Deploying Interactive Machine Learning Applications with Clipper

Joseph E. Gonzalez
Co-director of the RISE Lab
jegonzal@cs.berkeley.edu

Managing the Machine Learning Lifecycle

Joseph E. Gonzalez
Co-director of the RISE Lab
jegonzal@cs.berkeley.edu

About Me

- Co-director of the RISE Lab
- Co-founder of Turi Inc.
- Member of the Apache Spark PMC

Research

- Artificial Intelligence
- Data Science
- Distributed Data Systems
- Graph Processing Systems

riselab
UC Berkeley

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

BERKELEY
DeepDrive

APACHE
Spark™

turi
create intelligence™

Conjecture

Machine learning models are the next “big data”

Evidence

1. Everyone is talking about models but few have them.
2. They have the opportunity to transform industries.
3. They are a consequence of mastering big data.
4. Today, their full value is only realized with advanced skills and technologies

Conjecture

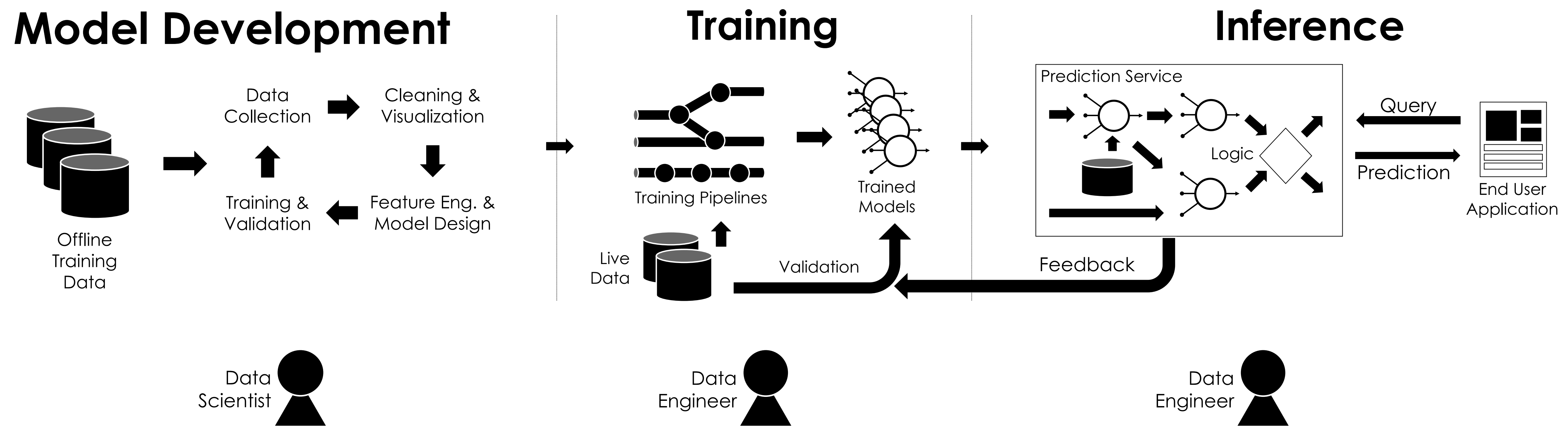
Machine learning models are the next "big data"

Corollaries

*Data Engineers will need to manage data & **machine learning models***

*We need new technologies to manage the **machine learning lifecycle***

What is the *Machine Learning Lifecycle*?



Model Development

Offline Training Data

Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Data Scientist

Training

Training Pipelines

Trained Models

Live Data

Validation

Data Engineer

Inference

Prediction Service

Logic

Query

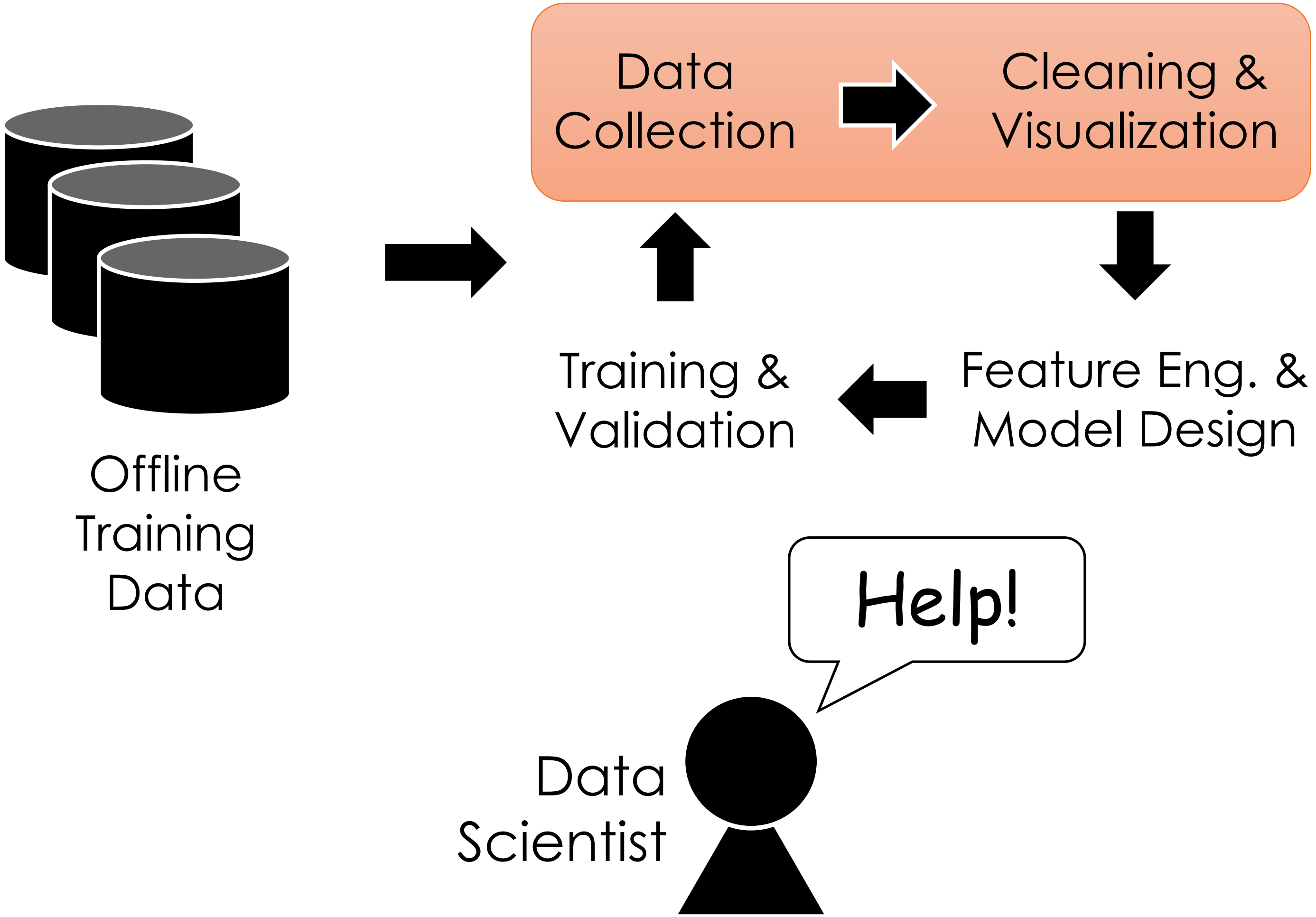
Prediction

End User Application

Feedback

Data Engineer

Model Development



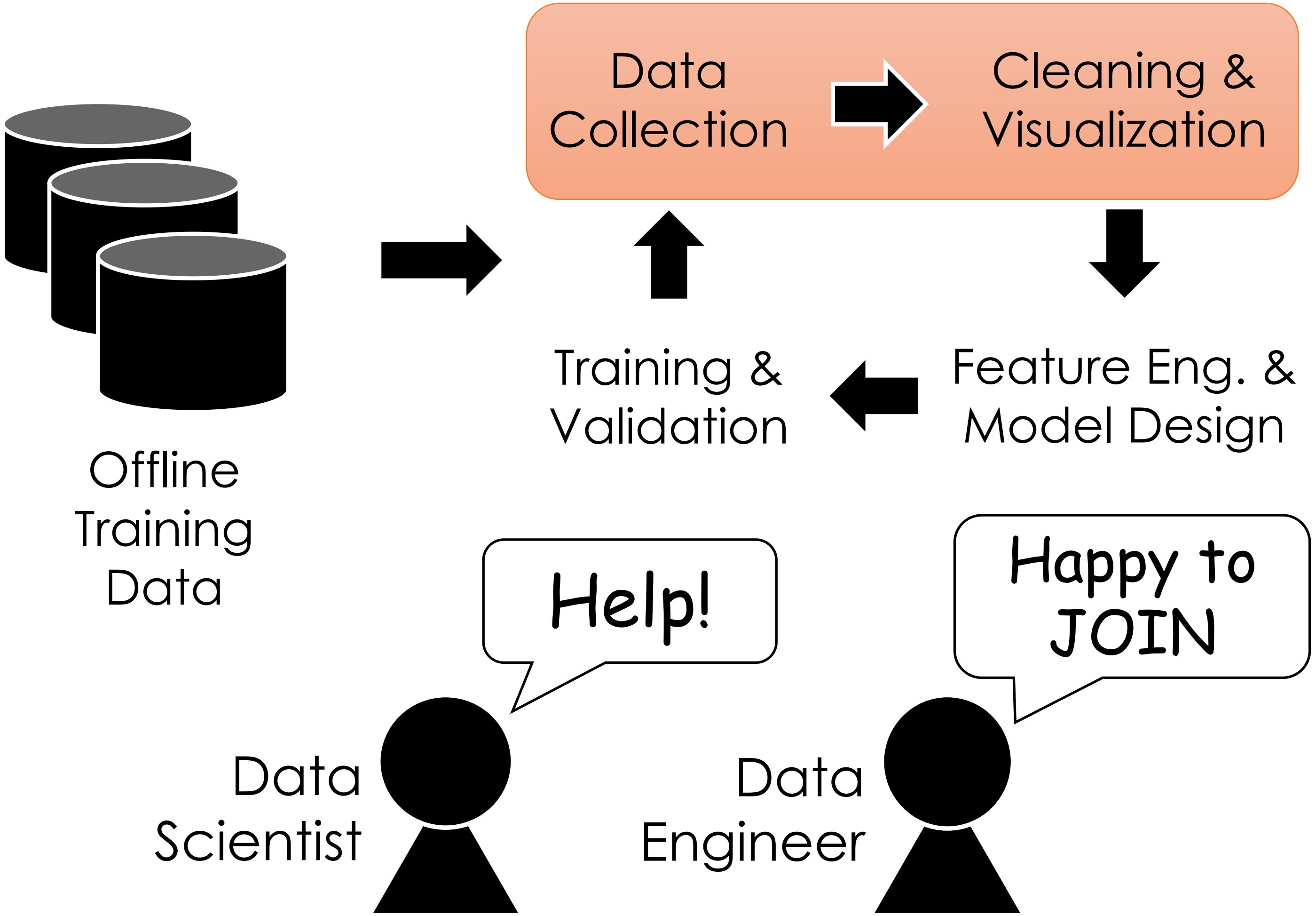
Identifying potential sources of data

Joining data from multiple sources

Addressing **missing values** and **outliers**

Plotting trends to identify **anomalies**

Model Development



Offline Training Data

Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Data Scientist

Data Engineer

Identifying potential sources of data

Joining data from multiple sources

Addressing **missing values** and **outliers**

Plotting trends to identify **anomalies**

Big Data Borat
@BigDataBorat

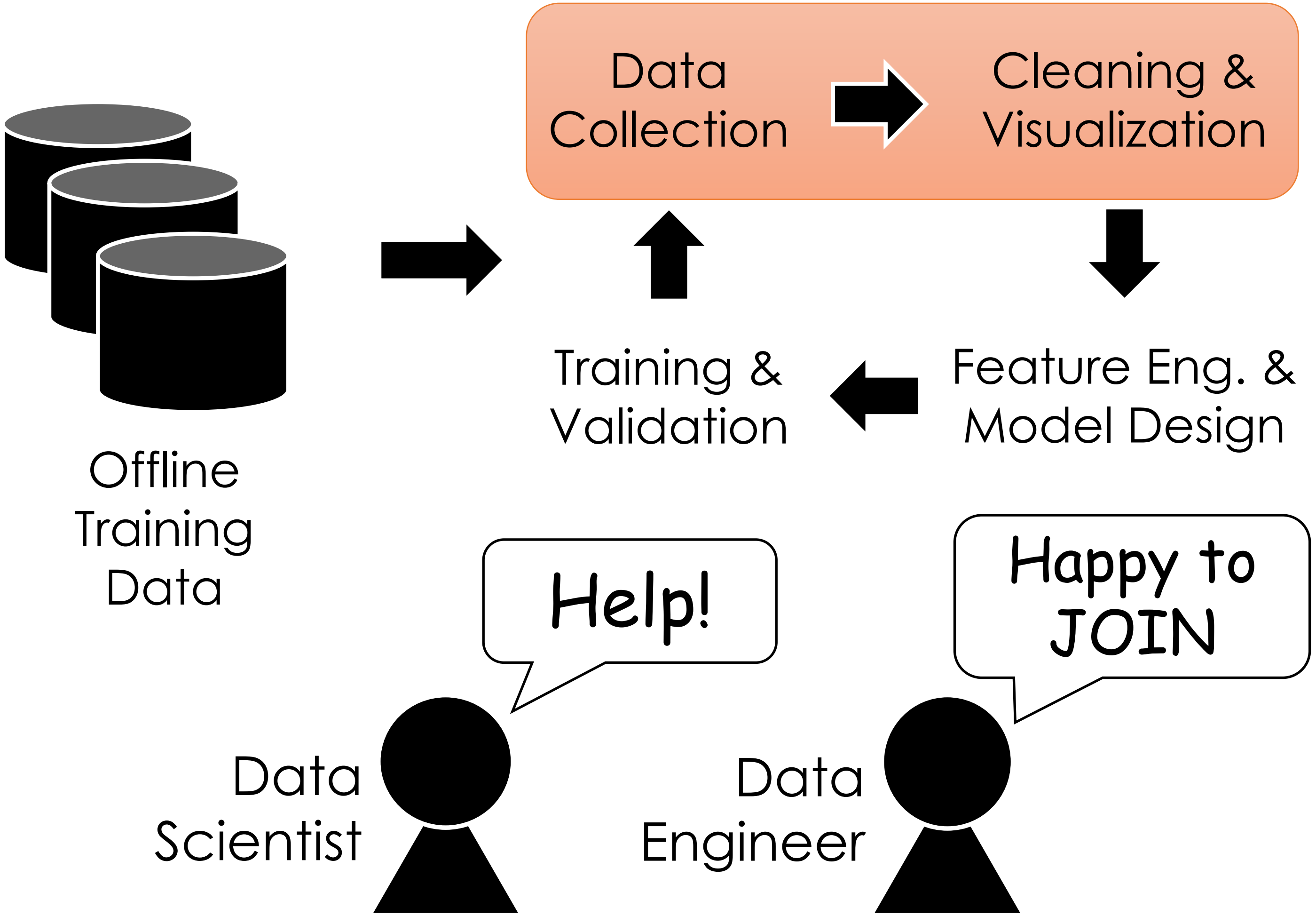
In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

6:47 PM - 26 Feb 2013

533 Retweets **330** Likes

12 533 330

Model Development


Offline Training Data

Data Collection → Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Data Scientist

Data Engineer

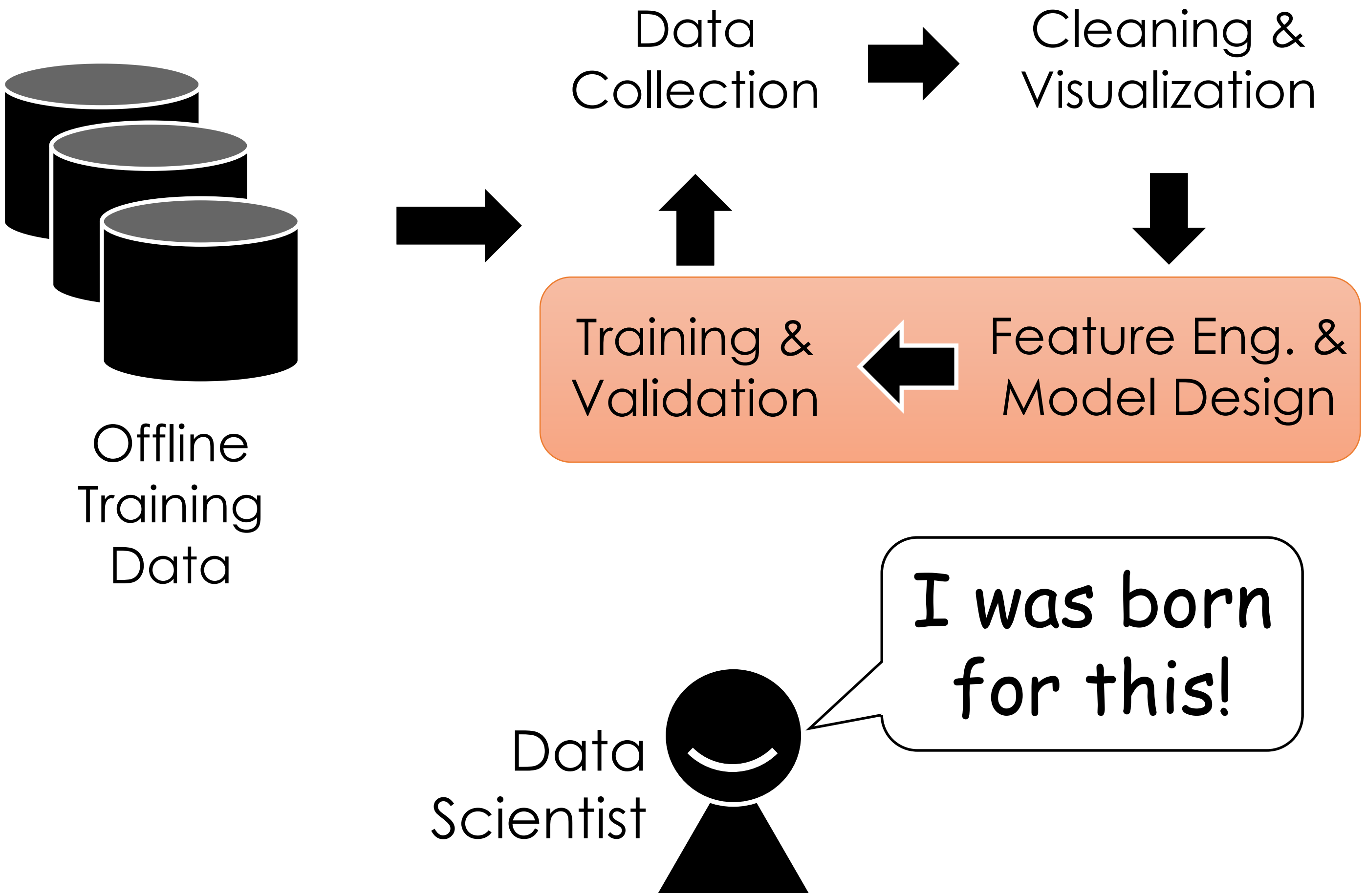
Identifying potential sources of data

Joining data from multiple sources

Addressing **missing values** and **outliers**

Plotting trends to identify **anomalies**

Model Development



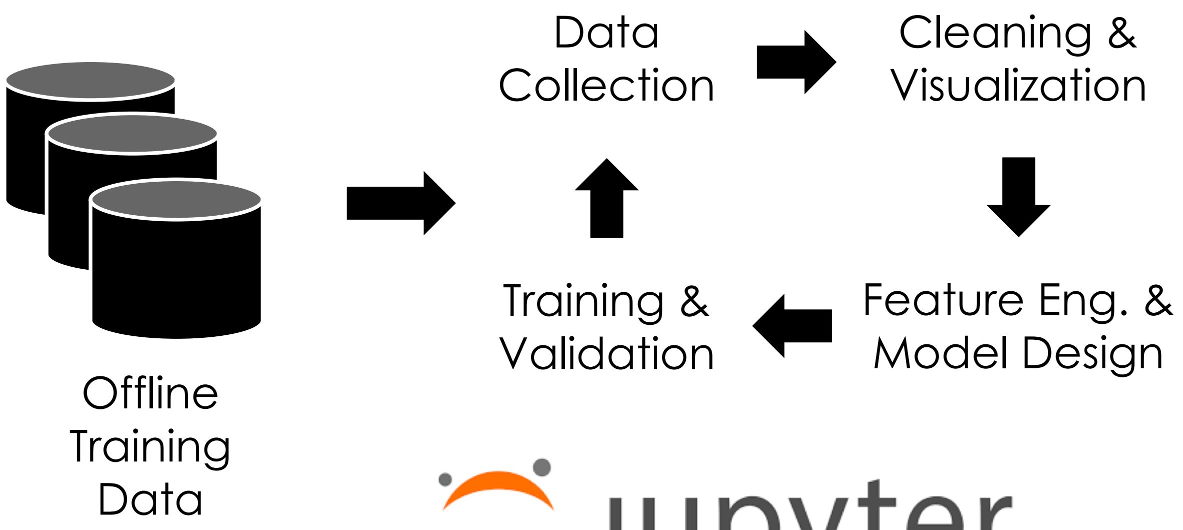
Building informative **features functions**

Designing new **model architectures**

Tuning training algos.

