# Machine Learning Model Serving @Twitter

Zhiyong (Joe) Xie
Tech Lead & Sr. Software Engineer
Twitter

# Short Bio

- BS & MS @ Nanjing University
- MS @ University of Washington
- Intern @ Microsoft + Facebook
- Fulltime @ Amazon + Twitter

# Outline

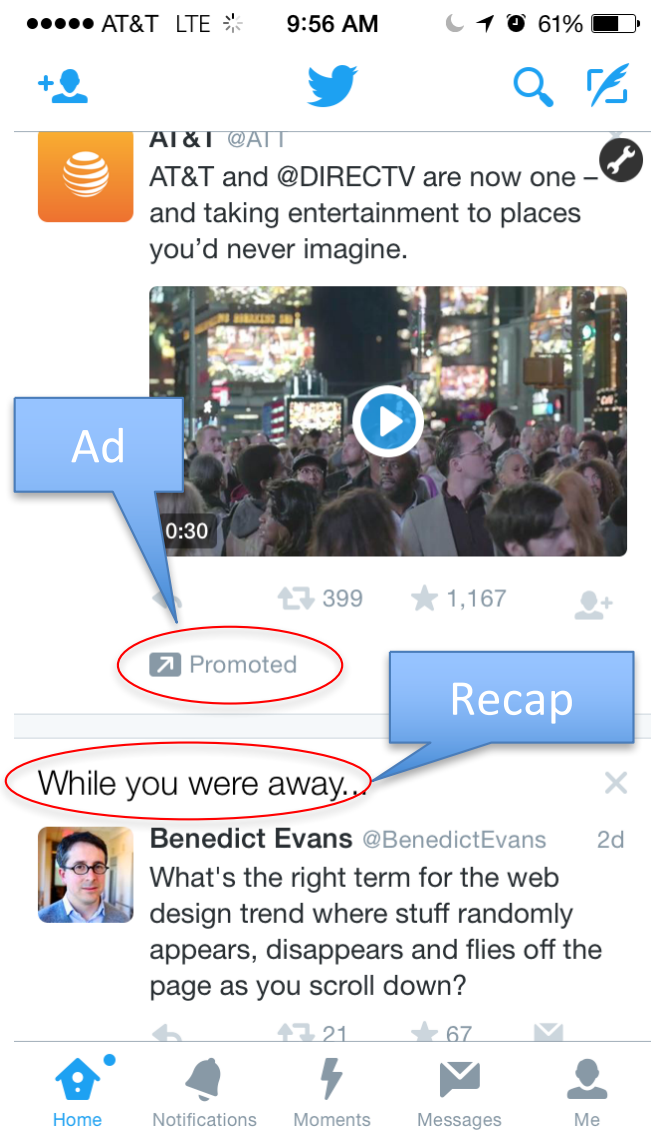
- Machine Learning (ML) Infra Overview
- Model Serving Challenges
  - –Performant
  - –Resilient & Robust
  - –Real-time
  - –Scalability
- Deep Dive into Solutions
- Model Serving Scenarios
- Case Study

