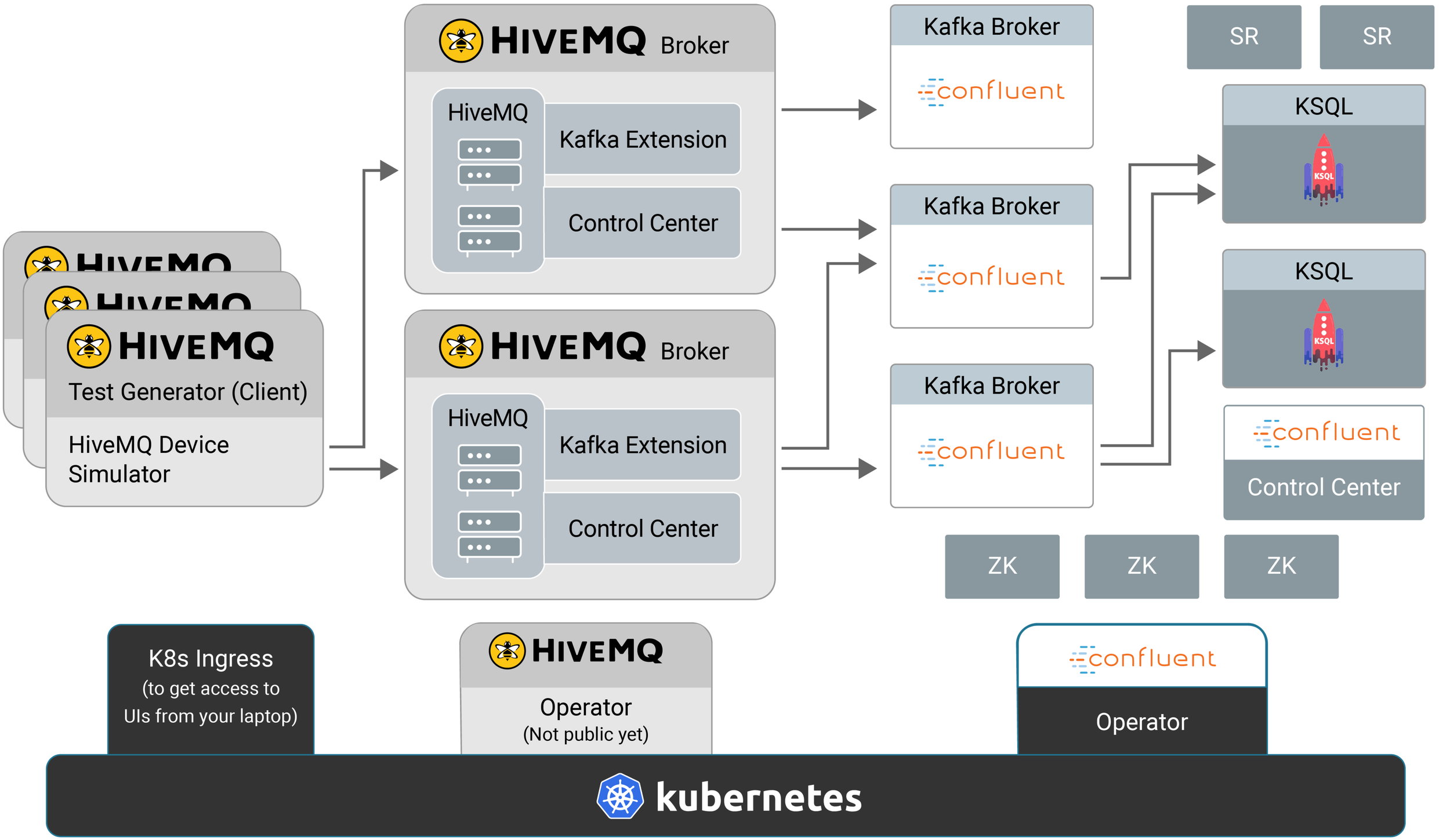
confluent

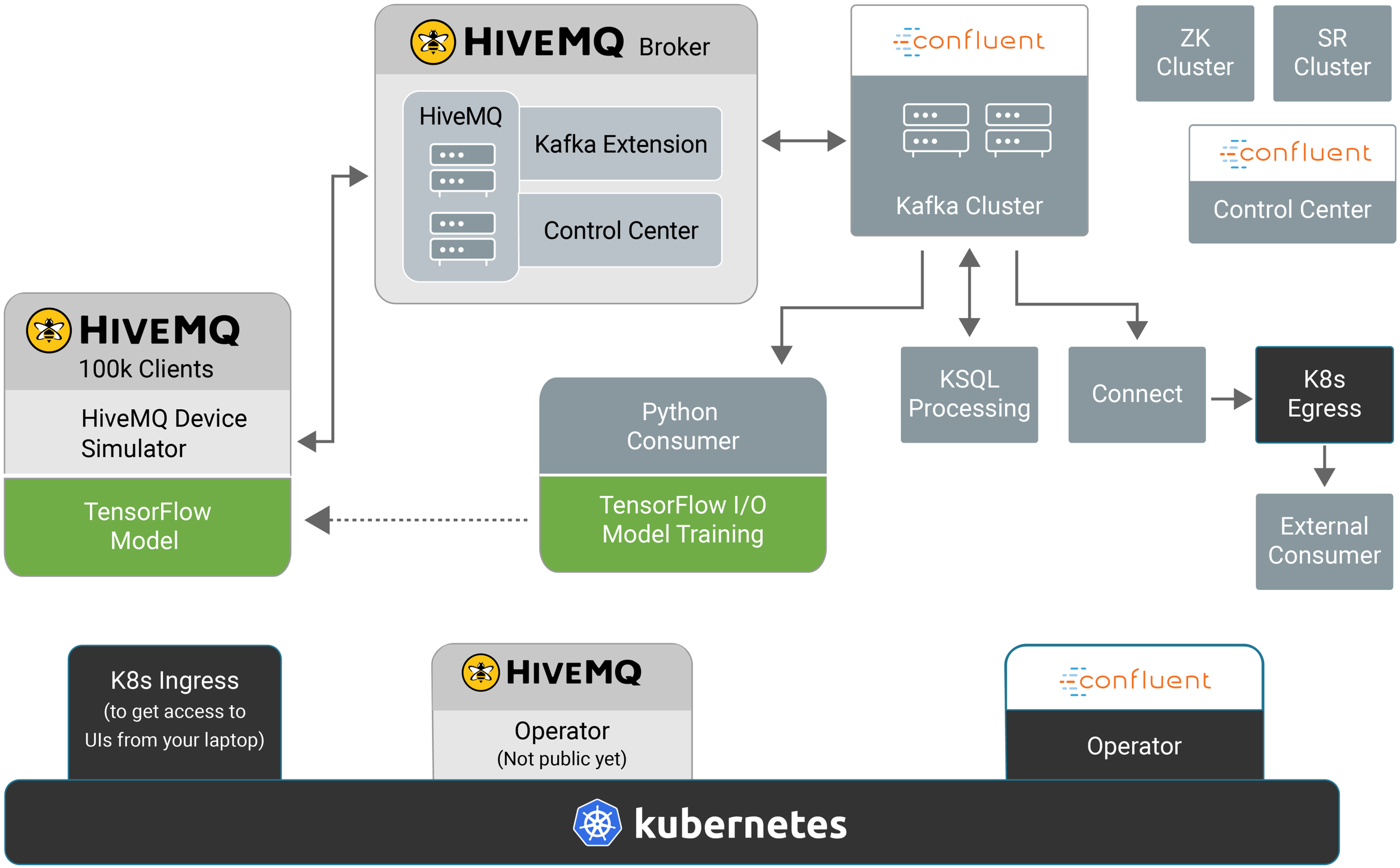
HIVEMQ

Agenda

- Use Case
- Architecture
- **Live Demo**
- Best Practices
- Next steps




HIVEMQ Broker
HiveMQ
Kafka Extension
Control Center
HIVEMQ Broker
HiveMQ
Kafka Extension
Control Center
HIVEMQ
Test Generator (Client)
HiveMQ Device Simulator
Kafka Broker
confluent
Kafka Broker
confluent
Kafka Broker
confluent
SR
SR
KSQL
KSQL
KSQL
confluent
Control Center
ZK
ZK
ZK
K8s Ingress
(to get access to UIs from your laptop)
HIVEMQ
Operator
(Not public yet)
confluent
Operator
kubernetes