Validating prediction accuracy

Model Development *Technologies*



Offline Training Data

Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

scikit learn

TensorFlow

R

PYTORCH

Keras

jupyter

mxnet

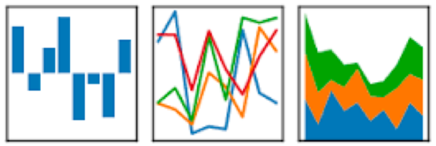
dmlc XGBoost

matplotlib

Caffe2

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



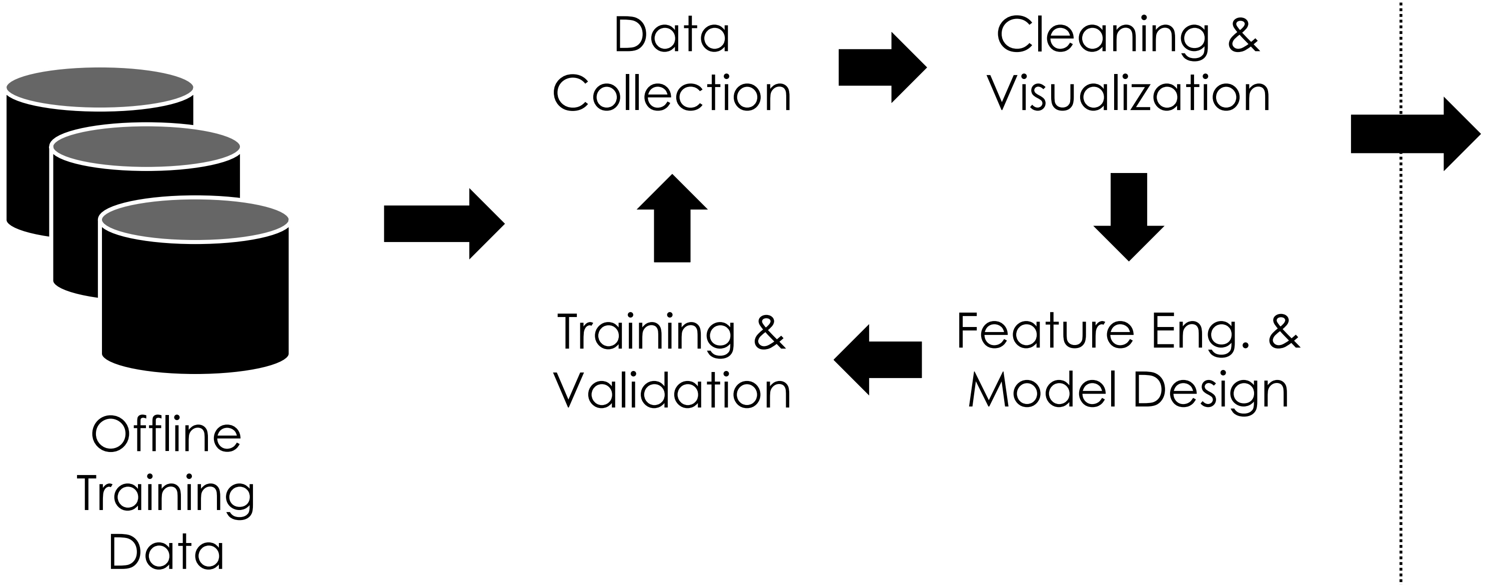
NumPy

APACHE Spark™

DASK

HIVE

What is the output of Model Development



Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

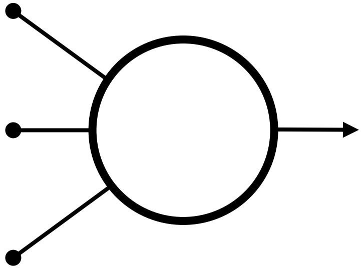
Training & Validation

Offline Training Data

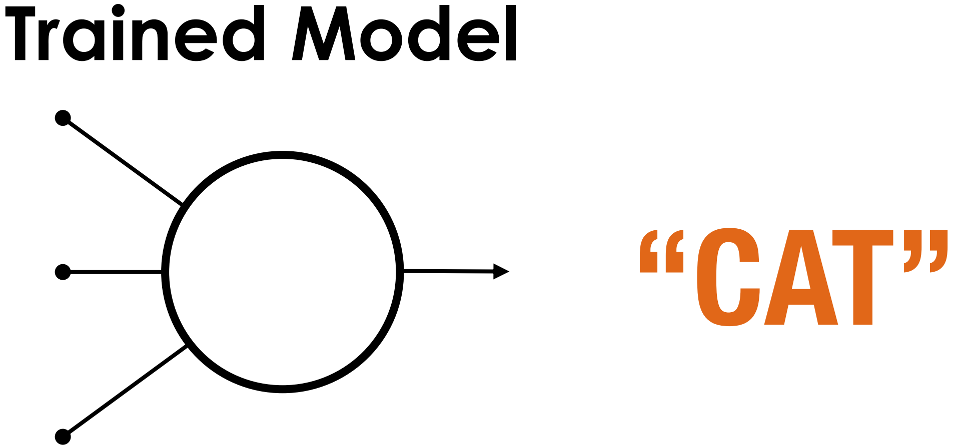
Reports & Dashboards

(insights …)

Trained Model



A learned function from a **query** to a **prediction**



Trained Model

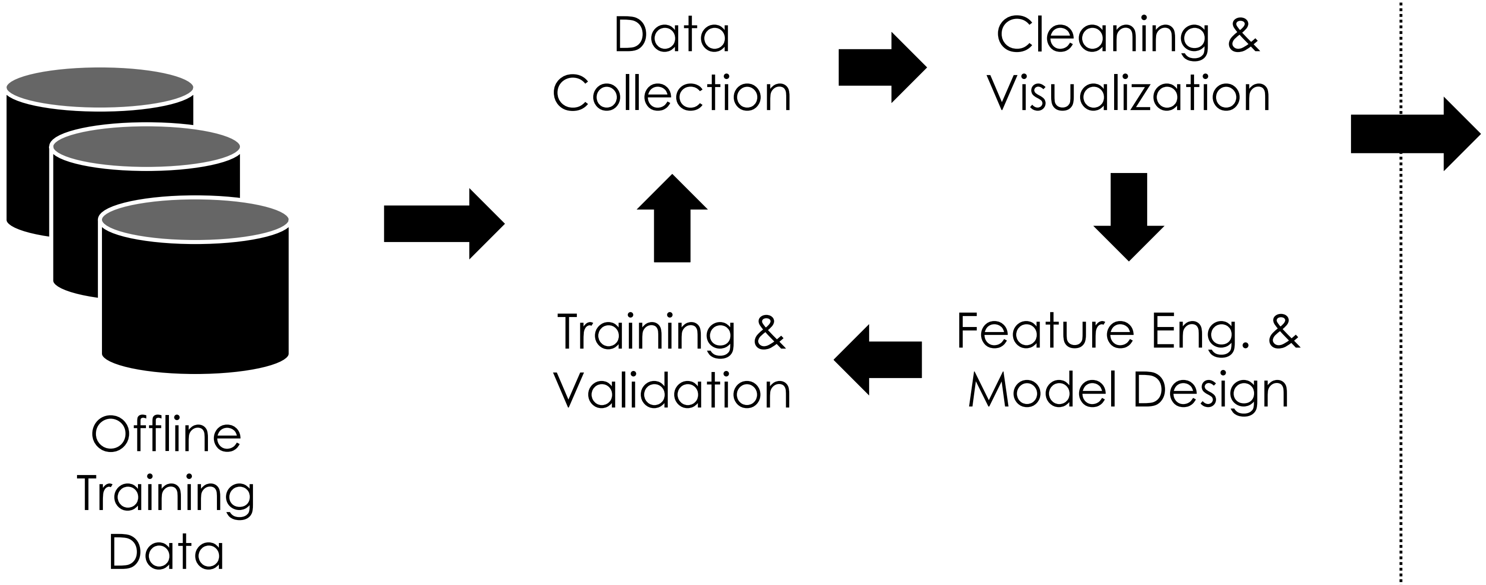
“CAT”

consisting of **parameters** and **model structure**.

Data
(10B to 10GB)

How to use the parameters…

What is the output of Model Development



Offline Training Data

Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Reports & Dashboards

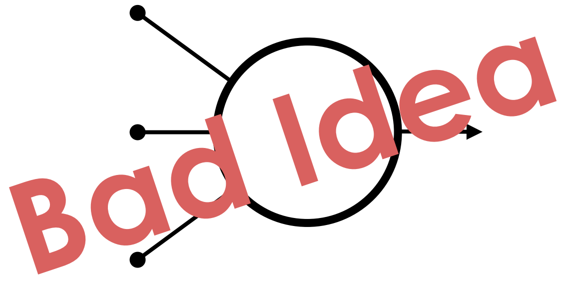
View of notebook: Dashboard Demo Notebook

Showing the first 1000 rows.

Showing sample based on the first 1000 rows. Plot over all results.

(insights …)

Trained Model

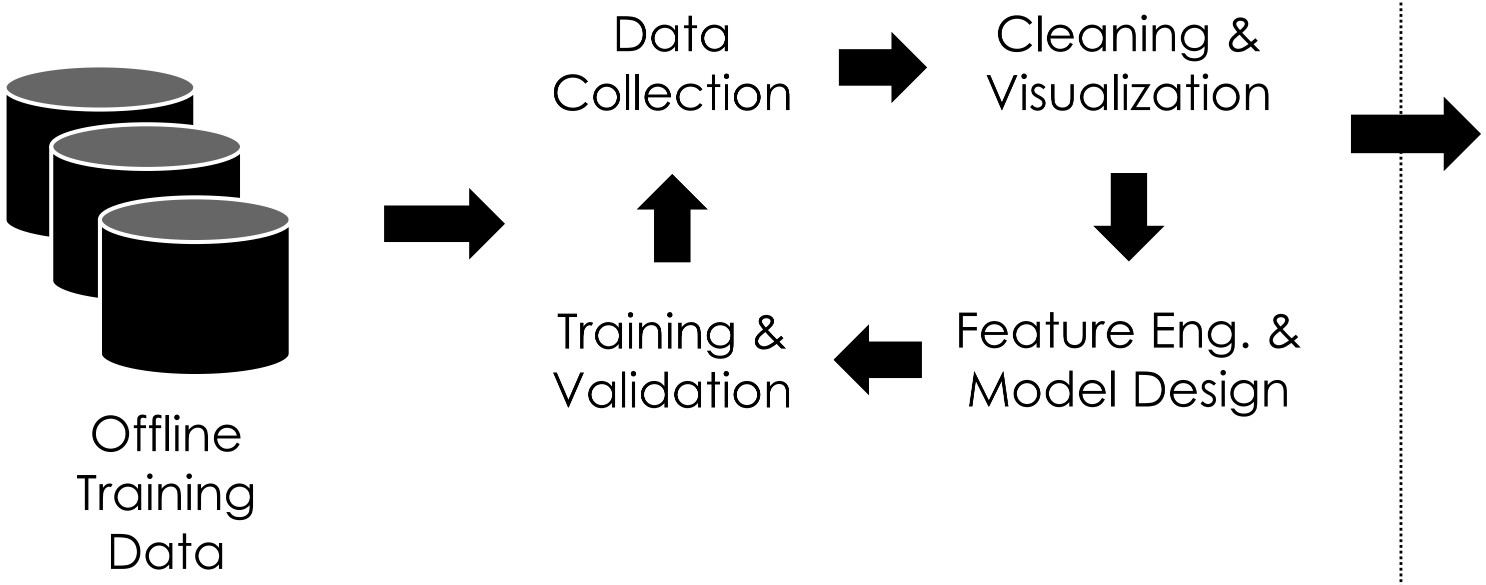


Why is it a **Bad Idea** to directly produce **trained models** from **model development**?

With just a trained model we are **unable to**

1. **retrain** models with new data
2. track data and code for **debugging**
3. capture **dependencies** for deployment
4. audit training for **compliance** (e.g., GDPR)

What is the output of Model Development



Offline Training Data

Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Reports & Dashboards

Showing the first 1000 rows.

Showing sample based on the first 1000 rows. Plot over all results.

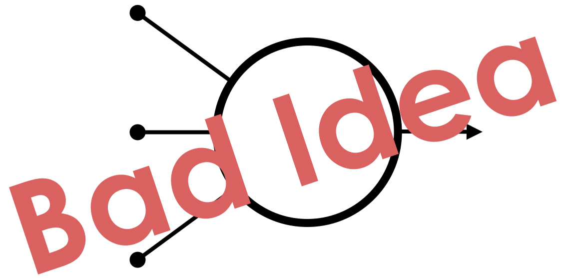
View of notebook: Dashboard Demo Notebook

Layout option: Stack

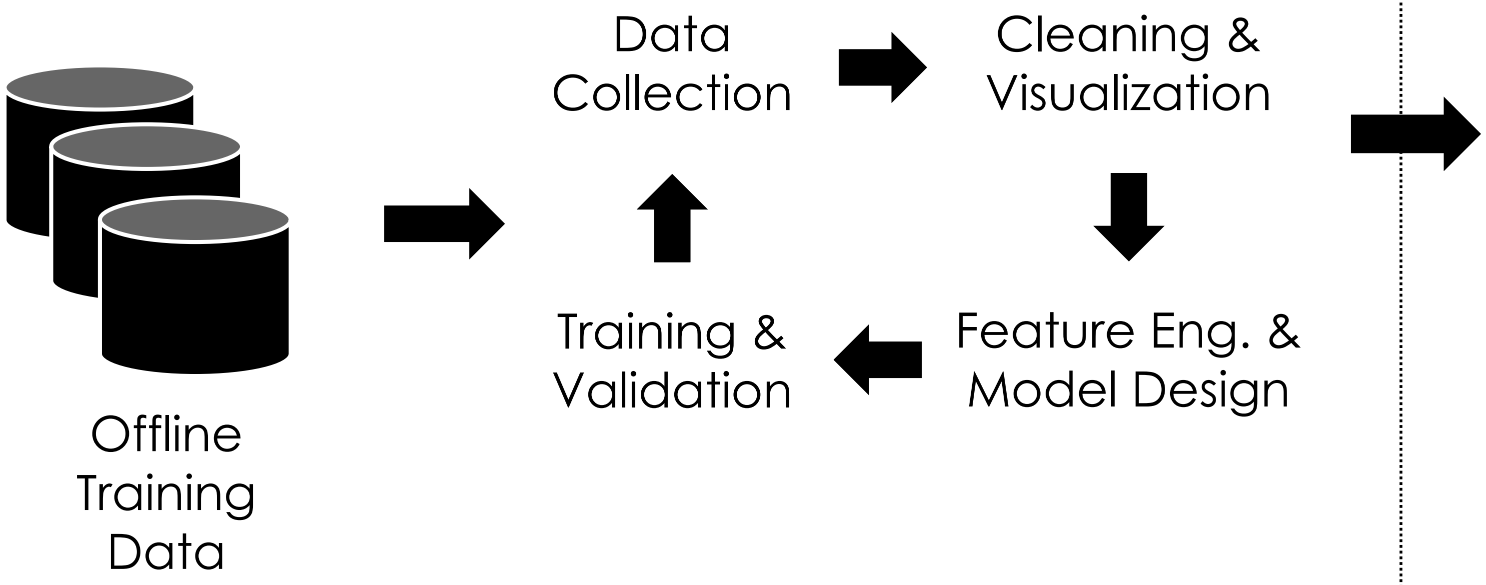
Dashboard width: 1024px

(insights …)

Trained Models



What is the output of Model Development



Offline Training Data

Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Reports & Dashboards

View of notebook: Dashboard Demo Notebook

Layout option: Stack Float

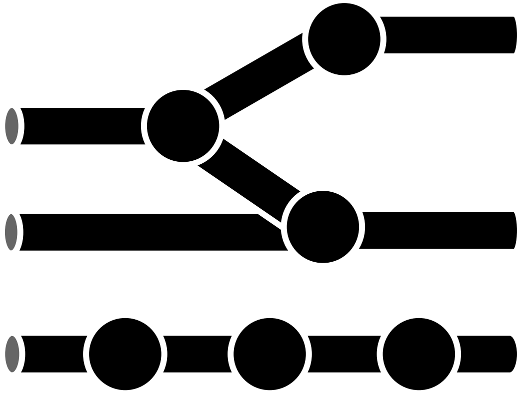
Dashboard width: 1024px

Showing the first 1000 rows.

Showing sample based on the first 1000 rows. Plot over all results.