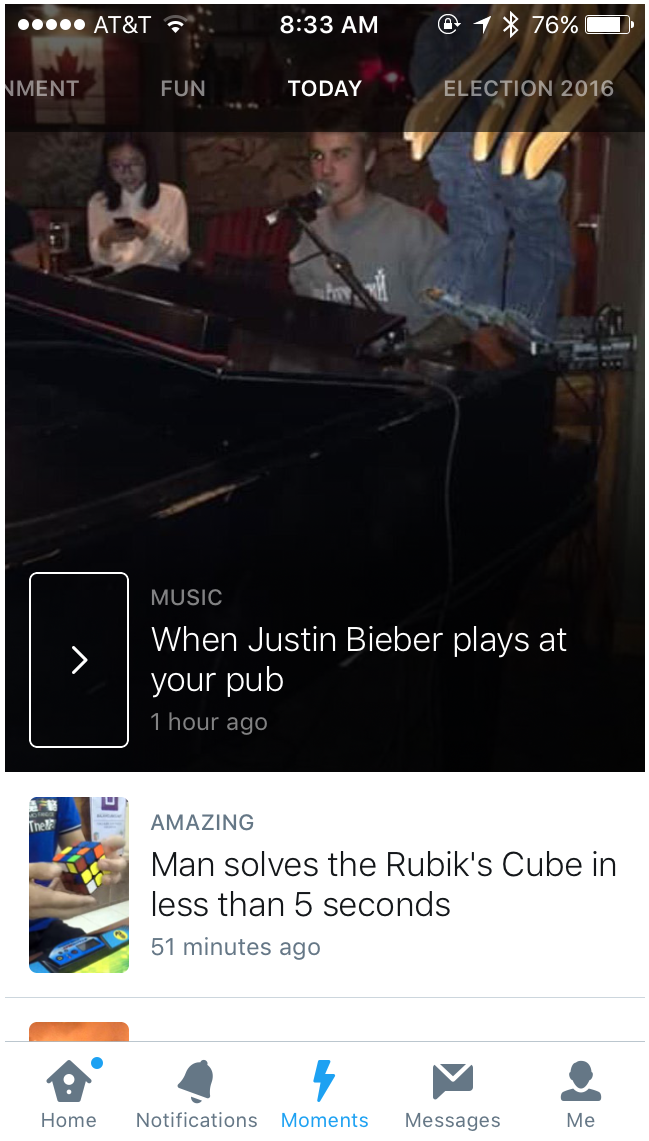
# Backgrounds

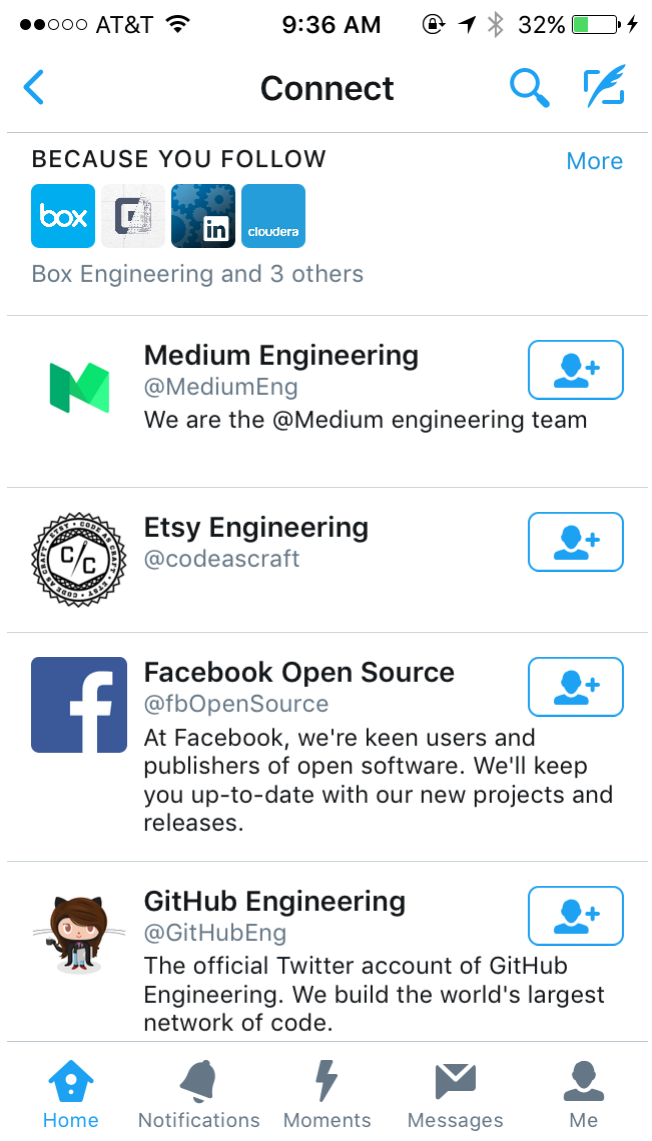
# ML Infra - Overview

- ML is increasingly at the core of everything we build at Twitter
- ML infra supports many product teams
  - ads ranking, ads targeting, timeline ranking, product safety, recommendation, moments ranking, trends

