

WOTD

World Of Tech
2017年12月1-2日

全球软件开发技术峰会

[深圳站]

报名咨询：010-68478816

议题提交：wot@51cto.com

市场合作：yangxh@51cto.com

商务合作：songjc@51cto.com

媒体合作：yankk@51cto.com

在线咨询（微信）：18401576051

团·购·享·受·更·多·优·惠

5折

优惠（截止8月31日）

现在报名，立省1400元 / 张

Agenda

- How ML works
- Uber's ML platform
- Case Study: UberEATS in ML
- Key Challenges: Deep dive
- System Architectures

The Life of a Model

- Data
 - Data sources, batch or streaming, data aggregation...
- Training in various environments
 - Fast iteration, traditional ML vs DL, training environments ...
- Model evaluations
 - Standard evaluation vs. customized eval
- Model deployment
 - Versioning, production
- Inference
 - Batch predictions vs online predictions, scaling out, SLA ...
- Monitoring
 - Signal selection





ML PLATFORM MISSION

Enable engineers and data scientists across the company to easily build and deploy machine learning solutions at scale

Uber ML Platform

- ML as a Service
- Scalable infrastructure for training and serving
- Workflow tools for prototyping, iteration, and productionization
- Model and data serving with full monitoring for batch and realtime
- Scope
 - Traditional ML & Deep Learning
 - Supervised, Unsupervised and Semi-supervised
 - Online learning



Having trouble? →

> GO TO OLD UI



← BACK TO PROJECTS

EATS_ETD_Prediction

OWNER

ntl, alexnik, nico, myz

TEAM

eats_core

DOCUMENTATION

+ Add

TIER

3

ROLLOUT



Data Sources



Models



Deployments



Predictions

DATA SOURCES

+ ADD DATA SOURCE



Dynamic Data Source

SETTINGS



Type:

DYNAMIC

Source:

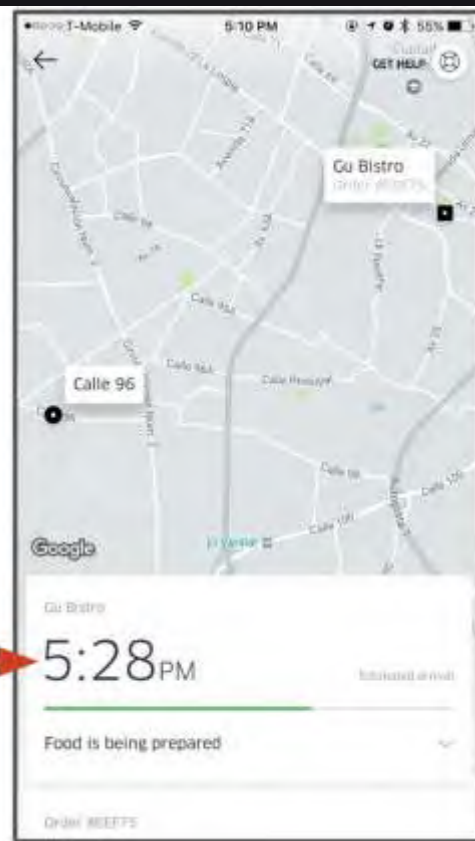
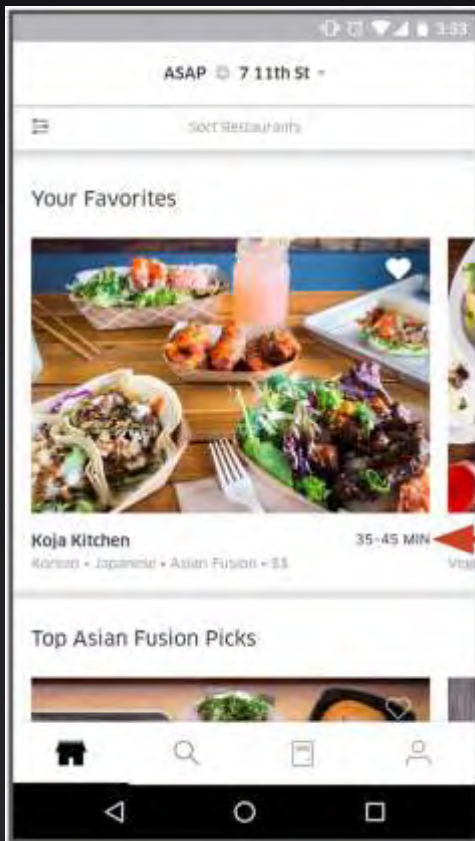
SQL Pipeline

Data Type:

spark_sql

Example Use Case: UberEATS

MEAL DELIVERY TIME



Uber EATs Delivery Time Models

- Features
 - Curated features
 - Request Level Features – *user's current location*
- Models
 - Several models for different stages of order
 - GBDT Regression
 - Different versions of each for experimentation

Key Challenges

- Guarantee same data for batch training and online scoring
- Train and deploy separate model per city
- One-click deploy & easy scale out
- Live monitoring of model performance

Challenge 1:

Same data for train & predict

Data Sources: Problems

	Request Level Features	Aggregated Feature	Near Realtime Aggregated Feature
Training	In Batch	In Batch	In Batch
Online scoring	Given by user	Curated in batch Consumed by query	Curated in streaming, consumed by query

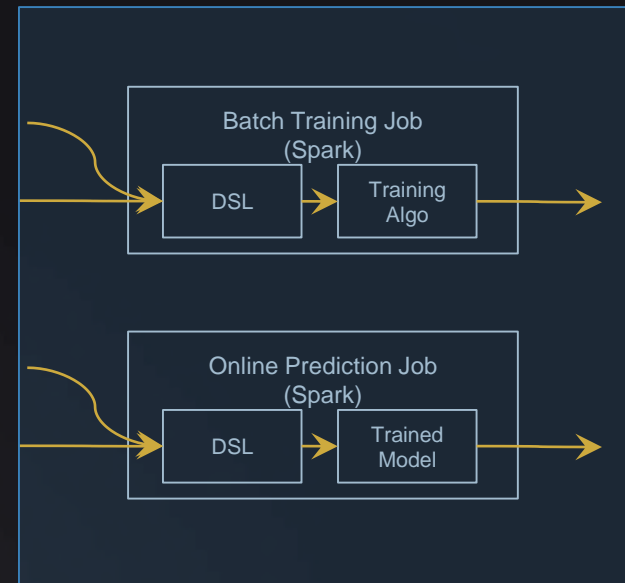
Generation Pattern: Batch and streaming
Consuming Pattern: Batch and Query