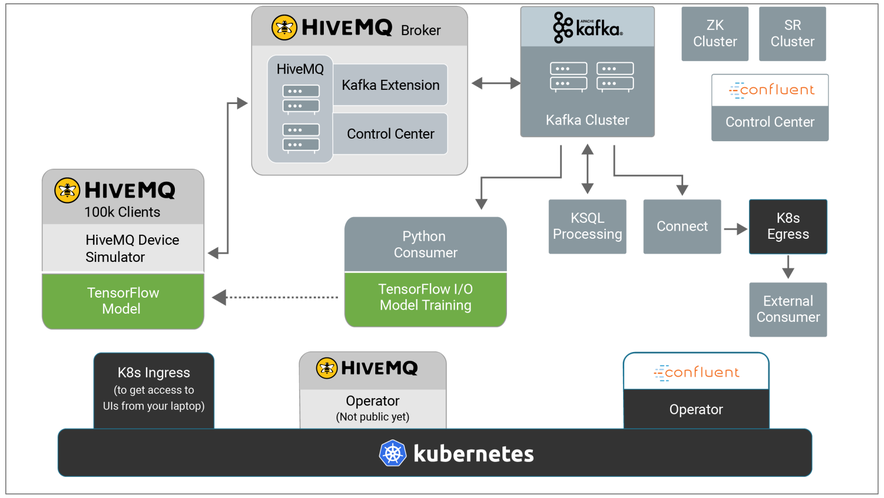

HIVEMQ Broker
HiveMQ
Kafka Extension
Control Center
confluent
Kafka Cluster
ZK Cluster
SR Cluster
confluent
Control Center
HIVEMQ
100k Clients
HiveMQ Device Simulator
TensorFlow Model
Python Consumer
TensorFlow I/O Model Training
KSQL Processing
Connect
K8s Egress
External Consumer
K8s Ingress
(to get access to UIs from your laptop)
HIVEMQ
Operator
(Not public yet)
confluent
Operator
kubernetes

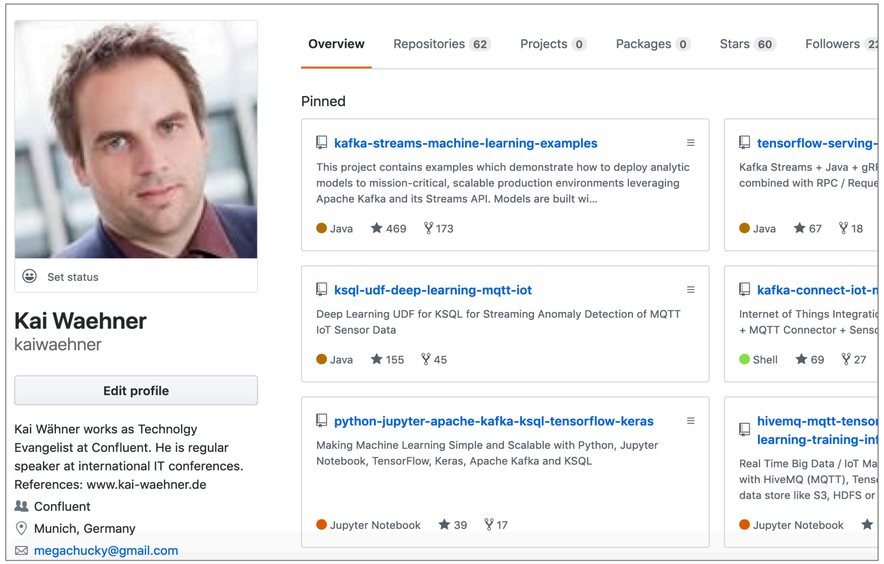
Live Demo

End-to-End Integration and Data Processing for 100000 Connected Cars

Demo 100.000 Connected Cars

(Kafka + MQTT + TensorFlow)





confluent

HIVEMQ

https://github.com/kaiwaehner/hivemq-mqtt-tensorflow-kafka-realtime-iot-machine-learning-training-inference