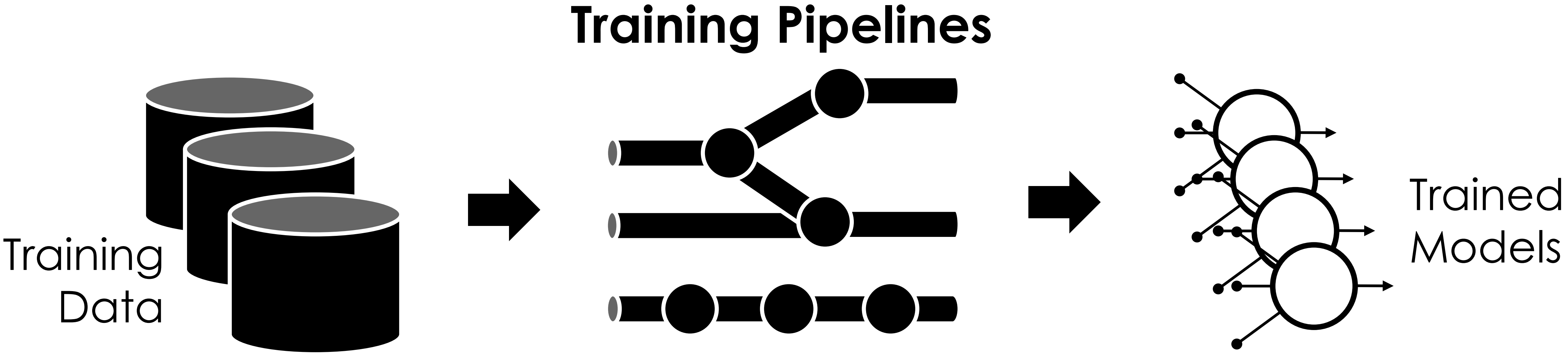
(insights …)

Training Pipelines



Training Pipelines Capture the **Code** and **Data Dependencies**

- ➢ Description of how to train the model from data sources

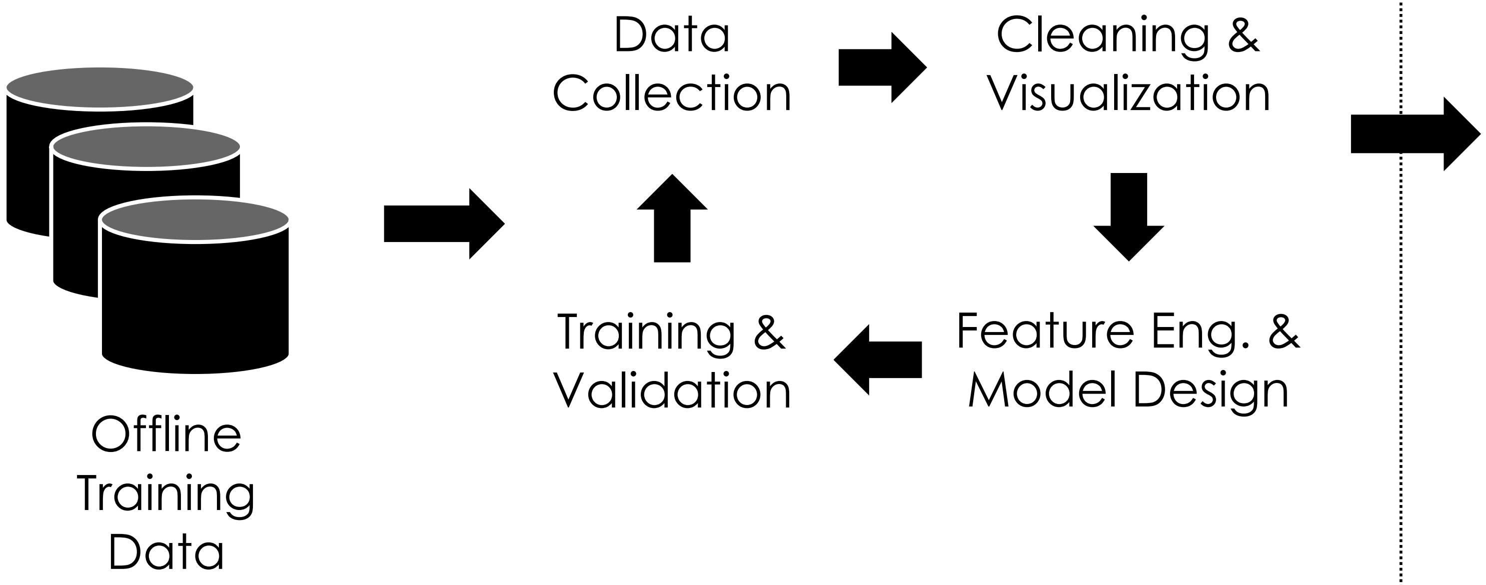


Software Engineering Analogy

Training Pipelines → Code

Trained Models → Binaries

What is the output of Model Development



Data Collection

Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Offline Training Data

Reports & Dashboards

View of notebook: Dashboard Demo Notebook

Layout option: Stack

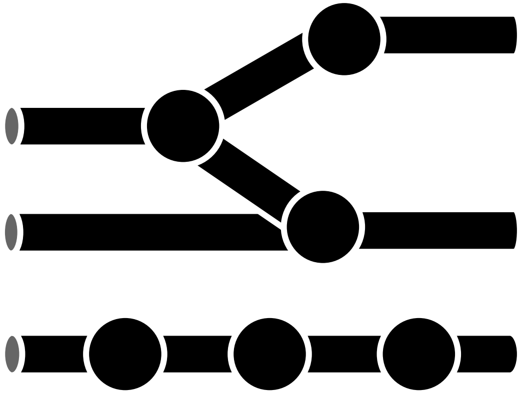
Dashboard width: 1024px

Showing the first 1000 rows.

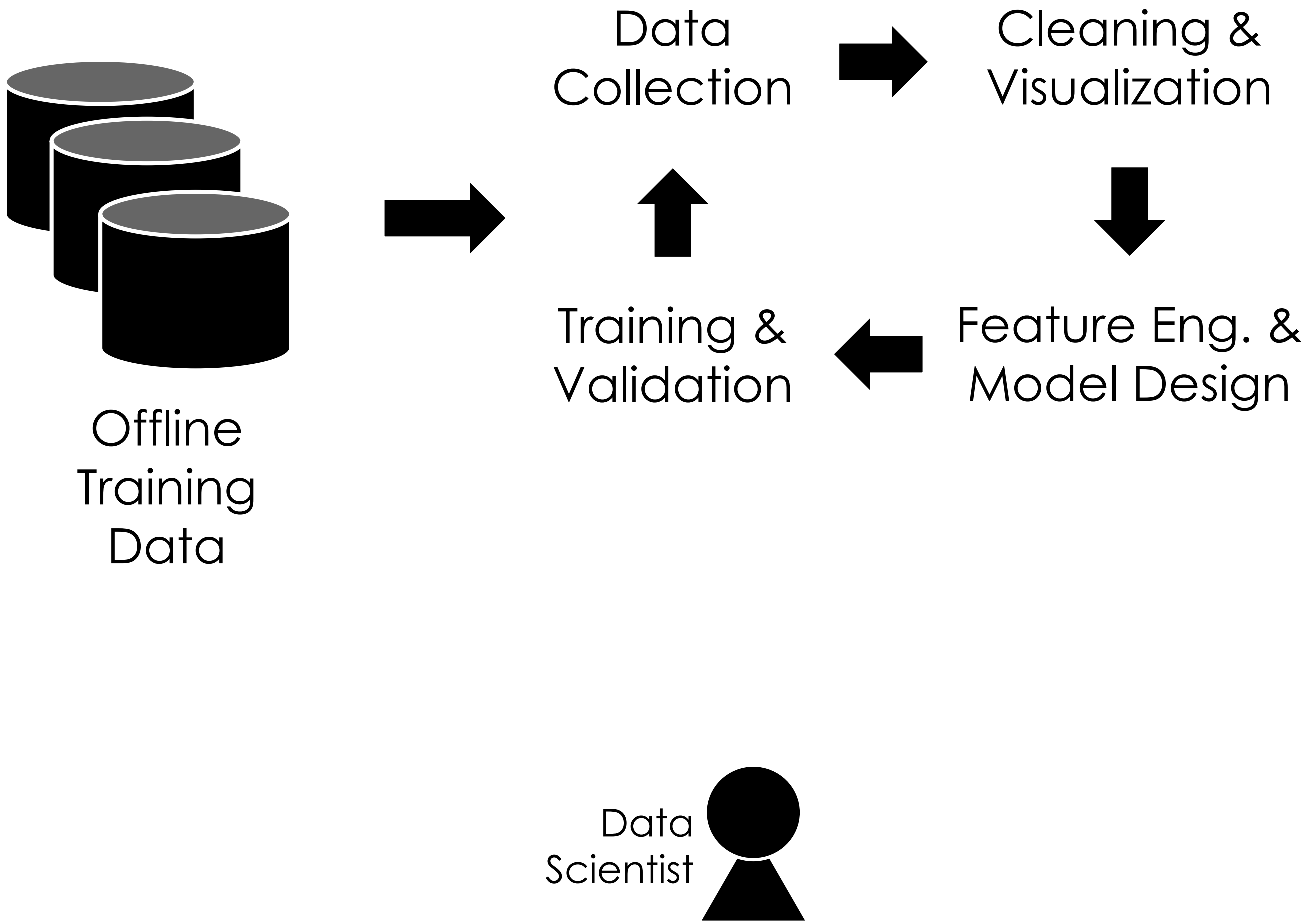
Showing sample based on the first 1000 rows. Plot over all results.

(insights …)

Training Pipelines



Model Development



Offline Training Data

Data Collection

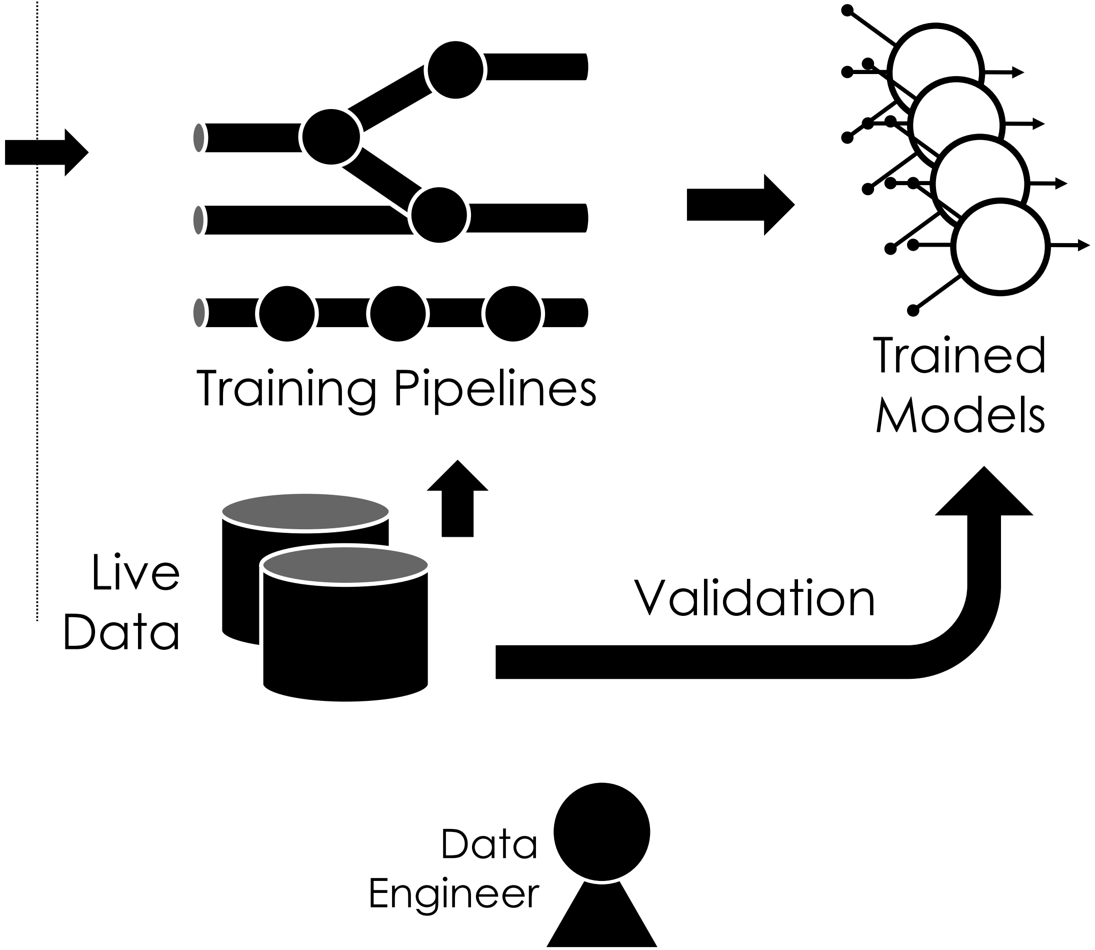
Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Data Scientist

Training



Training Pipelines

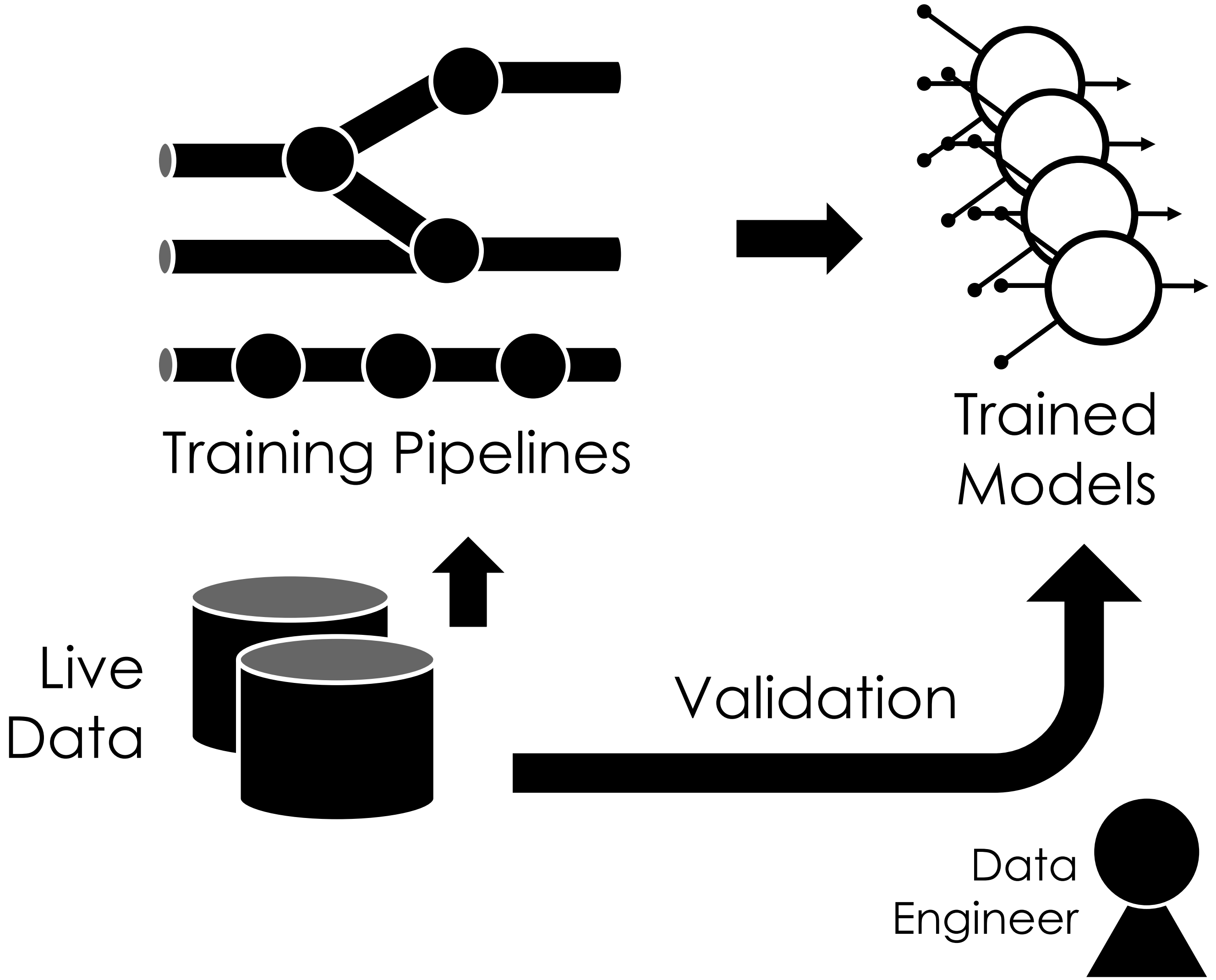
Trained Models

Live Data

Validation

Data Engineer

Training



Training models **at scale** on **live data**

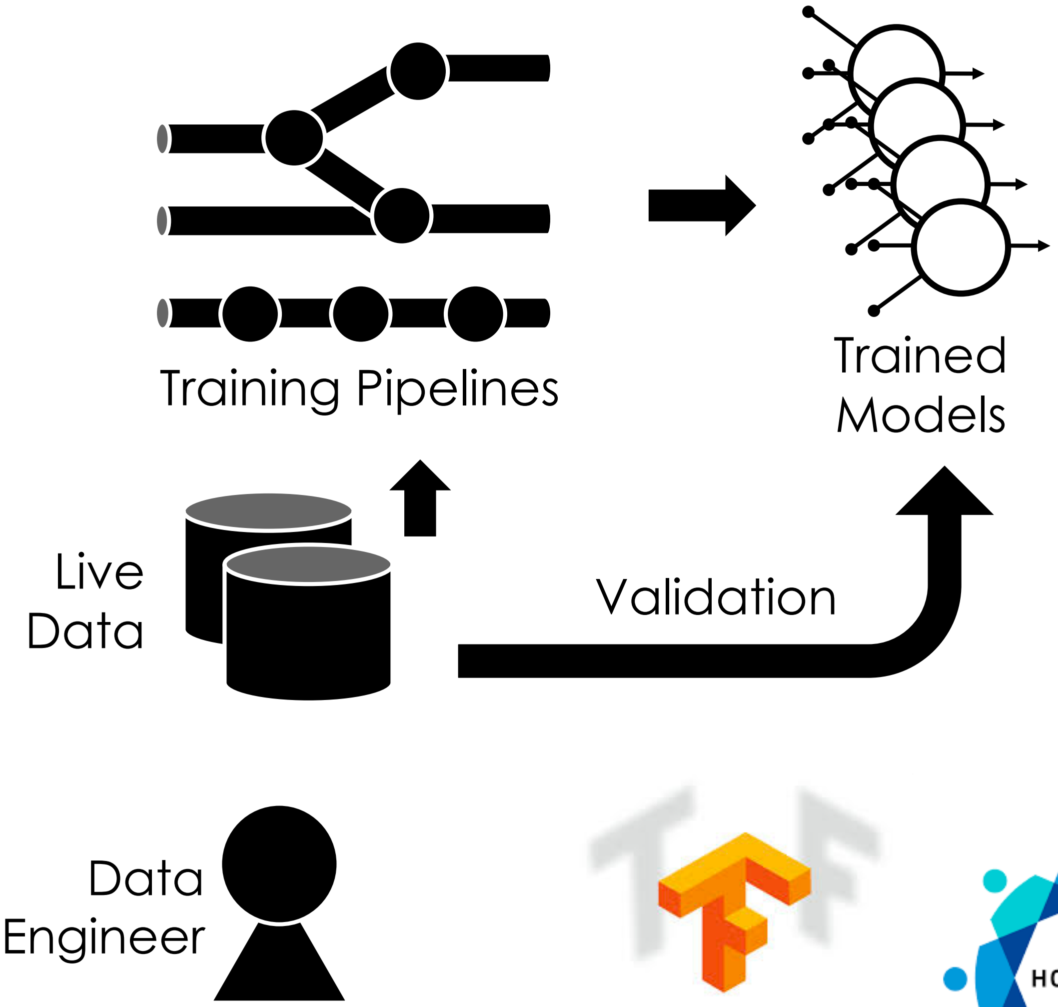
Retraining on new data

Automatically **validate** prediction accuracy

Manage model **versioning**

Requires **minimal expertise** in machine learning

Training *Technologies*



Training Pipelines

Trained Models

Live Data

Validation

Data Engineer

Workflow Management:

