Machine Learning with the Apache Kafka Open Source Ecosystem

How to build a Scalable, Mission-Critical Machine Learning Infrastructure?

Kai Waehner

Technology Evangelist

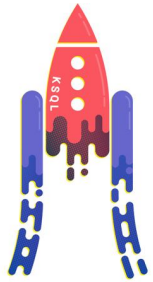
kontakt@kai-waehner.de

LinkedIn

@KaiWaehner

www.confluent.io

www.kai-waehner.de



What is eXtreme Scale?

- High Volume of Events (millions, billions, trillions)
- Big Data Sets for Analytics (GB, TB, PB)
- Dynamic Scalability for Training (minutes, hours, days)
- Real Time Prediction Process for Deployment (ms)
- Hybrid Deployments (different frameworks and clouds)

Agenda

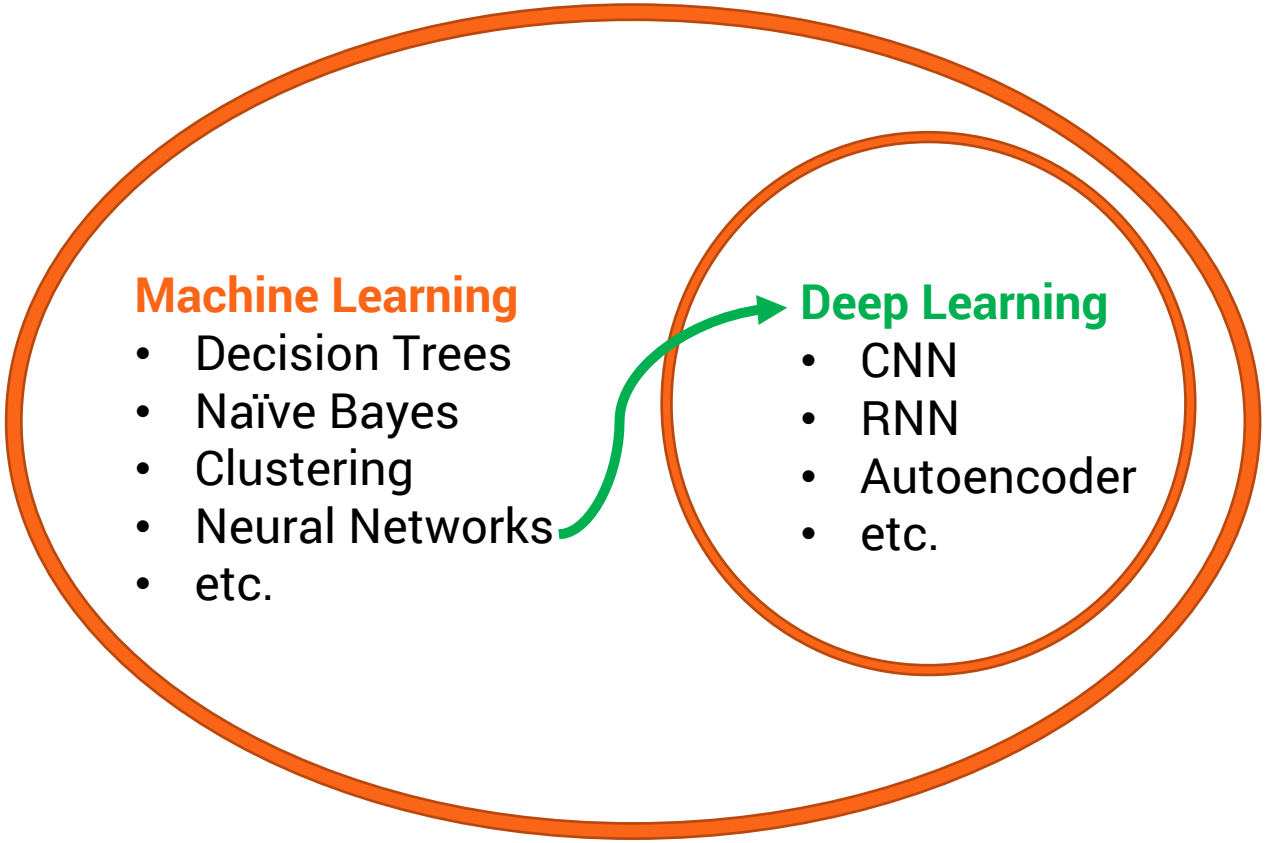
1) Added Business Value via Machine Learning
2) Apache Kafka Ecosystem as Infrastructure for Machine Learning
3) Data Ingestion and Preprocessing with Apache Kafka for Model Training
4) Predictions in Real Time with Kafka Streams and KSQL
5) DevOps and Monitoring of a Machine Learning Infrastructure

Agenda

1) **Added Business Value via Machine Learning**
2) Apache Kafka Ecosystem as Infrastructure for Machine Learning
3) Data Ingestion and Preprocessing with Apache Kafka for Model Training
4) Predictions in Real Time with Kafka Streams and KSQL
5) DevOps and Monitoring of a Machine Learning Infrastructure

Machine Learning

... allows computers to find hidden insights without being explicitly programmed where to look.



Real World Examples of Machine Learning

Spam Detection

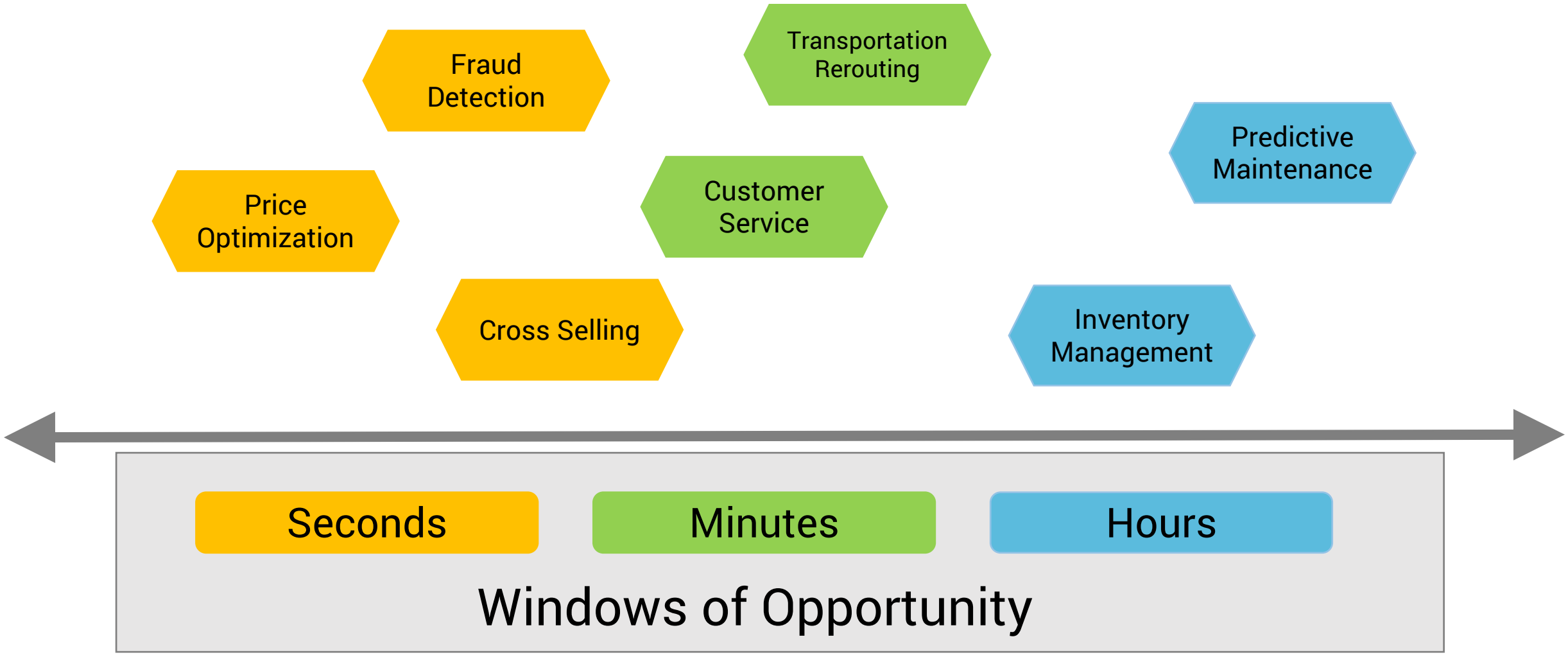
Search Results +
Product Recommendation

Picture Detection
(Friends, Locations, Products)

The Next Disruption:
Google Beats Go Champion

Your Company

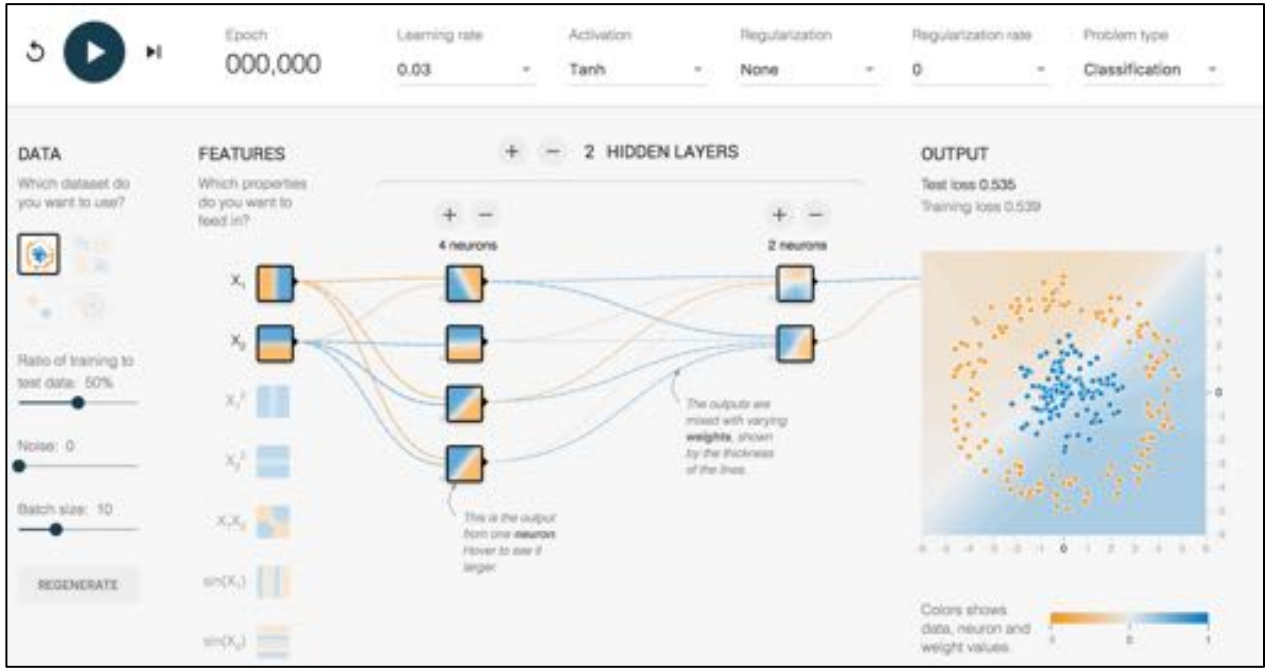
Leverage Machine Learning to Analyze and Act on Critical Business Moments



Live Demo – Building an Analytic Model

Neural Networks in Action

http://playground.tensorflow.org/



Languages, Frameworks and Tools for Machine Learning

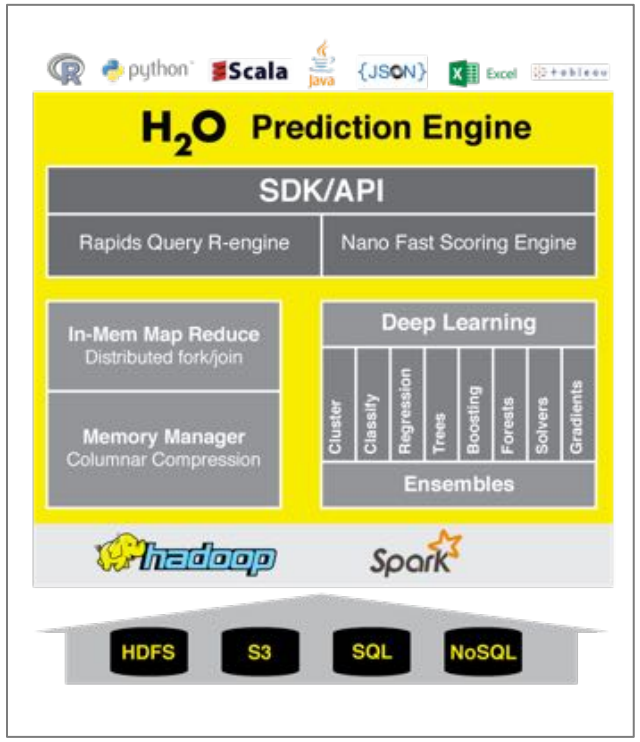
theano

DL4J
DEEPLEARNING4J

Portable Format for Analytics (PFA)

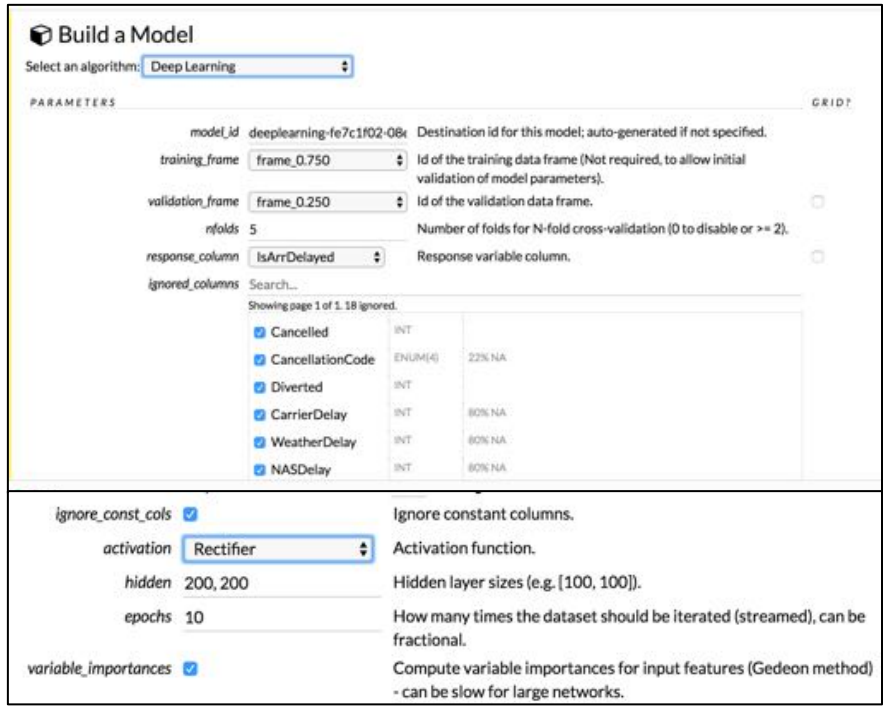
There is no Allrounder → ML-independent infrastructure needed!

Machine Learning with H2O.ai



H2O Engine

R / Python /
Scala / Flow UI



Java Code

```
@ModelPojo(name="deeplearning_fe7c1f02_08ec_4070_b784_c2531147e451", algorithm="deeplearning")
public class deeplearning_fe7c1f02_08ec_4070_b784_c2531147e451 extends GenModel {
  public hex.ModelCategory getModelCategory() { return hex.ModelCategory.Binomial; }
  public boolean isSupervised() { return true; }
  public int nfeatures() { return 12; }
  public int nclasses() { return 2; }
  // Thread-local storage for input neuron activation values.
  final double[] NUMS = new double[10];
  static class NORMMUL implements java.io.Serializable {
    public static final double[] VALUES = new double[10];
    static {
      NORMMUL_0.fill(VALUES);
    }
    static final class NORMMUL_0 implements java.io.Serializable {
      static final void fill(double[] sa) {
        sa[0] = 0.1573591362493411;
        sa[1] = 0.5316756588306932;
        sa[2] = 0.10894648014128883;
        sa[3] = 0.5257610635956896;
        sa[4] = 0.00209932098808304;
```

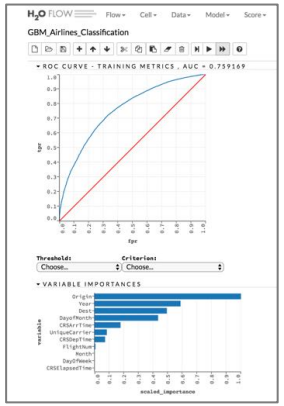
confluent

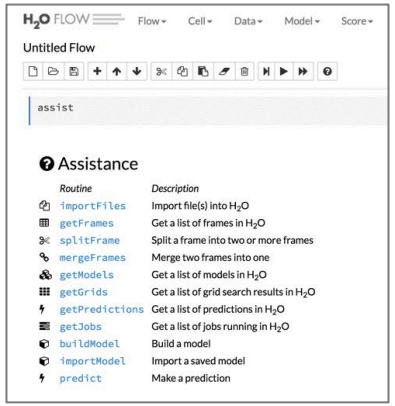
Live Demo – Building an Analytic Model

Use Case:
Airline Flight Delay Prediction

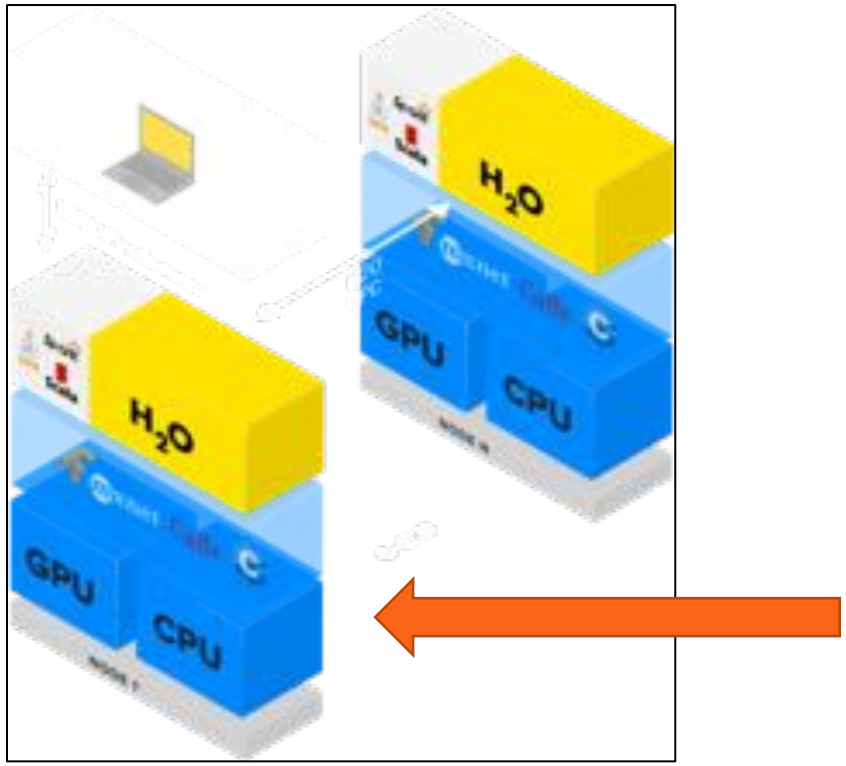
Machine Learning Algorithm:
Deep Learning
using Neural Networks