Data Sources (Solutions)

- Data Storage
 - Spark for batch jobs
 - Cassandra for online jobs
 - Streaming jobs
- Data Accessors
 - Own DSL
 - Access basis features, curated features, and column stats
- Data Transformation
 - Standard transformation functions + UDFs
- Examples

@palette:store:orders:prep_time_avg_1week:rs_uuid

nFill(@basis:distance, mean(@basis:distance))



Challenge 2:

Separate model per city

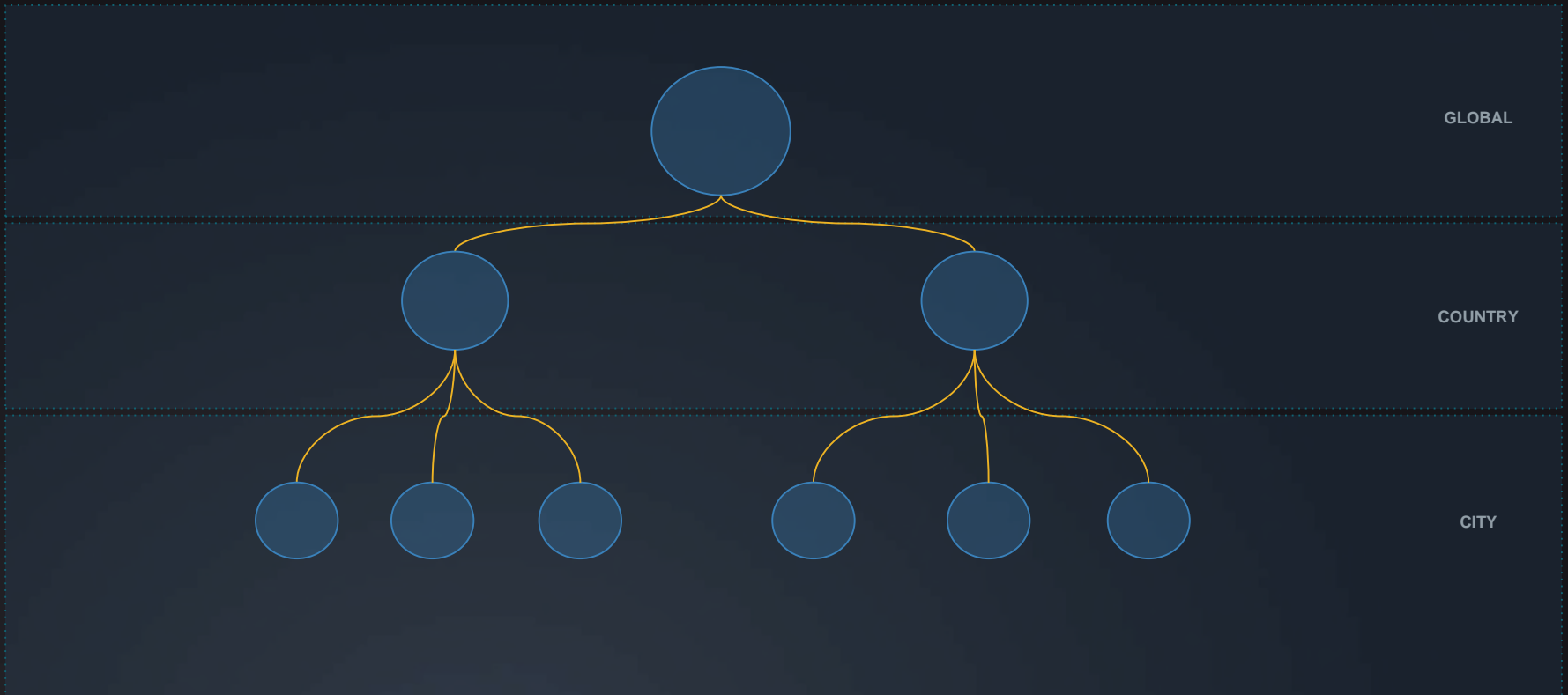
PROBLEM

- - Often you want to train a model per city
 - Hard to train and deploy 400+ individual models