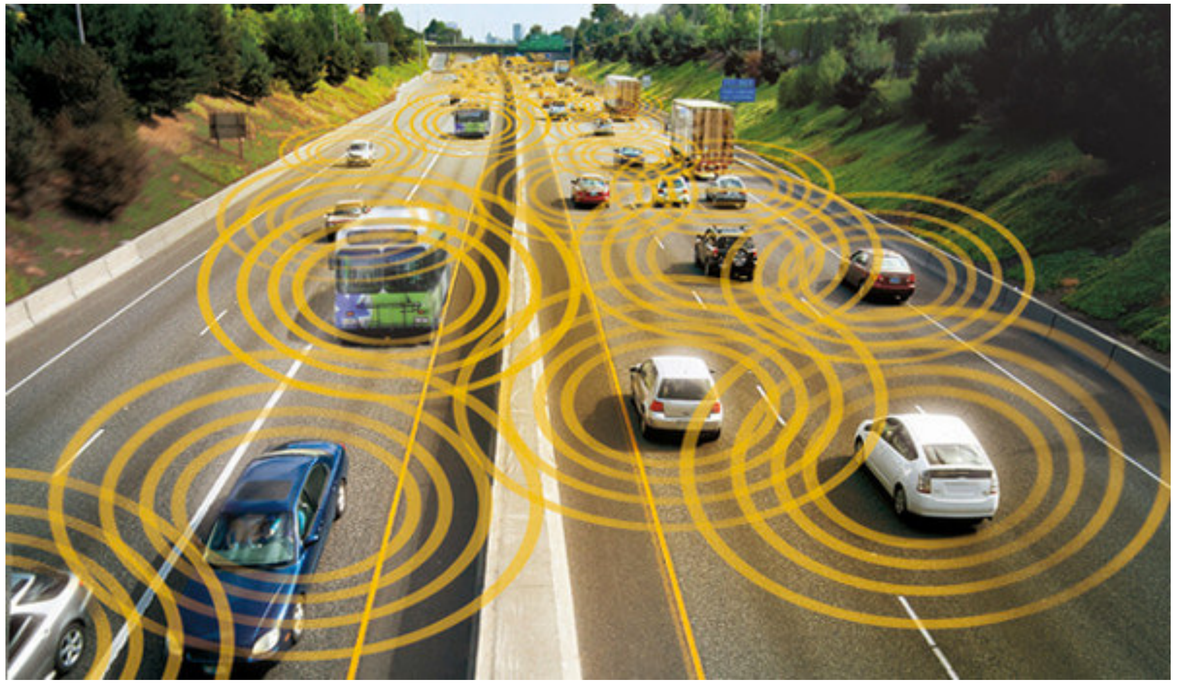
or

http://bit.ly/kafka-mqtt-ml-demo

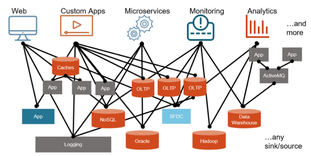
=> Try it out in 30 minutes!

Agenda

- Use Case
- Architecture
- Live Demo
- **Best Practices**
- Next steps



Typical Journey



Pre-Streaming

Value

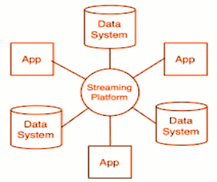
Developer Interest

1

Pub + Sub
Store
Process

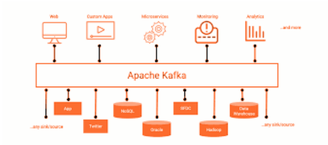
Enterprise Streaming Pilot / Early Production

2



SLA Ready, Integrated Streaming

3



Global Streaming

4

Central Nervous System

5

Maturity (Investment & time)

Projects

Platform

confluent

HIVEMQ

Start Small, but prepare for Scalability from Beginning

1. Use cloud native and scalable components

- Confluent Platform is cloud native and built for scale
- HiveMQ is cloud native and built for scale

2. Don’t deep dive too much in the beginning – but understand options

- HiveMQ Kafka Extension?
- Confluent MQTT connectors?
- Customer Integration?

3. Plan for Enterprise-readiness

- Security
- Monitoring
- Operations tooling
- Bi-directional communication

confluent HIVEMQ

Choose the right tool stack and infrastructure

Understand Trade-Offs and choose the right options for deployments

- Edge
- On Premise
- Cloud

Use the best tools for the job

- Confluent Platform for Event Streaming
- HiveMQ for MQTT messaging and connectivity

confluent HIVEMQ

Separation of concerns

1. **Devices**
2. **Gateway**
3. **Integration**
4. **Data Streaming**
5. **Consumer Apps**