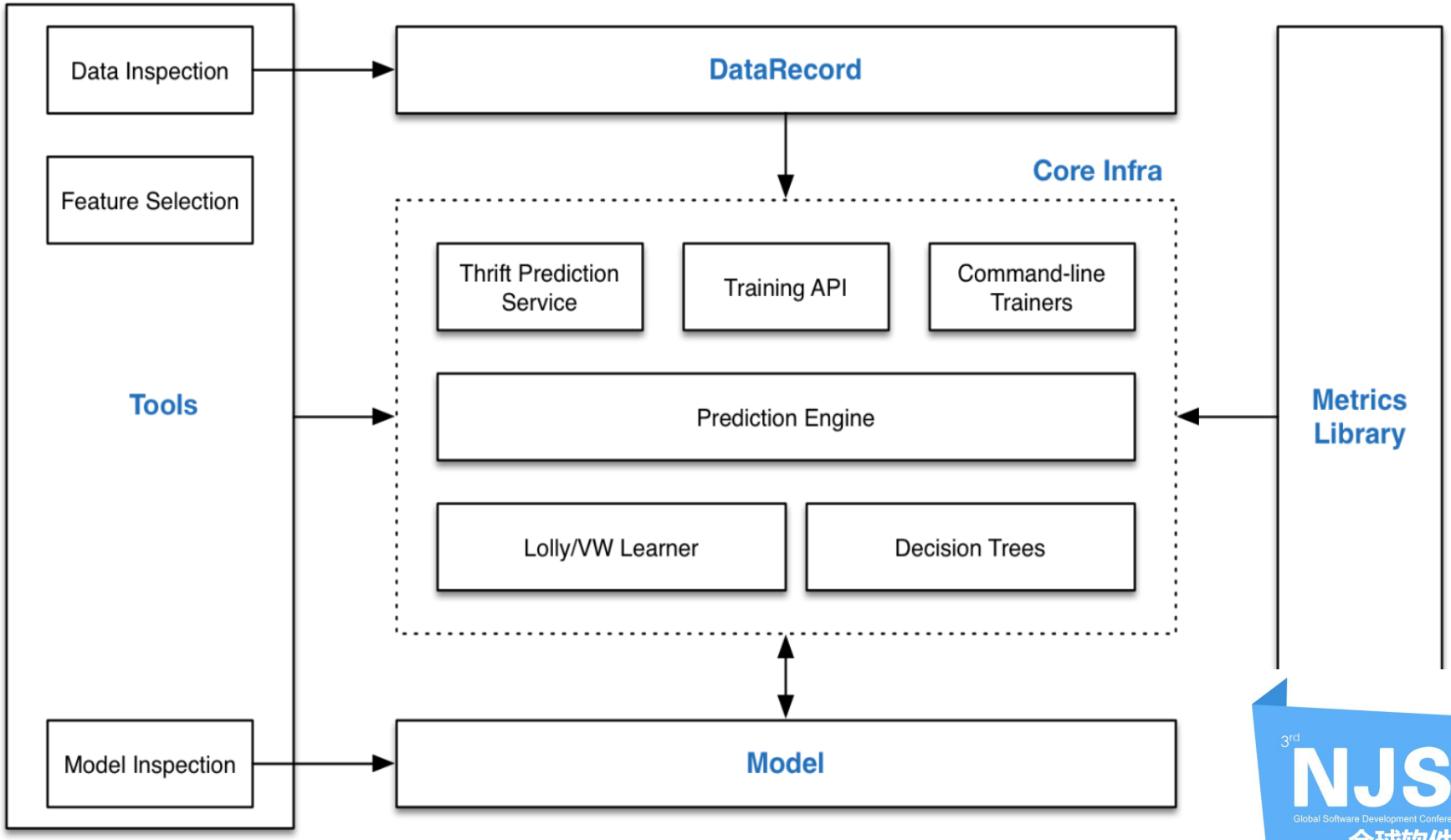
# ML Infra – Product Examples



# ML Infra - High-level Architecture

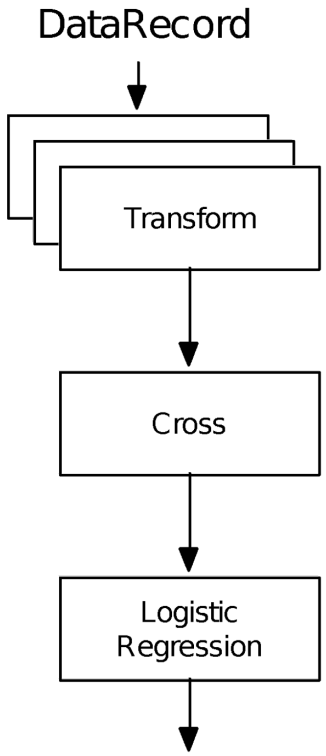


# ML Infra – Data Record

- Unified data representation shared across teams
- Data format
  - –Support 4 dense, two sparse features
  - –Use hashed feature id instead of string name for efficient serialization, storage and computation
  - –Data schema for feature id to name mapping

# ML Infra – Core Prediction Engine

- Large scale generalized linear learning with nonlinear feature representations
- Architecture
  - –Nonlinear transform: Minimum description length (MDL), decision trees, neural network
  - –Feature crossing
  - –Logistic regression: In-house JVM learner



# Challenges

# Challenges - Performant

- Trillions of predictions served daily
- Thousands of features per example
- Milliseconds latency per request
- ...

# Challenges – Resilient & Robust

- Traffic spike during events, etc. Super bowl, Oscar award, world cup
- Traffic corruption due to upstream issue
- Machine failure

# Challenges – Real Time

- Twitter is all about real-time: news, events, trends, hashtags.
- Advertising campaign targets real-time event spanning short period of time
- ML model dynamically adapts to changes spanning as short as a few hours even minutes

# Challenges - Scalability

- Horizontal scaling to handle organic growth, new features and advanced modeling
- Hundreds of millions of weights per model
- TBs of training data