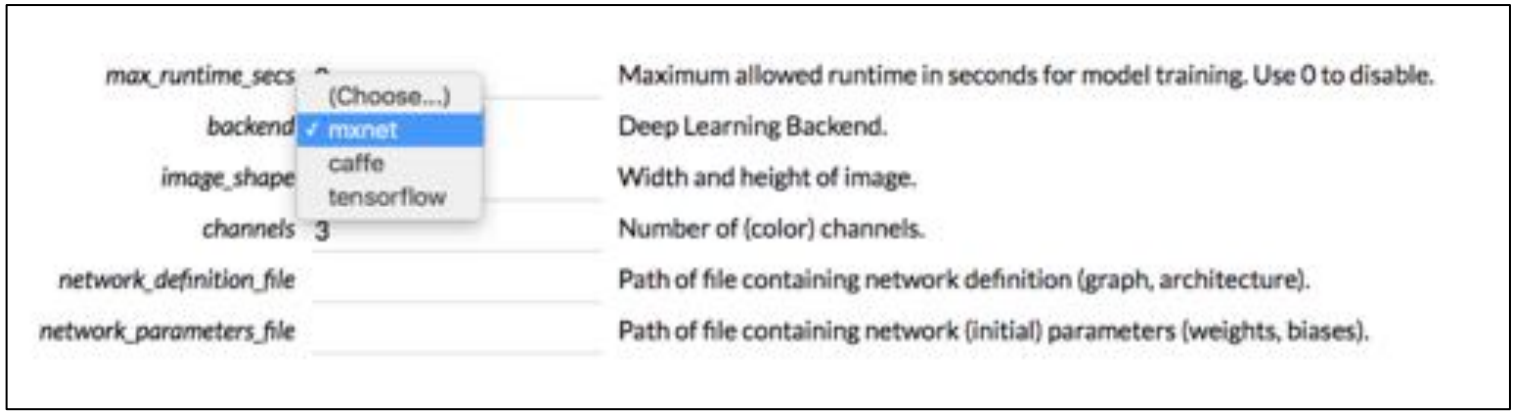
Technology:
H2O.ai, TensorFlow





H2O Deep Water (TensorFlow, MXNet, …)



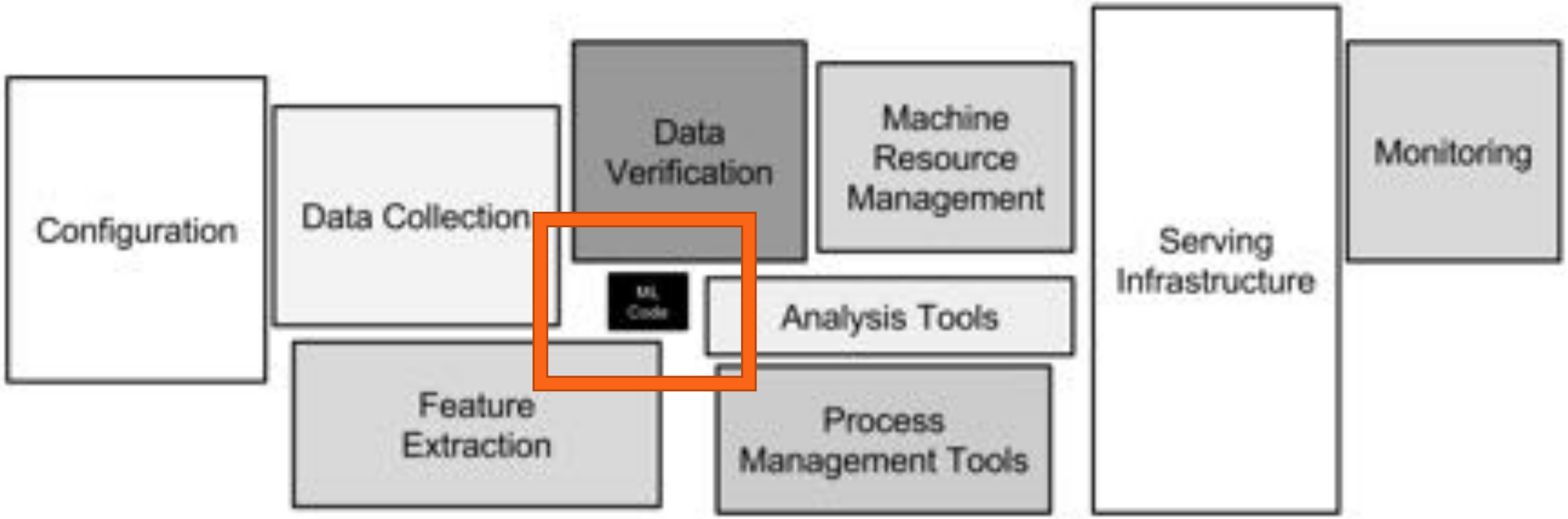


Deep Water
(H2O + TensorFlow)

Pre-Defined Networks
+
User-Defined Networks

https://h2o-release.s3.amazonaws.com/h2o/rel-vapnik/1/docs-website/h2o-docs/booklets/DeepWaterBooklet.pdf

Hidden Technical Debt in Machine Learning Systems

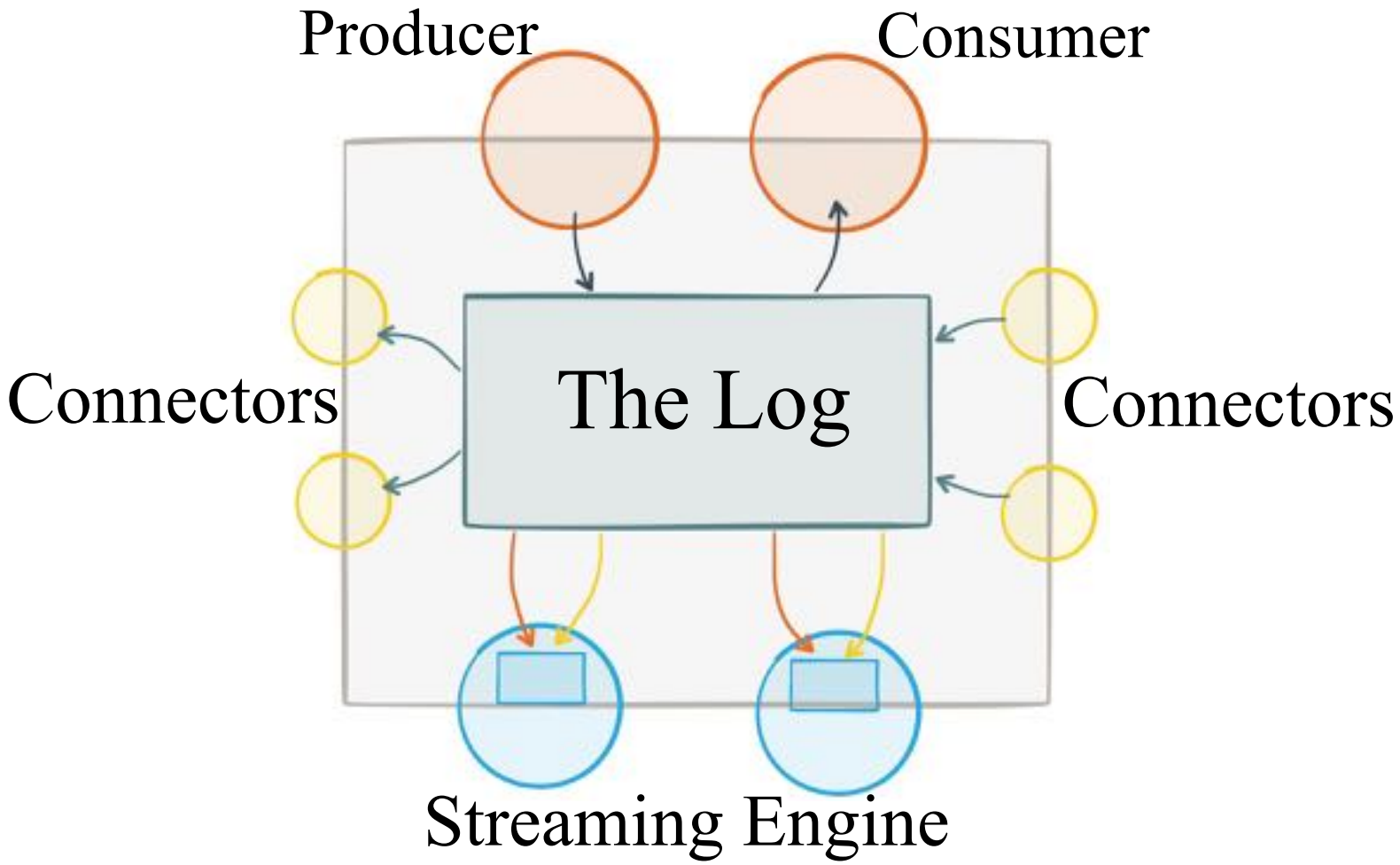


https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

Agenda

1) Added Business Value via Machine Learning
2) **Apache Kafka Ecosystem as Infrastructure for Machine Learning**
3) Data Ingestion and Preprocessing with Apache Kafka for Model Training
4) Predictions in Real Time with Kafka Streams and KSQL
5) DevOps and Monitoring of a Machine Learning Infrastructure

Apache Kafka – The Rise of a Streaming Platform

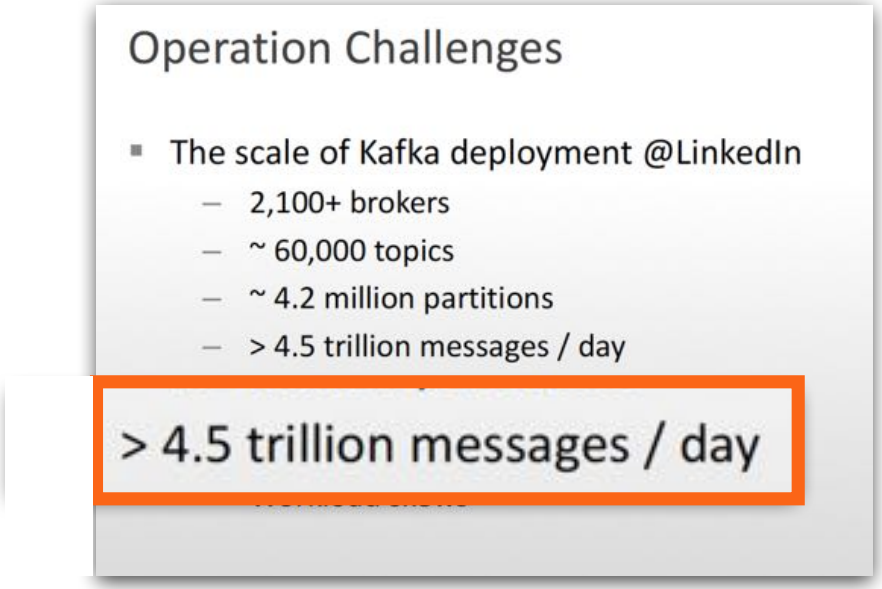


kafka

1 PUB/SUB
2 STORE
3 PROCESS

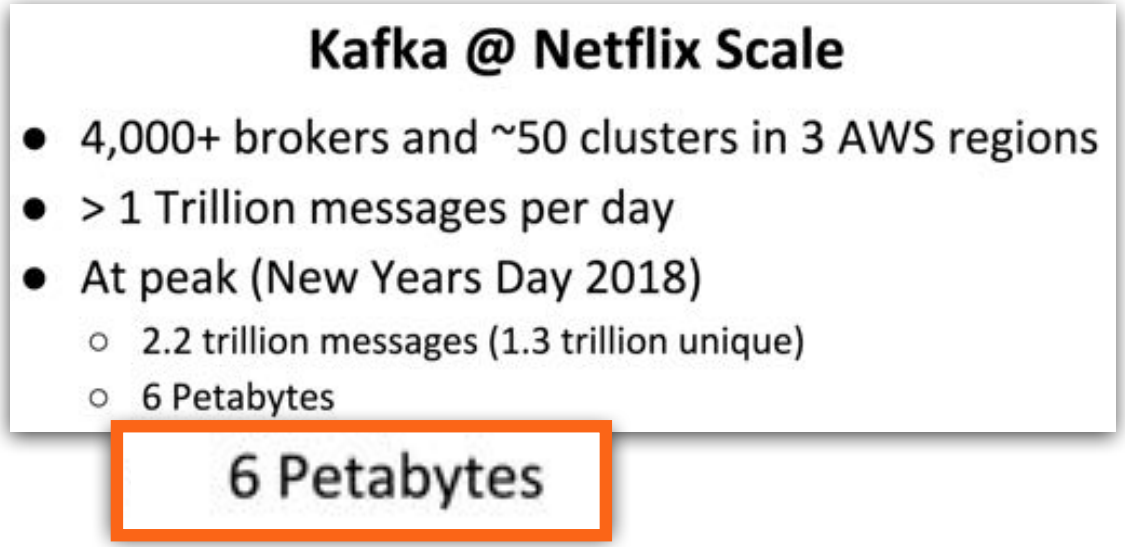
Apache Kafka at Scale

LinkedIn (2018)



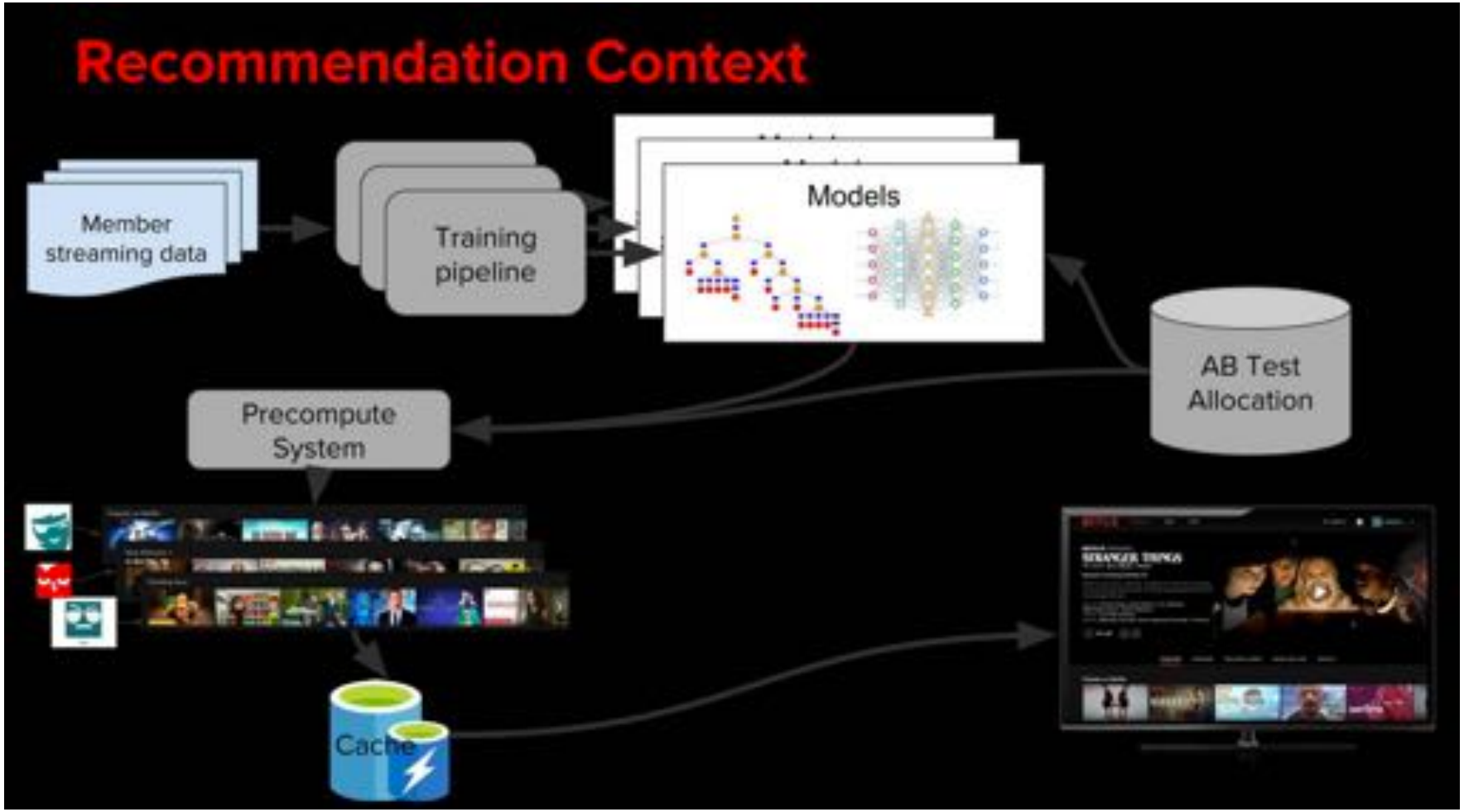
Strata DATA CONFERENCE

NETFLIX (2018)



