Unleashing Apache Kafka and TensorFlow in Hybrid Architectures

Kai Waehner
Technology Evangelist
kontakt@kai-waehner.de
LinkedIn
@KaiWaehner
www.confluent.io
www.kai-waehner.de
Poll

In which of these environments, if at all, are you using Apache Kafka today?

1. Self managed - On premise (Bare metal, VM)
2. Self managed - Public cloud (AWS EC2, etc.)
3. Self managed - Virtual private cloud (Kubernetes, Mesos, etc.)
4. Kafka as a Service – Public cloud
5. We do not currently use Apache Kafka
Disclaimer: This is a fictional story (but not far from reality)…
Global automotive company builds connected car infrastructure

Digital Transformation
- Improve customer experience
- Increase revenue
- Reduce risk

3 years ago
Project begins

Today
Connected car infrastructure in production for first use cases

2 years in the future
Improved processes leveraging machine learning
Analyze and act on critical business moments

Windows of Opportunity

- Fraud Detection
- Real Time Tracking
- Cross Selling
- Transportation Rerouting
- Customer Service
- Predictive Maintenance
- Inventory Management

Seconds
Minutes
Hours
Machine Learning (ML)

...allows computers to find hidden insights without being explicitly programmed where to look.
The First Analytic Models

How to deploy the models in production?
...real-time processing?
...at scale?
...24/7 zero downtime?
Hidden Technical Debt in Machine Learning Systems

Impedance mismatch between model development and model deployment

https://www.slideshare.net/NickPentreath/productionizing-spark-ml-pipelines-with-the-portable-format-for-analytics-100788521
Scalable, Technology-Agnostic ML Infrastructures

Meet Michelangelo: Uber’s Machine Learning Platform
By Jeremy Hermann & Mike Del Balso
November 8, 2017

https://www.infoq.com/presentations/netflix-ml-meson
https://eng.uber.com/michelangelo
https://www.infoq.com/presentations/paypal-data-service-fraud

What is this **kafka** thing used everywhere?
Apache Kafka—The Rise of a Streaming Platform
Apache Kafka at Scale at Tech Giants

* Kafka is not just used by tech giants
** Kafka is not just used for big data

> 4.5 trillion messages / day

> 6 Petabytes / day

“You name it”
Confluents - Business Value per Use Case

Key Drivers:
- Increase Revenue (make money) $↑
- Decrease Costs (save money) $↓
- Mitigate Risk (protect money) $

Strategic Objectives (sample):
- Core Business Platform
- Improve Customer Experience (CX)
- Increase Operational Efficiency
- Migrate to Cloud
- Fraud Detection
- Regulatory

Example Use Cases:
- Website / Core Operations (Central Nervous System)
- Real-time app updates
- Customer 360
- IoT sensor ingestion
- Microservices Architecture
- Digital replatforming/Mainframe Offload
- Middleware replacement
- Online Security (syslog, log aggregation, Splunk replacement)
- Online Fraud Detection
- Faster transactional processing / analysis incl. Machine Learning / AI

Example Case Studies (of many):
- The [Silicon Valley] Digital Natives; LinkedIn, Netflix, Uber, Yelp...
- Real Time Streaming Platform for Communications and Beyond: Capital One
- Simplifying Omni-channel Retail at Scale: Target
- Connected Car: Navigation & improved in-car experience: Audi
- Predictive Maintenance: Audi
- Developer Velocity - Building Stateful Financial Applications with Kafka Streams: Funding Circle
- Mainframe Offload: RBC
- Application Modernization: Multiple Examples
- Kafka as a Service - A Tale of Security and Multi-Tenancy: Apple
- Detect Fraud & Prevent Fraud in Real Time: PayPal
- Streaming Platform in a regulated environment (e.g. Electronic Medical Records): Celmatix
Apache Kafka’s Open Source Ecosystem as Infrastructure for ML
Apache Kafka’s **Open Ecosystem** as Infrastructure for ML

- **Kafka Connect**
- Go/.NET /Python Kafka Producer
- **Rest Proxy**
- **Schema Registry**
- **Kafka Streams**
- **KSQL**
Okay, let’s build our own ML infrastructure step by step. Where do we start?
Real time tracking of the cars to enable new, innovative digital services

The big data team has the data already.
Why did they choose Confluent Cloud?