# Solutions

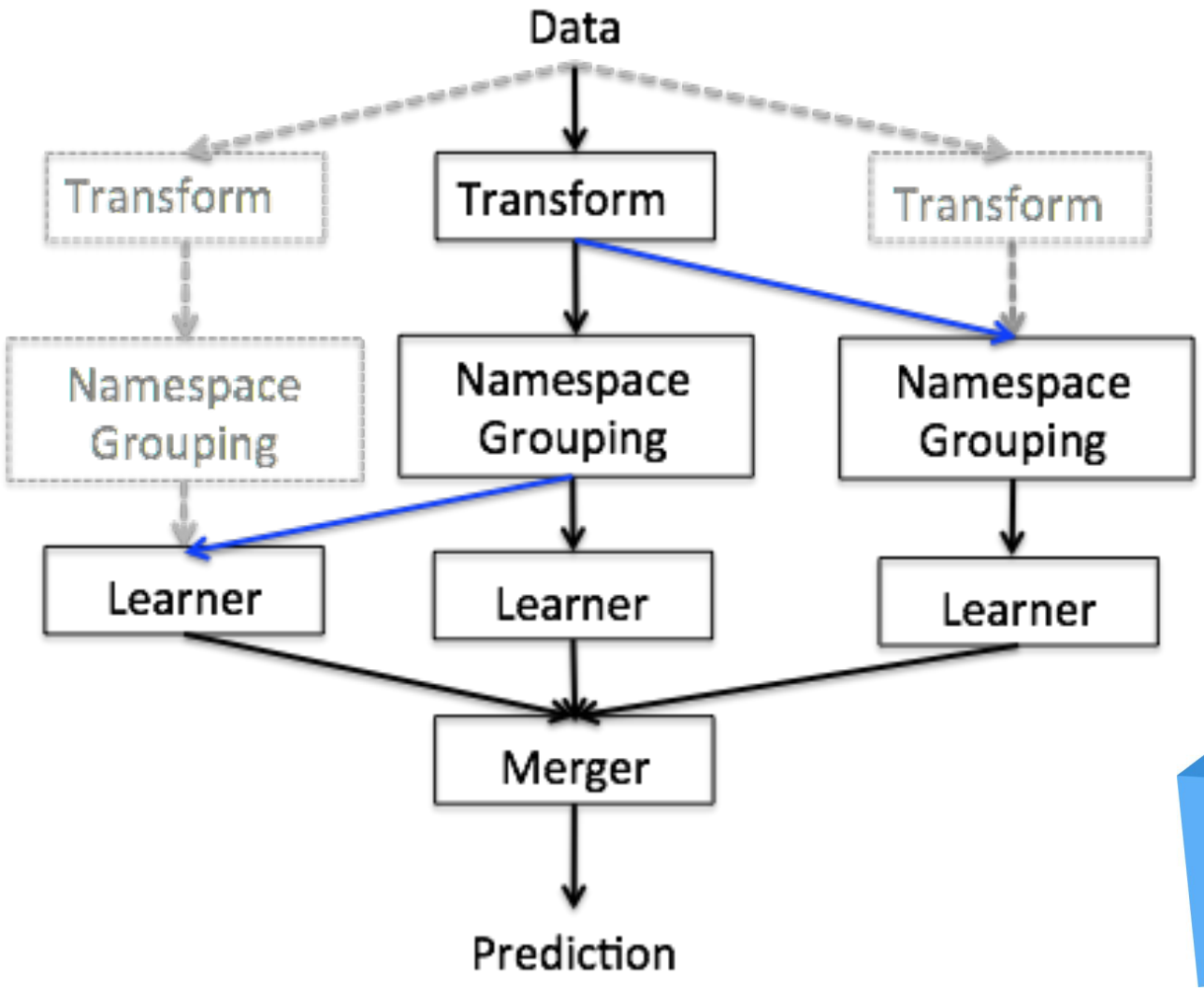
# Solutions - Performant

- Reduce serialization cost
  - Model collocation
  - Batch request API
  - Compressed request API

# Solutions - Performant

- Reduce computational cost
  - Feature id instead of string name
  - Transform sharing across models
  - Feature cross done on the fly