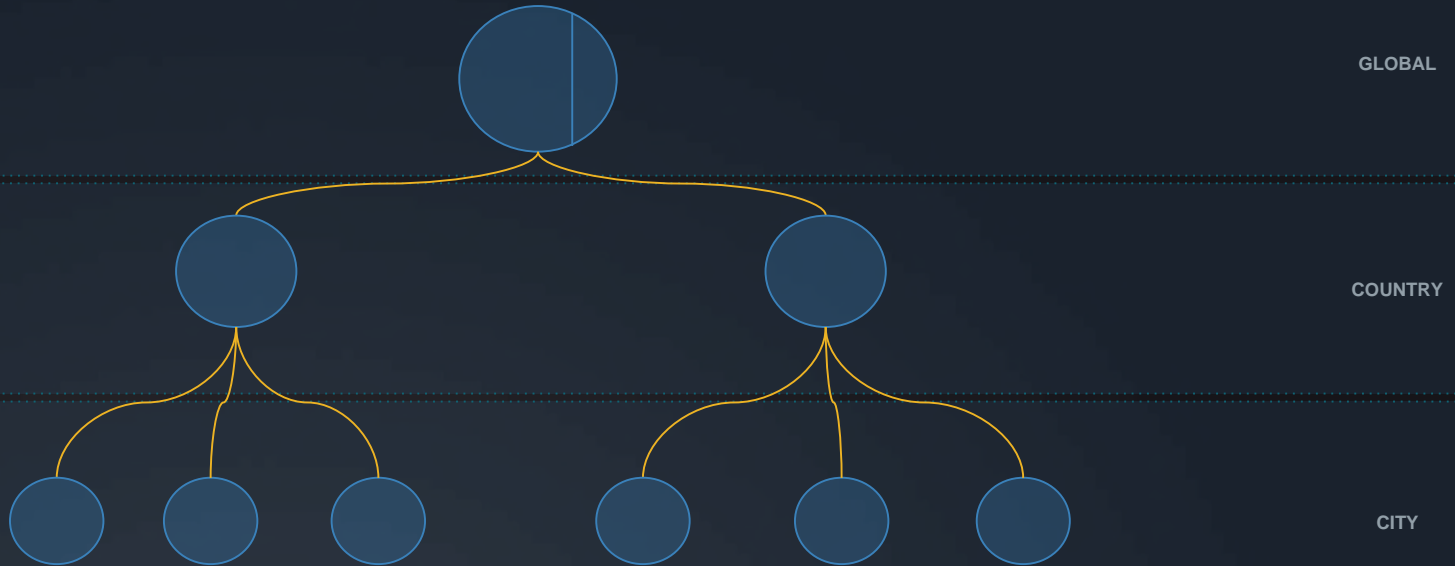
SOLUTION

- - Let users define hierarchical partitioning scheme
 - Automatically train model per partition
 - Manage and deploy as single logical model

1 Define partition scheme



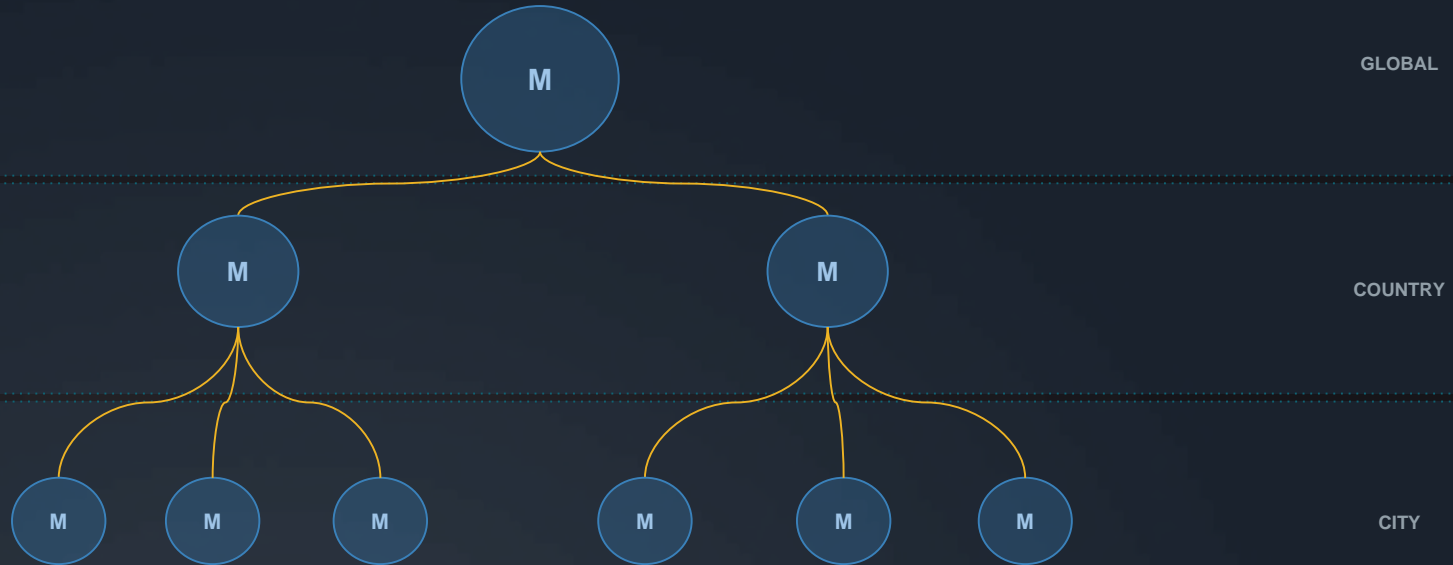
2 Make train / test split



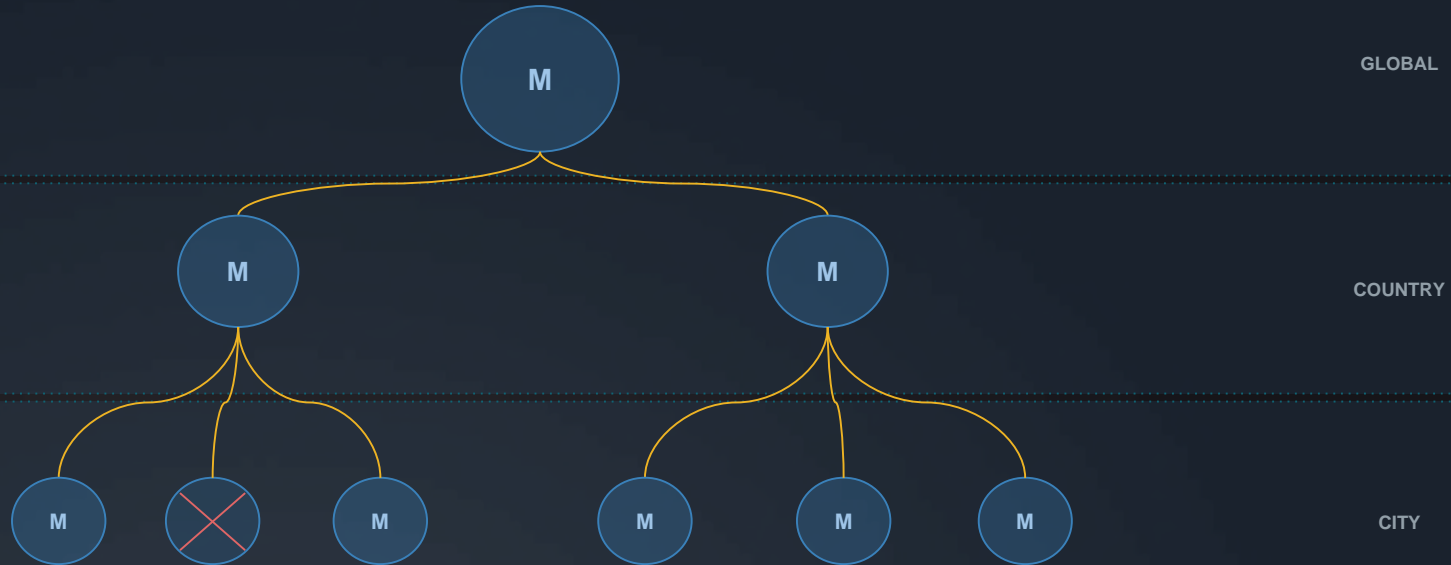
3 Keep same split and partition for each level