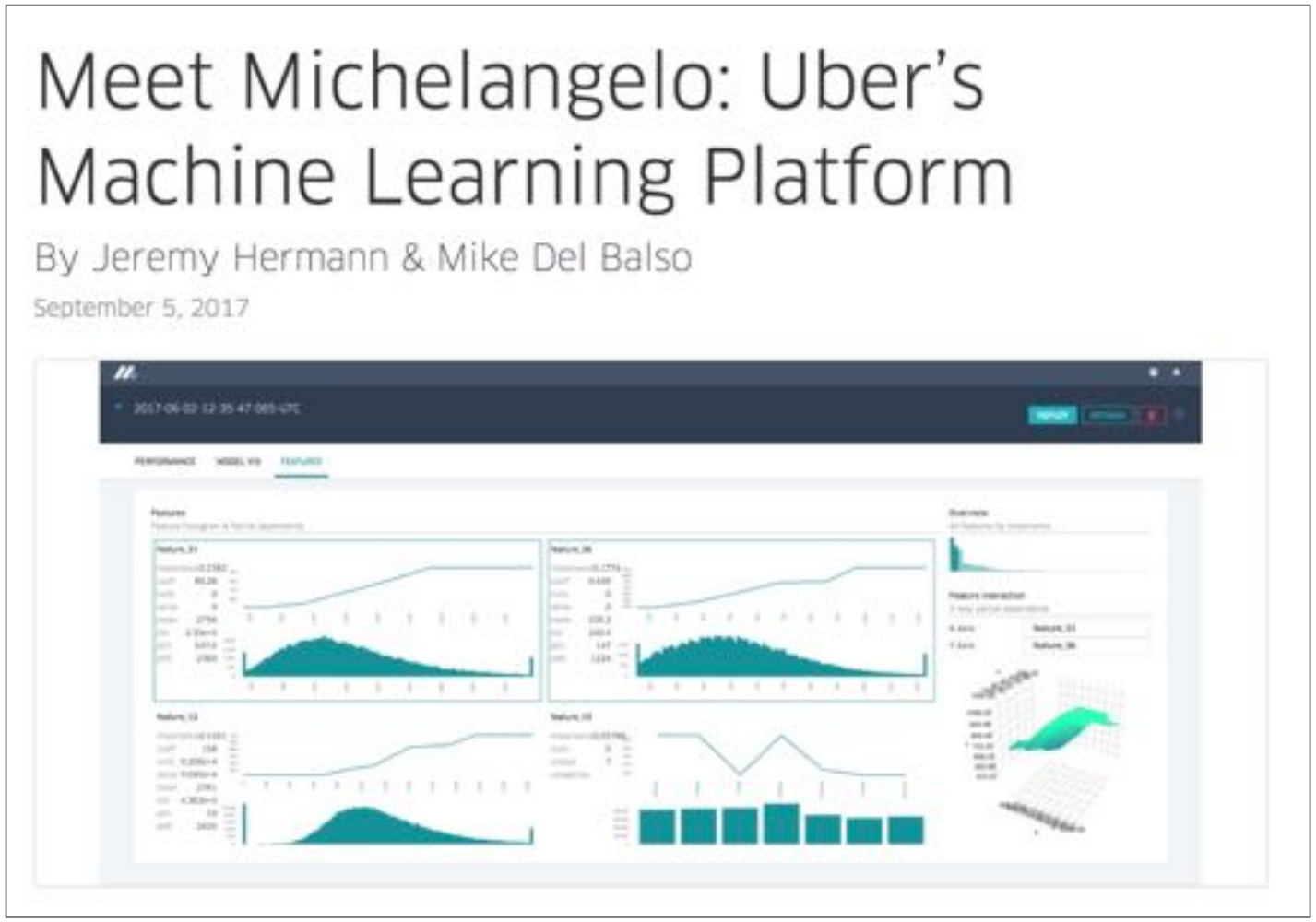
QCon

Netflix' Meson: Automation Engine for ML Pipelines



https://www.infoq.com/presentations/netflix-ml-meson

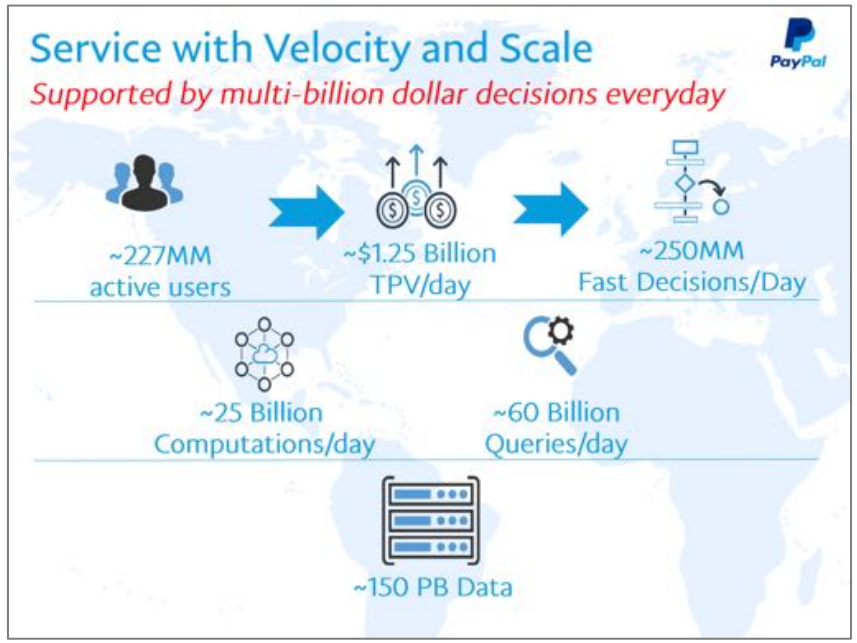
Uber's internal ML-as-a-Service Platform

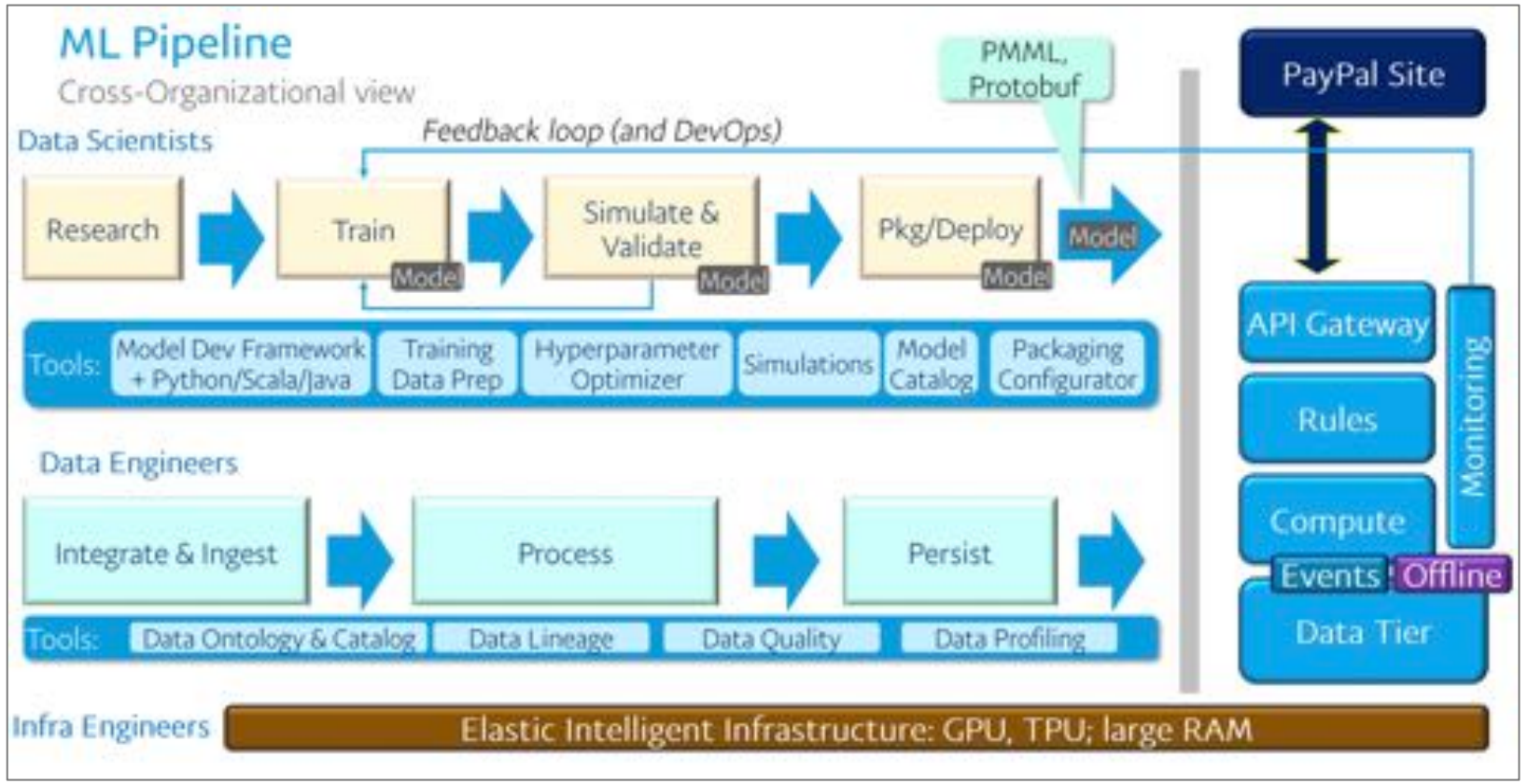


https://eng.uber.com/michelangelo

- Cover the **end-to-end ML workflow**: manage data, train, evaluate, and deploy models, make predictions, and monitor predictions
- **Supports various AI technologies**: Traditional ML models, time series forecasting, and deep learning

Paypal: Real Time Fraud Detection at Scale

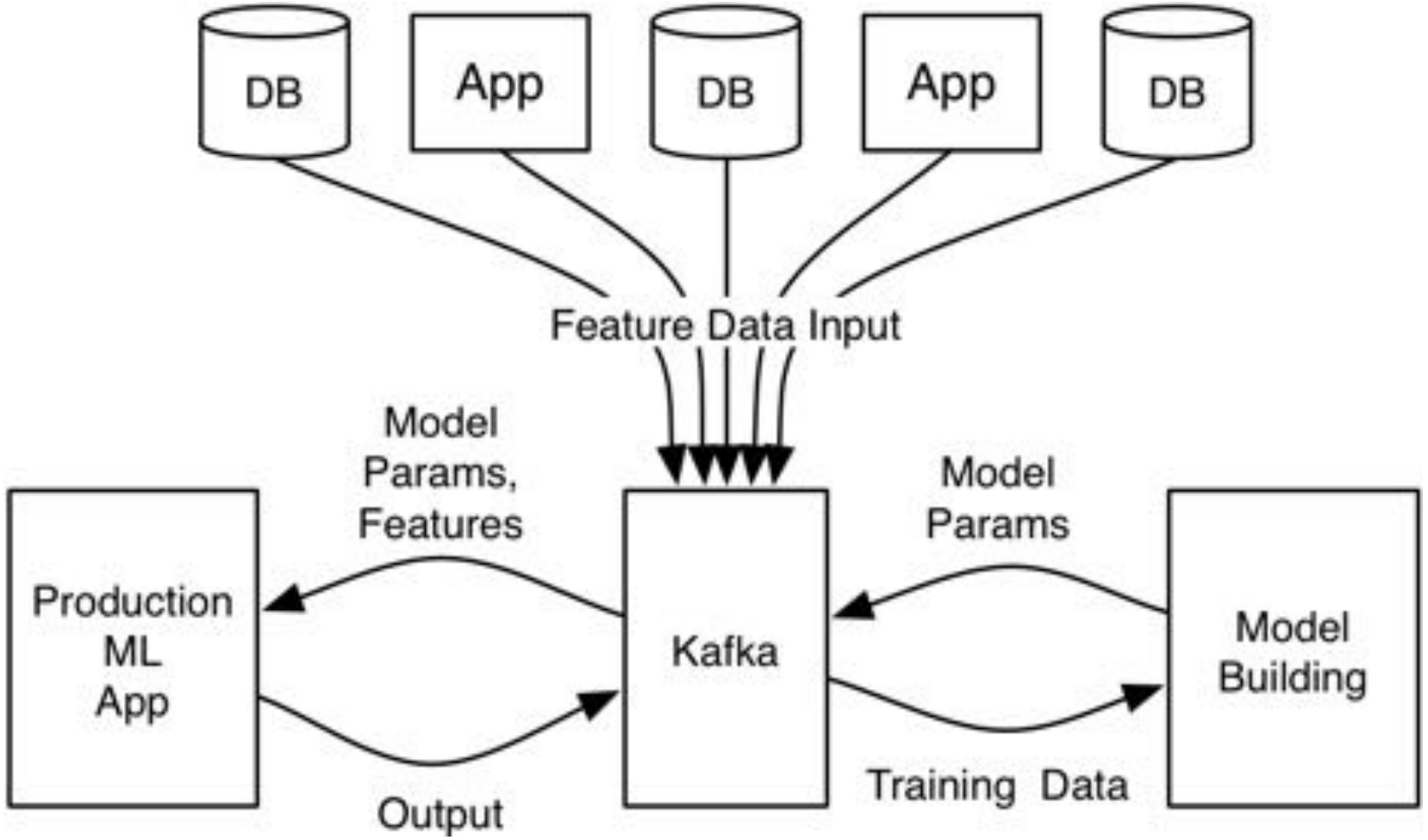




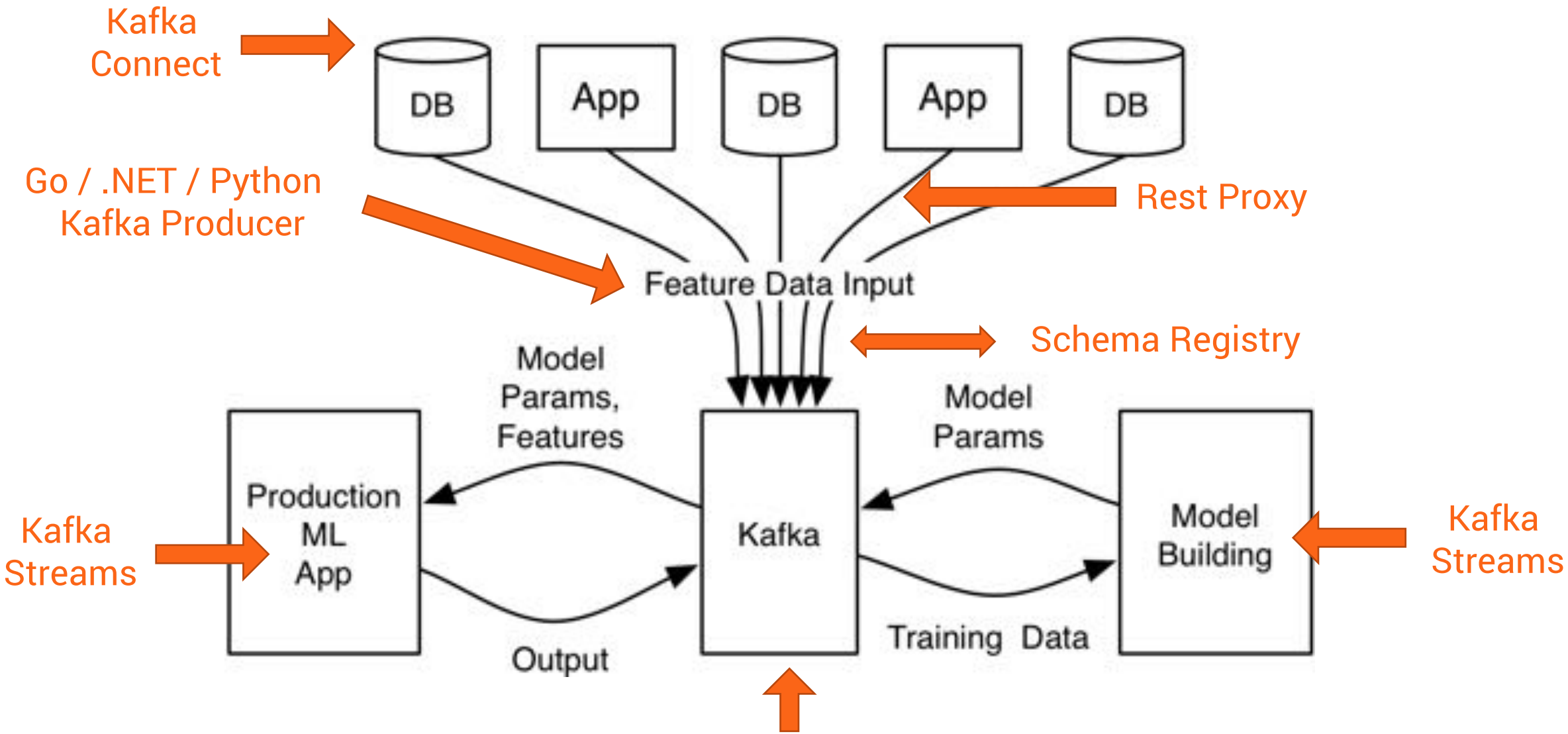
"Scalable model / framework independent infrastructure for fraud detection"

https://www.infoq.com/presentations/paypal-data-service-fraud

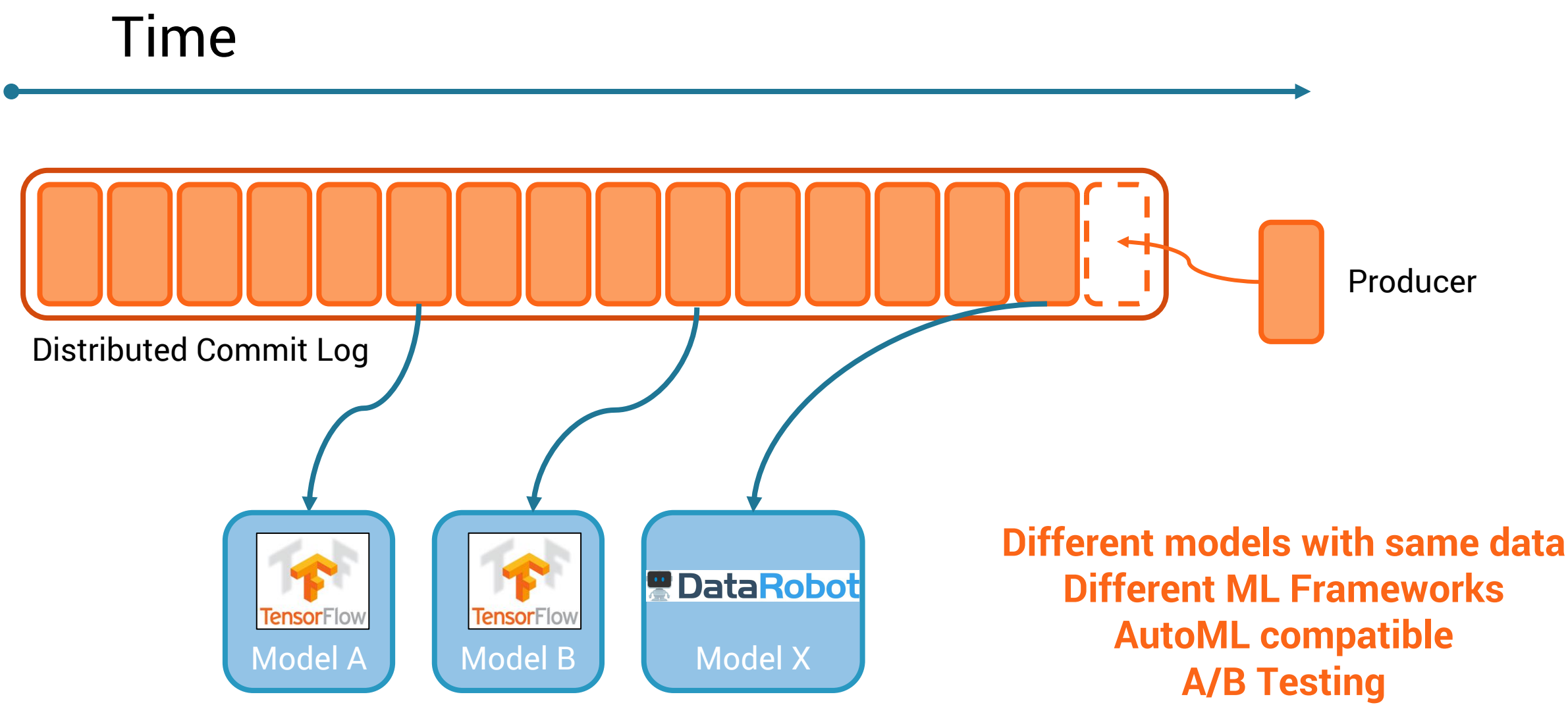
Apache Kafka's Open Source Ecosystem as Infrastructure for Machine Learning