# Solutions - Performant




Multiple Model Flow with Topology Sharing

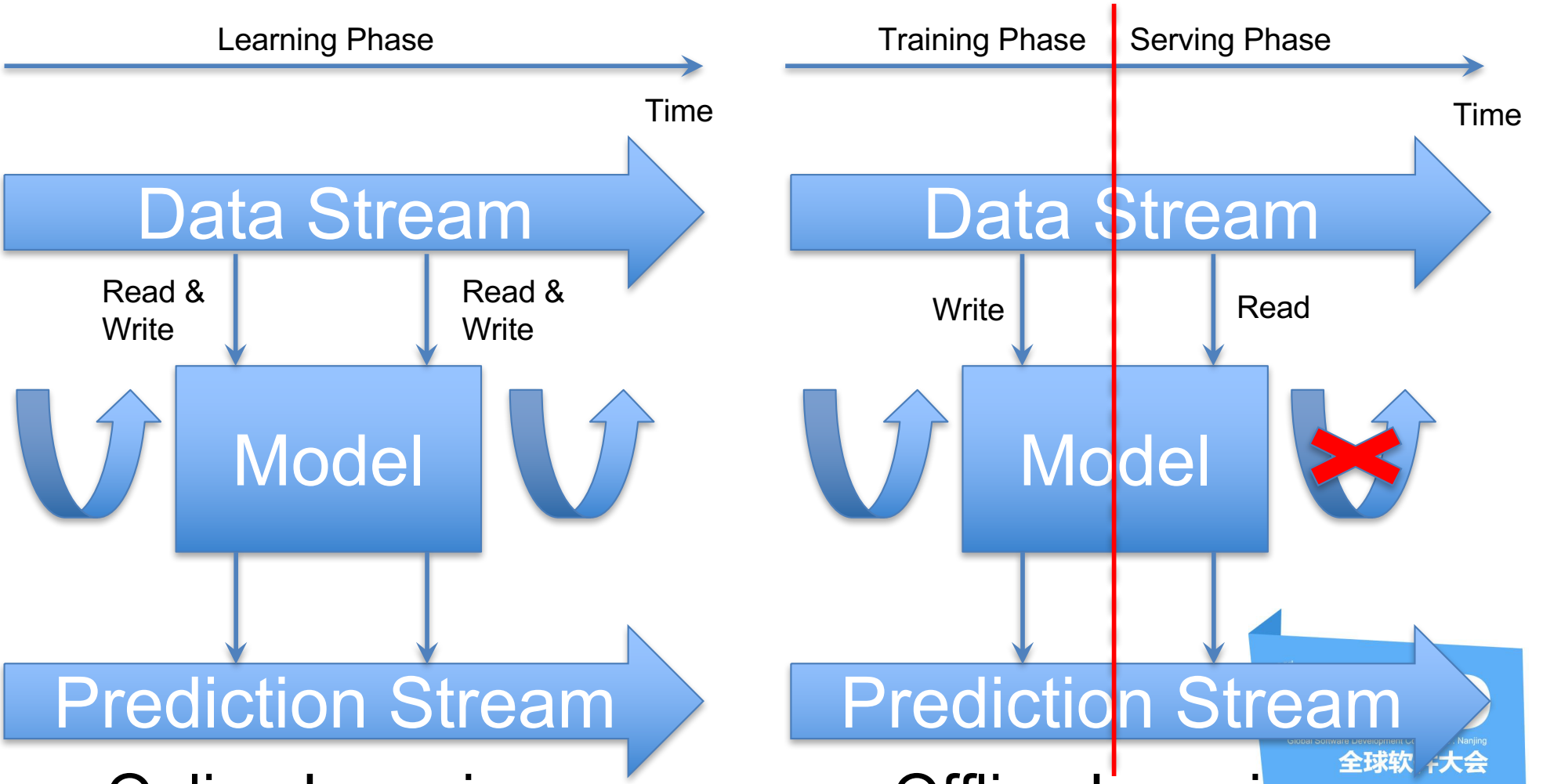
# Solutions - Resilient

- Load factor to control the traffic at the client side based on the success rate of the requests