Apache Airflow

Luigi

Azkaban
Open-source Workflow Manager

OOZIE

Scalable Training:

PYTORCH

mxnet

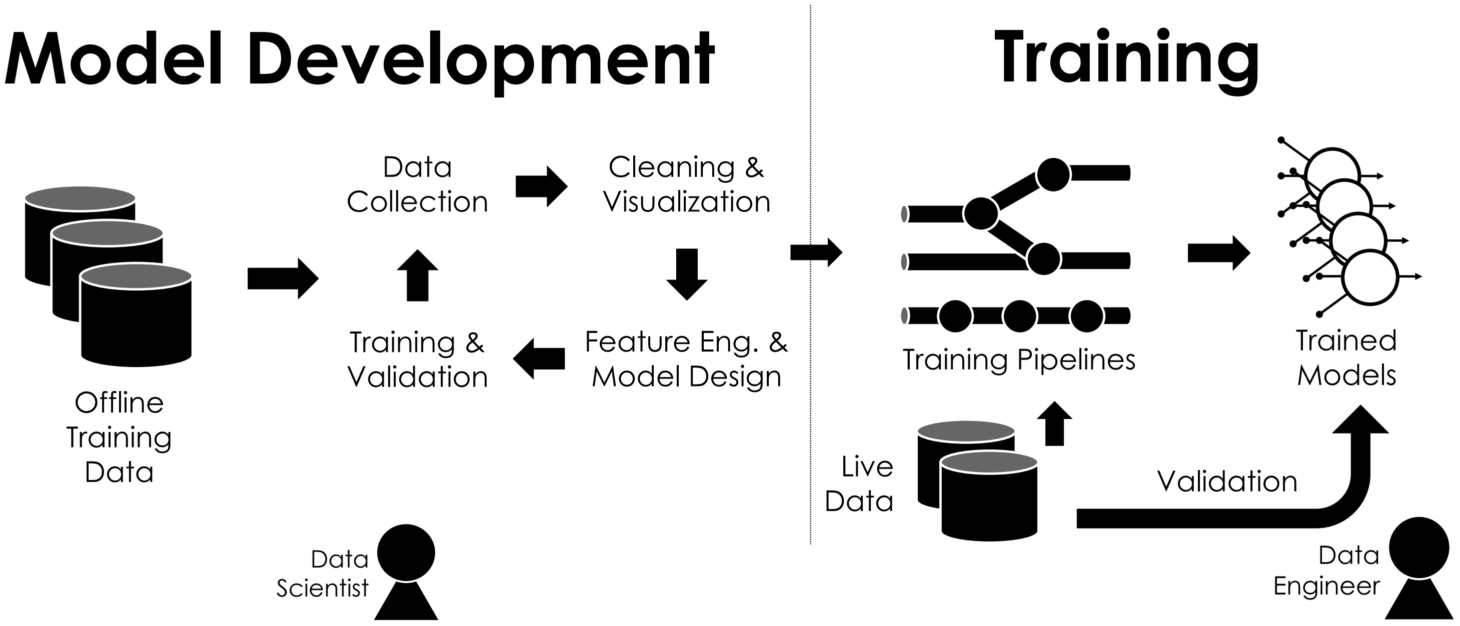
TensorFlow

HOROVOD

APACHE Spark™

dmlc XGBoost

Open Problems

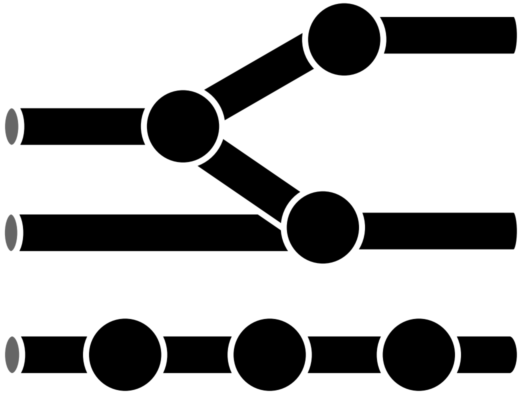


Context & Composition

Context How, What, & Who?

- **How** was the model or data created?
- **What** is the latest or best version?
- **Who** is responsible? (blame...)

Partial Solution



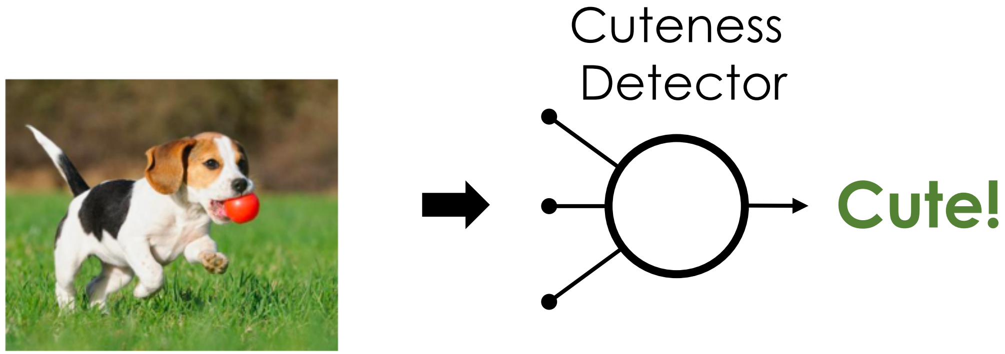
Training Pipelines

Track relationships between

1. **Code** versions ✓ git
2. **Model & Data** versions
3. **People** (versions?)

Composition

Models are being composed to solve new problems

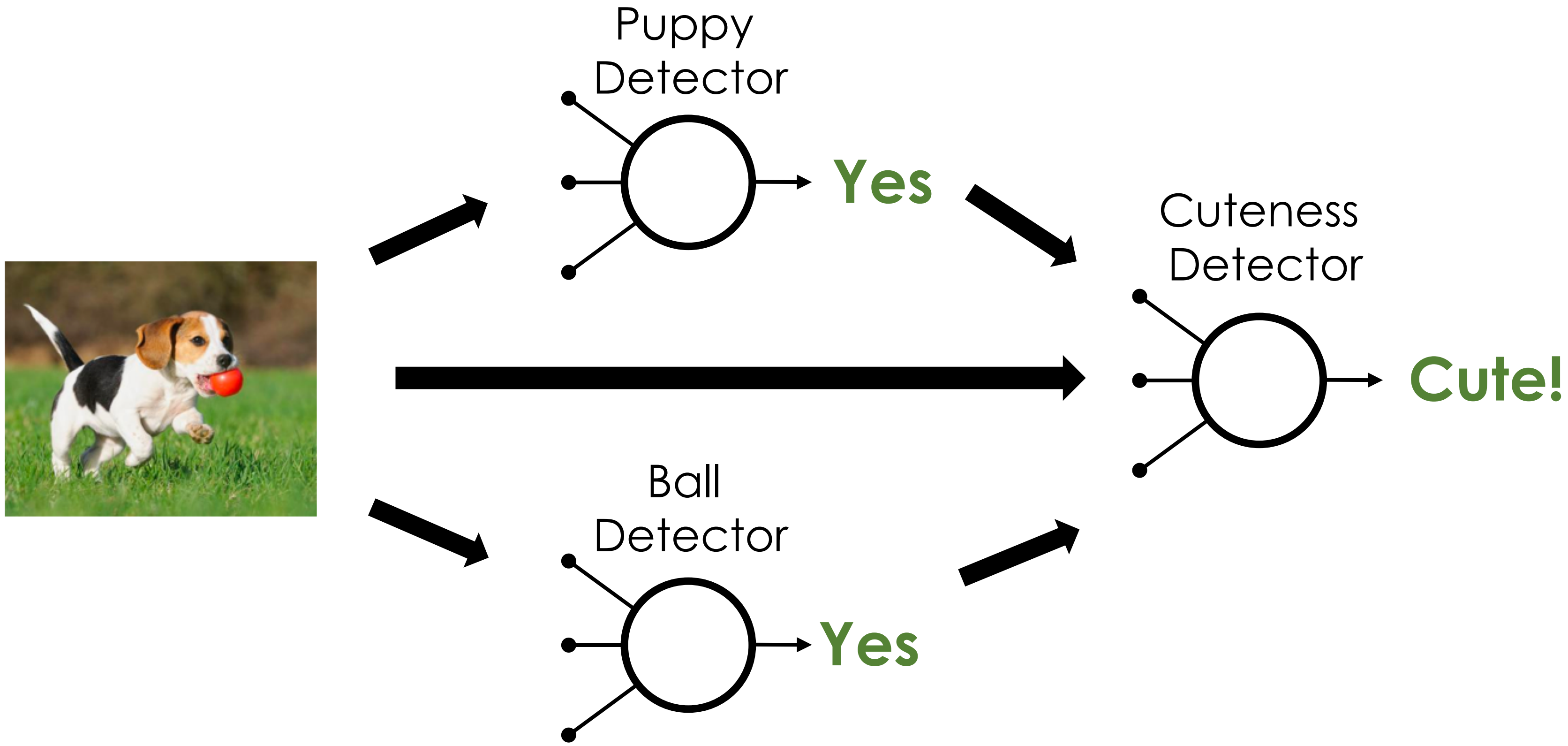


Cuteness Detector

Cute!

Composition

Models are being composed to solve new problems



Puppy Detector

Yes

Cuteness Detector

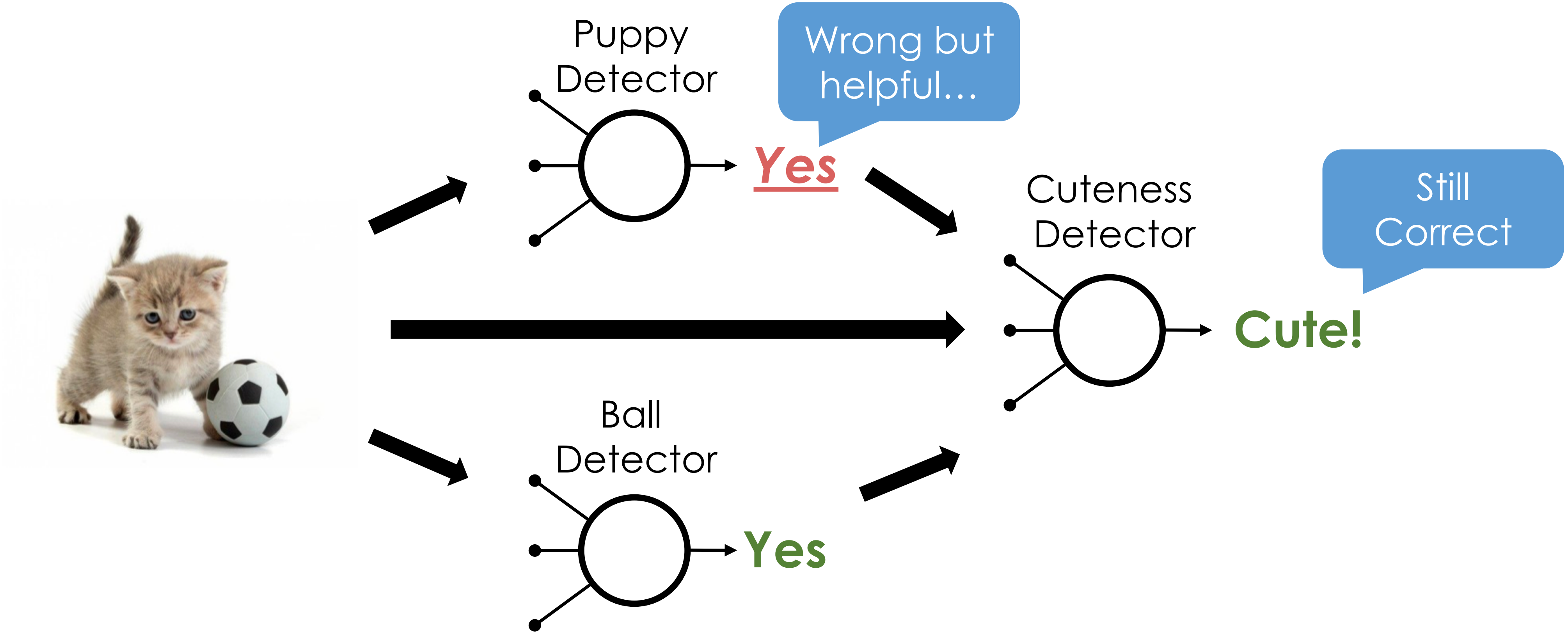
Cute!

Ball Detector

Yes

Composition

Models are being composed to solve new problems



Puppy Detector

Wrong but helpful…

Yes

Cuteness Detector

Still Correct

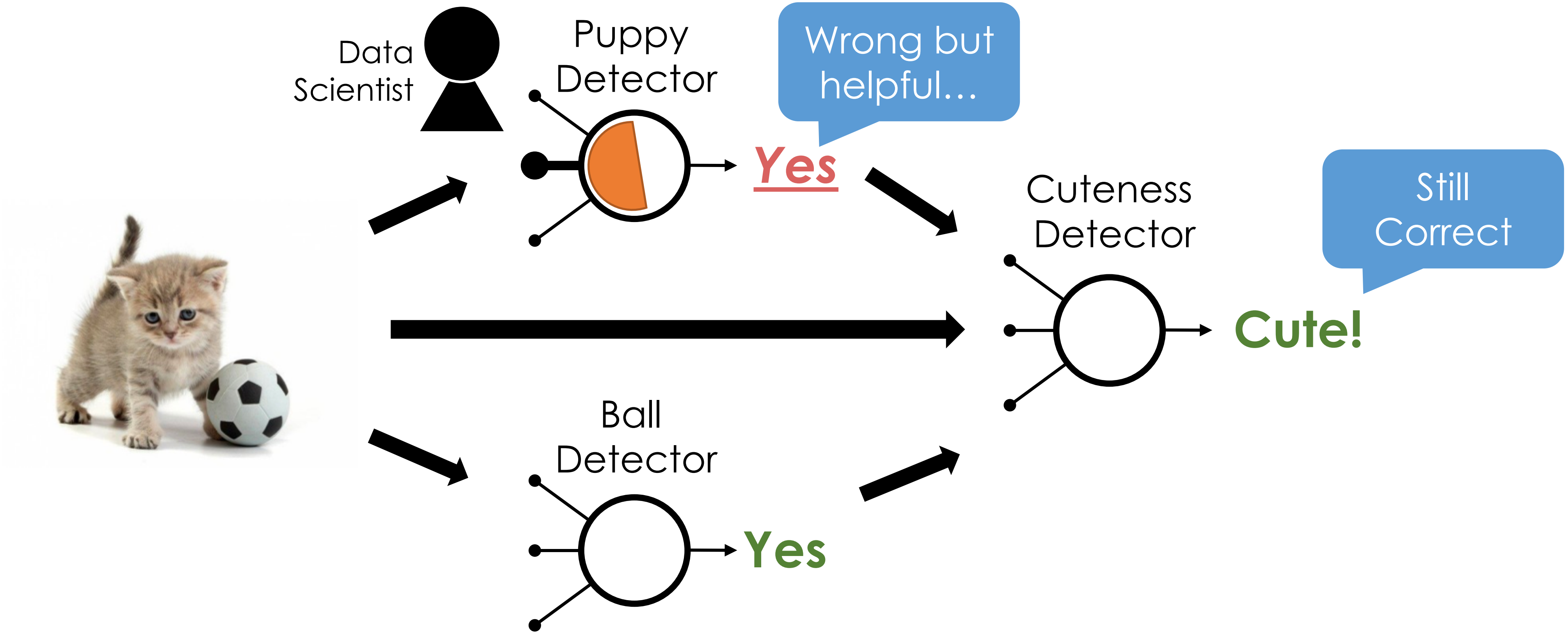
Cute!