Apache Kafka's **Open Source Ecosystem** as Infrastructure for Machine Learning

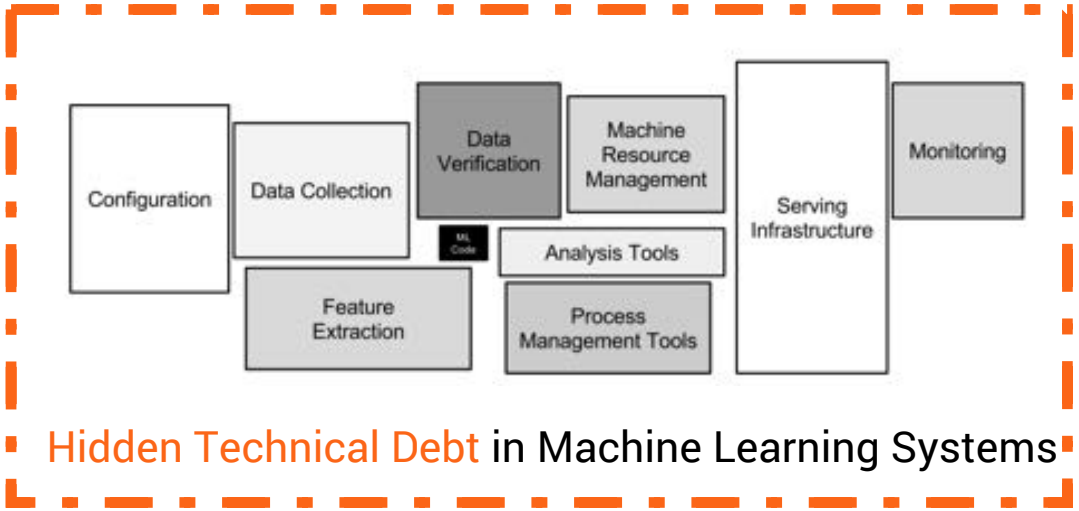


Replay-ability – A log never forgets!

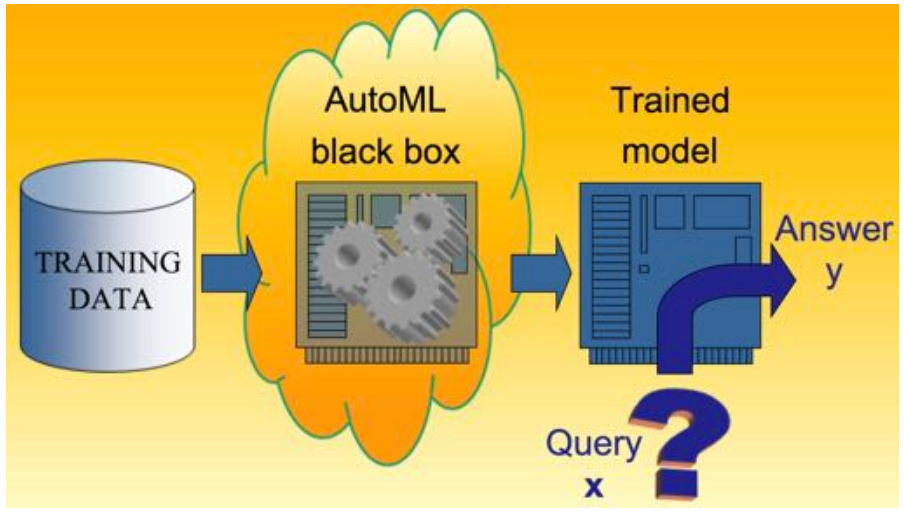


Different models with same data
Different ML Frameworks
AutoML compatible
A/B Testing

AutoML → No Data Scientist available for the ML Tasks?



Hidden Technical Debt in Machine Learning Systems



http://slideplayer.com/slide/10575150/

"One-Click Data-In Model-Out simplicity"

H2O.ai Driverless AI: Your Expert System for AI

DataRobot

Cloud AutoML

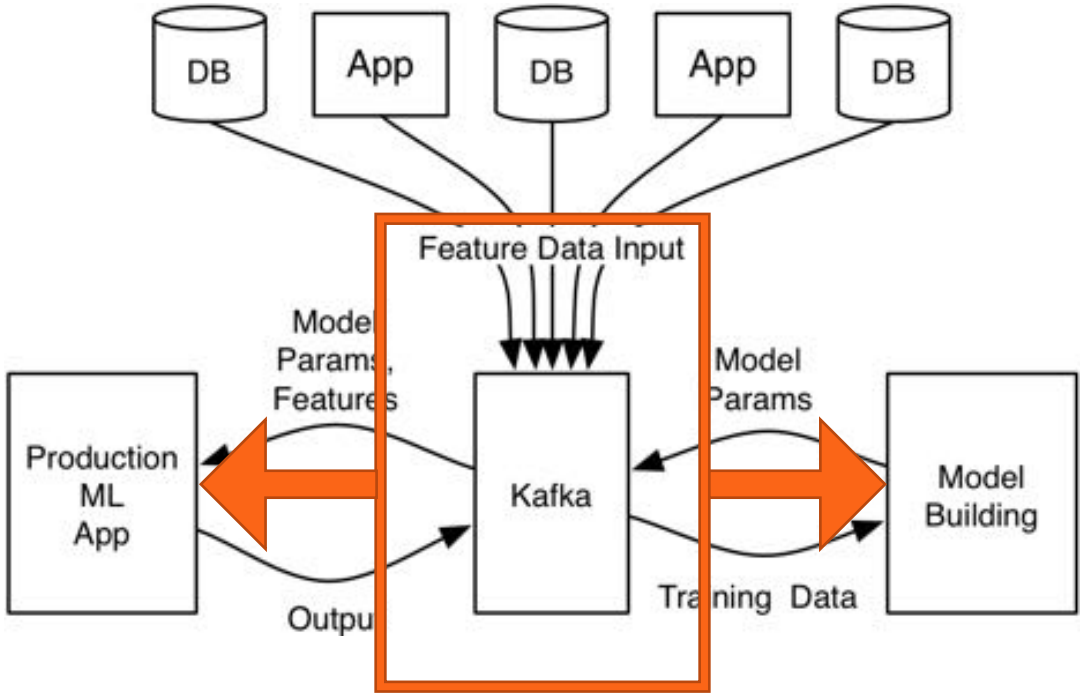
tazi.ai

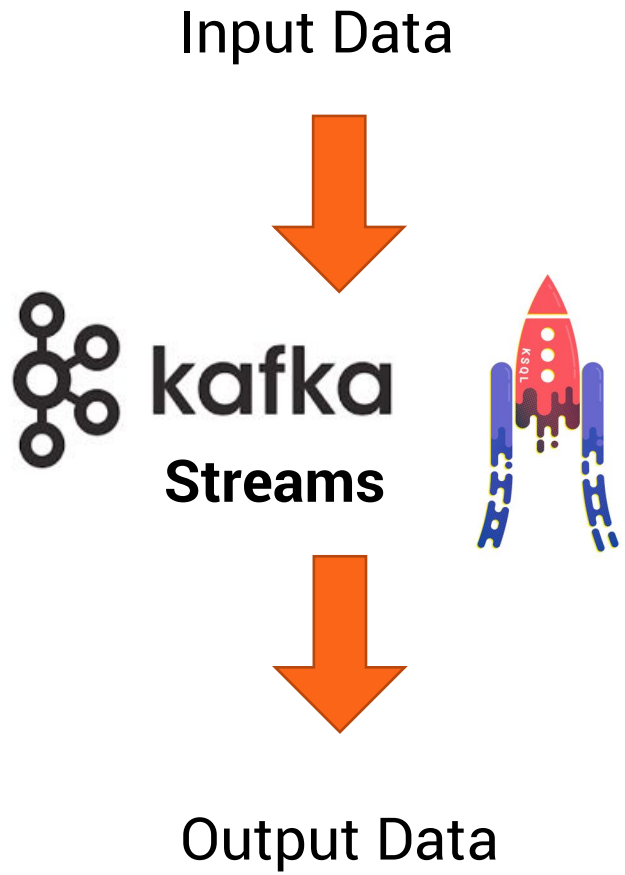
This talk does not focus on building models, but scalable infrastructure for ML

Agenda

1) Added Business Value via Machine Learning
2) Apache Kafka Ecosystem as Infrastructure for Machine Learning
3) Data Ingestion and Preprocessing with Apache Kafka for Model Training
4) Predictions in Real Time with Kafka Streams and KSQL
5) DevOps and Monitoring of a Machine Learning Infrastructure

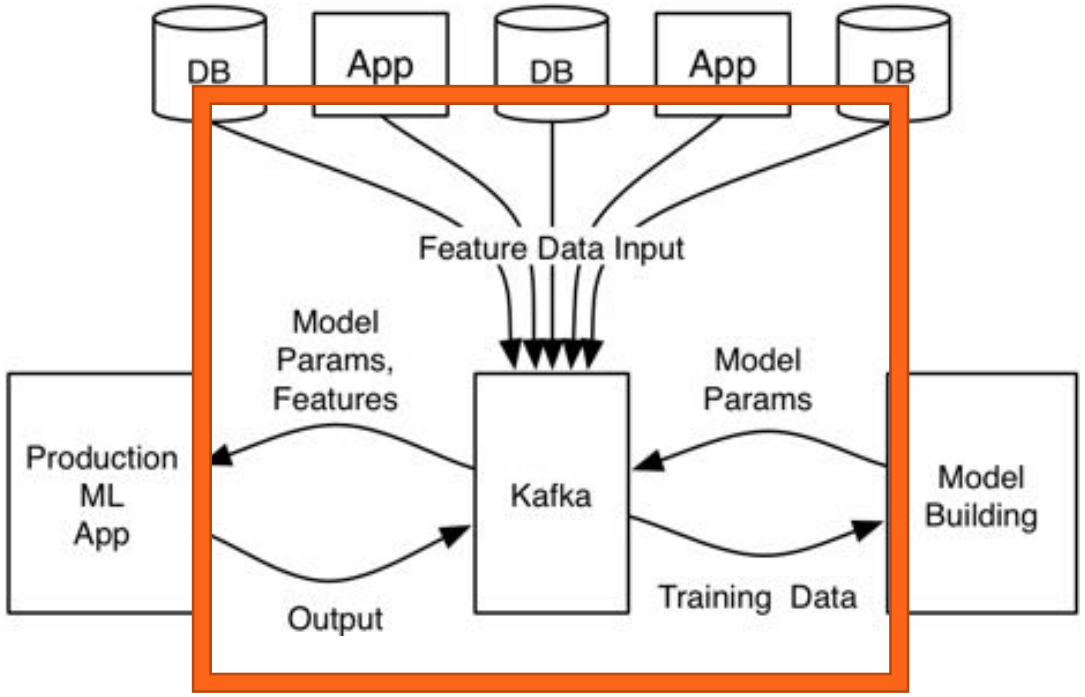
Kafka Streams / KSQL for Data Preprocessing

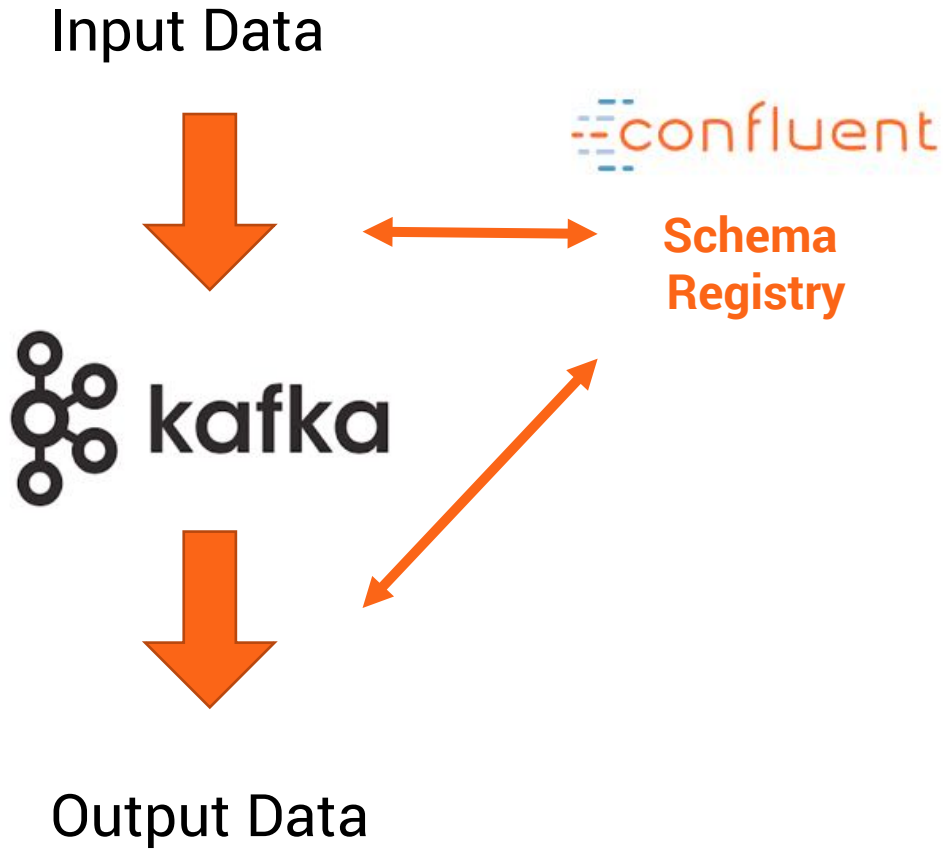




“Kafka benefits under the hood”
Streaming ETL
Same Pipeline for Training and Serving

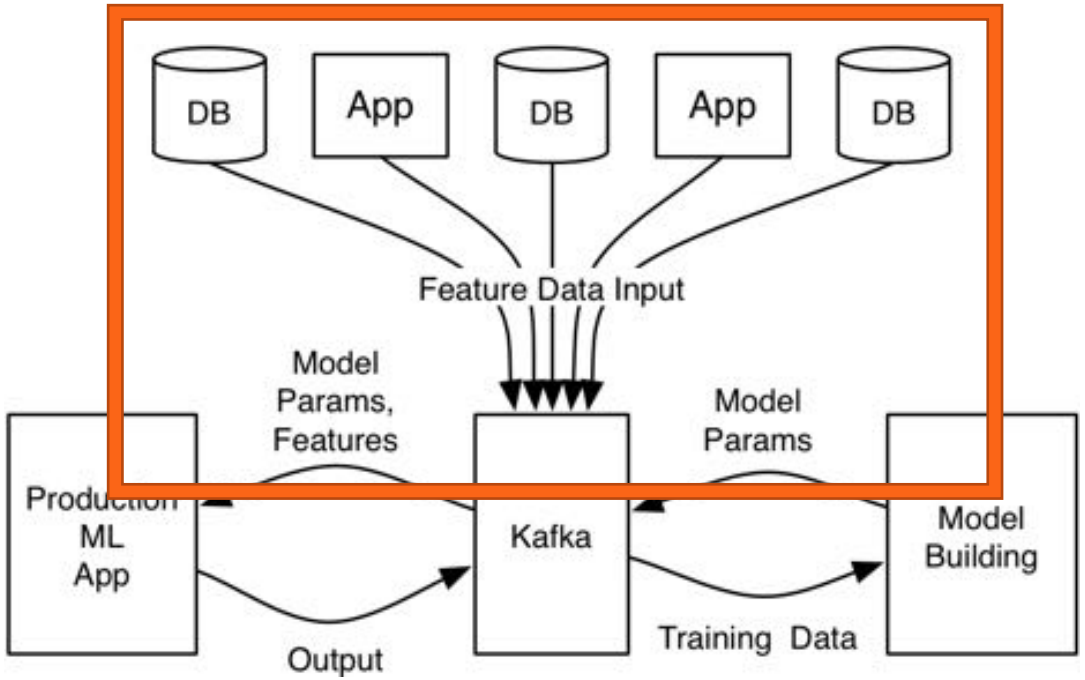
Confluent Schema Registry for Message Validation





“Kafka benefits under the hood”
Schema Definition + Evolution
Forward and Backward Compatibility