- QPS limiter to control the traffic at the service side

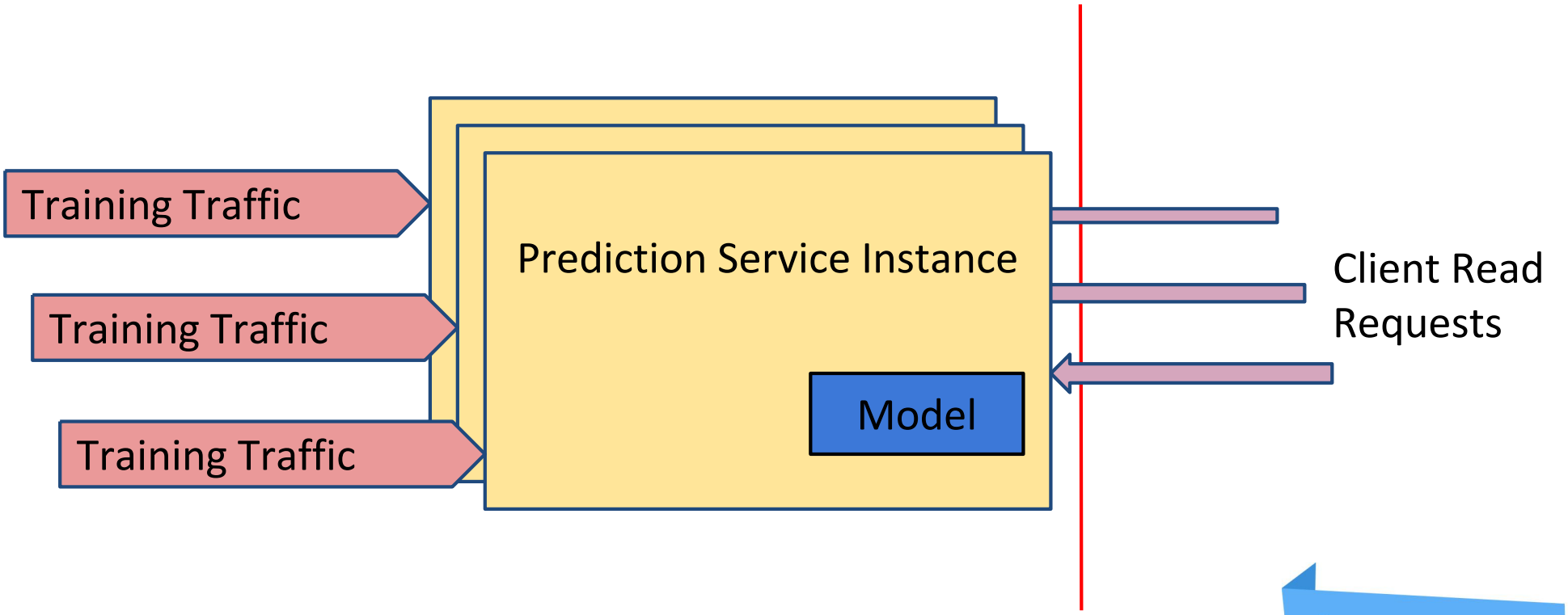
# Solutions - Robust

- Snapshot models at fixed interval
- Abnormal detection based on traffic pattern
- Controller to turn on / off the traffic