Ball Detector

Yes

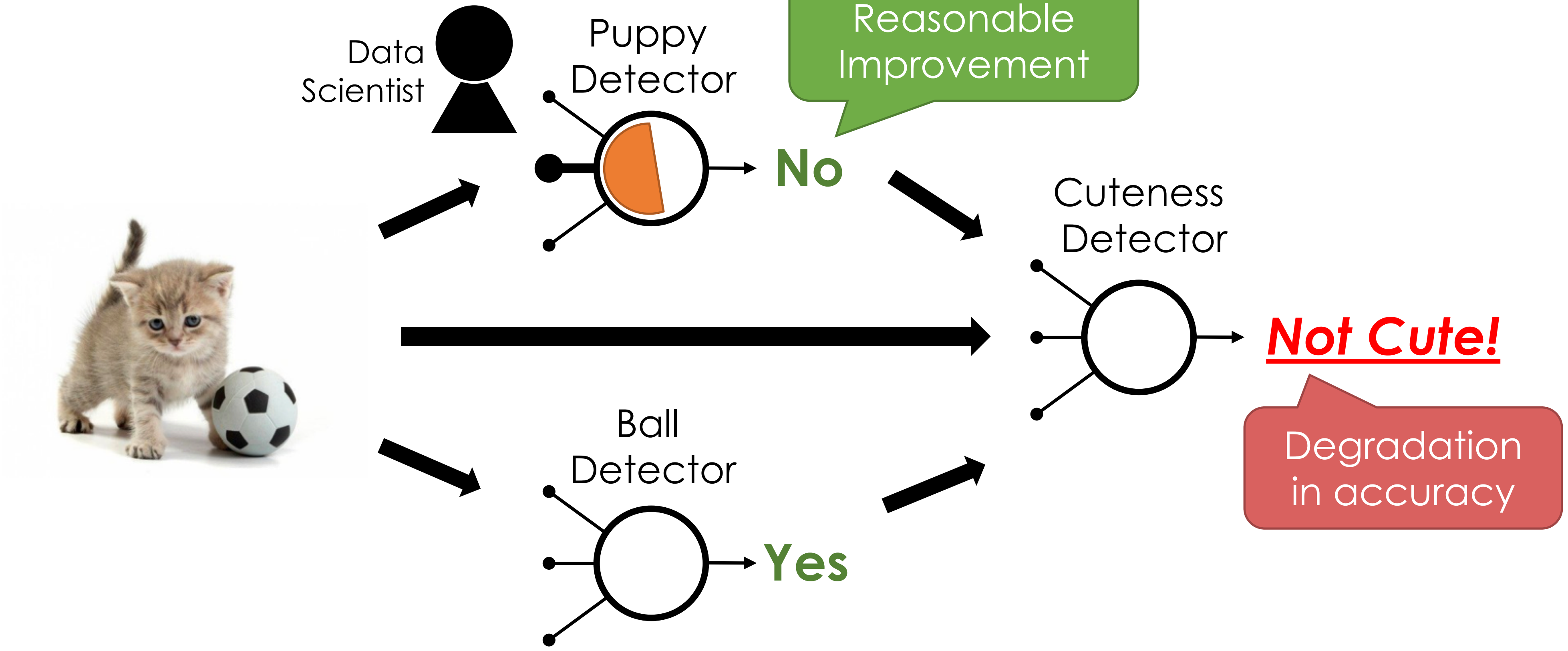
Composition

Models are being composed to solve new problems



Composition

Models are being composed to solve new problems

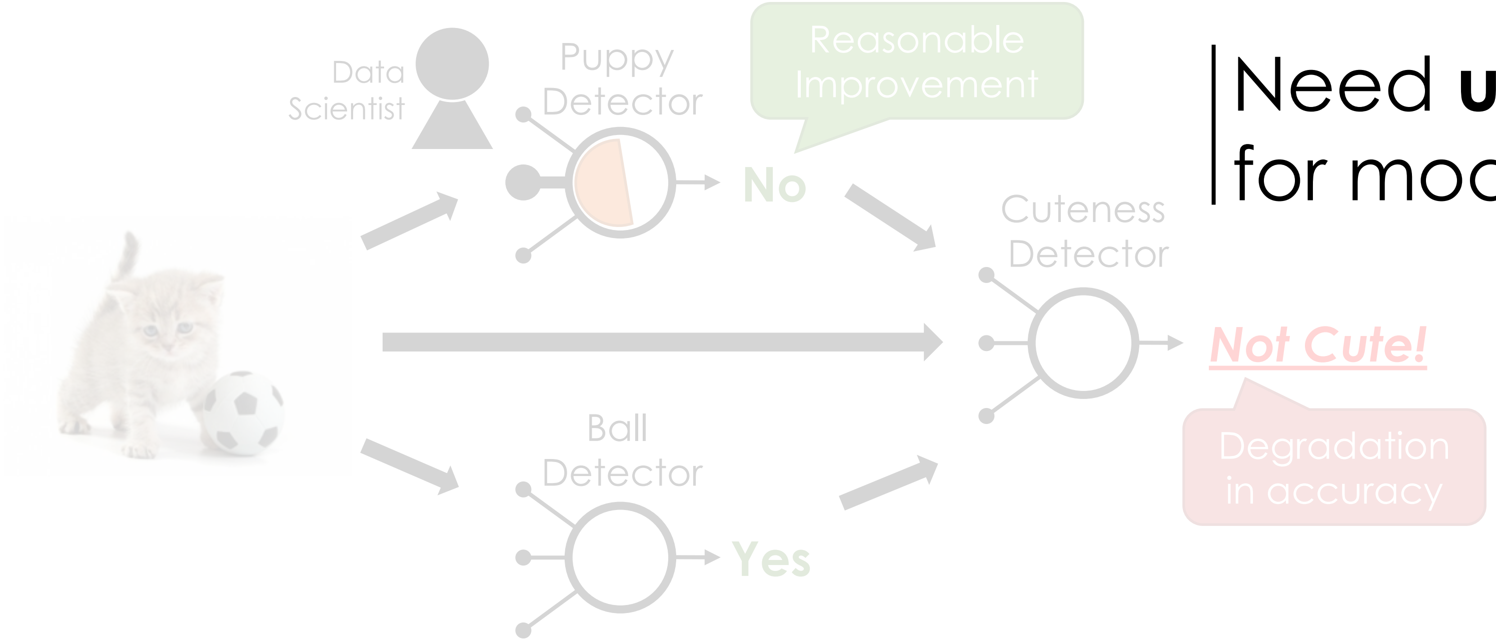


Reasonable Improvement

Degradation in accuracy

Composition

Models are being composed to solve new problems



Need to track composition and validate **end-to-end accuracy**.

Need **unit** and **integration** testing for models.

Active **Research** in the riselab UC Berkeley for Model Development and Training

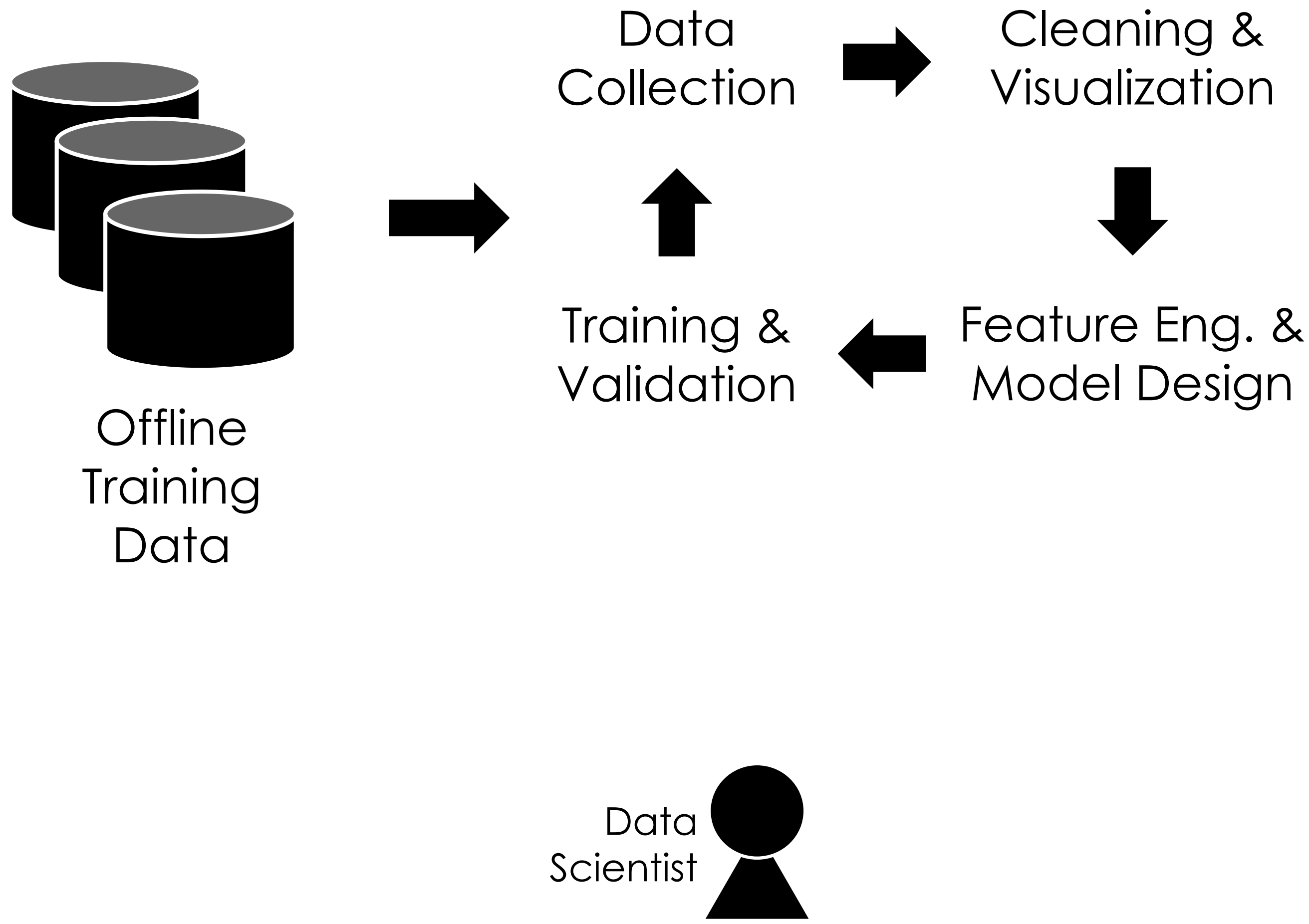
ground

A an open source
context management service
that spans multiple
data systems
http://www.ground-context.org/

flor

An **experiment management**
designed to track
data, code, and people and
address reproducibility
https://github.com/ucbrise/flor