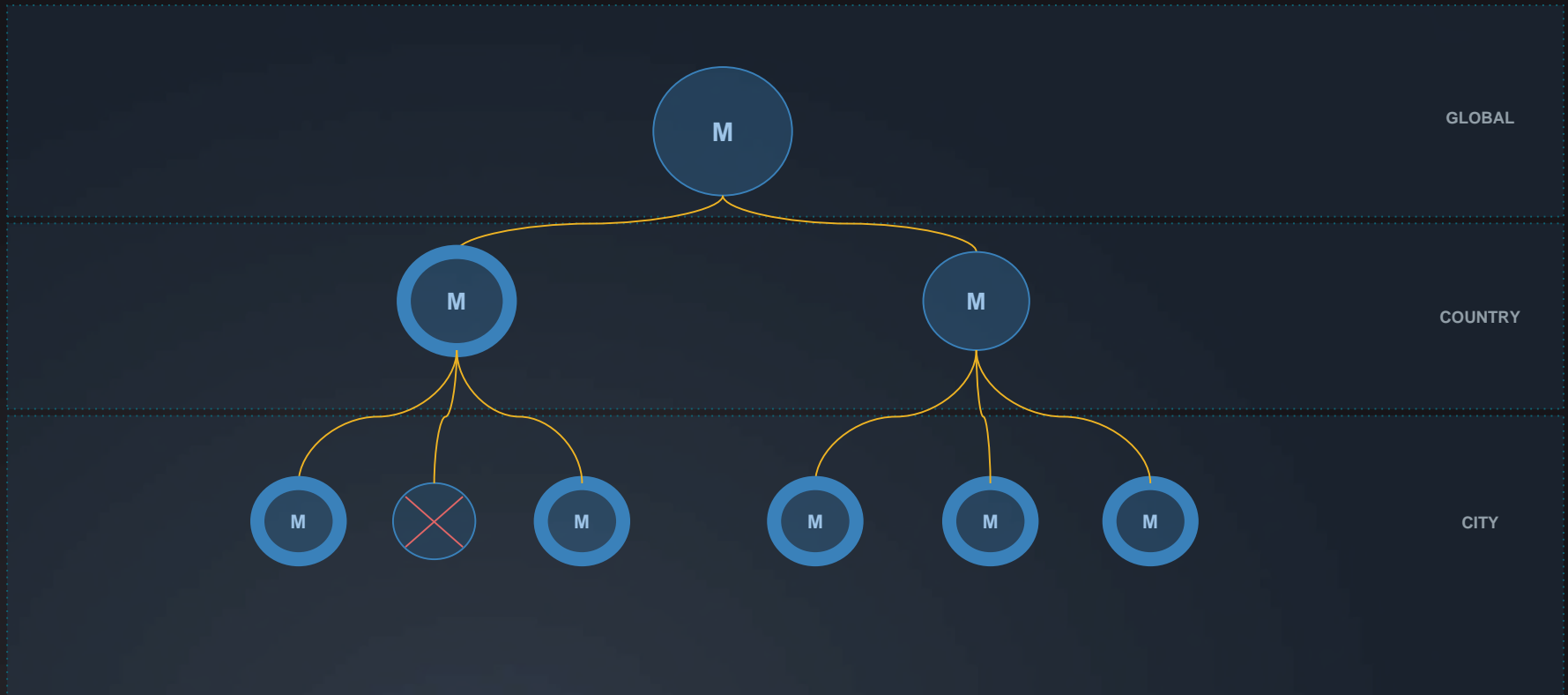
4 Train model for every node



Prune bad models



6 At serving time, route to best model for each node



Challenge 3:

One-click deploy and scale out

REALTIME PREDICT SERVICE

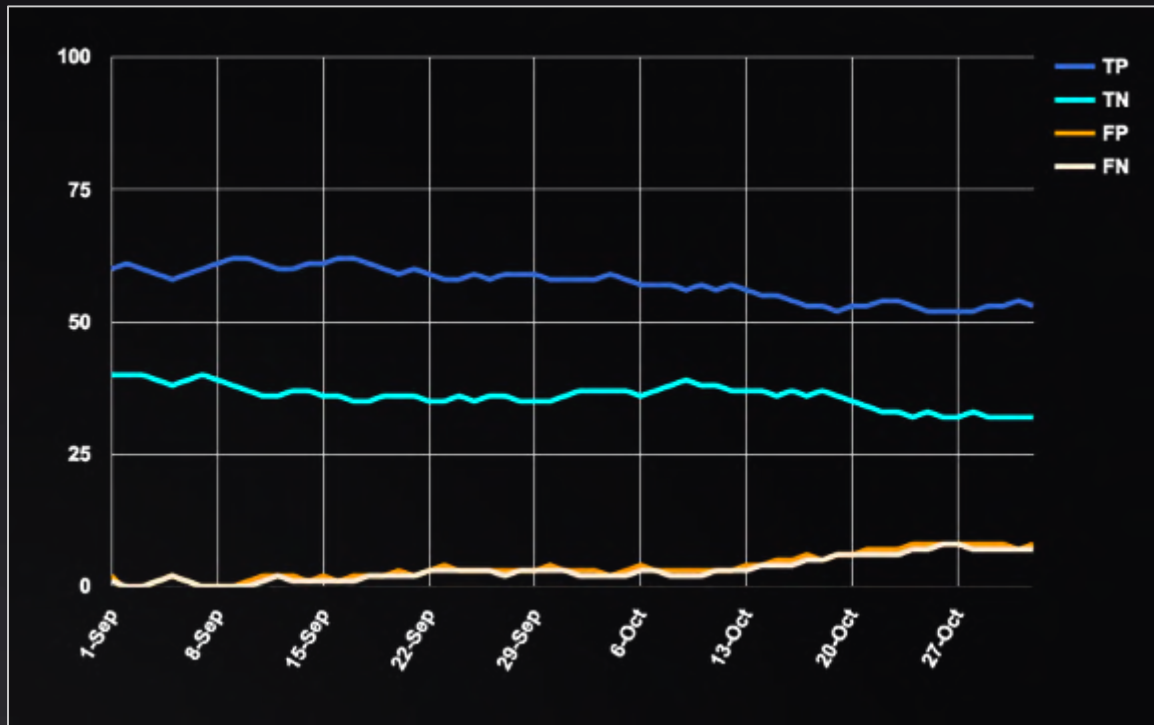
- Predict service
 - RPC service container for one or more models
 - Scale out in Docker on Mesos
 - Single- or multi-tenant deployments
 - Connection management and batched/parallelized queries to Cassandra
 - Monitoring & alerting
- Deployment
 - Each model is packed individually
 - One click deploy across DCs via standard Uber deployment infrastructure
 - Health checks and rollback

Challenge 4:

Live model performance monitoring

LIVE PREDICTION MONITORING

- Problem
 - Ensure deployed model is making good predictions
- Solution
 - Log predictions
 - Join logged predictions to actual outcomes
 - Publish metrics for monitoring and alerting
 - Optionally hold back logged predictions



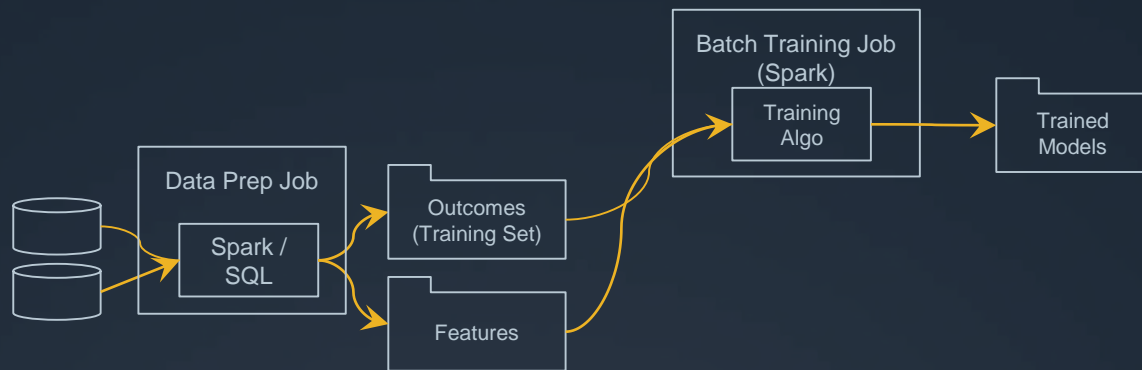
System Architecture

GET DATA

TRAIN MODELS

EVAL MODELS

DEPLOY, PREDICT & MONITOR



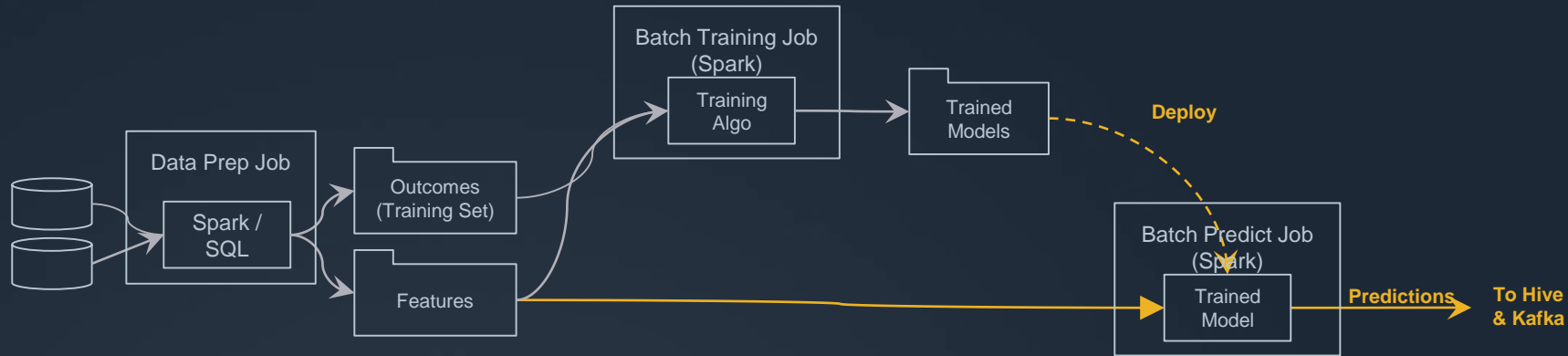
HADOOP / YARN (Batch)