Decouple tasks

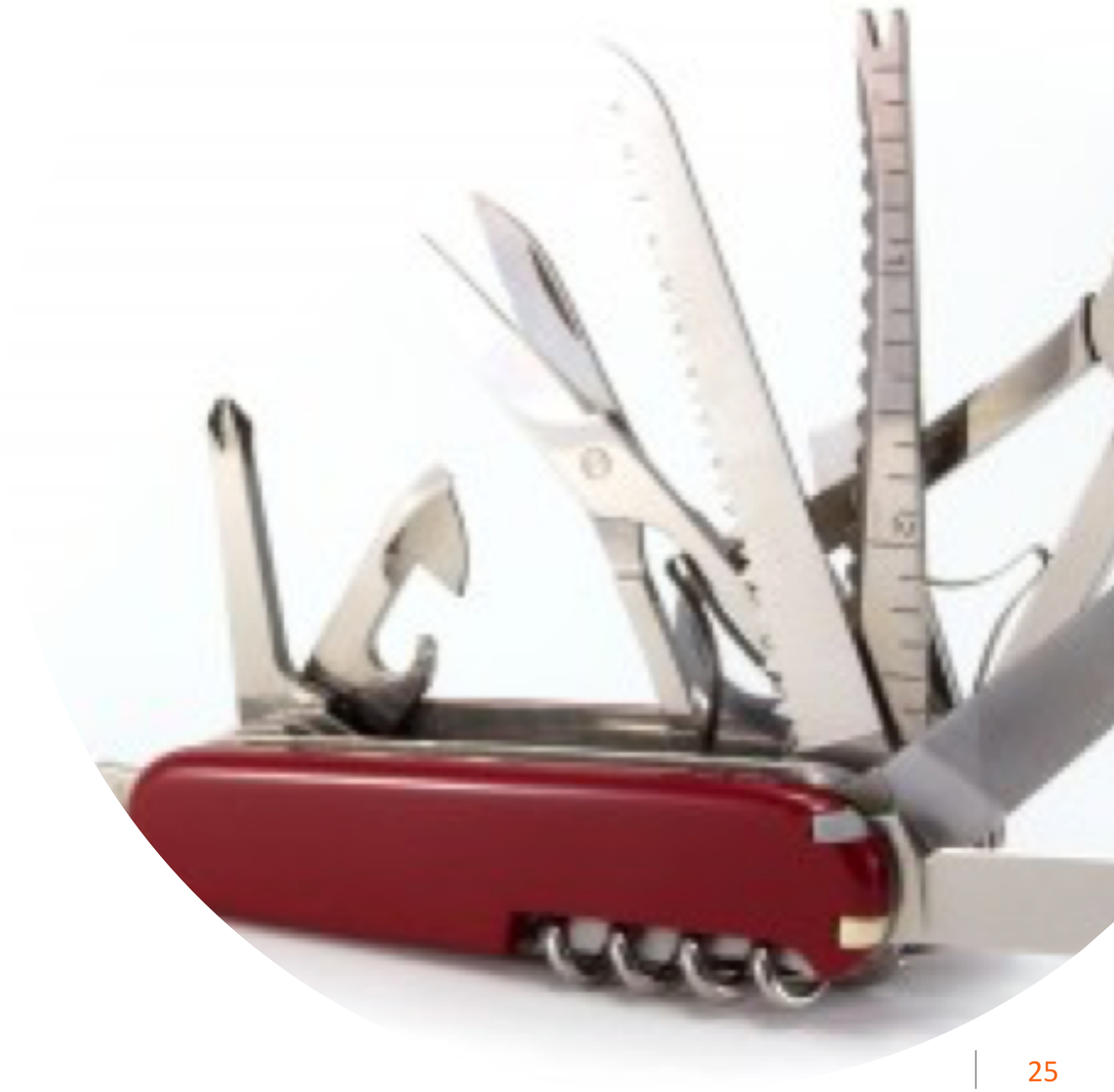
- Source integration
- Data processing
- Business logic
- Sink integration
- Analytics
- …

confluent HIVEMQ

Different data for different use cases

- Database, Data Lake
- Search
- Real time, Near Real Time, Batch
- Streaming, Request-Response
- CQRS, Event Sourcing
- Machine Learning