Kafka Connect for Data Ingestion



"Kafka benefits under the hood"
Out-of-the-Box Connectivity
Data Format Conversion
Simple Message Transformation

kafka Connect

amazon web services S3

HDFS

Google BigQuery

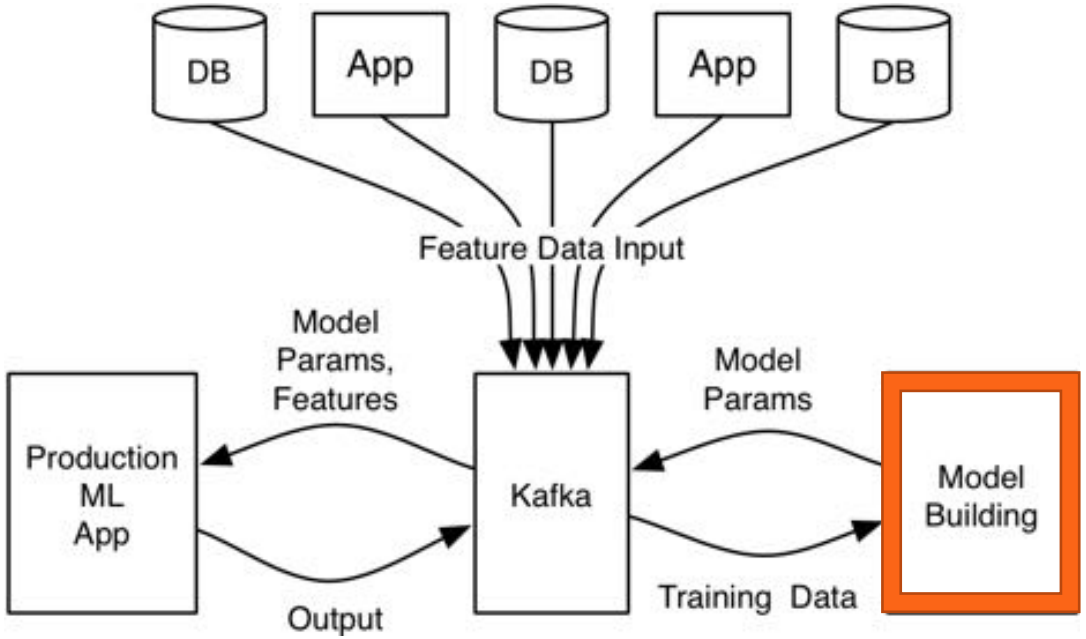
Amazon SageMaker

Apache Spark

Apache Flink

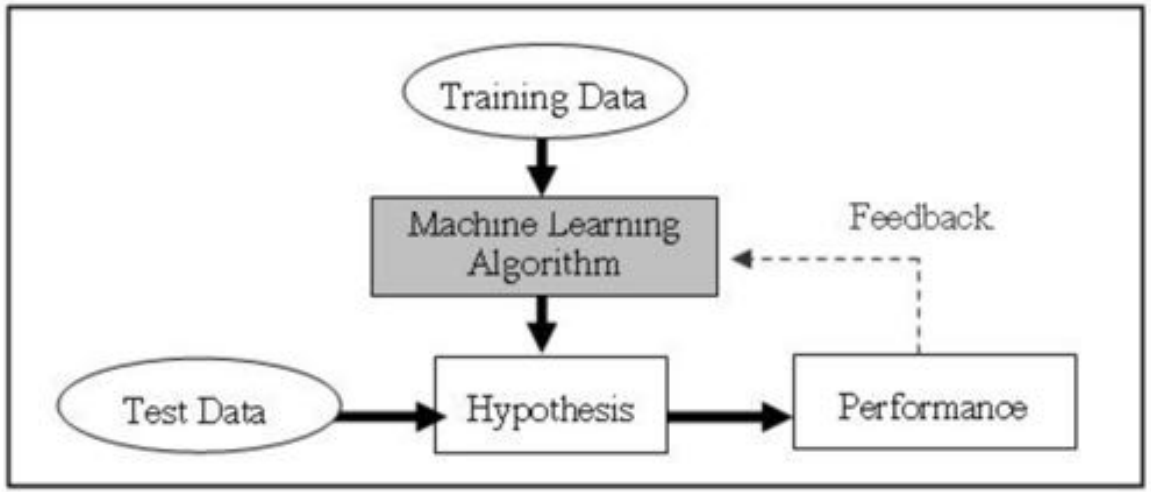
TensorFlow

Model Training



Benefits of Public Cloud

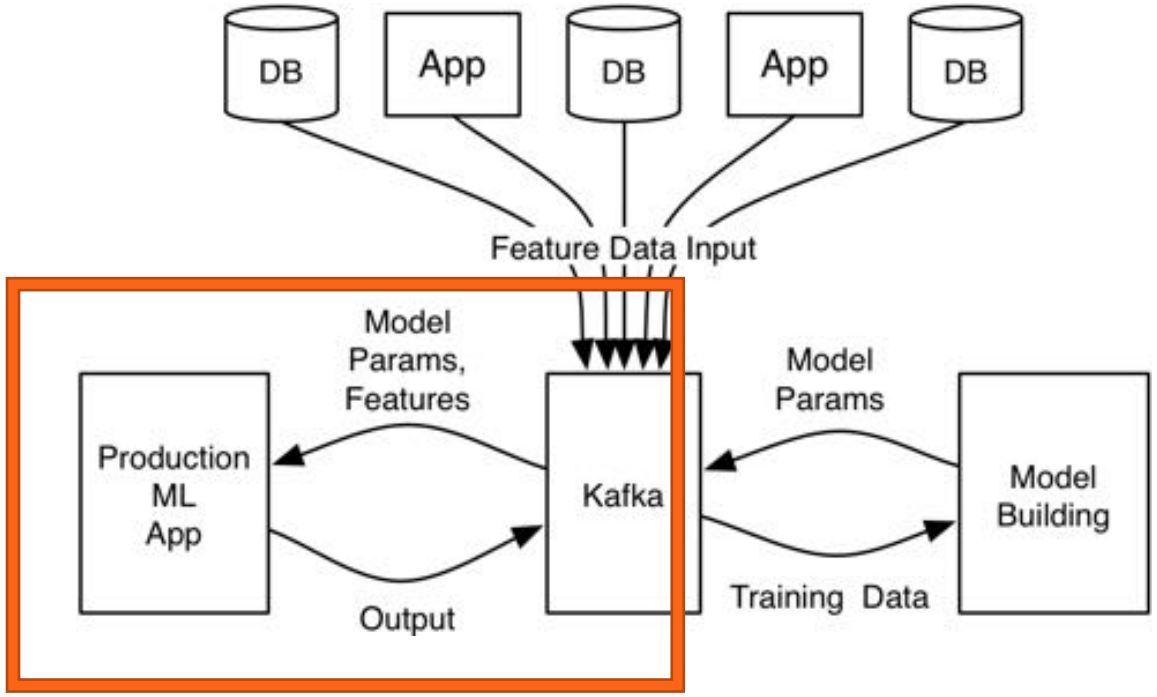
Extreme Scale
Dynamic Instances
Special Hardware



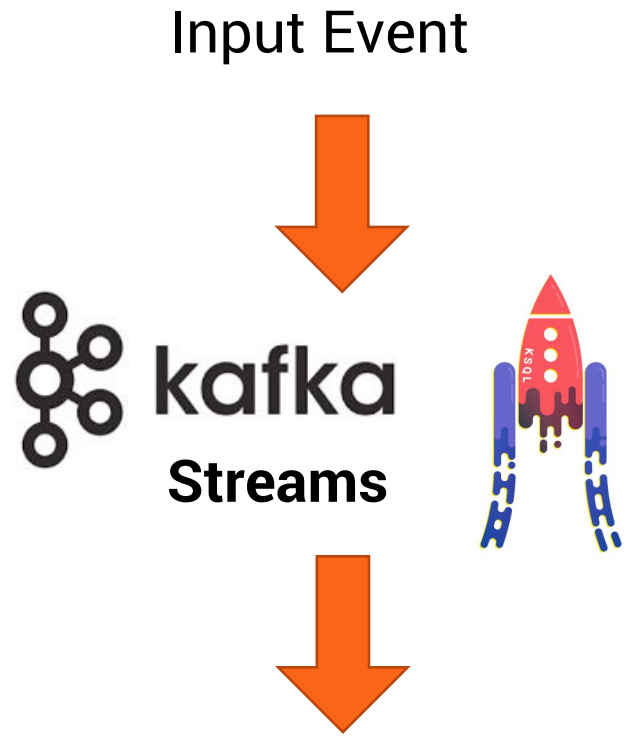
Agenda

1) Added Business Value via Machine Learning
2) Apache Kafka Ecosystem as Infrastructure for Machine Learning
3) Data Ingestion and Preprocessing with Apache Kafka for Model Training
4) **Predictions in Real Time with Kafka Streams and KSQL**
5) DevOps and Monitoring of a Machine Learning Infrastructure

Model Serving / Inference / Deployment / Scoring

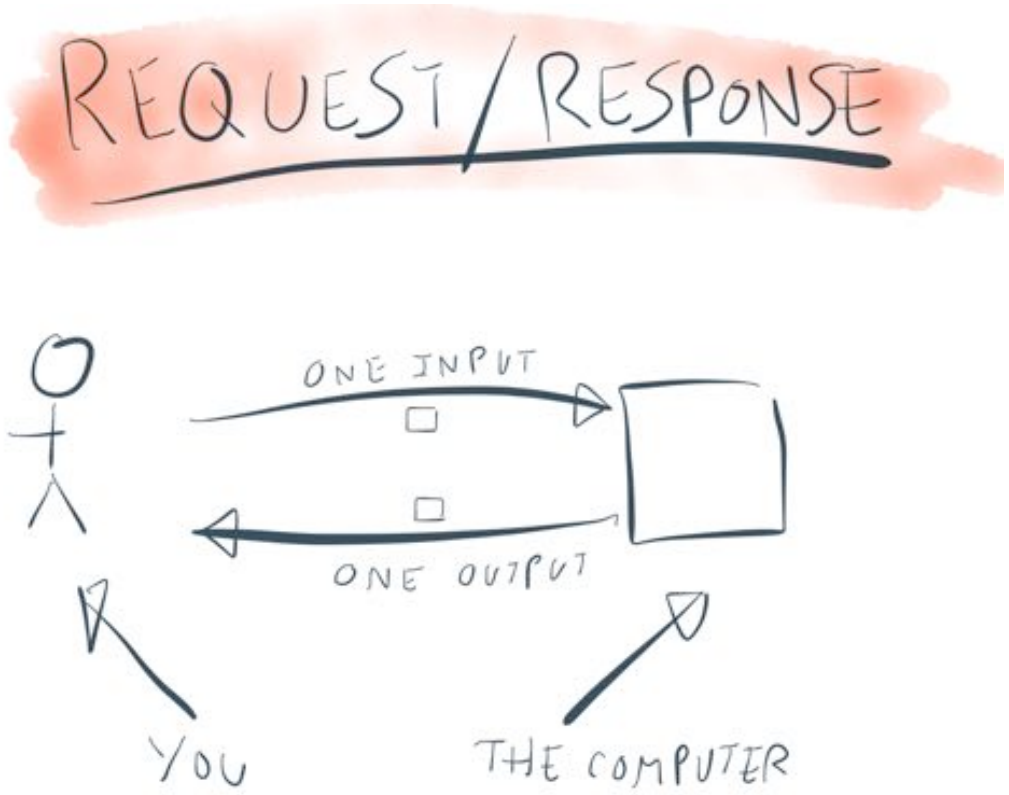


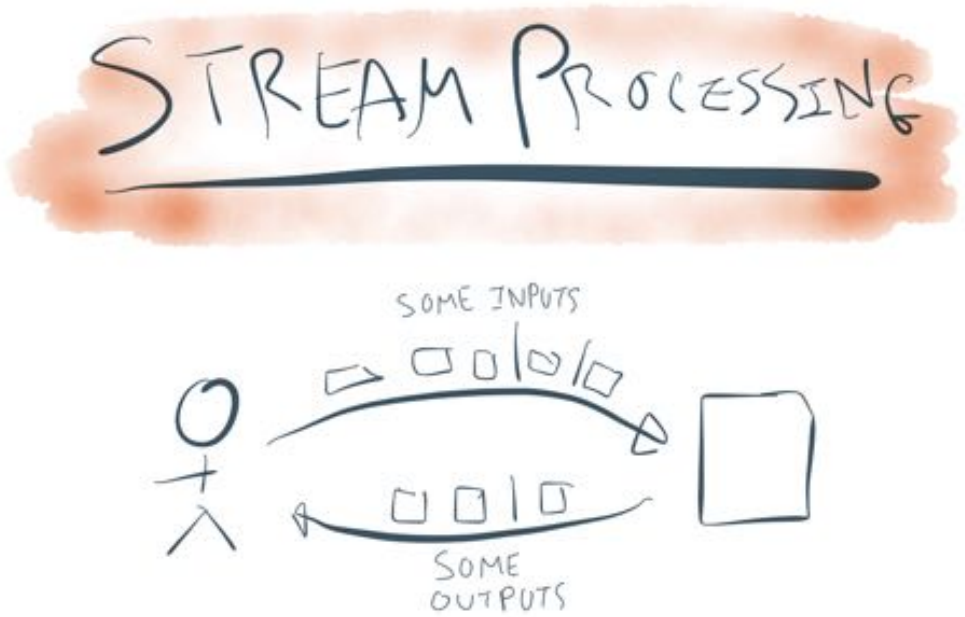
Kafka Streams
KSQL



"Kafka benefits under the hood"
Continuous Stream Processing
Reuse Preprocessing Logic from Ingestion Pipeline
Serving within the application (not via REST interface)
Predictions in real time

Stream Processing

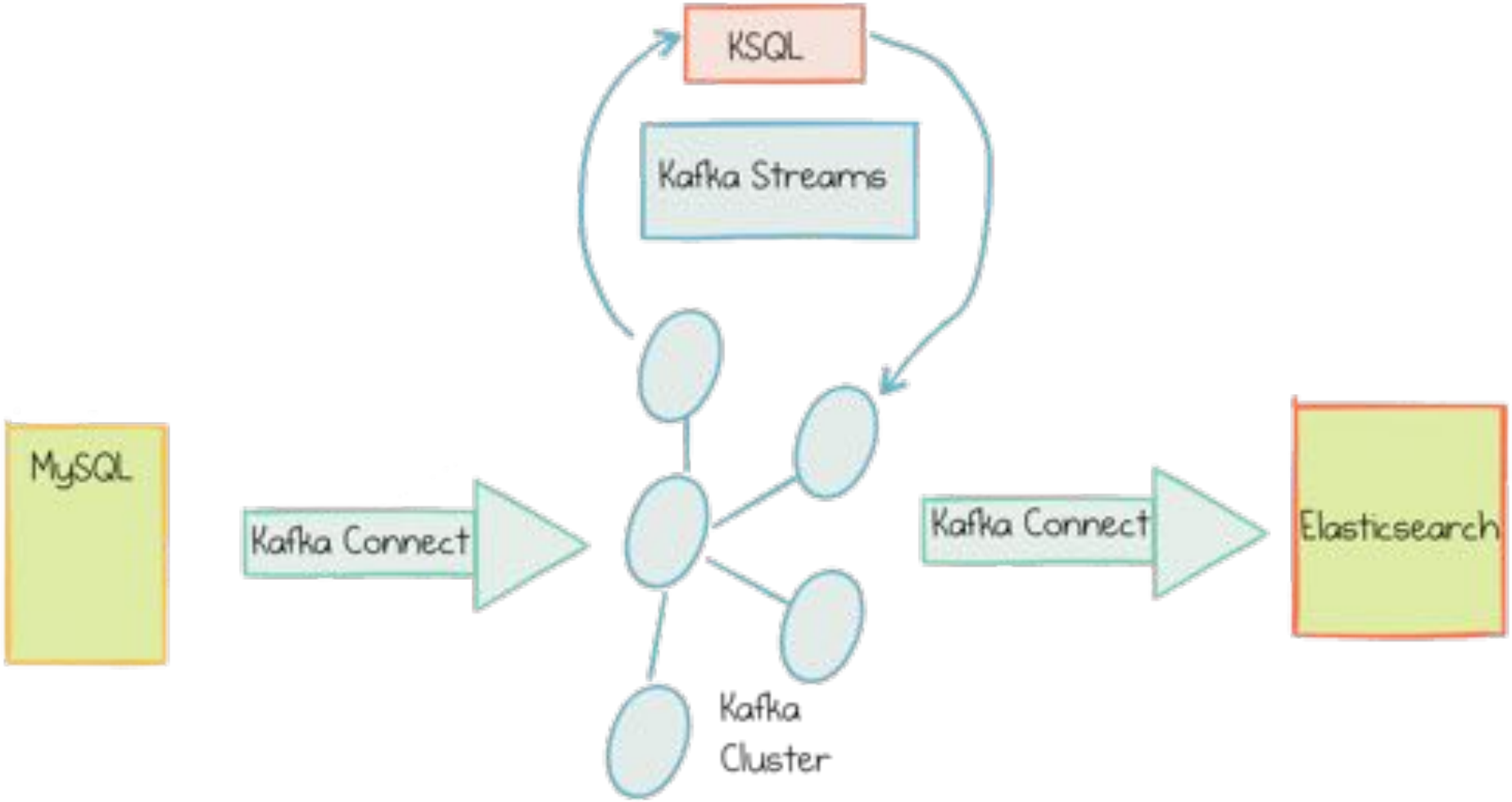




Data at Rest

Data in Motion

Kafka Streams (shipped with Apache Kafka) / KSQL (Confluent Open Source)



Stream Processing Pipeline

Applying an Analytic Model is just a piece of the puzzle!