GET DATA

TRAIN MODELS

EVAL MODELS

DEPLOY, PREDICT & MONITOR



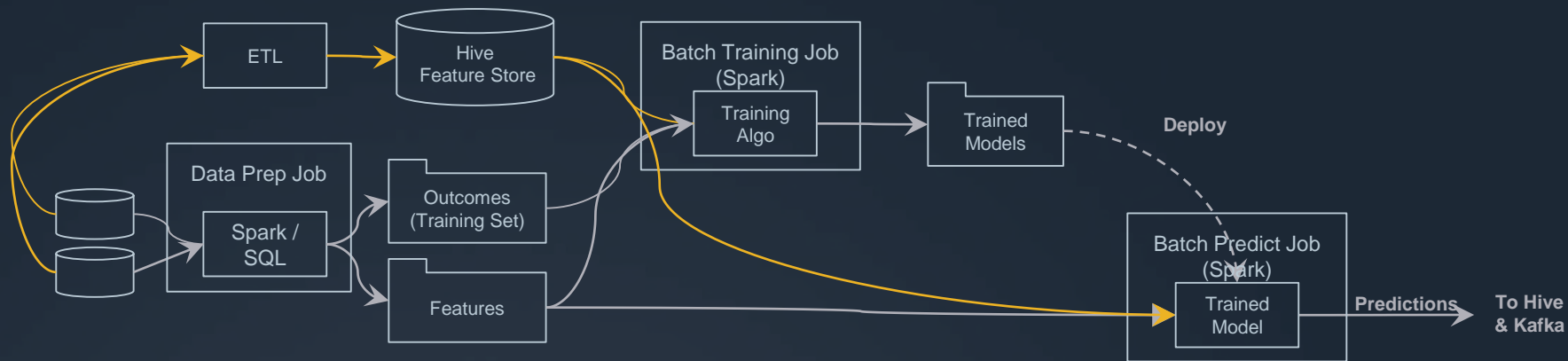
HADOOP / YARN (Batch)

GET DATA

TRAIN MODELS

EVAL MODELS

DEPLOY, PREDICT & MONITOR



HADOOP / YARN (Batch)

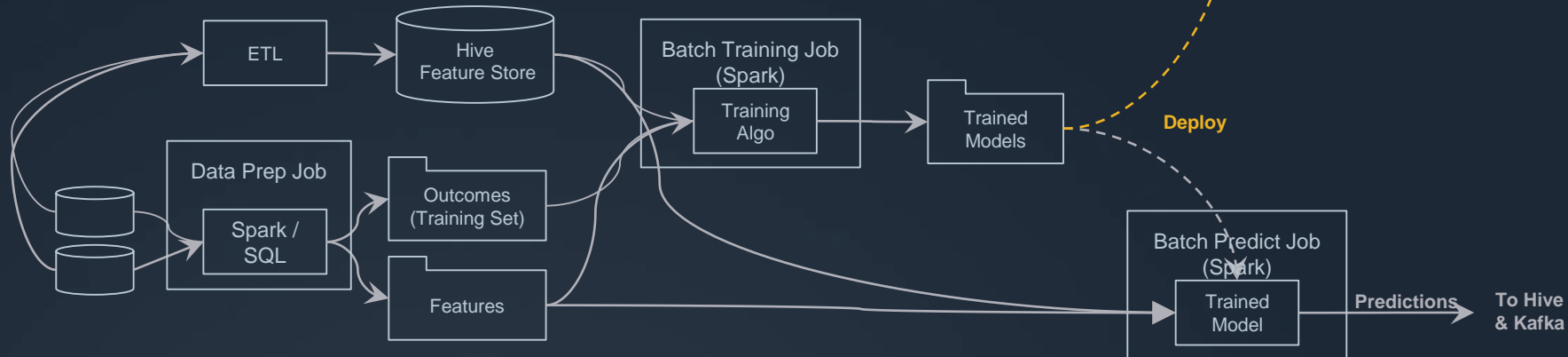
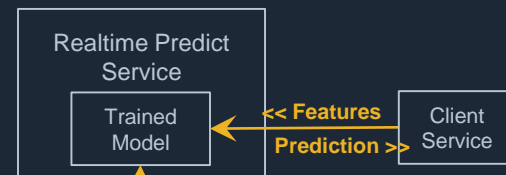
GET DATA

TRAIN MODELS

EVAL MODELS

DEPLOY, PREDICT & MONITOR

MESOS / DOCKER (Realtime)



HADOOP / YARN (Batch)