# Solutions - Real Time:
# Online vs. Offline Learning

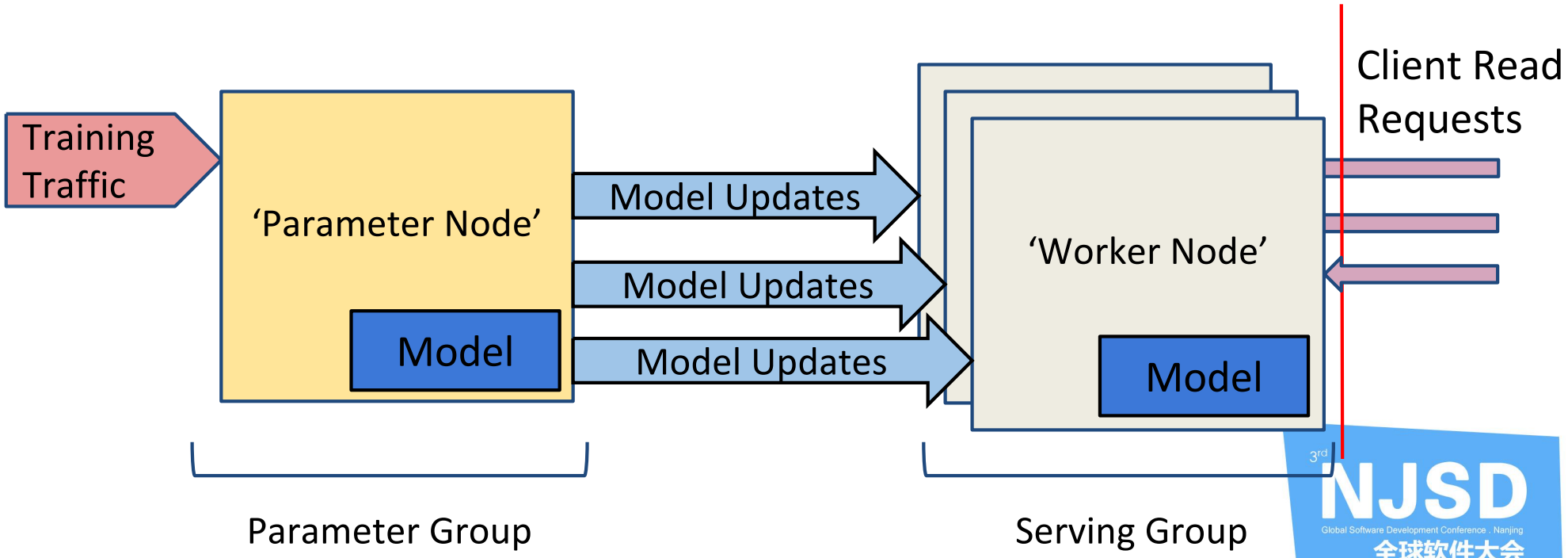


# Solutions - Real time: Online Learning Architecture



# Solutions - Scaling: Parameter Server

- Incremental model updates instead of integrated training



# Model Serving Scenarios

- Static model in-memory integration
- Static model standalone service
- Online learning service with integrated training
- Scaling Online Learning with Parameter server

# Scenarios: Service vs. Library

## When Service is a good fit?

- Easier to use / update / scale
- Separate the heavy CPU / memory loads from the client system
- Leverage existing tools (etc loadtest, dashboard, querying client) and batch compressed training