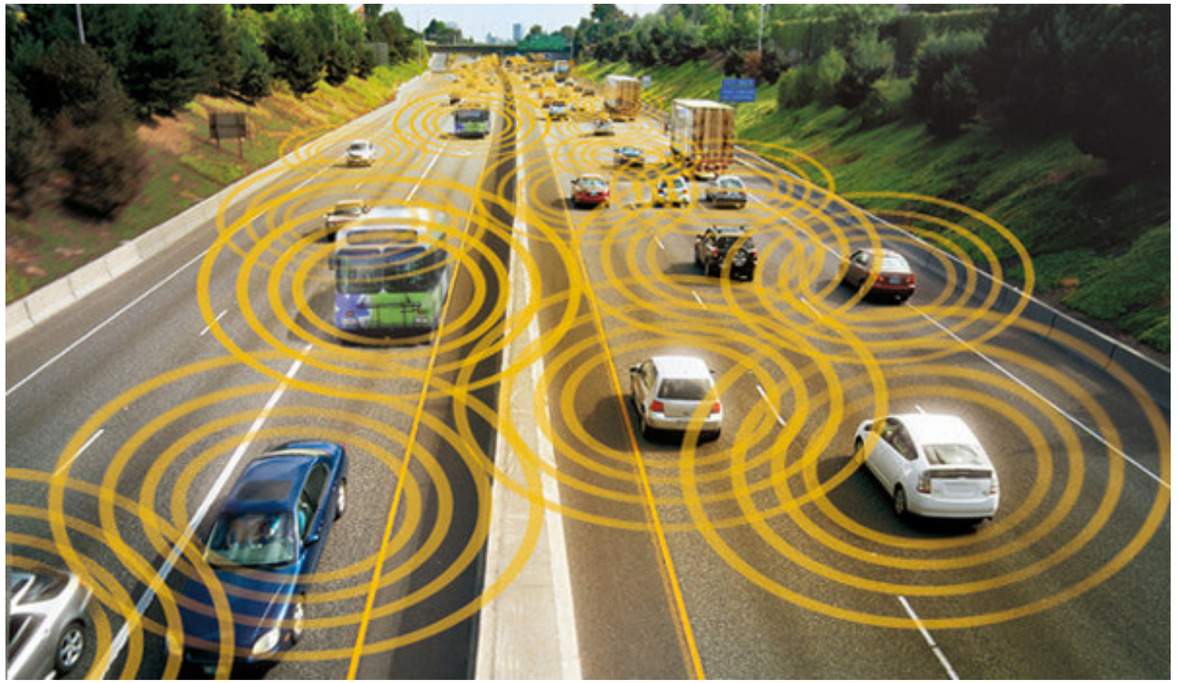
There is no single MASTER DATA EVENT…



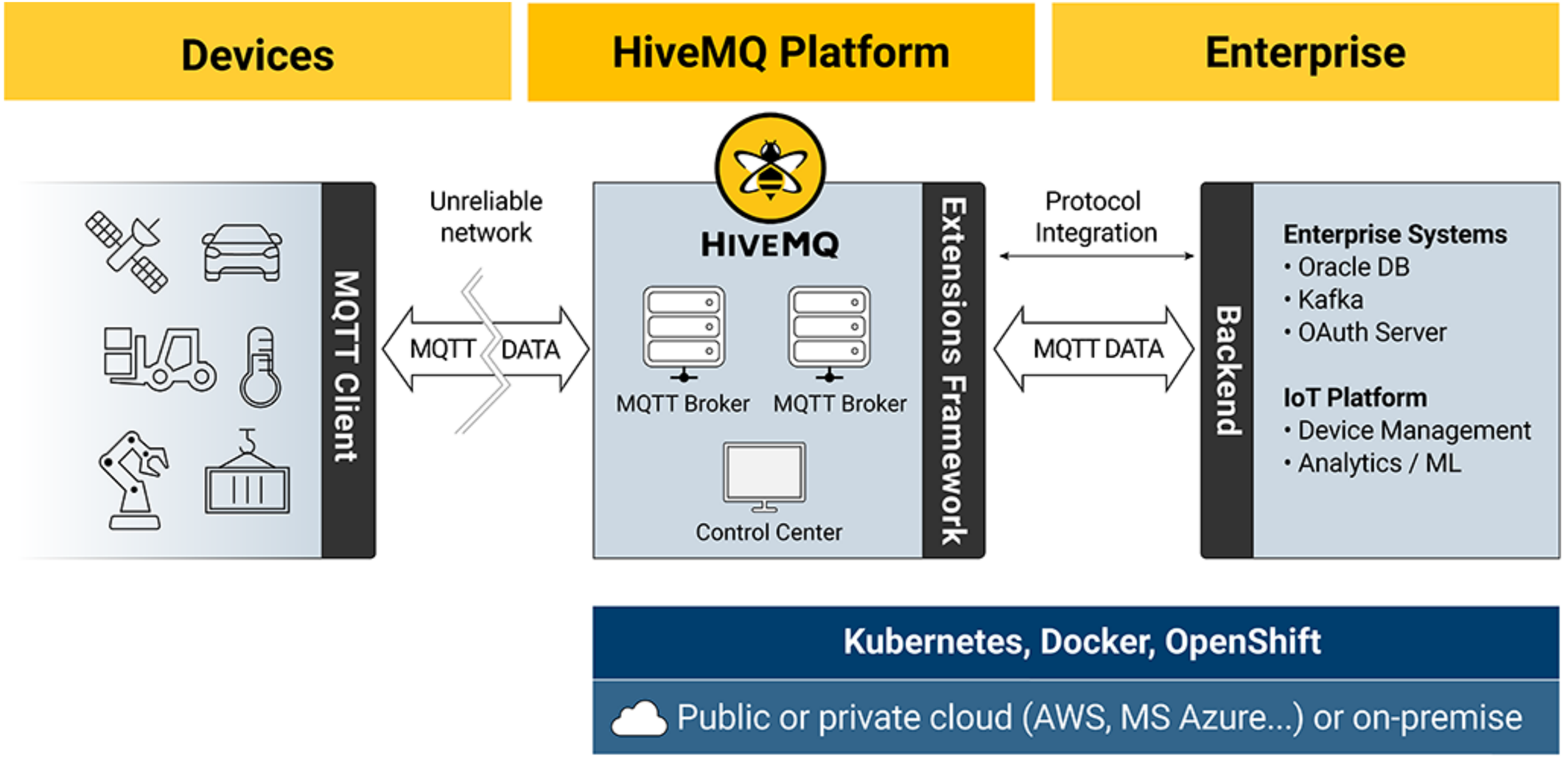