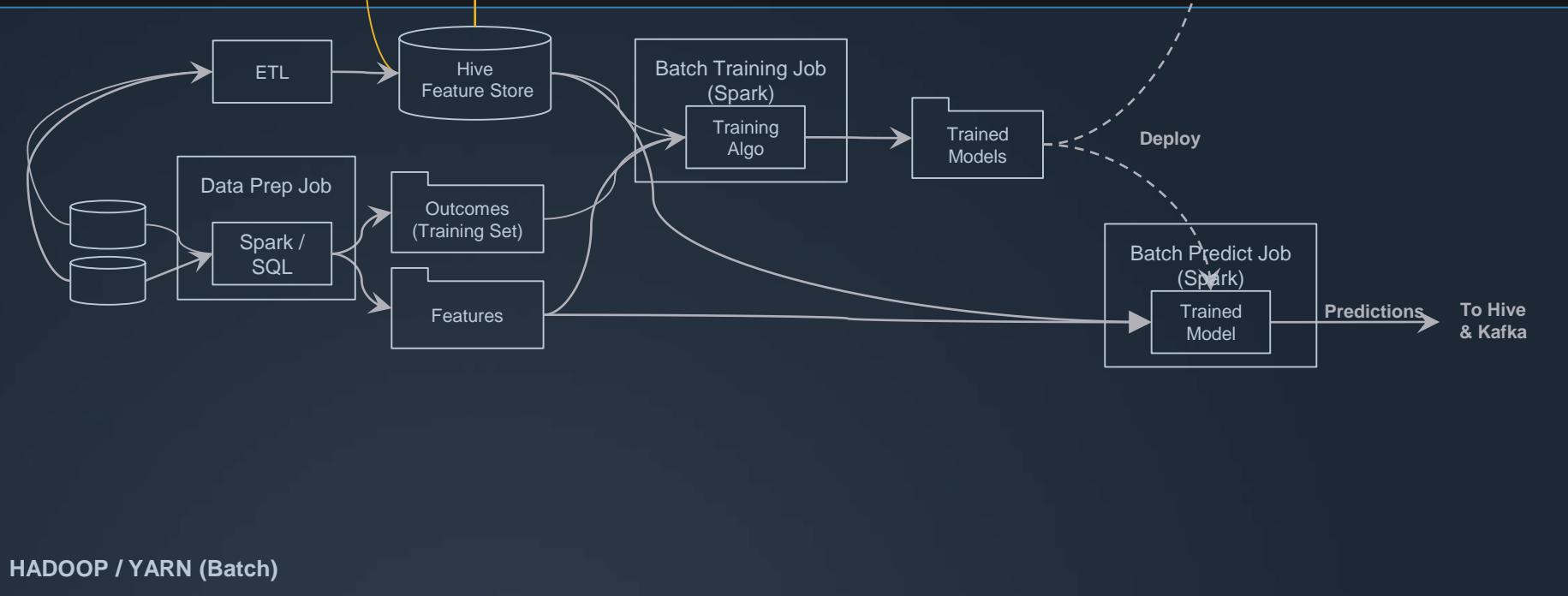
GET DATA

TRAIN MODELS

EVAL MODELS

DEPLOY, PREDICT & MONITOR

MESOS / DOCKER (Realtime)



HADOOP / YARN (Batch)

GET DATA

TRAIN MODELS

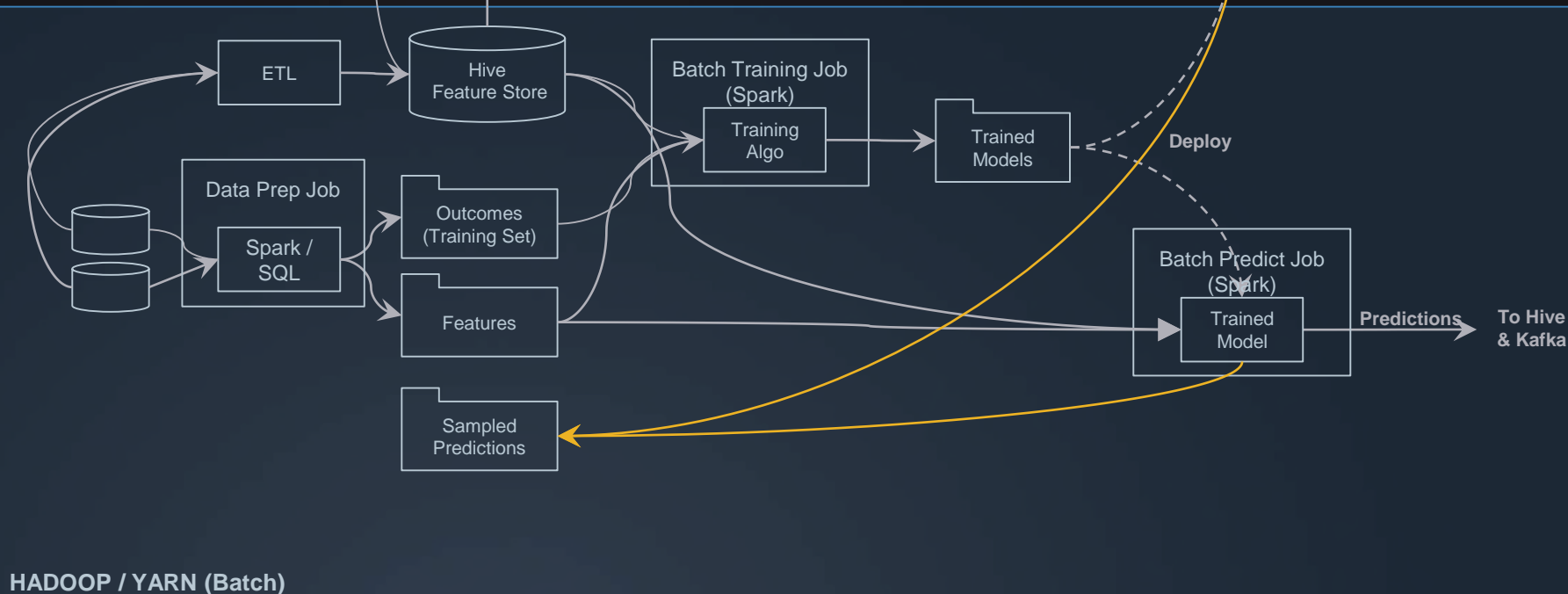
EVAL MODELS

DEPLOY, PREDICT & MONITOR

51CTO

WOT!

MESOS / DOCKER (Realtime)



HADOOP / YARN (Batch)