Model Development



Offline Training Data

Data Collection

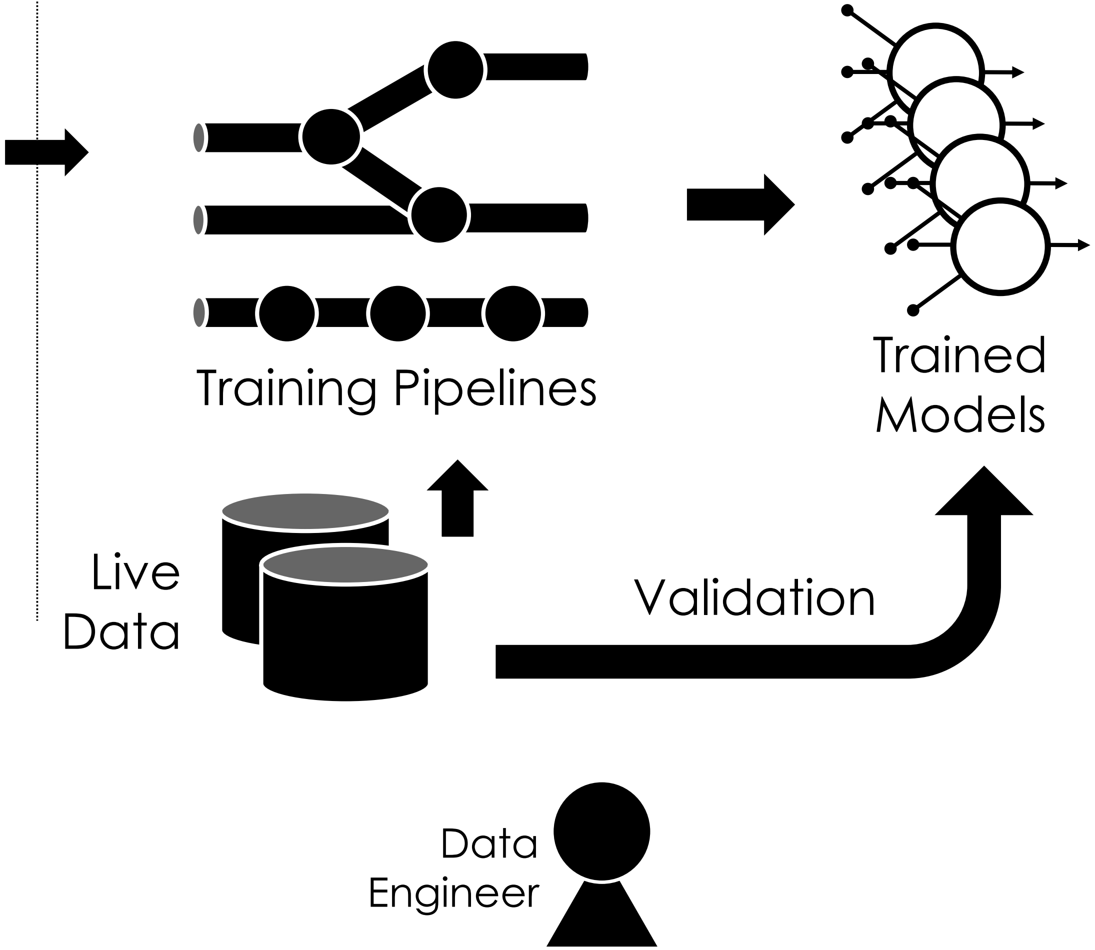
Cleaning & Visualization

Feature Eng. & Model Design

Training & Validation

Data Scientist

Training



Training Pipelines

Trained Models

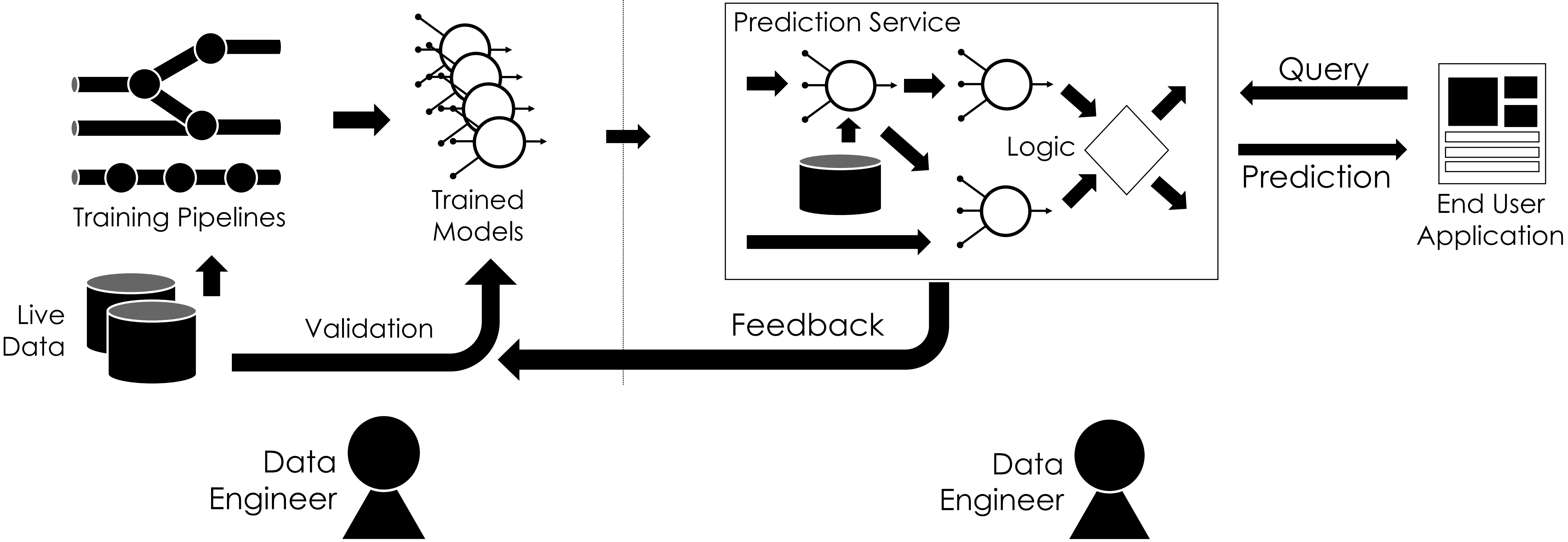
Live Data

Validation

Data Engineer

Training

Inference



Training Pipelines

Trained Models

Live Data

Validation

Prediction Service

Logic

Query

Prediction

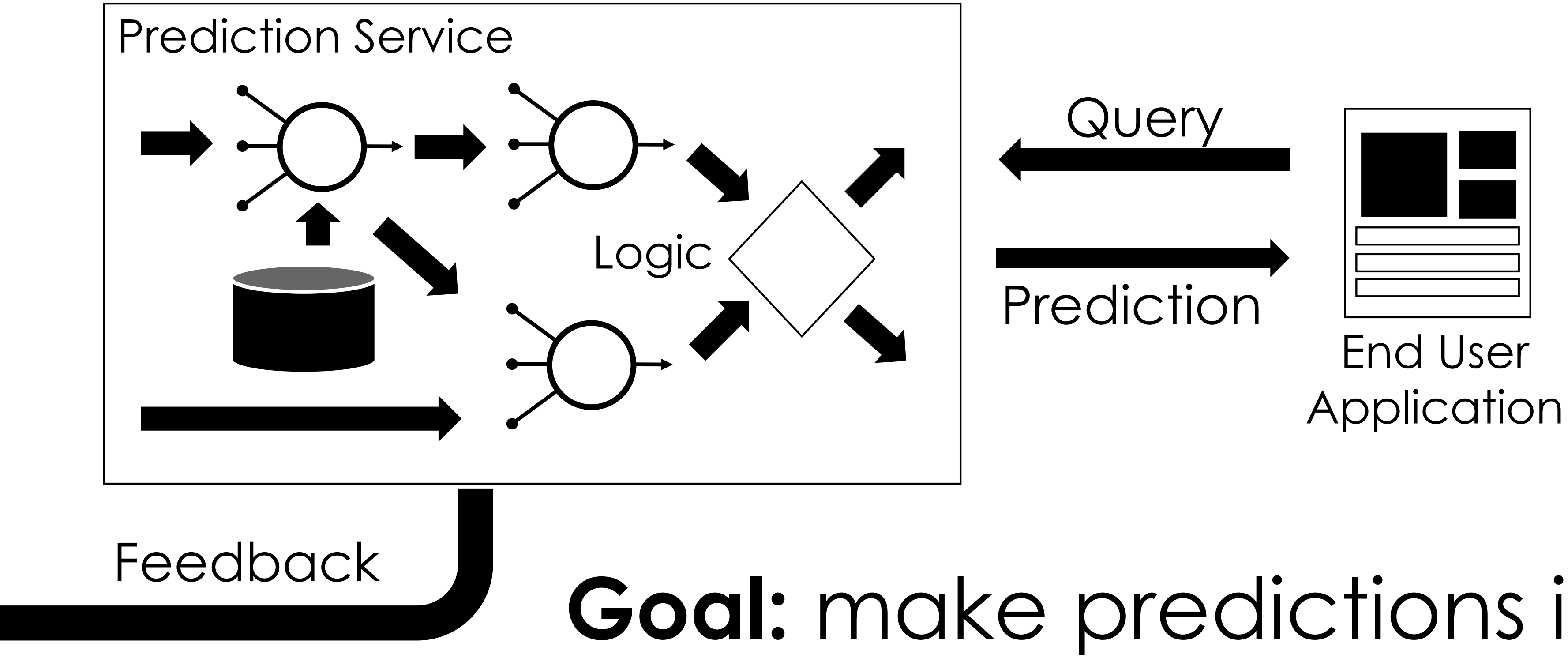
End User Application

Feedback

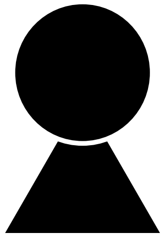
Data Engineer

Data Engineer

Inference

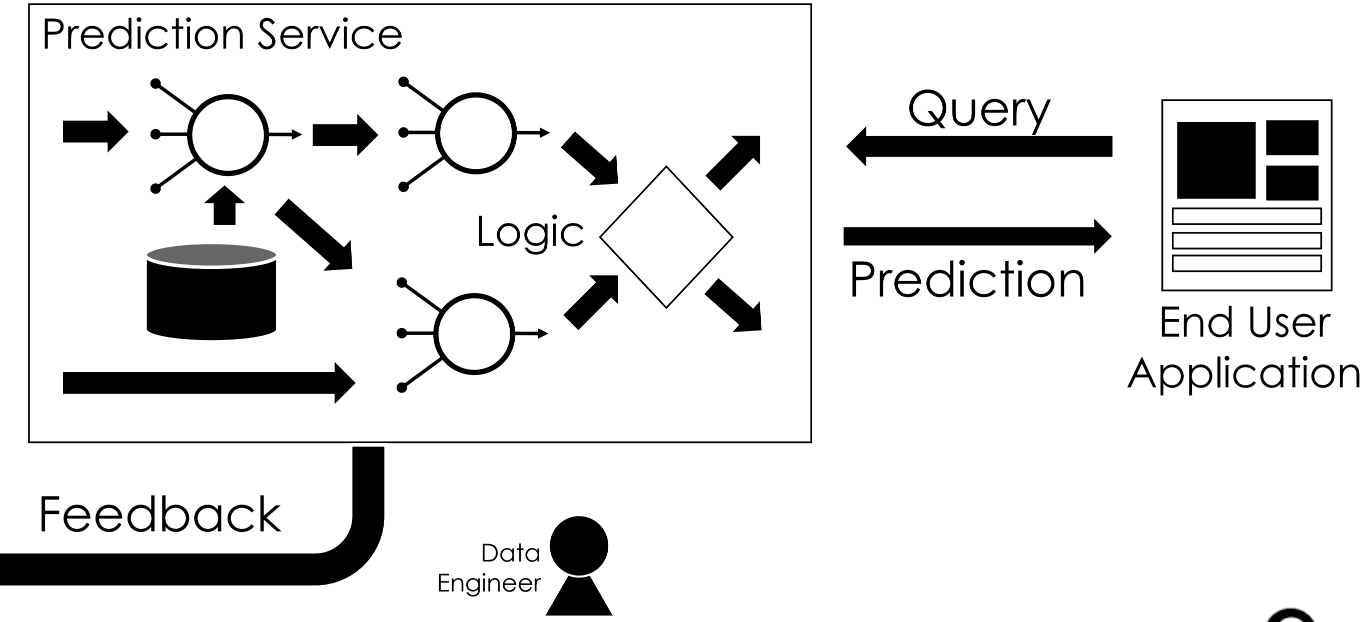


Goal: make predictions in ~10ms under heavy load



Complicated by **Deep Neural Networks**
➔ New **ML Algorithms** and **Systems**

Inference *Technologies*



Prediction Service

Logic

Query

Prediction

End User Application

Feedback

Data Engineer

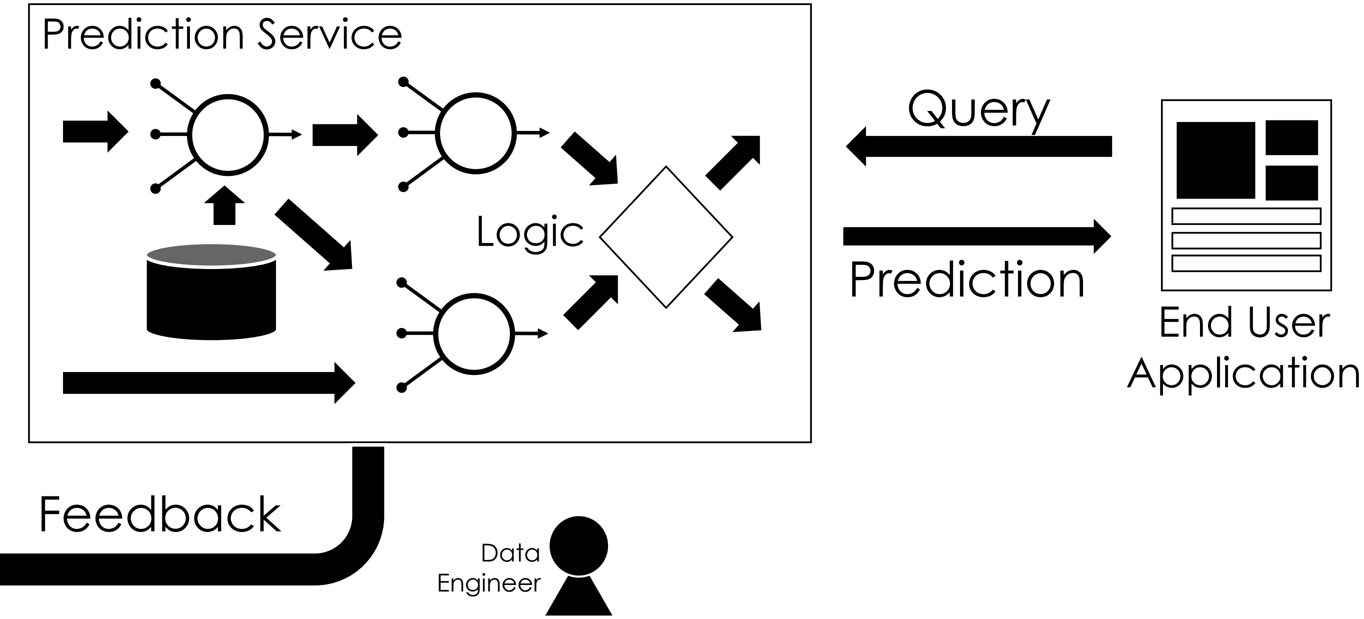
Clipper

PredictionIO

mxnet

TensorFlow

Inference *Technologies*



Prediction Service

Logic

Query

Prediction

End User Application

Feedback

Data Engineer

Our system →

Clipper

Specialized in Particular Models or Frameworks

mxnet

PredictionIO

TensorFlow

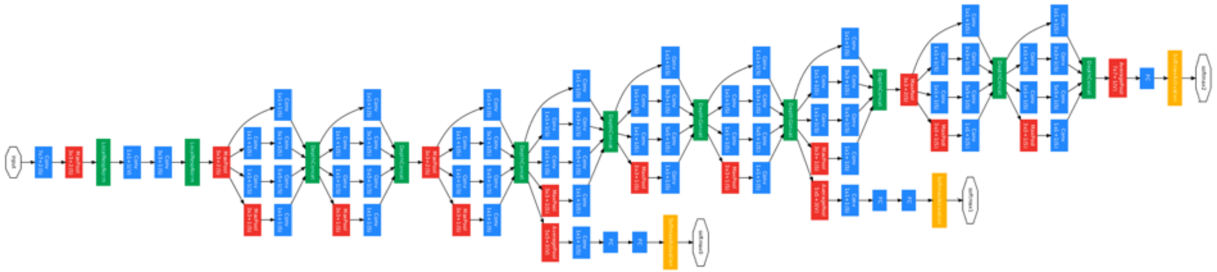
Deploying Interactive Machine Learning Applications with Clipper

Joseph E. Gonzalez
Co-director of the RISE Lab
jegonzal@cs.berkeley.edu

The remainder of this talk …

- ***Challenges*** *of prediction serving*
- ***Clipper architecture*** *overview*
- ***Open-source*** *system effort*

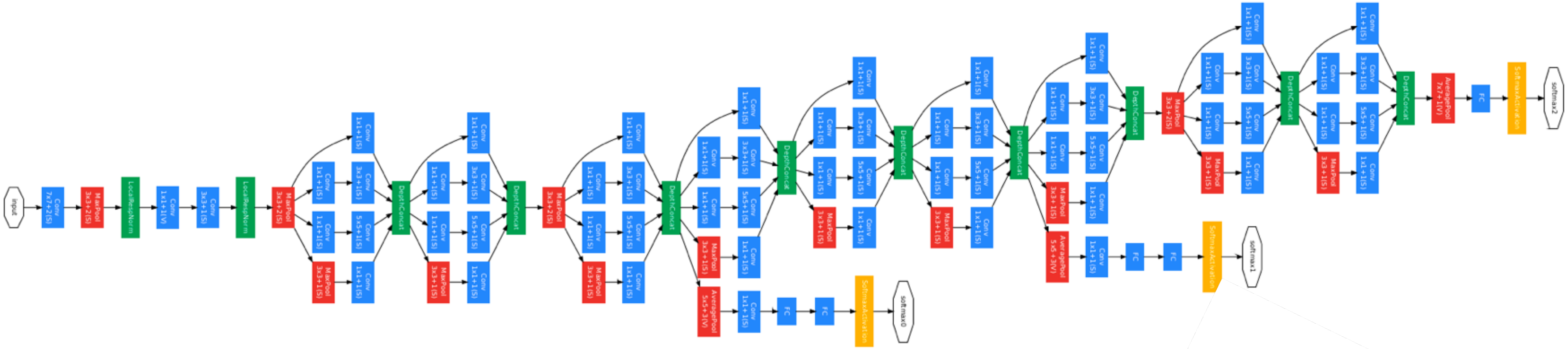
Prediction-Serving Challenges



Support low-latency, high-throughput serving workloads

Large and growing ecosystem of ML models and frameworks

Support low-latency, high-throughput serving workloads



Models getting more complex

- 10s of GFLOPs [1]

Deployed on critical path

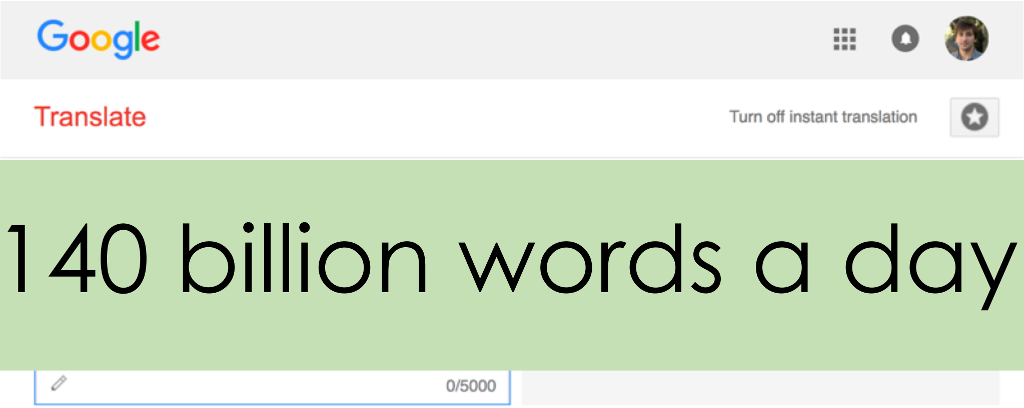
- Maintain SLOs under heavy load

Using specialized hardware for predictions

[1] *Deep Residual Learning for Image Recognition.* He et al. CVPR 2015.

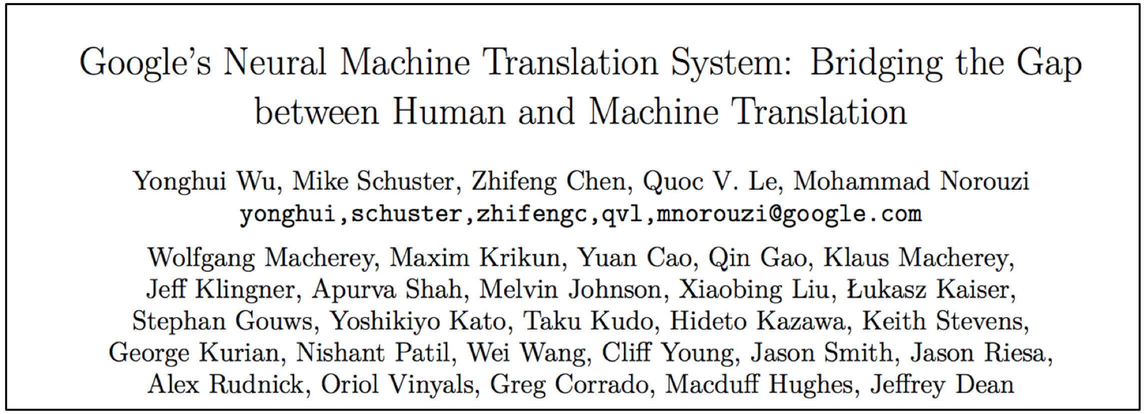
Google Translate

Serving



140 billion words a day[1]

82,000 GPUs
running 24/7



Google's Neural Machine Translation System: Bridging the Gap
between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
`yonghui,schuster,zhifengc,qvl,mnorouzi@google.com`

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey,
Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser,
Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens,
George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa,
Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

"If each of the ***world's Android phones*** *used the new Google voice search for just* ***three minutes a day****, these engineers realized, the company would* ***need twice as many data centers.****"*
– Wired

Designed New Hardware!
Tensor Processing Unit (TPU)

[1] https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

Building Application Specific Systems

YouTube

CONVIVA®

bing

OK GOOGLE

NETFLIX

Google Ads

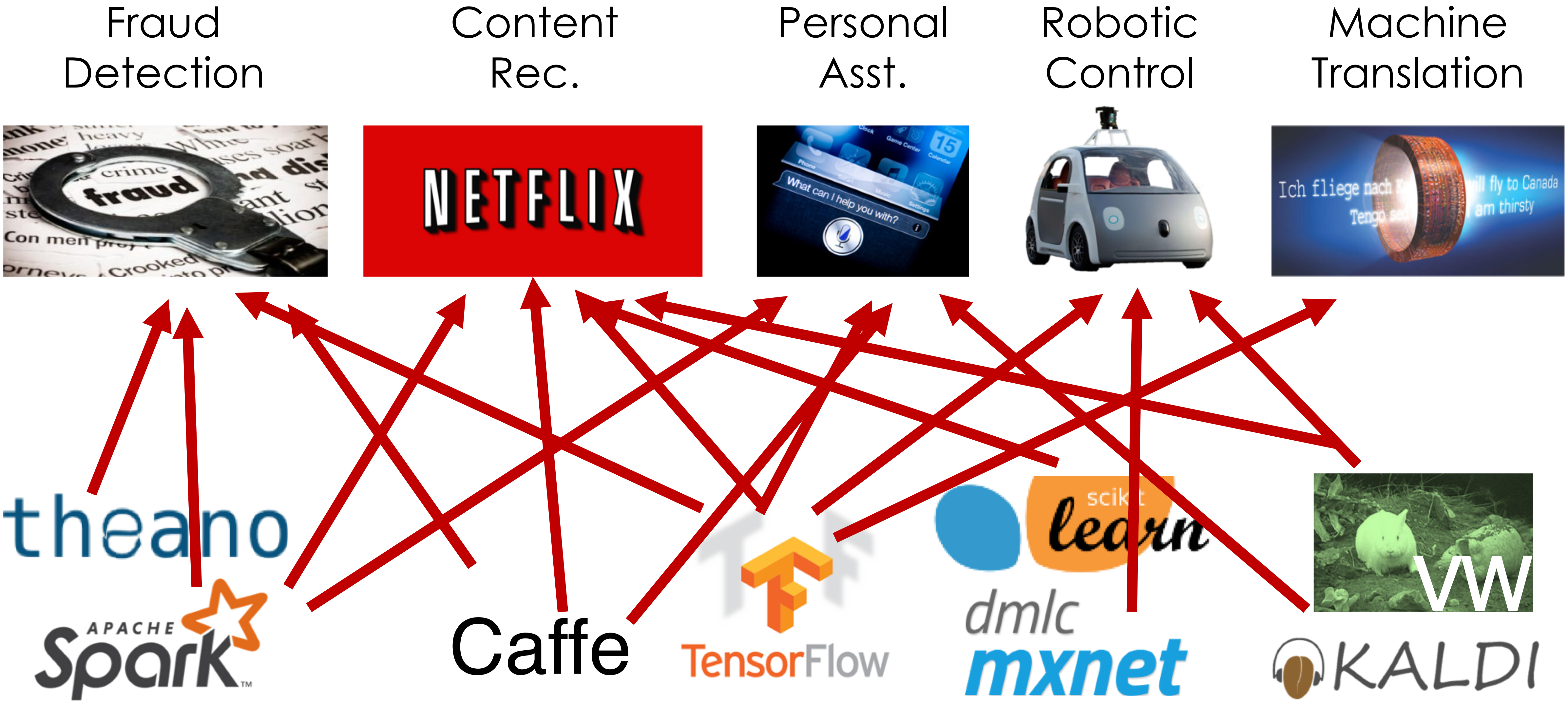
TensorFlow

Building Application Specific Systems

Problems:

- **Expensive** to **build** and **maintain**
  - Require **ML** and **systems expertise**
- **Tightly-coupled** model and application
  - Difficult to change or update application
- Only supports **single ML framework**

Growing ecosystem of ML Frameworks

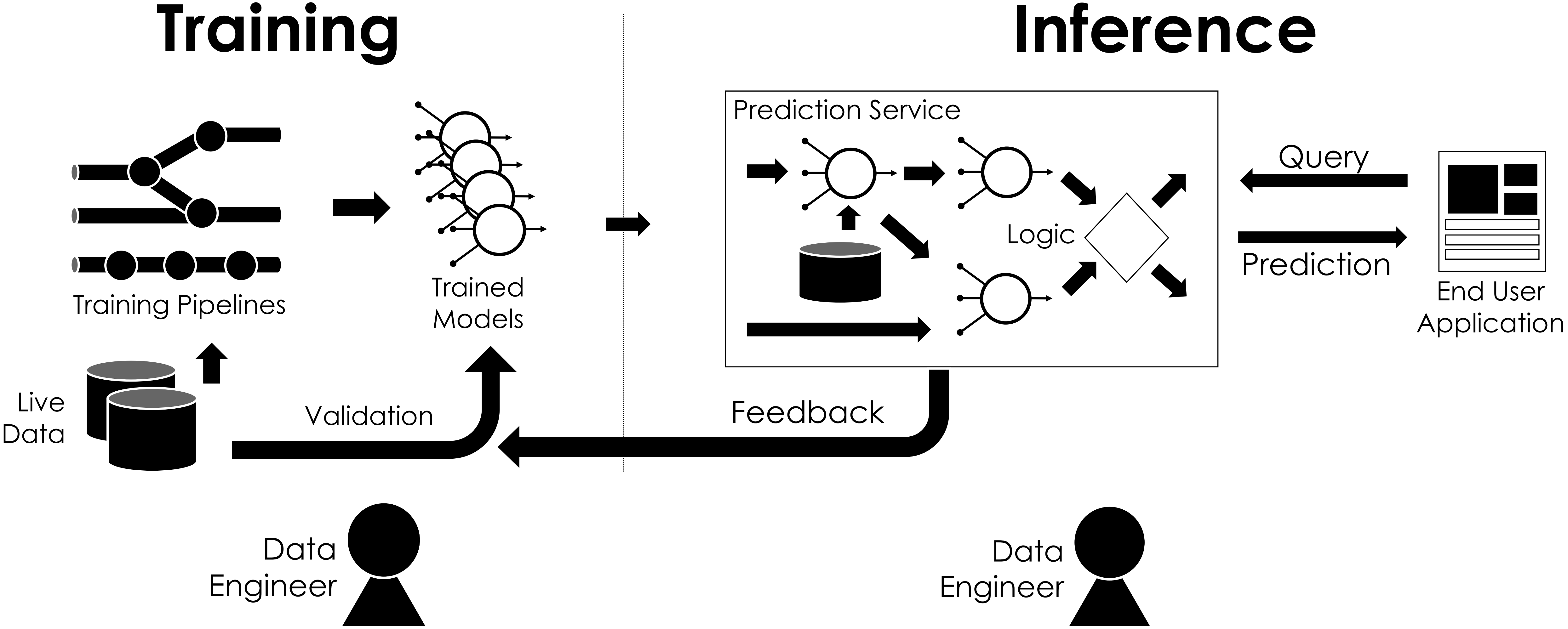


Building & maintaining **separate serving systems** for each framework is **expensive**!