confluent HIVEMQ

Agenda

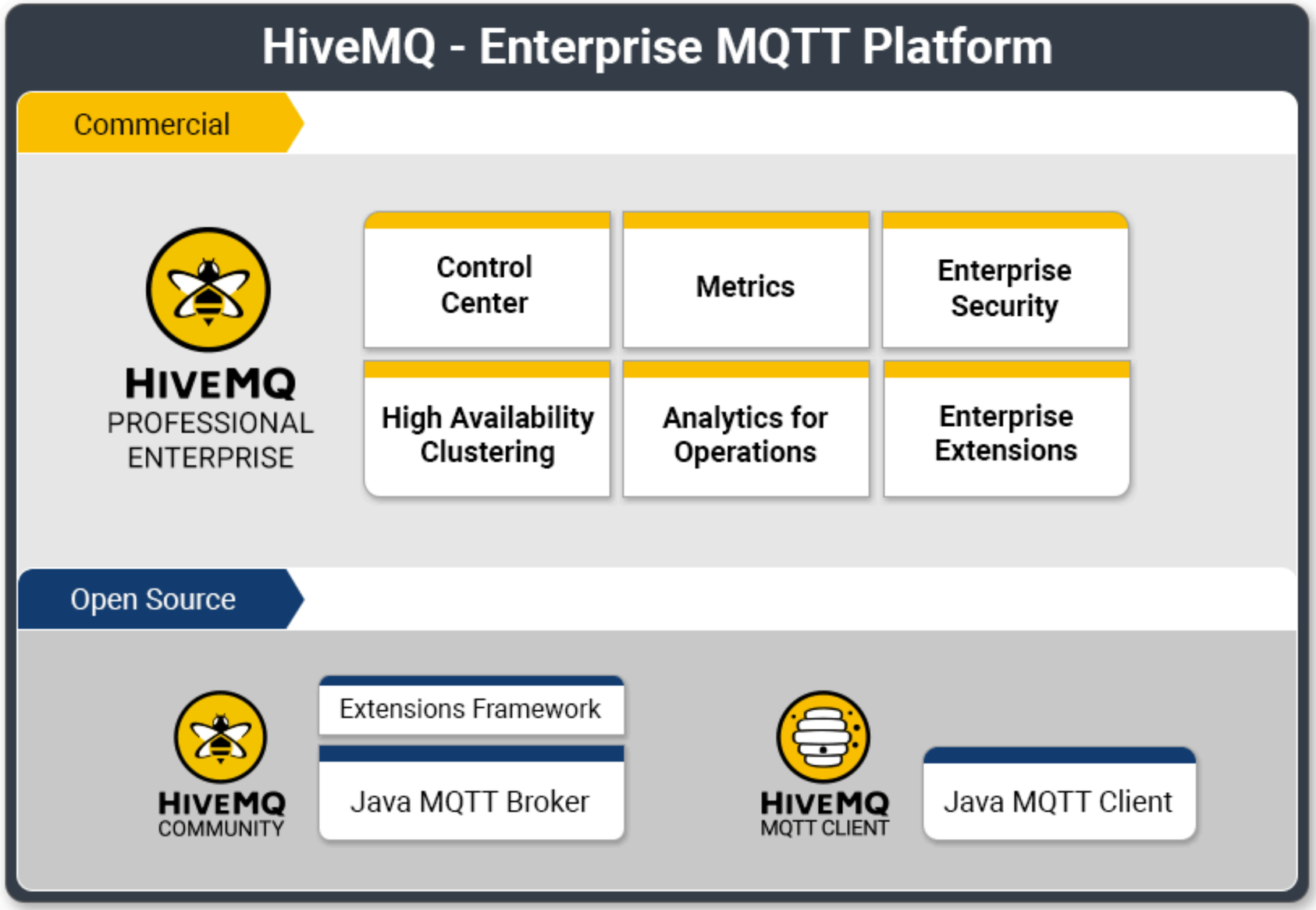
- Use Case
- Architecture
- Live Demo
- Best Practices
- **Next steps**