Stream Ingest

- Messaging
- APIs
- Integration
- Adapters / Channels

Stream Preprocessing

- Normalization
- Filtering
- Transformation
- Enrichment
- Aggregation

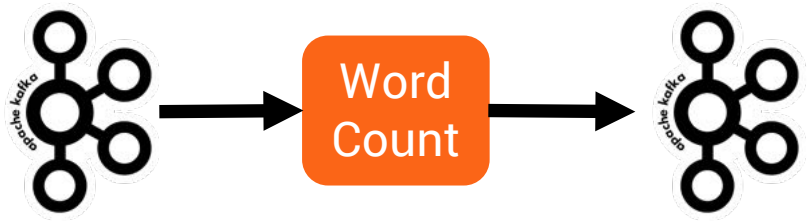
Stream Analytics

- Contextual Rules
- Windowing
- Patterns
- Analytics
- Machine Learning
- …

Stream Outcomes

- Index / Search
- Analytics / DW Reporting
- Process Management
- Applications & APIs
- Analytics (Real Time)

A complete streaming microservices, ready for production at large-scale




```
public static void main(final String[] args) throws Exception {
  Properties config = new Properties();
  config.put(StreamsConfig.APPLICATION_ID_CONFIG, "wordcount-example");
  config.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "kafka-broker1:9092");
  config.put(StreamsConfig.KEY_SERDE_CLASS_CONFIG, Serdes.String().getClass().getName());
  config.put(StreamsConfig.VALUE_SERDE_CLASS_CONFIG, Serdes.String().getClass().getName());

  KStreamBuilder builder = new KStreamBuilder();
  KStream<String, String> textLines = builder.stream("TextLinesTopic");
  KStream<String, Long> wordCounts = textLines
    .flatMapValues(value -> Arrays.asList(value.toLowerCase().split("\\W+")))
    .groupBy((key, word) -> word)
    .count("Counts")
    .toStream();
  wordCounts.to(Serdes.String(), Serdes.Long(), "WordsWithCountsTopic");

  KafkaStreams streams = new KafkaStreams(builder, config);
  streams.start();
}
```

App configuration

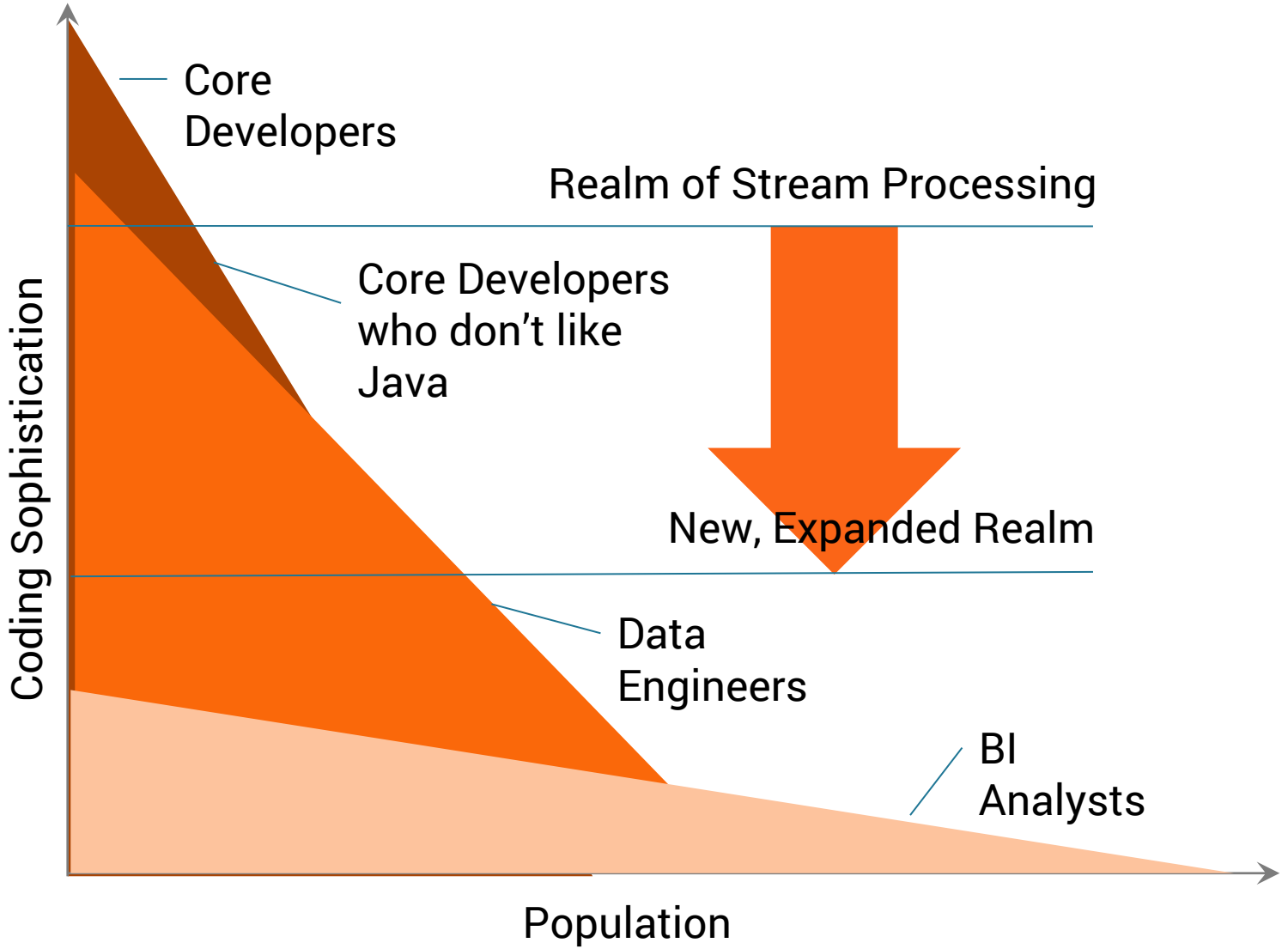
Define processing
(here: WordCount)

Start processing

Why KSQL?

Kafka Streams

KSQL



Trade-Offs

Kafka / Consumer Producer

- subscribe()
- poll()
- send()
- flush()

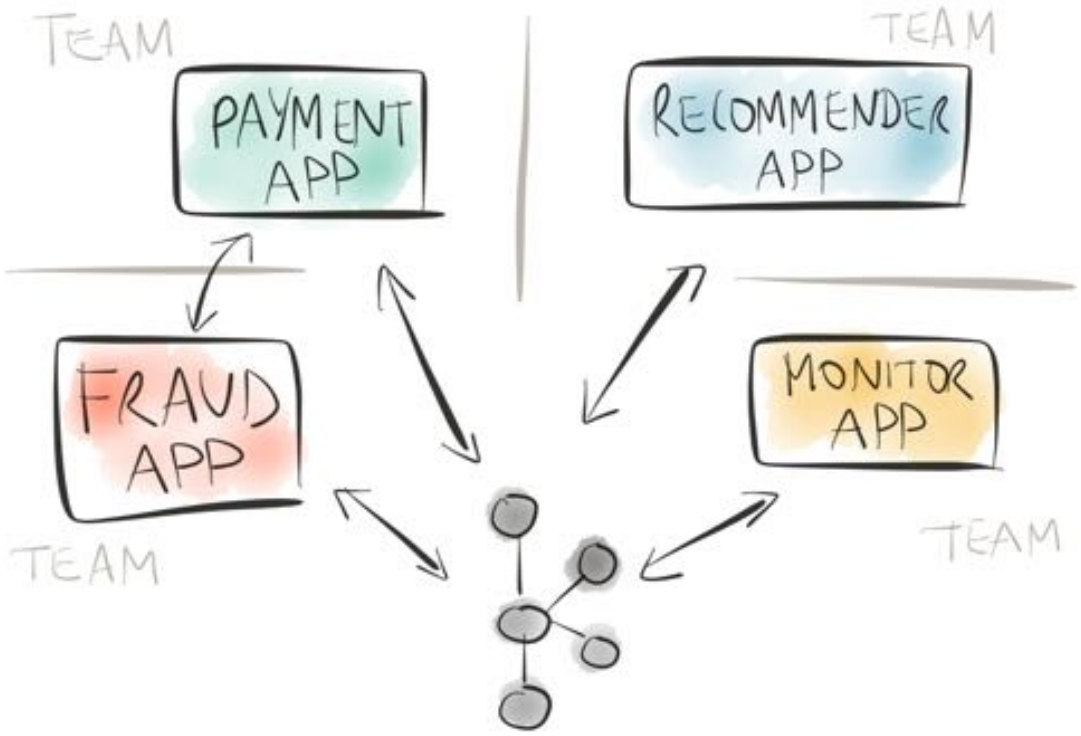
Kafka Streams

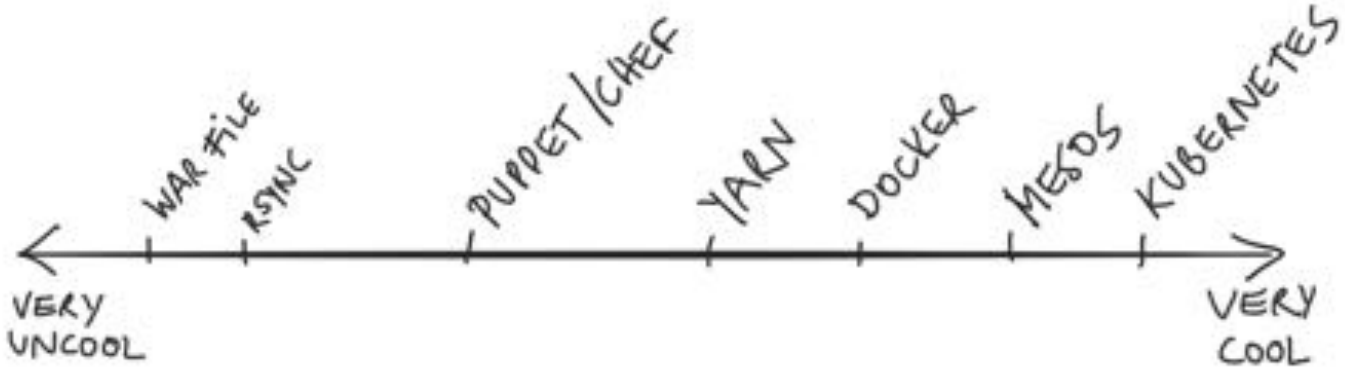
- mapValues()
- filter()
- punctuate()

KSQL

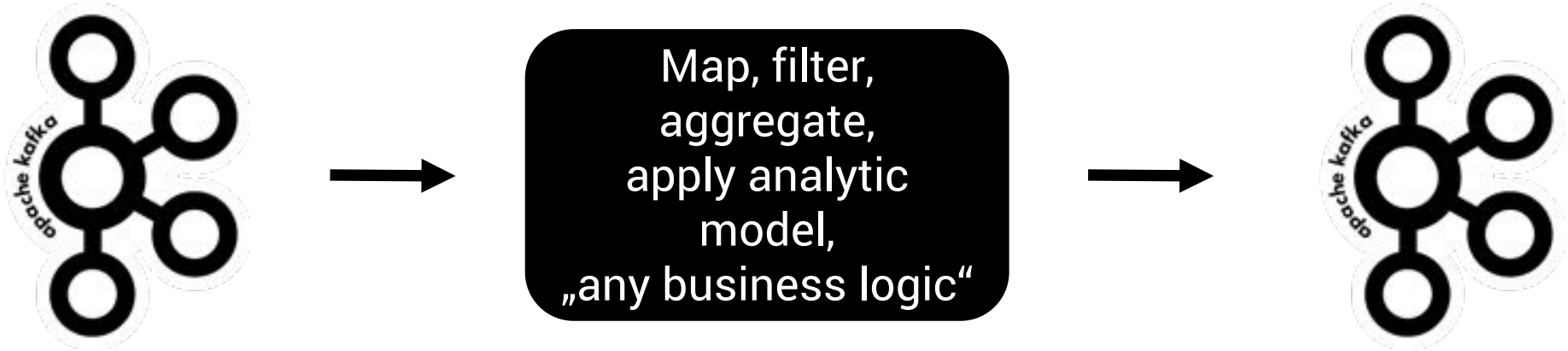
- Select…from…
- Join…where…
- Group by..

When to use Kafka Streams or KSQL for Stream Processing?





Kafka Streams (shipped with Apache Kafka) / KSQL (Confluent Open Source)



Input Stream
(Kafka Topic)

Stream Processing Microservice
(Kafka Streams / KSQL)

Output Stream
(Kafka Topic)

Kafka Cluster

Deployed Anywhere

Java App, Docker,
Kubernetes, Mesos,
“you-name-it”

Kafka Cluster

Kafka Streams and KSQL

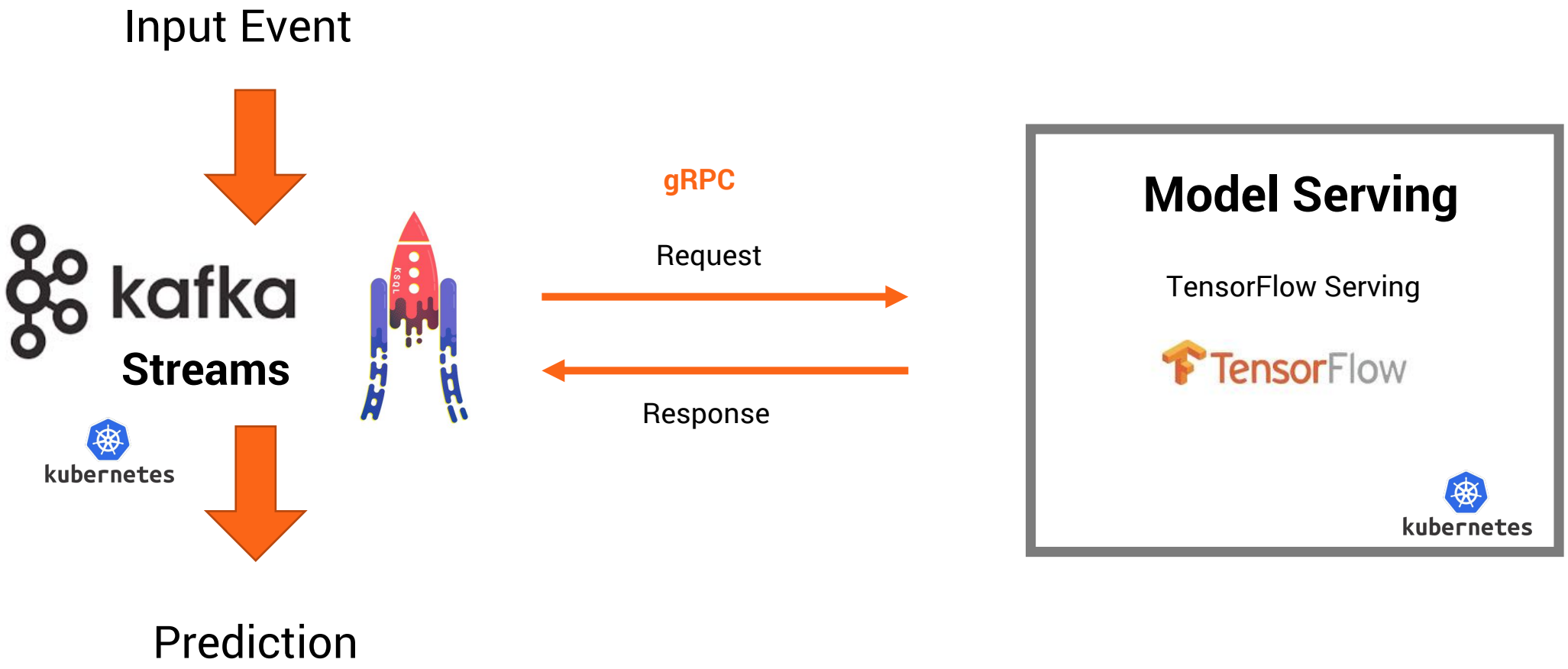
are viable for S / M / L / XL / XXL use cases

Ok.

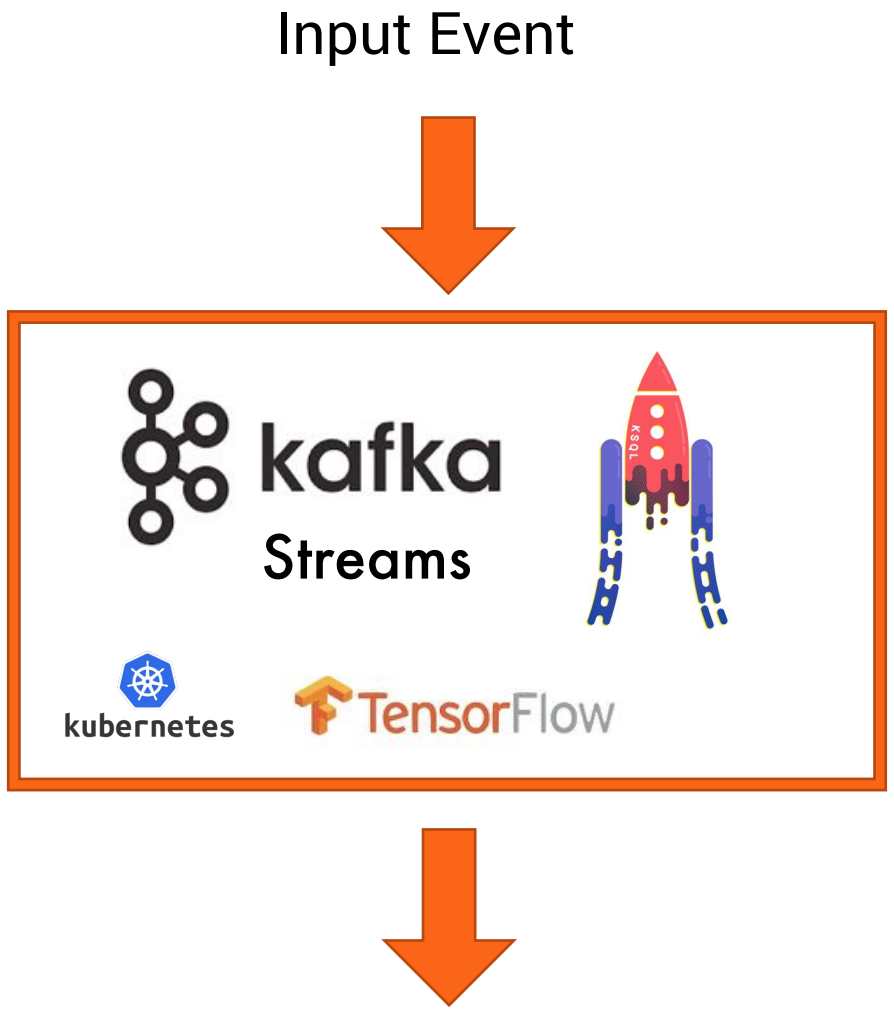
Ok.

Ok.