Solution

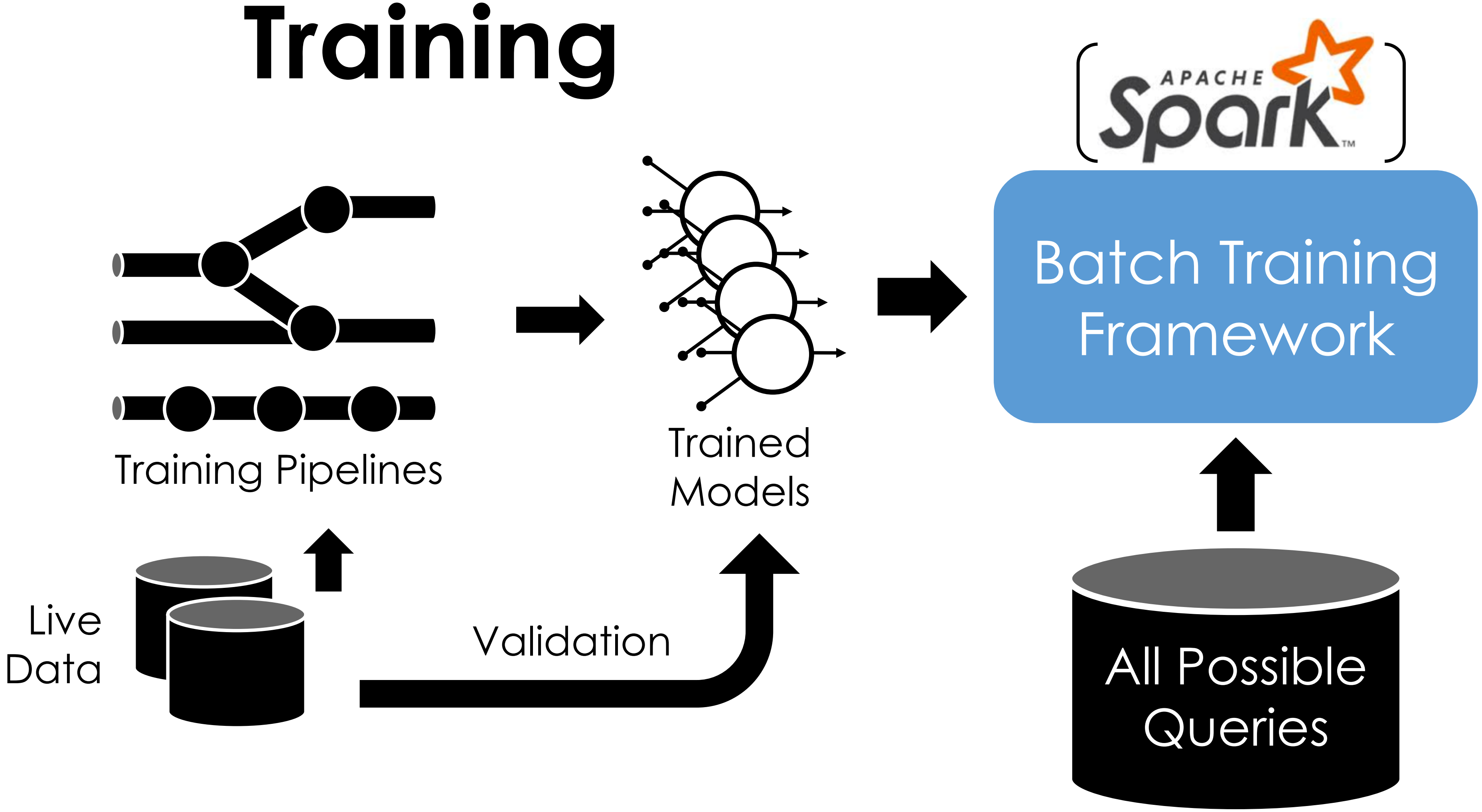
Pre-materialize predictions into a low latency **Data Management System**

Pre-materialized Predictions



Pre-materialized Predictions

Training



Training Pipelines

Trained Models

APACHE Spark

Batch Training Framework

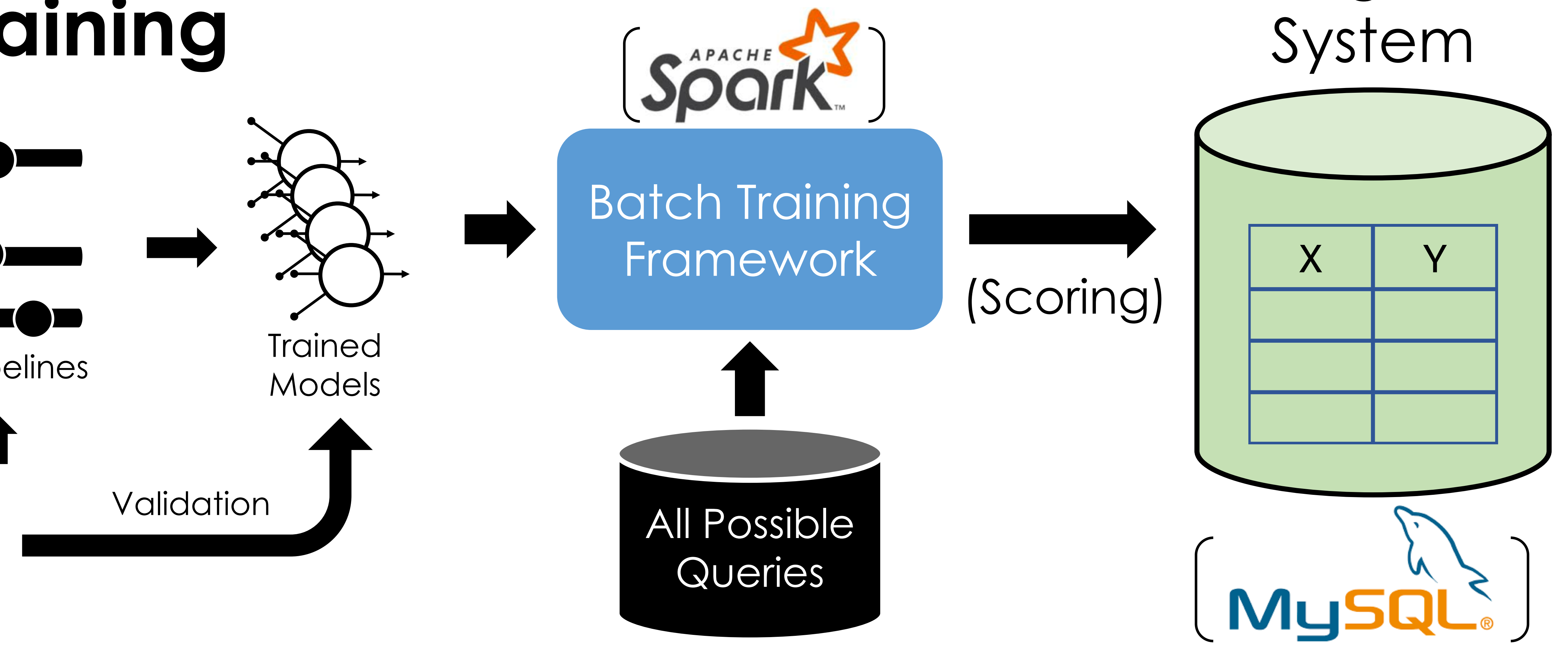
Live Data

Validation

All Possible Queries

Pre-materialized Predictions

aining



Data Management System

elines

Trained Models

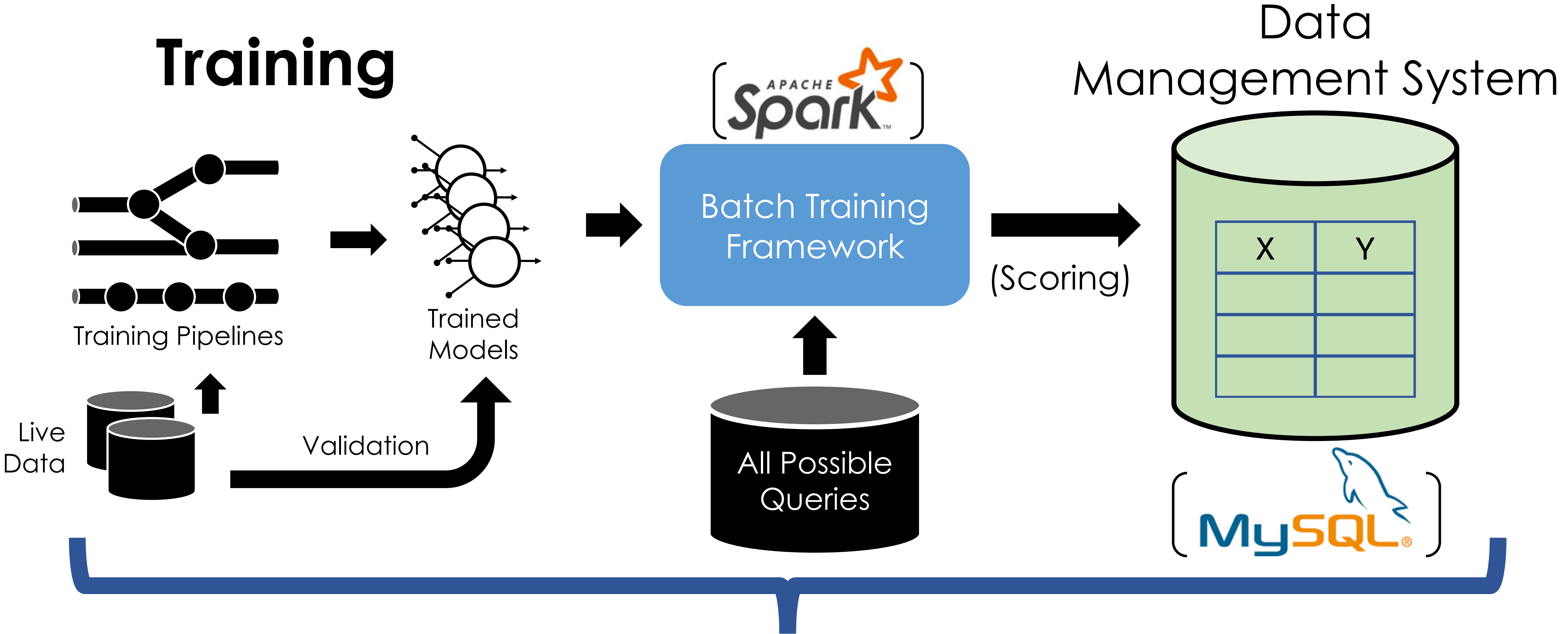
Validation

Batch Training Framework

All Possible Queries

(Scoring)

Pre-materialized Predictions



Training

Training Pipelines

Trained Models

Live Data

Validation

Apache Spark

Batch Training Framework

All Possible Queries

(Scoring)

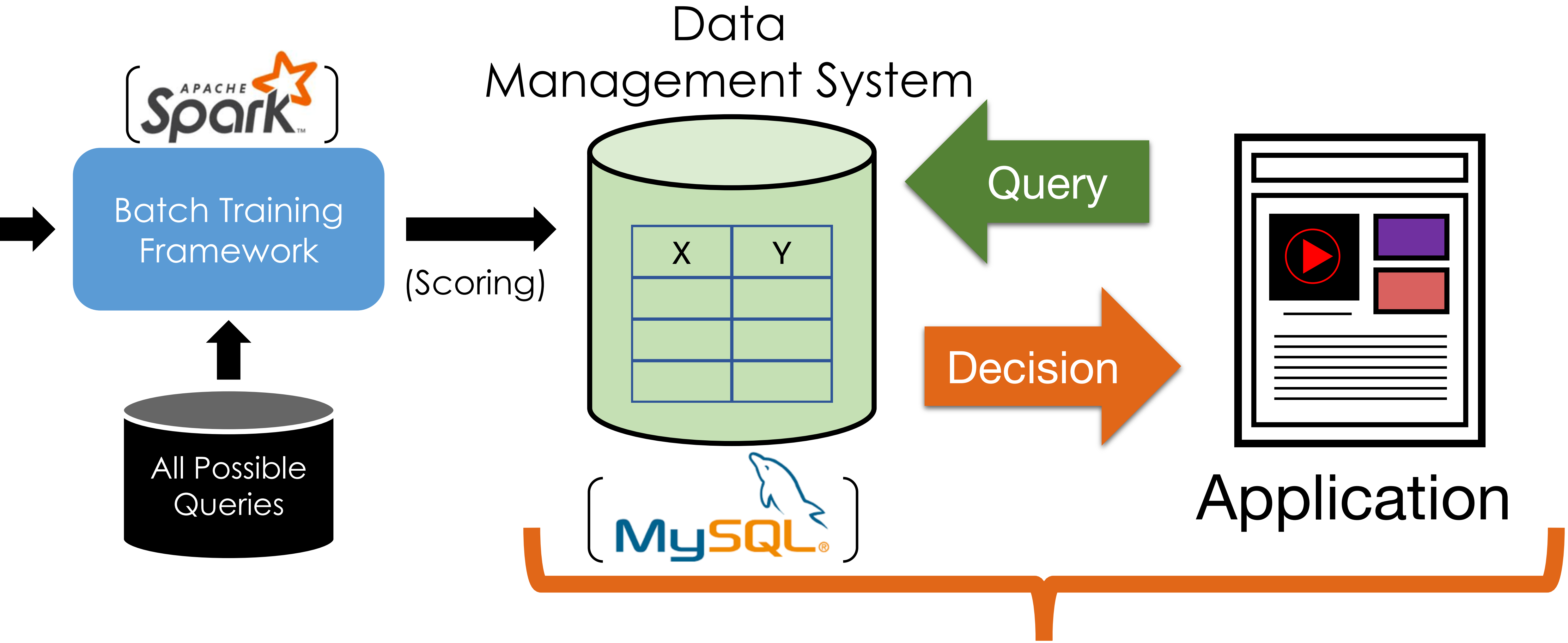
Data Management System

X	Y

MySQL

Standard Data Eng. Tools

Serving **Pre-materialized** Predictions



Data Management System

Batch Training Framework

(Scoring)

All Possible Queries

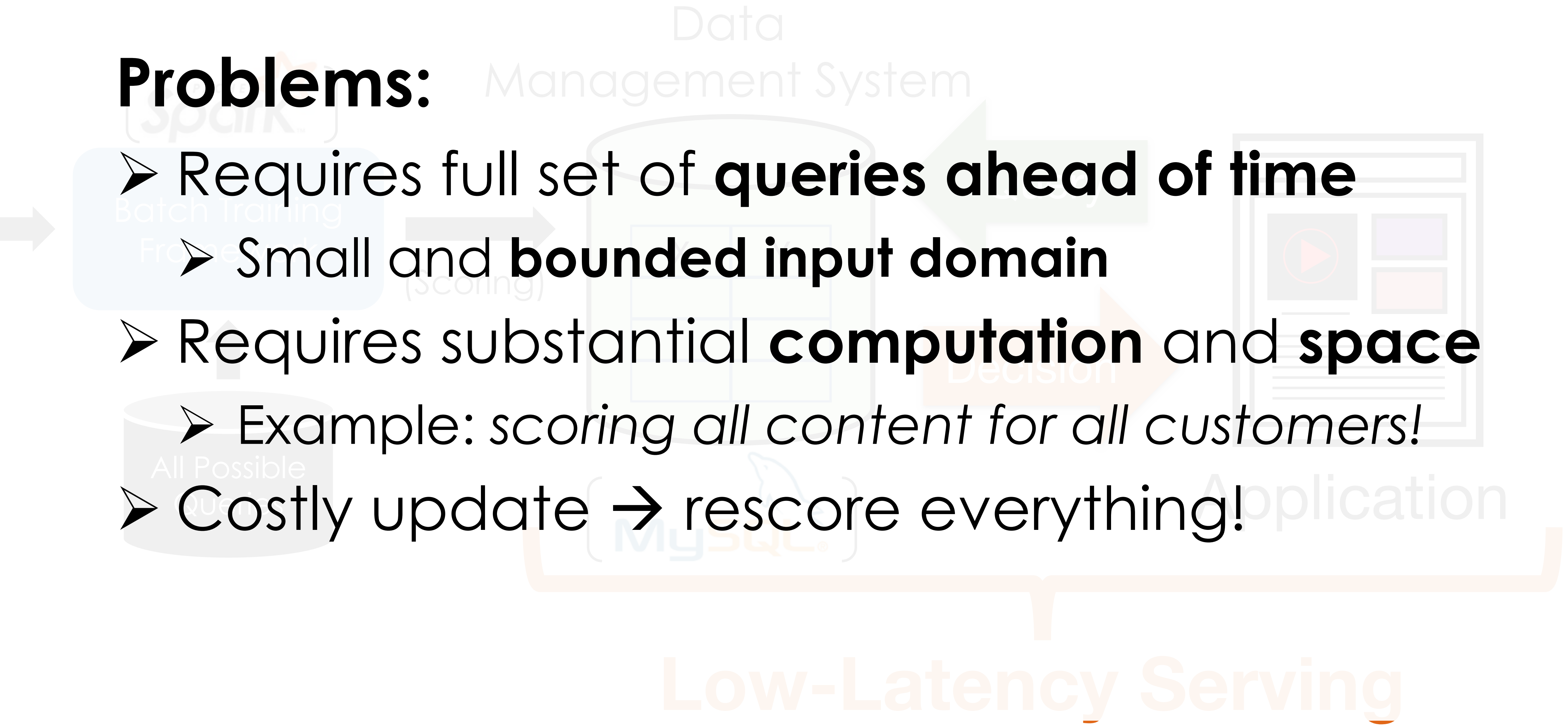
Query

Decision

Application

Low-Latency Serving

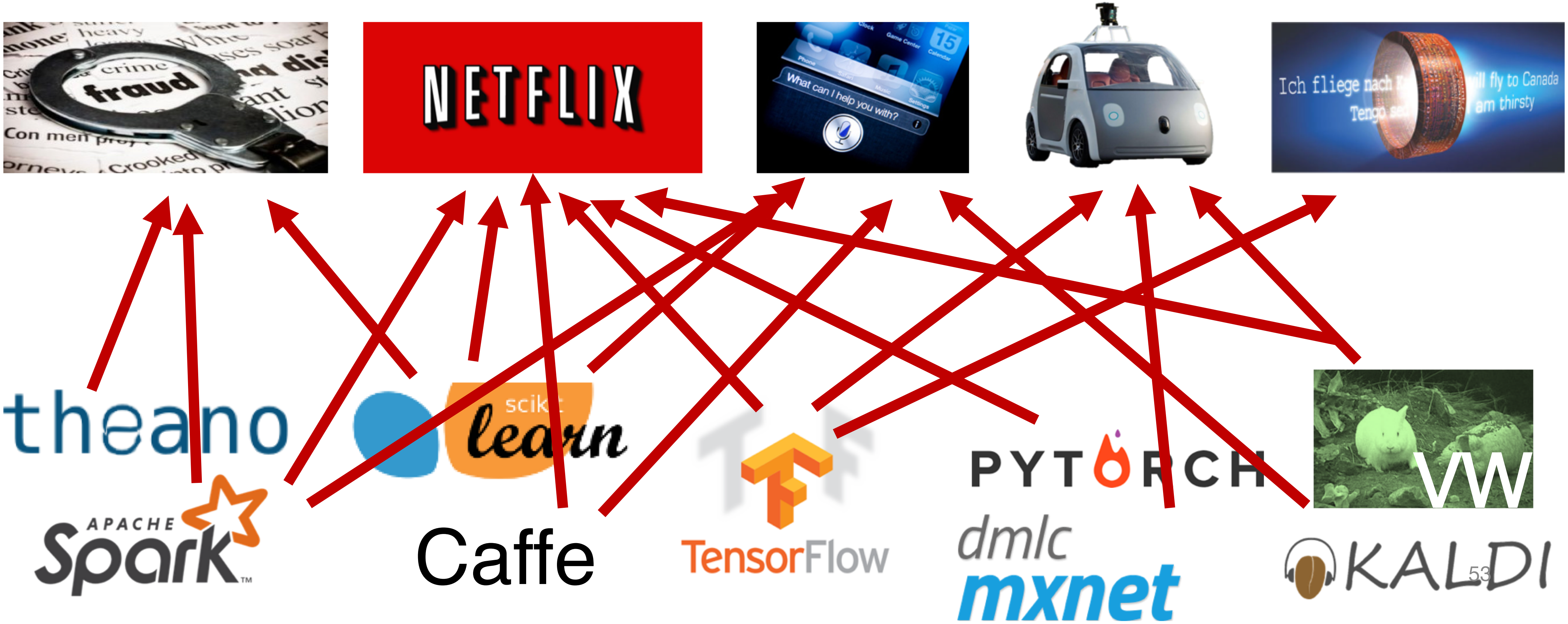
Serving **Pre-materialized** Predictions



Problems:

- Requires full set of **queries ahead of time**
  - Small and **bounded input domain**
- Requires substantial **computation** and **space**
  - Example: *scoring all content for all customers!*
- Costly update → rescore everything!