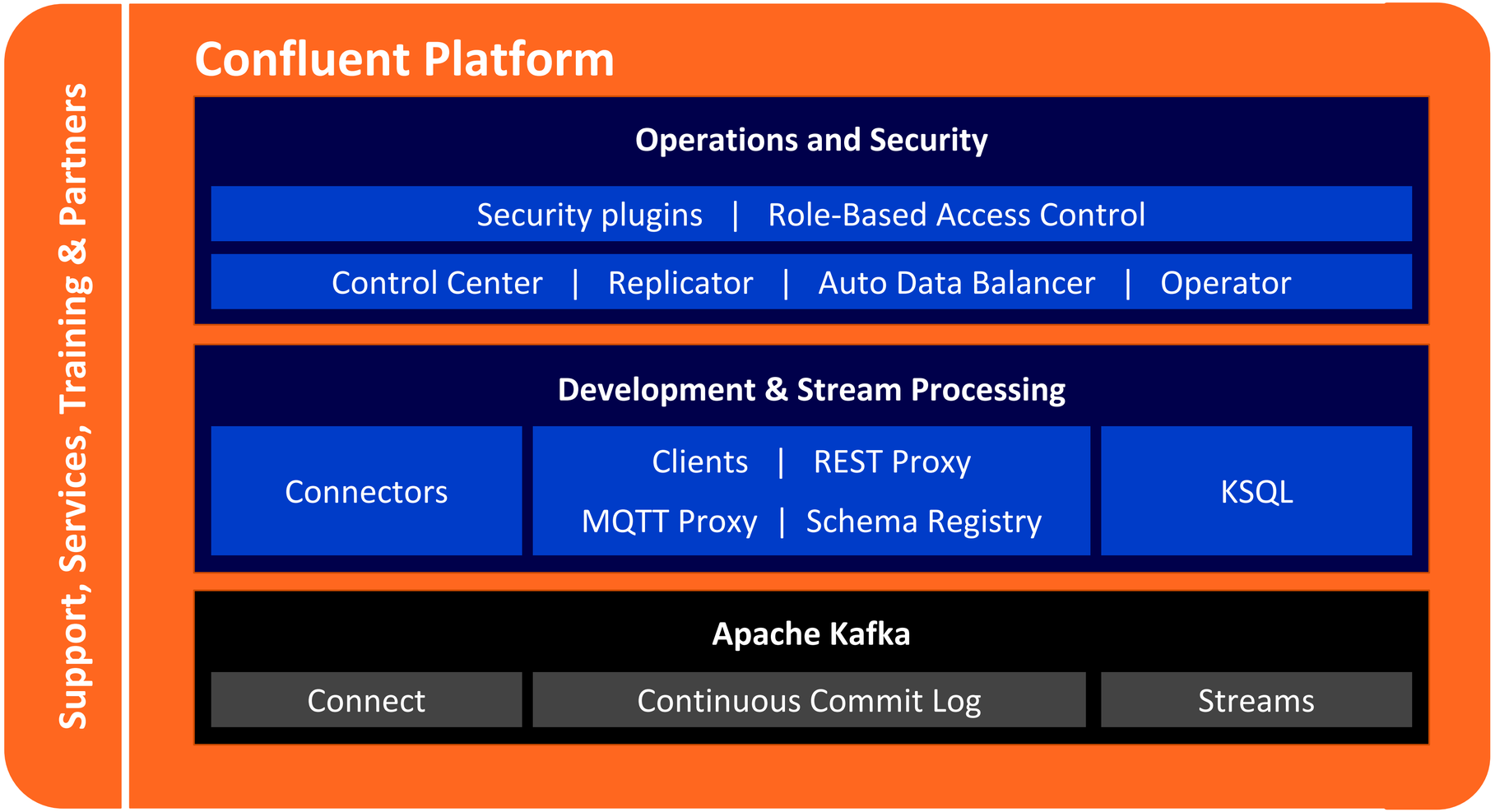


The HiveMQ Platform



The HiveMQ Platform – Open Source and Enterprise-grade





Mission-critical Reliability

Complete Event Streaming Platform

Freedom of Choice

Self-Managed Software

Fully-Managed Service

Datacenter | **Public Cloud** | **Confluent Cloud**

Spend your time on your applications!

Confluent Cloud

Cloud-Native Confluent Platform **Fully-Managed Service**

Available on the leading public clouds with **mission-critical SLAs** and **consumption-based pricing**.

aws

Google Cloud Platform

Microsoft Azure

Serverless Kafka characteristics:

Pay-as-you-go, elastic auto-scaling, abstracting infrastructure (topics not brokers)

confluent HIVEMQ

Next steps…

Try out the demo in 30 minutes:

https://github.com/kaiwaehner/hivemq-mqtt-tensorflow-kafka-realtime-iot-machine-learning-training-inference

http://bit.ly/kafka-mqtt-ml-demo

Check out the **documentation and blog posts**

- HiveMQ and Apache Kafka - Streaming IoT Data and MQTT Messages: https://www.hivemq.com/blog/streaming-iot-data-and-mqtt-messages-to-apache-kafka/
- Internet of Things (IoT) and Event Streaming at Scale with Apache Kafka and MQTT: https://www.confluent.io/blog/iot-with-kafka-connect-mqtt-and-rest-proxy

Contact us for questions or any other feedback:

- Website, Email, Slack, Phone, …
- Dominik: dominik@hivemq.com , Kai: kai.waehner@confluent.io

Questions? Feedback?

Please contact us!

Kai Waehner
Technology Evangelist

kai.waehner@confluent.io
LinkedIn
@KaiWaehner
www.confluent.io
www.kai-waehner.de

Dominik Obermaier
CTO HiveMQ

dominik.obermaier@hivemq.com
www.linkedin.com/in/dobermai
www.hivemq.com
www.twitter.com/dobermai