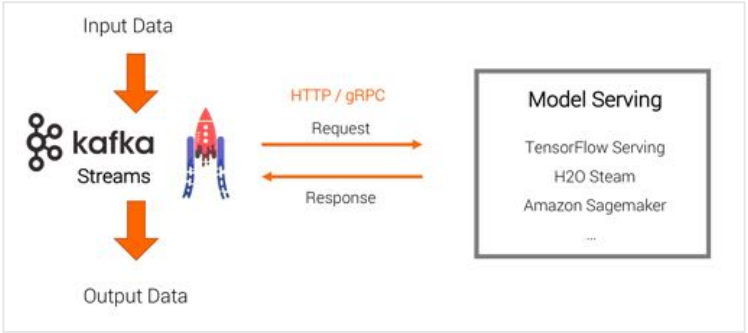
Option 1: gRPC communication to do model inference



Option 2: Model interference natively integrated into the App



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)

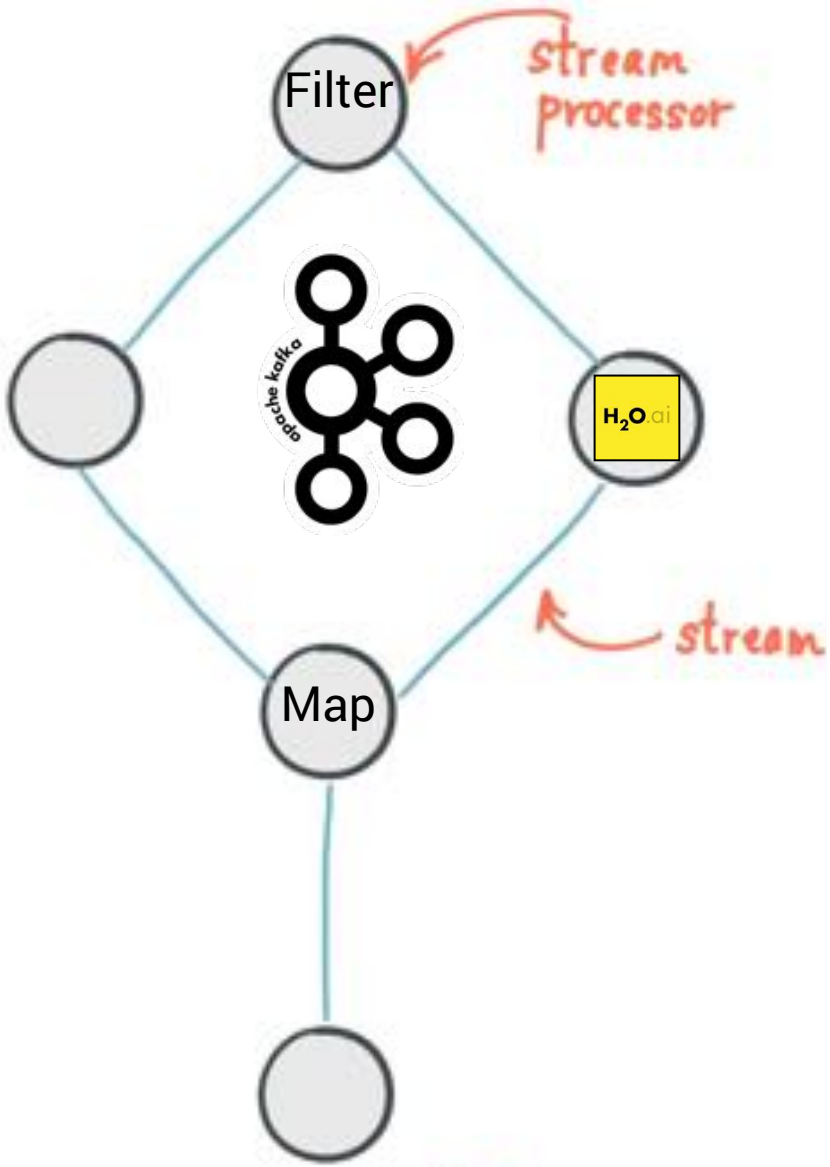
Live Demo – Deployment of a Trained Model

Use Case:
Airline Flight Delay Prediction

Machine Learning Algorithm:
Neural Network
built with H2O and TensorFlow

Streaming Platform:
Apache Kafka and Kafka Streams

H2O.ai Model + Kafka Streams



1) Create H2O DL model

```
// Create H2O object (see deeplearning_fe7c1f02_08ec_4070_b784_c2531147e451.java)
hex.genmodel.GenModel rawModel;
rawModel = (hex.genmodel.GenModel) Class.forName(modelClassName).newInstance();
EasyPredictModelWrapper model = new EasyPredictModelWrapper(rawModel);
```

2) Configure Kafka Streams Application

```
// Configure Kafka Streams Application
final String bootstrapServers = args.length > 0 ? args[0] : "localhost:9092";
final Properties streamsConfiguration = new Properties();
// Give the Streams application a unique name. The name must be unique
// in the Kafka cluster
// against which the application is run.
streamsConfiguration.put(StreamsConfig.APPLICATION_ID_CONFIG, "machine-learning-example");
// Where to find Kafka broker(s).
streamsConfiguration.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
```

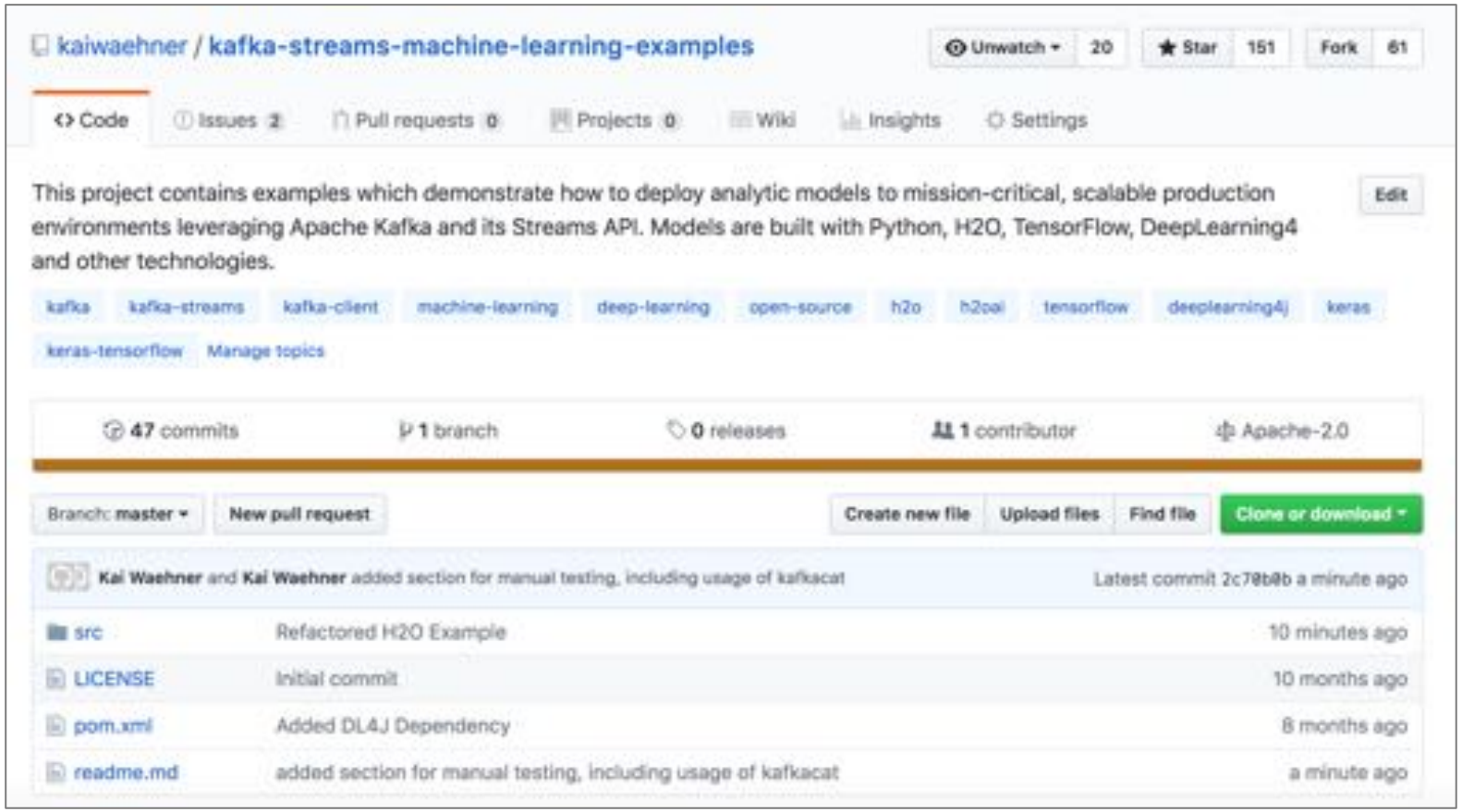
3) Apply H2O DL model to Streaming Data

```
airlineInputLines.foreach(new ForeachAction<String, String>() {
    public void apply(String key, String value) {

        // Year,Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,Actu
        // value:
        // 1987,10,14,3,741,730,912,849,PS,1451,NA,91,79,NA,23,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,YES,YES

        if (value != null && !value.equals("")) {
            System.out.println("####################");
            System.out.println("Flight Input:" + value);

            String[] valuesArray = value.split(",");

            RowData row = new RowData();
            row.put("Year", valuesArray[0]);
            row.put("Month", valuesArray[1]);
            row.put("DayofMonth", valuesArray[2]);
            row.put("DayOfWeek", valuesArray[3]);
            row.put("CRSDepTime", valuesArray[5]);
            row.put("UniqueCarrier", valuesArray[8]);
            row.put("Origin", valuesArray[16]);
            row.put("Dest", valuesArray[17]);
            BinomialModelPrediction p = null;
            try {
                p = model.predictBinomial(row);
            } catch (PredictException e) {
                e.printStackTrace();
            }

            System.out.println("Label (aka prediction) is flight departure delayed: " + p.label);
            System.out.print("Class probabilities: ");
            for (int i = 0; i < p.classProbabilities.length; i++) {
                if (i > 0) {
                    System.out.print(",");
                }
                System.out.print(p.classProbabilities[i]);
            }
```

4) Start Kafka Streams App

```
// Start Kafka Streams Application to process new incoming messages from Input Topic
final KafkaStreams streams = new KafkaStreams(builder, streamsConfiguration);
streams.cleanUp();
streams.start();
```

Github Examples: Kafka + Machine Learning



kafka

TensorFlow

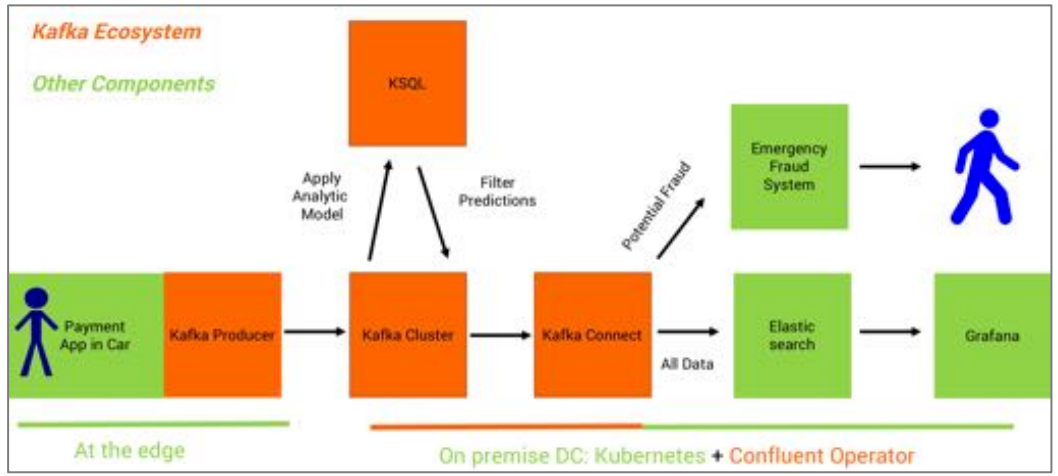
H2O.ai

DL4J
DEEPLEARNING4J

https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/
https://github.com/kaiwaehner/kafka-streams-machine-learning-examples

1) git clone → 2) mvn clean package → 3) look at implementations and unit tests

KSQL and Deep Learning (Autoencoder) for Fraud Detection



"CREATE STREAM FraudDetection AS
SELECT payment_id, applyFraudModel(payment_input)
FROM payment_engine;"

User Defined Function (UDF)

H2O.ai TensorFlow

https://www.confluent.io/blog/build-udf-udaf-ksql-5-0
https://github.com/kaiwaehner/ksql-udf-deep-learning-mqtt-iot

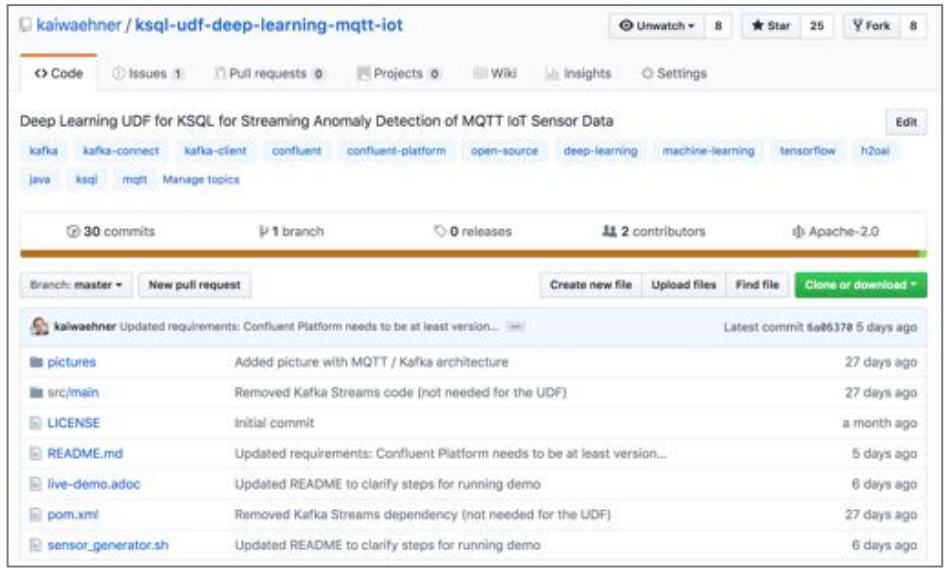
Live Demo – Prebuilt Model Embedded in KSQL Function

Use Case:
Anomaly Detection
(Payment Fraud Detection)

Machine Learning Algorithm:
Autoencoder built with H2O

Streaming Platform:
Apache Kafka and KSQL

Github Examples: KSQL + Deep Learning



kafka

+ Kafka Connect
+ Elasticsearch

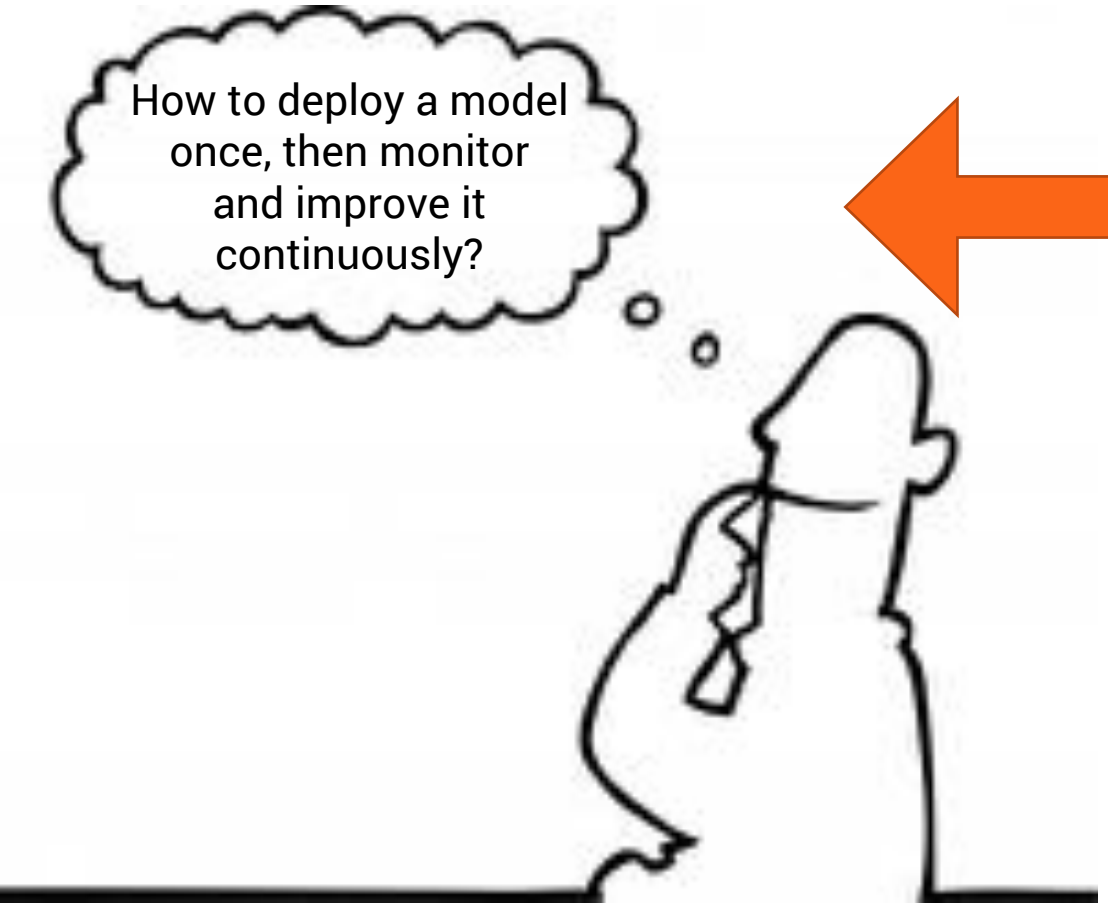
H2O.ai

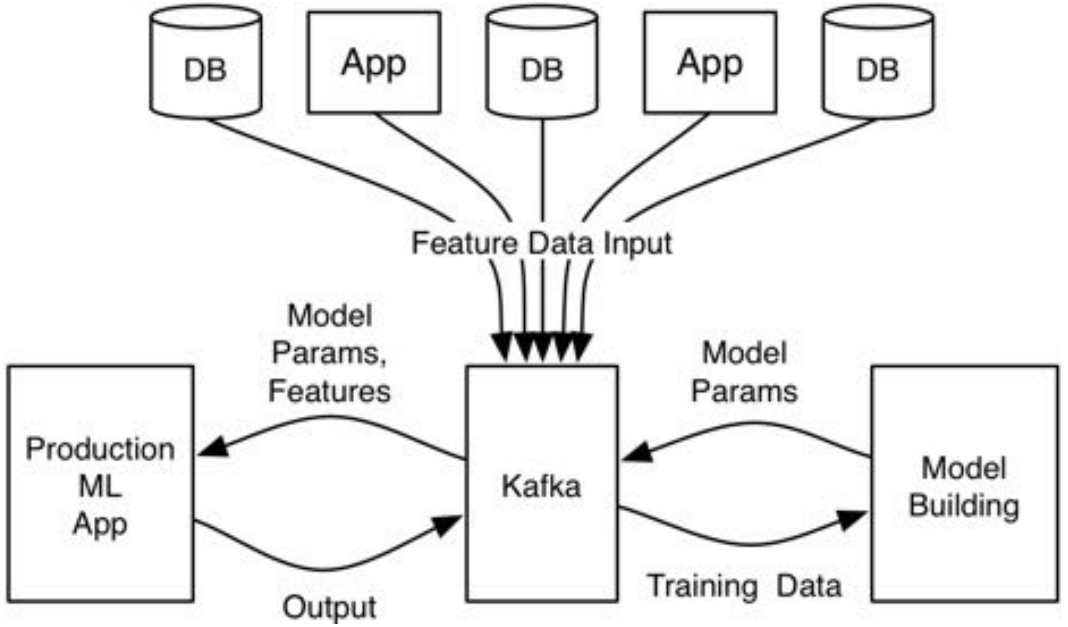
https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/
https://github.com/kaiwaehner/ksql-udf-deep-learning-mqtt-iot
https://github.com/kaiwaehner/ksql-fork-with-deep-learning-function

Agenda

1) Added Business Value via Machine Learning
2) Apache Kafka Ecosystem as Infrastructure for Machine Learning
3) Data Ingestion and Preprocessing with Apache Kafka for Model Training
4) Predictions in Real Time with Kafka Streams and KSQL
5) **DevOps and Monitoring of a Machine Learning Infrastructure**

Automated Model Improvement with Apache Kafka and Kafka Streams



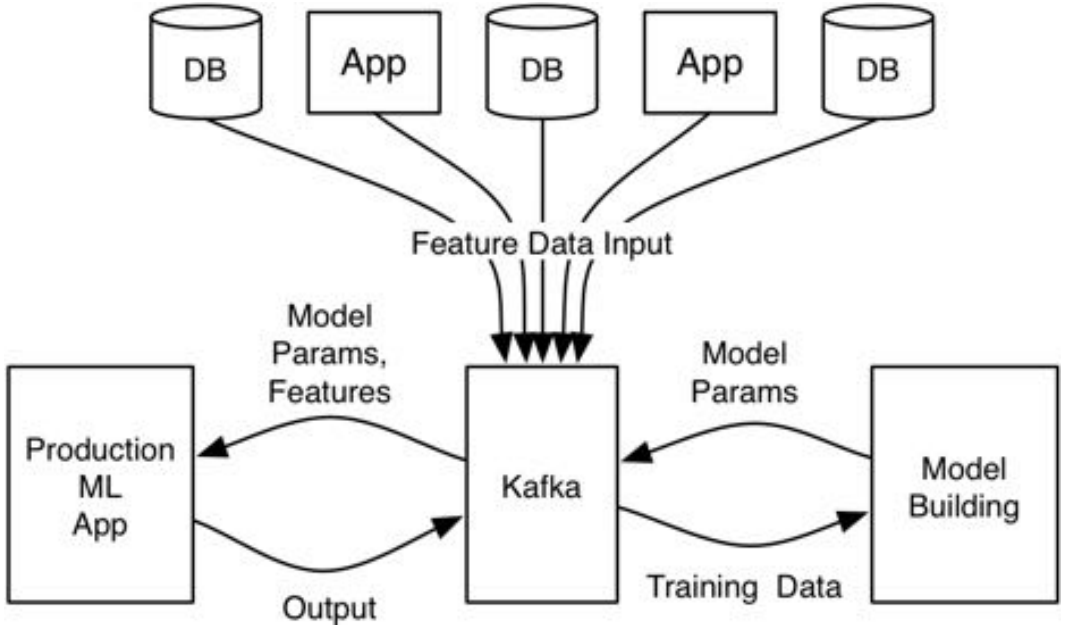


Automated Model Improvement with Apache Kafka and Kafka Streams

How to improve models?