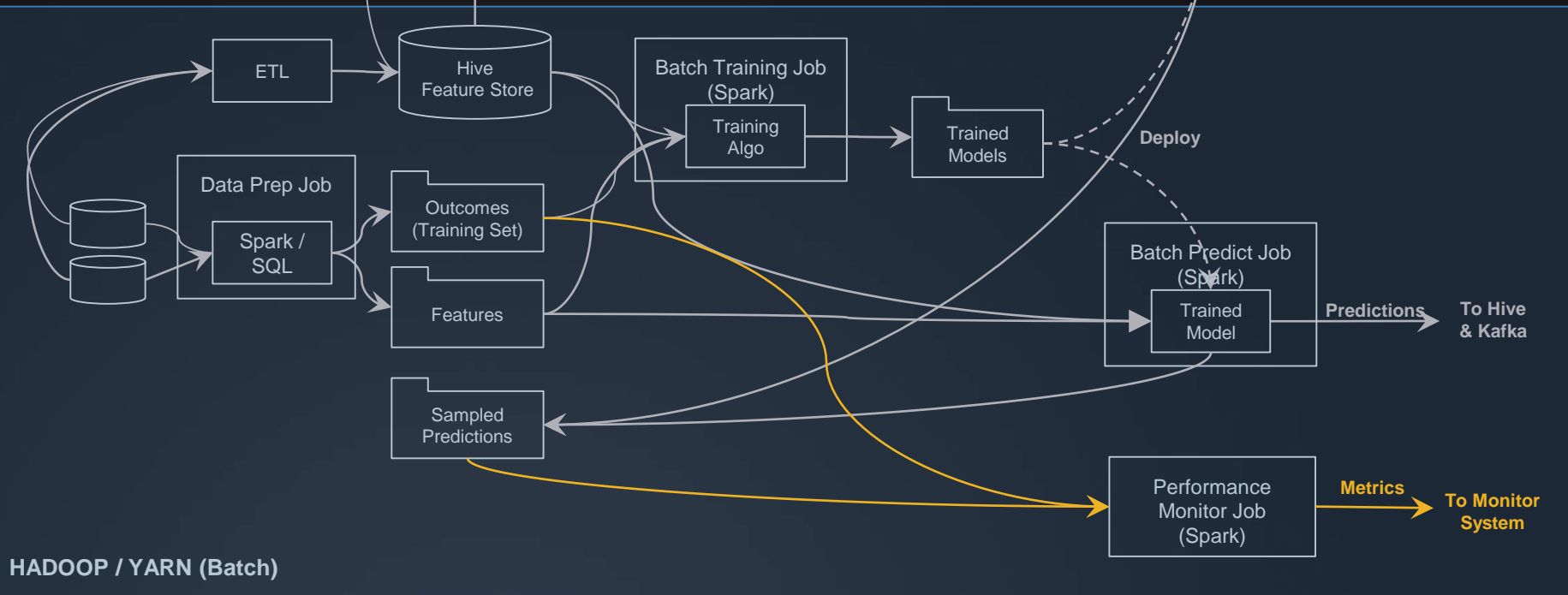
GET DATA

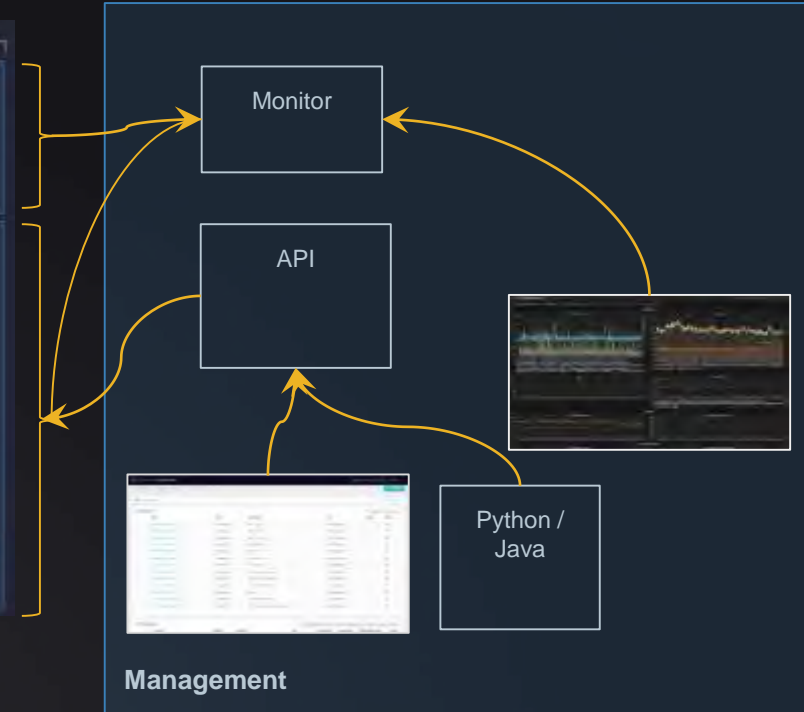
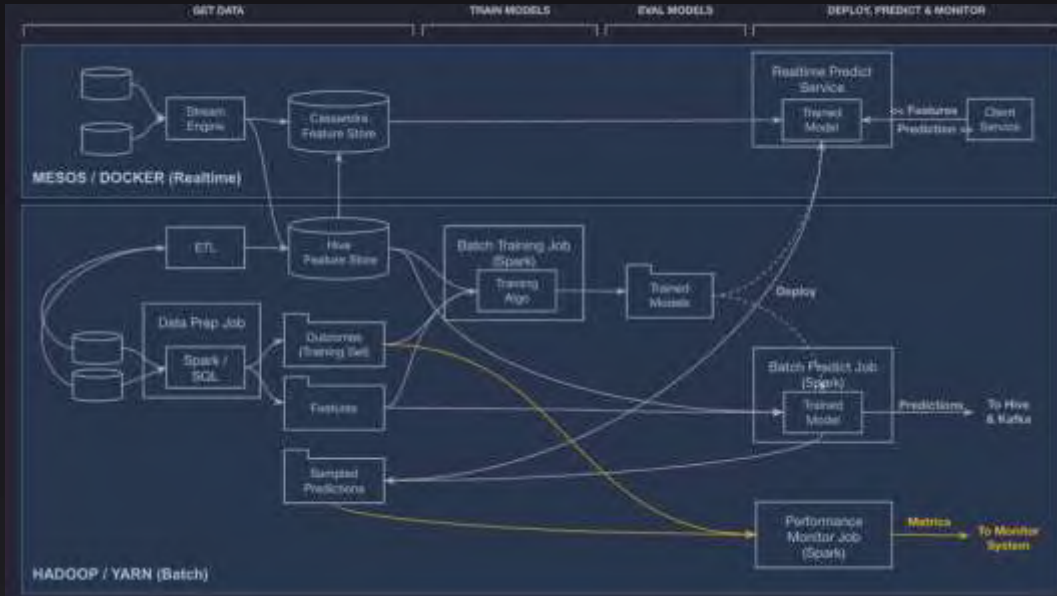
TRAIN MODELS

EVAL MODELS

DEPLOY, PREDICT & MONITOR

MESOS / DOCKER (Realtime)





Management

Thank you!