**Why Cloud?**
- Extreme Scale
- Dynamic Instances
- Special Hardware (GPUs, TPUs,)

**Why Confluent Cloud?**
- No operations burden
- 99.95 Enterprise SLA, guaranteed high throughput, low ms latency end-to-end
- Confluent Ecosystem, Multi-Cloud + on premise Deployments,
- End-to-End monitoring with Confluent Control Center
Replication of IoT Data from AWS to GCP

We should also use Kafka, but—oh no... GCP is the strategic cloud for the analytics team!
Poll

Which of the following use cases would you expect for hybrid Kafka deployments and replication of data?

1. Backup / Separation of Concerns

2. Disaster recovery

3. Active / active deployments

4. Cloud migration

5. We don't anticipate using a hybrid deployment
Data Preprocessing

- Use KSQL to preprocess data at scale without coding
- Use SQL statements for interactive analysis
  + deployment to production at scale
- Leverage e.g. Python with KSQL REST interface

Preprocessing
Filter, transform, anonymize, extract features

Data Ready for Model Training

Data needs to be preprocessed at scale and reusable!
Preprocessing with KSQL

```
SELECT car_id, event_id, car_model_id, sensor_input
FROM car_sensor c
LEFT JOIN car_models m ON c.car_model_id = m.car_model_id
WHERE m.car_model_type = 'Audi_A8';
```
Data Ingestion into a Data Store

Preprocessed Data → Kafka Connect → Data Store

- “Kafka Benefits Under the Hood”
- Out-of-the-box connectivity
- Data format conversion
- Single message transformation (including error-handling)

There isn’t just one ML solution. We need to be flexible!
Let's build some models at extreme scale using TensorFlow and TPUs!
Model Training without additional Data Store

https://github.com/tensorflow/io/tree/master/tensorflow_io/kafka

- Native integration between Kafka and TensorFlow
- KafkaDataSet and KafkaOutputSequence for TensorFlow
- Written in C++ (linked with librdkafka)
- Part of the graph in TensorFlow
- Direct training and inference from streaming data
- No data storage like S3 or HDFS needed
Analytic Model (Autoencoder for Anomaly Detection)
AutoML

Hold on. This is still too complex for many of our use cases! We don’t have many data scientists. How can we help developers build models?

“One-click Data-in Model-out Simplicity”
Replayability — a log never forgets!

Time

Distributed Commit Log

Producer

Distributed Commit Log

Google Cloud Storage

HDFS

Different models with same data
Different ML frameworks
AutoML compatible
A/B testing

Model A
TensorFlow

Model B
TensorFlow

Model X
DataRobot

Apache Kafka and Machine Learning – Kai Waehner
The Need for Local Data Processing

We are ready to use our models for predictions, BUT all the PII data needs to be processed in our local data center!
Self managed on premise deployment for model deployment and monitoring

Confluent Operator takes over the challenge of operating Kafka on Kubernetes!
(Automated provisioning, scaling, fail-over, partition rebalancing, rolling updates, monitoring, …)

Oh no... self-managed Kubernetes + Kafka ecosystem = operations nightmare

What about scaling brokers, external clients, persistent volumes, failover, rolling upgrades, and so on?
Okay, we deploy locally.

But how to do the model inference?

Can Kafka and Kubernetes help here?
Model Deployment - Option 1: RPC communication to do model inference

Input Event

kafka Streams

Prediction

Model Serving

TensorFlow Serving

gRPC

Request

Response
Model Deployment - Option 2:
Model interference natively integrated into the App

Input Event

Prediction
Stream Processing vs. Request-Response for Model Serving

Pros of a Model Server:
- **Simple integration** with existing technologies and organizational processes
- **Easier to understand** if you come from non-streaming world
- **Later migration** to real streaming is also possible
- **Model management built-in** for different models, versioning and A/B testing

Cons (== Pros of Deployment in the Streaming App):
- **Worse latency** as remote call instead of local inference
- **No offline inference** (devices, edge processing, etc.)
- **Coupling** the availability, scalability, and latency/throughput of your Kafka Streams application with the **SLAs of the RPC interface**
- **Side-effects** (e.g., in case of failure) **not covered by Kafka processing** (e.g., exactly once)
Confluent Schema Registry for Message Validation

- “Kafka Benefits Under the Hood”
- Schema definition + evolution
- Forward and backward compatibility
- Multi data center deployment

I am a little bit worried. How can we ensure every team in every data center produces and consumes correct data?
Monitoring the infrastructure for ML

Build vs. Buy
Hosted vs. Managed
Basic vs. Advanced
KSQL and Deep Learning (Auto Encoder) for Anomaly Detection