1. Manual Update
2. Continuous Batch Updating
3. Real Time → Online Model Training

Your choice… All possible with Kafka!



Caveats for Online Model Training

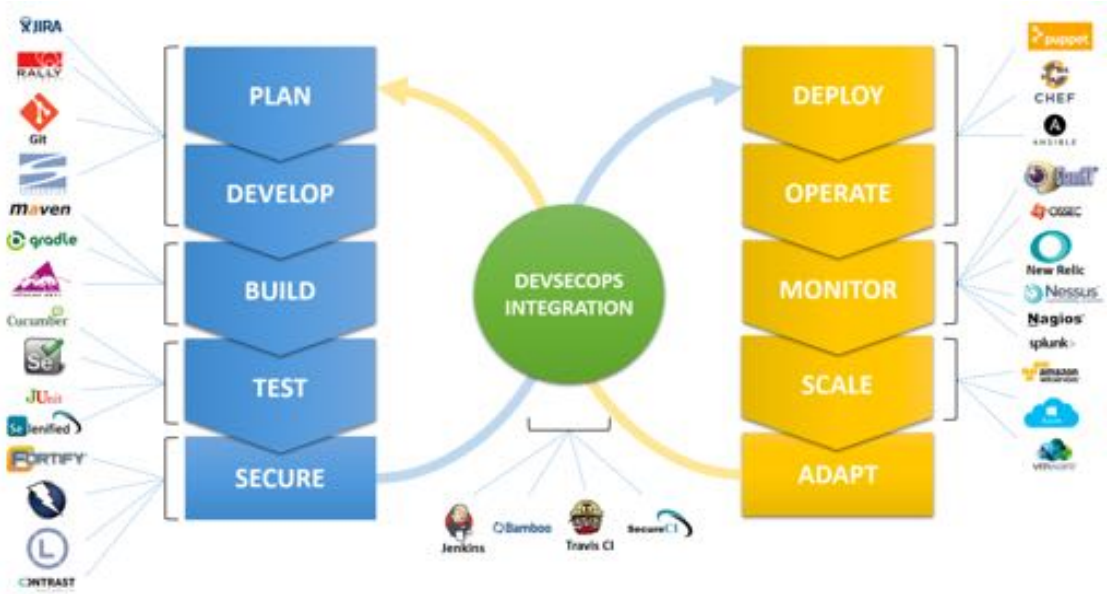
- Processes and infrastructure not ready
- Validation needed before production
- Slows down the system
- Only a few ML implementations → Build your own!
- Only possible for unsupervised ML (e.g. clustering)
- Many use cases do not need it

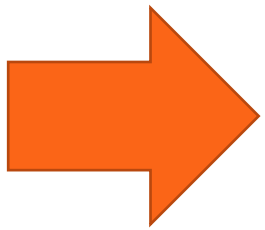
→ Do it only when feasible!

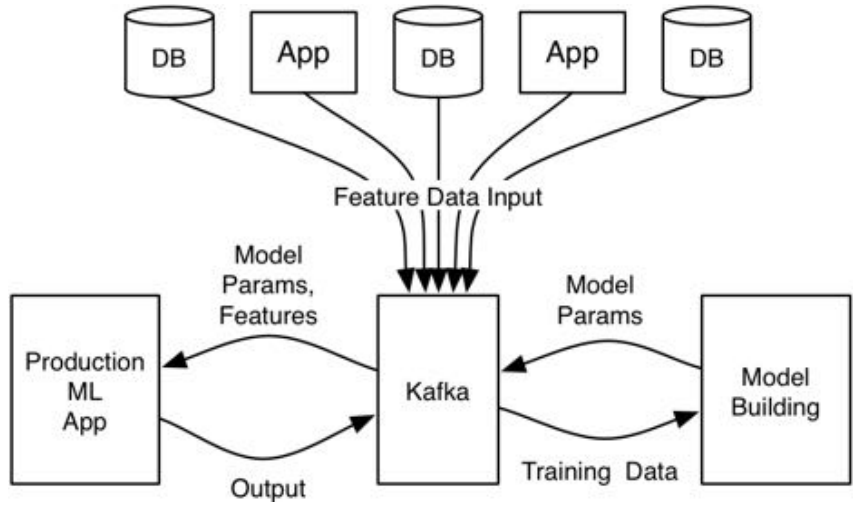
Continuous Batch Updating as "Best Feasible Option"

DevOps Pipeline

1. Apply the model online to make predictions
2. Collect data and train a new model
3. Automated Re-Deployment (e.g. via a Kafka Topic)







https://www.confluent.io/blog/predicting-flight-arrivals-with-the-apache-kafka-streams-api/
https://www.coveros.com/services/devops/

Kubernetes – The Winner of the Container and DevOps Wars!

kubernetes

Google Cloud Platform

Amazon EKS

RED HAT OPENSHIFT

Microsoft Azure

CLOUDFOUNDRY

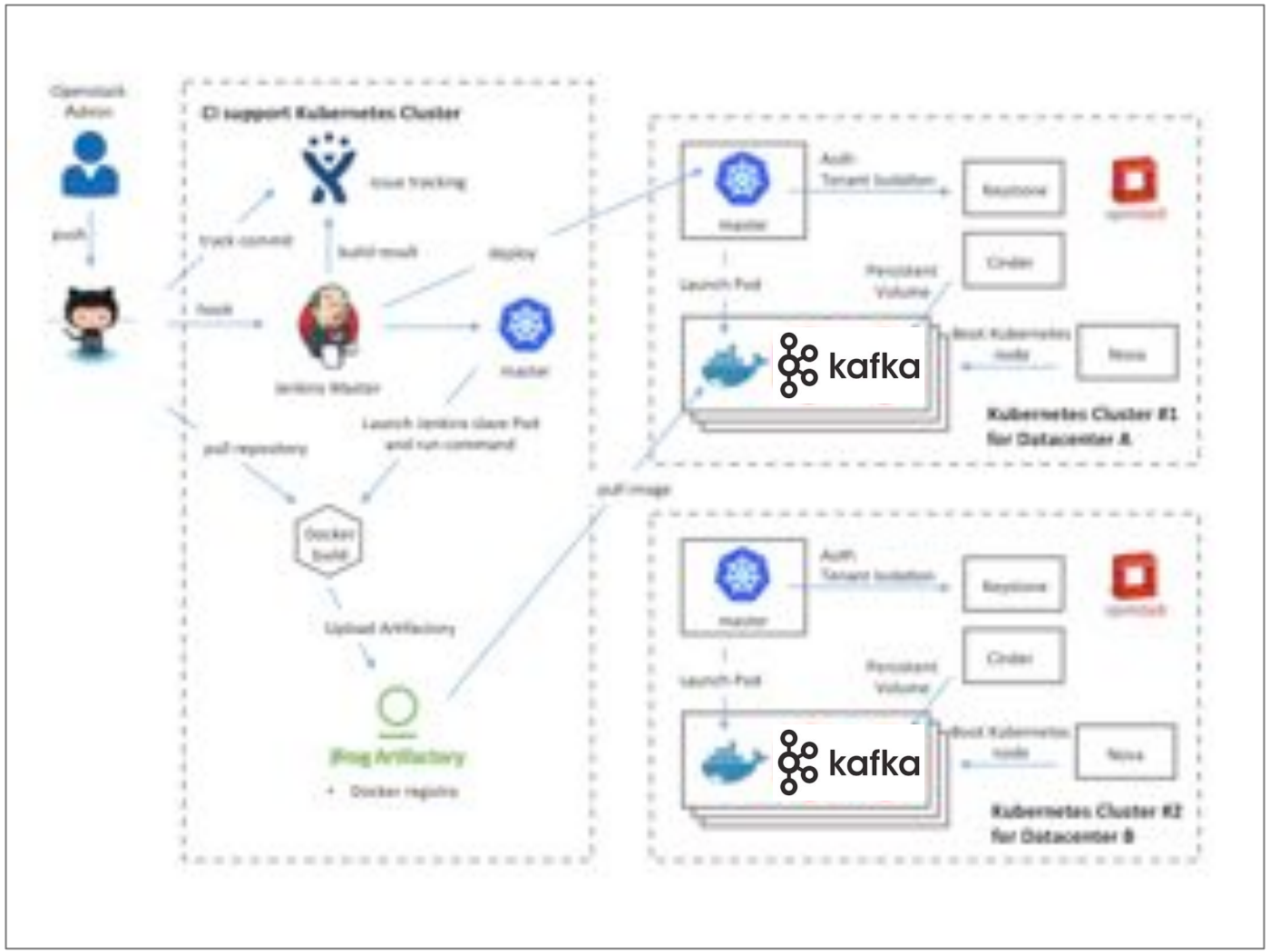
docker

Docker, Inc

MESOSPHERE

https://www.infoworld.com/article/3118345/cloud-computing/why-kubernetes-is-winning-the-container-war.html
http://techgenix.com/year-of-kubernetes/

Kubernetes for Infrastructure Deployment



https://kubernetes.io/blog/2016/10/kubernetes-and-openstack-at-yahoo-japan

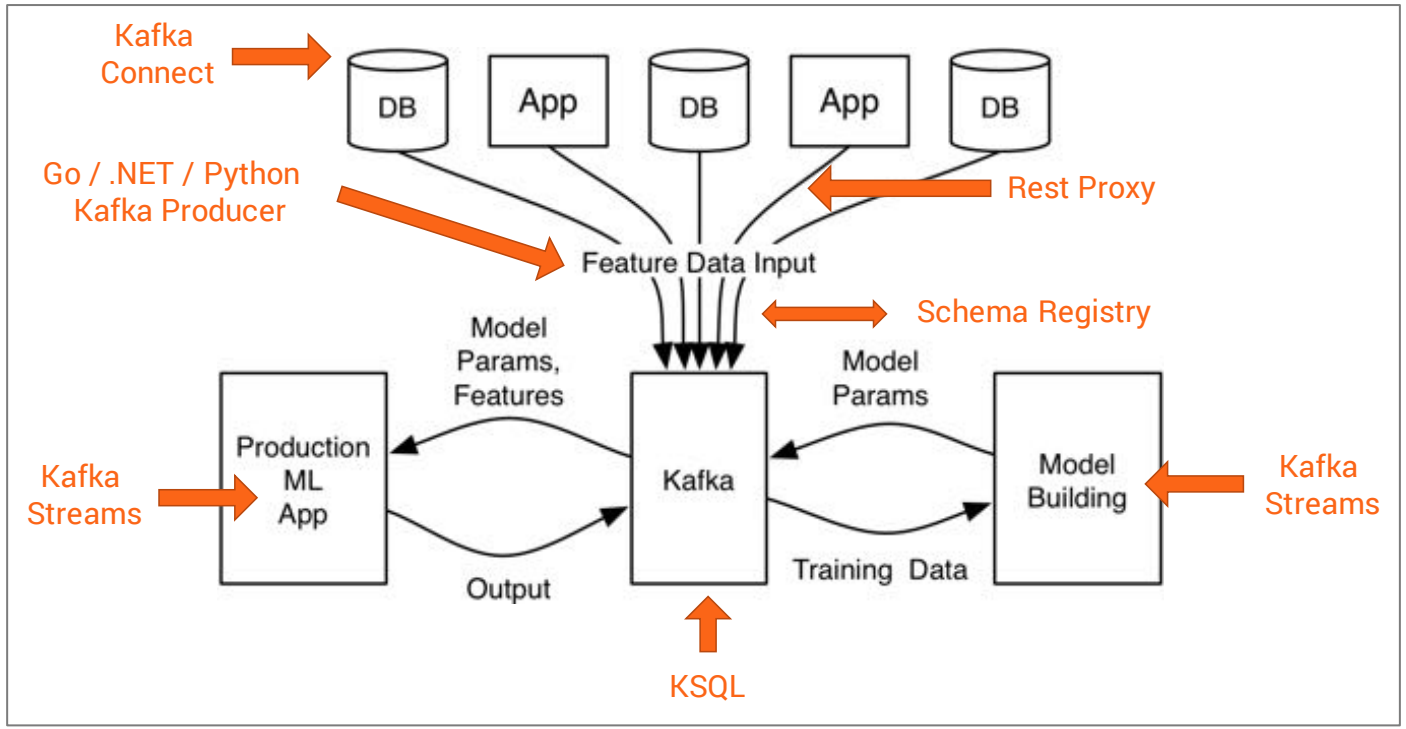
Stateful Backend

- Zookeeper and Kafka Broker Pods
- REST Proxy, Schema Registry
- Persistent Volumes
- Kubernetes Operator

Stateless and Stateful Clients

- Java / .Net / Go / Python Kafka Clients
- Kafka Streams / KSQL Apps
- Scalability and Elasticity

Monitoring the Infrastructure for Machine Learning



Grafana

Prometheus

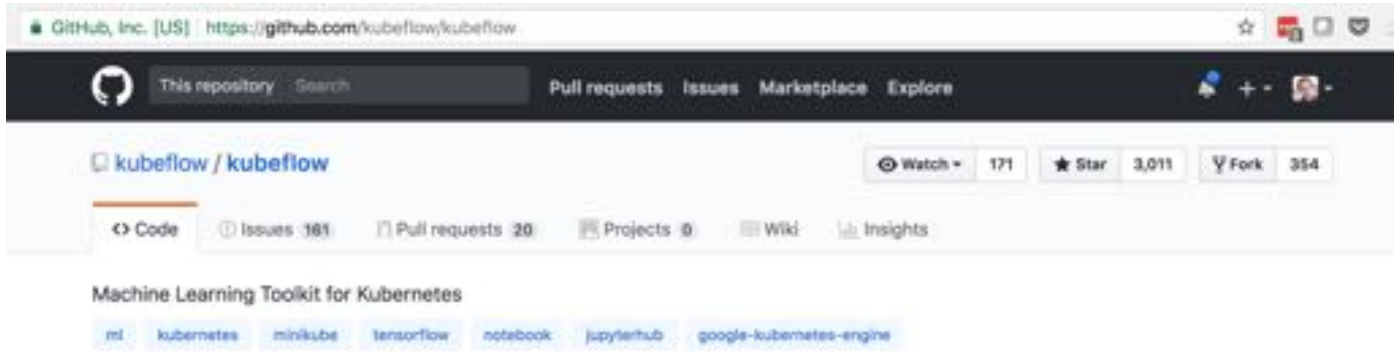
New Relic

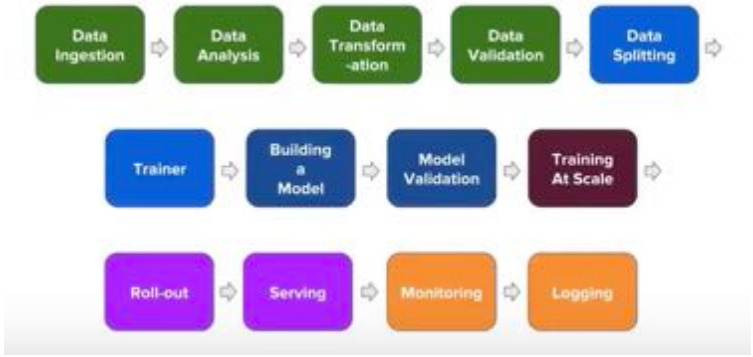
confluent Control Center

Java

Build vs. Buy
Hosted vs. Managed
Basic vs. Advanced

Kubernetes Deployment of ML Workflows





Kubeflow

The Kubeflow project is dedicated to making **deployments** of machine learning (ML) workflows on Kubernetes simple, portable and scalable. Our goal is **not** to recreate other services, but to provide a straightforward way to deploy best-of-breed open-source systems **for ML** to diverse infrastructures. Anywhere you are running Kubernetes, you should be able to run Kubeflow.

Warning:

Early Stage with focus on TensorFlow Training, TensorFlow Serving, Jupyter…
Bigger ecosystem expected soon… Including Kafka components for ingestion, serving, monitoring…

https://github.com/kubeflow/kubeflow

Key Take-Aways

→ Data Scientist and Developers have to work together continuously (org + tech!)

→ Mission critical, scalable production infrastructure is key for success of Machine Learning projects

→ Apache Kafka Ecosystem + Cloud = Machine Learning at Extreme Scale
(Ingestion, Processing, Training, Inference, Monitoring)

Questions? Feedback? Please contact me!

Kai Waehner

Technology Evangelist

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