## When Library might be a good fit?

- Small model with limited features

# Scenarios: Online vs. Offline Learning

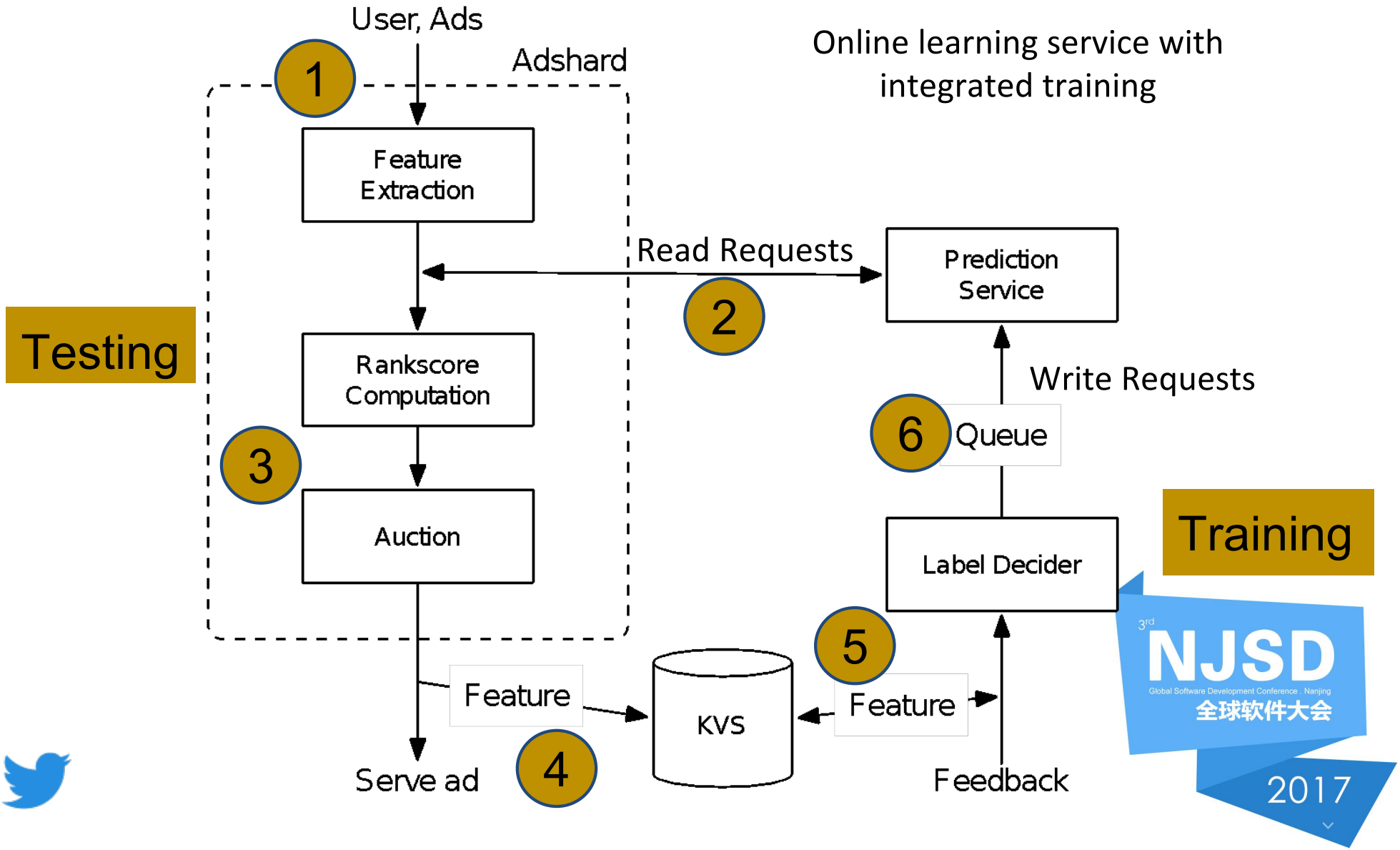
- When online learning is a good fit?
  - Capture the real-time info naturally
  - Improve the prediction quality continuously
  - Adapt to adversarial / competitive settings

- When offline learning is a good fit?
  - Data is scarce with high acquisition cost
  - Label is not immediately available

# Scenarios: Parameter Server

- Low serving efficiency due to high ratio of training / prediction traffic
- High network usage due to training traffic fan-out

# Case Study – Ads Prediction



# Work In Progress

- Deep learning as feature transform
- Distributed training for scaling online learning
- …

# Thank you!