Kafka Ecosystem

Other Components

Car Sensors

MQTT Proxy

Kafka Cluster

KSQL

Filter Anomalies

Apply Analytic Model

Kafka Connect

Potential Defect

Elastic search

All Data

Real Time Emergency System

Grafana

At the edge

On premise DC

Apply Analytic Model

Filter Anomalies

Potential Defect

On premise DC

At the edge

Real Time Emergency System

Grafana
Model Training with Python, KSQL, TensorFlow, Keras and Jupyter

https://github.com/kaiwaehner/python-jupyter-apache-kafka-ksql-tensorflow-keras
"CREATE STREAM AnomalyDetection AS
SELECT sensor_id, detectAnomaly(sensor_values)
FROM car_engine;"
End-to-End Sensor Analytics…

Python, Jupyter Notebook, TensorFlow, Keras, Apache Kafka, KSQL and MQTT
Model Training with Python, KSQL, TensorFlow, Keras and Jupyter

Use Case: Fraud Detection for Credit Card Payments

We use test data set from Kaggle as foundation to train an unsupervised autoencoder to detect anomalies and potential fraud in payments.

Focus of this project is not just model training, but the whole Machine Learning infrastructure including data ingestion, data preprocessing, model training, model deployment and monitoring. All of this needs to be scalable, reliable and performant.

Technology: Python, Jupyter, TensorFlow, Keras, Apache Kafka, KSQL

This project shows a demo which combines

- simplicity of data science tools (Python, Jupyter notebooks, NumPy, Pandas)
- powerful Machine Learning / Deep Learning frameworks (TensorFlow, Keras)
- reliable, scalable event-based streaming technology for production deployments (Apache Kafka, Kafka Connect, KSQL).

https://github.com/kaiwaehner/python-jupyter-apache-kafka-ksql-tensorflow-keras
Deep Learning UDF for KSQL for Streaming Anomaly Detection of MQTT IoT Sensor Data

https://github.com/kaiwaehner/ksql-udf-deep-learning-mqtt-iot
Comparing our current project status to others

Well, we are not there yet, but getting closer every month!
Which of the following use cases are you most likely to utilize Kafka for over the next year?

1. Data ingestion (processing e.g. batch in Spark)
2. Data pipeline (processing with Kafka)
3. Stream processing (e.g. Kafka Streams, KSQL)
4. Stream processing with machine learning
5. Other (like microservices, event sourcing, storage)
Confluent Delivers a Mission-Critical Event Streaming Platform

Confluent Platform

Management & Monitoring
Control Center | Security

Enterprise Operations
Replicator | Auto Data Balancer | Connectors | MQTT Proxy | Kubernetes Operator

Data Compatibility
Schema Registry

Development & Connectivity
Clients | Connectors | REST Proxy | KSQL

Apache Kafka®
Core | Connect API | Streams API

DATA INTEGRATION
- Hadoop
- Database
- Data Warehouse
- CRM
- other

REAL-TIME APPLICATIONS
- Transformations
- Custom Apps
- Analytics
- Monitoring
- other

CUSTOMER SELF-MANAGED
- Datacenter
- Public Cloud

CONFLUENT FULLY-MANAGED
- Confluent Cloud
Best-of-breed Platforms, Partners and Services for Multi-cloud Streams

**Private Cloud**
Deploy on bare-metal, VMs, containers or Kubernetes in your datacenter with Confluent Platform and Confluent Operator

**Hybrid Cloud**
Build a persistent bridge between datacenter and cloud with Confluent Replicator

**Public Cloud**
Implement self-managed in the public cloud or adopt a fully managed service with Confluent Cloud
Resources and Next Steps

https://confluent.io

https://confluent.io/cloud

https://slackpass.io/confluentcommunity
#confluent-cloud

@confluentinc
Cyberconflict: A new era of war, sabotage, and fear

9:55am-10:10am Wednesday, March 27, 2019
Location: Ballroom
Secondary topics: Security and Privacy

Rate today’s session → Please rate my session!

We’re living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you’re often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we’re uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David Sanger is the national security correspondent for the New York Times as well as a national security and political contributor for CNN and a frequent guest on CBS This Morning, Face the Nation, and many PBS shows.
Questions? Feedback? Let’s connect!

Kai Waehner
Technology Evangelist
kontakt@kai-waehner.de
@KaiWaehner
www.kai-waehner.de
www.confluent.io
LinkedIn