Wide range of **application** and **frameworks**



NETFLIX

What can I help you with?

Middle layer for prediction serving.

Common Abstraction

System Optimizations

---

theano

scikit learn

TensorFlow

PYTORCH

VW

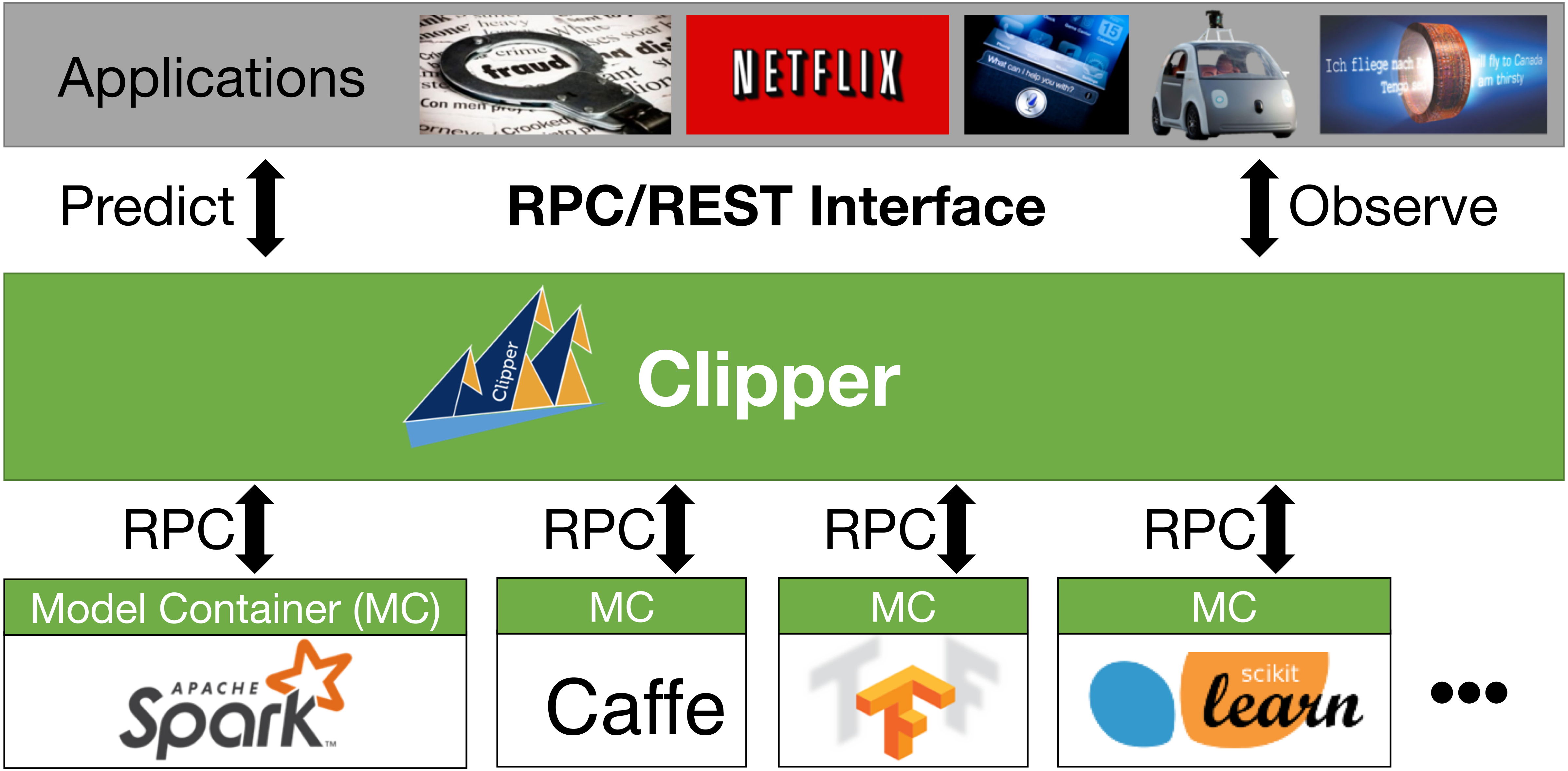
APACHE Spark

Caffe

dmlc mxnet

KALDI

Clipper Decouples Applications and Models



Applications

Predict

RPC/REST Interface

Observe

Clipper

RPC

RPC

RPC

RPC

Model Container (MC)

MC

MC

MC

Caffe

…

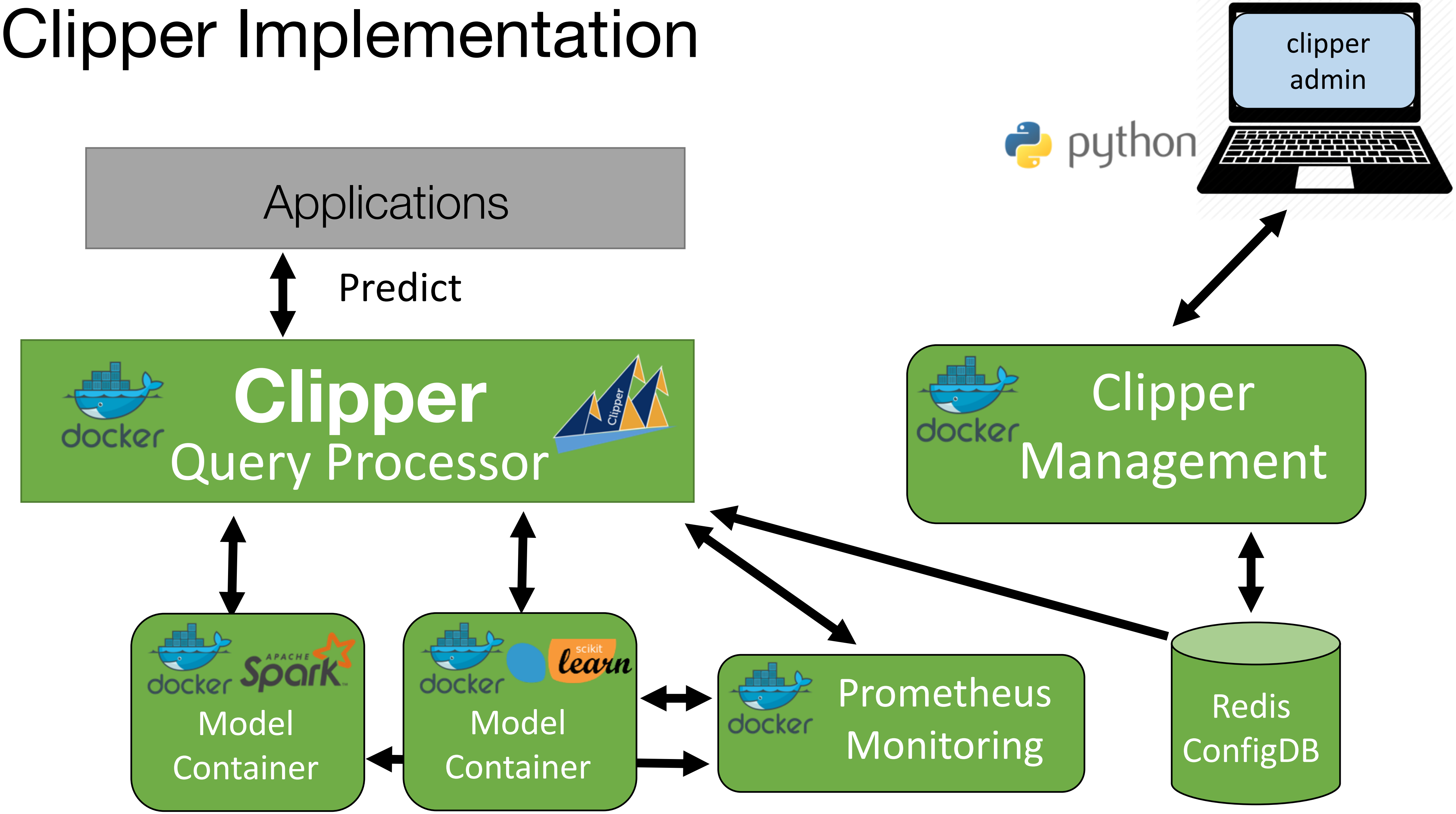
Predict ↕ **RPC/REST Interface** ↕ Observe

Clipper

- **Core system:** 10K lines of C++ and 8K lines of Python
- Open Source **(Apache License)** – http://clipper.ai
- Designed to support **production level** query traffic
  - Deliver low + predictable latency
  - Research goal: **study reality …**

RPC ↕ RPC ↕ RPC ↕ RPC ↕

Clipper Implementation



clipper admin

python

Applications

Predict

Clipper

Query Processor

Clipper Management

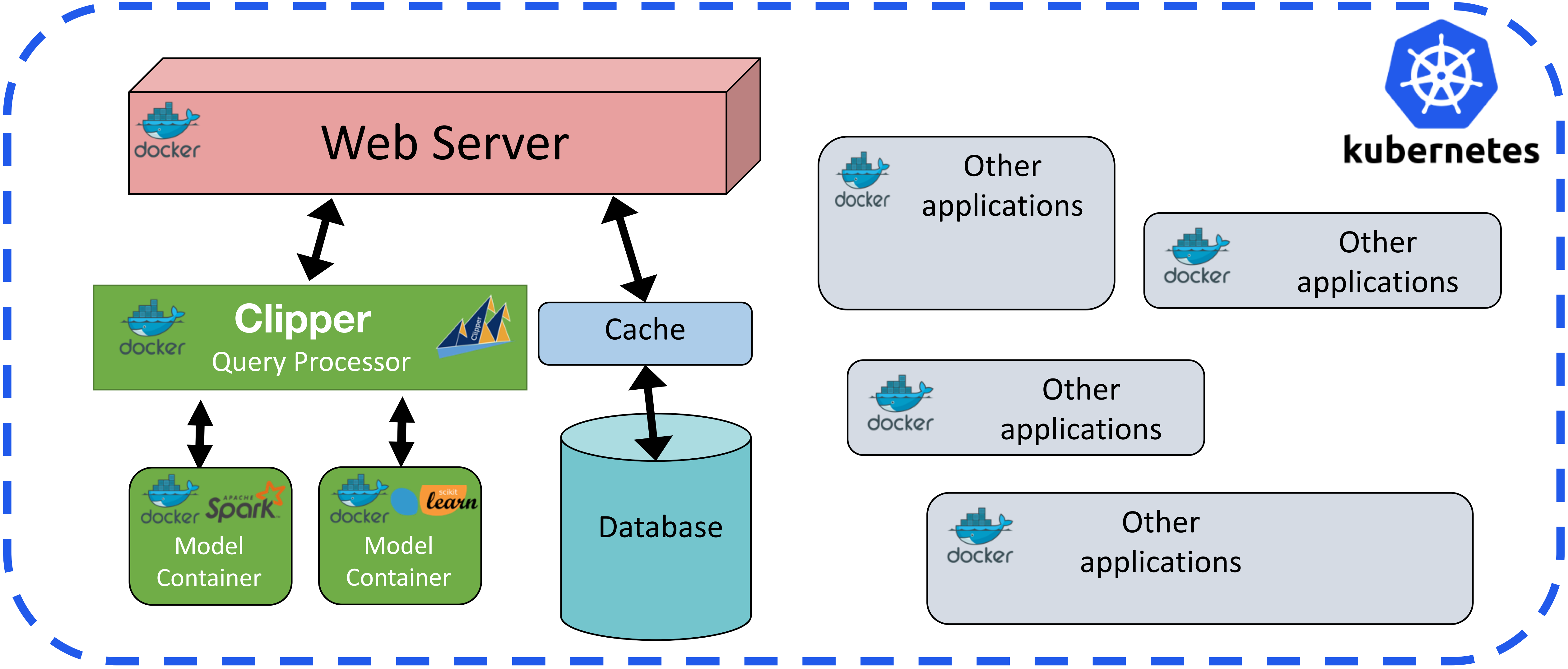
Model Container

Model Container

Prometheus Monitoring

Redis ConfigDB

Run alongside other applications with **Kubernetes**



Getting Started with Clipper

Tutorials at http://clipper.ai

Docker images available **on DockerHub**

Clipper admin is distributed as **pip package**:

```
pip install clipper_admin
```

Get up and running **without compiling**

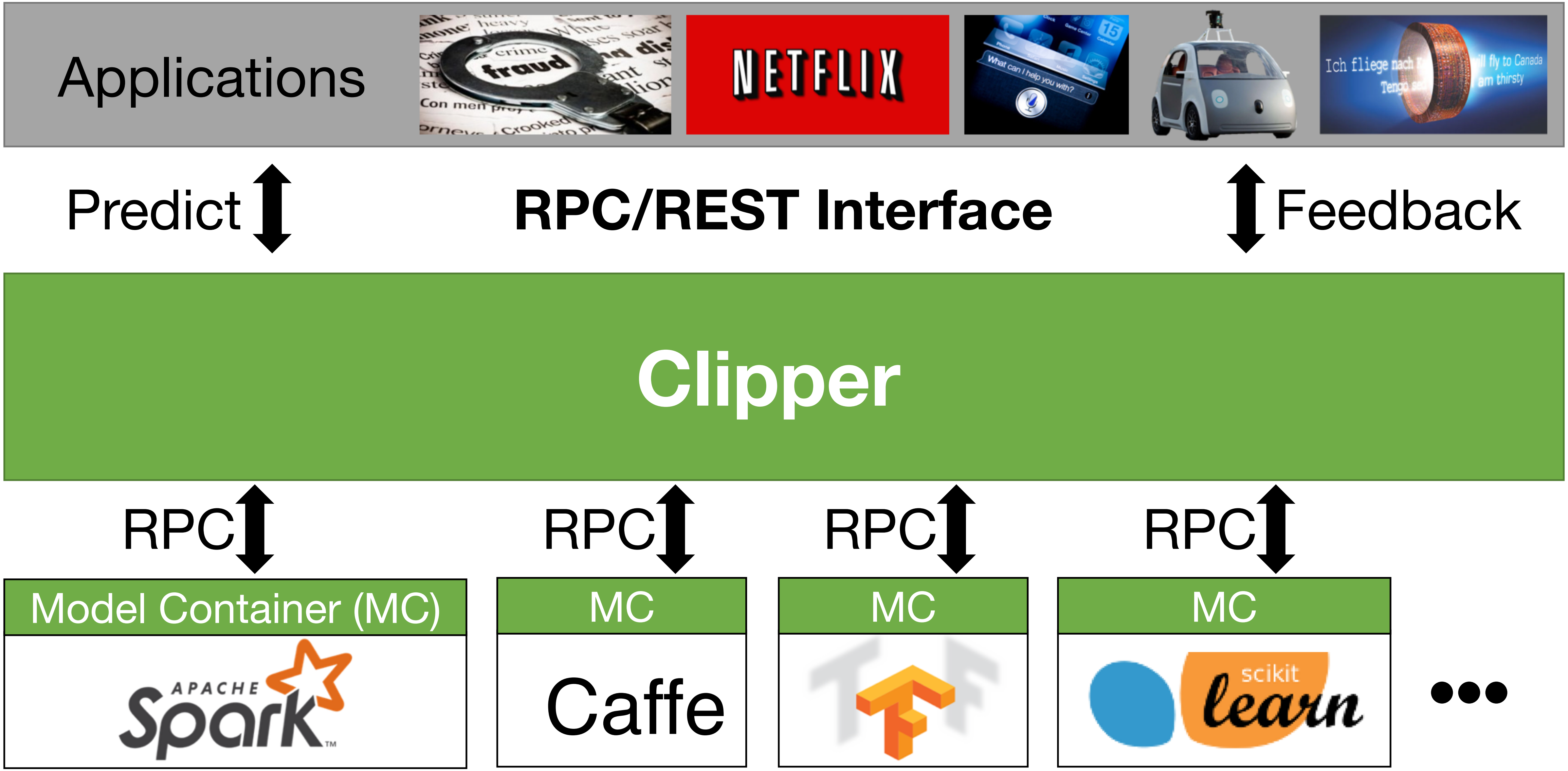
Clipper Design Innovations

Containerized frameworks: unified abstraction and framework level isolation and scaling

Cross-framework caching and batching: optimize throughput and latency

Cross-framework model composition: improved accuracy through ensembles and bandits

Clipper Architecture



Applications

Predict

RPC/REST Interface

Feedback

Clipper

RPC

RPC

RPC

RPC

Model Container (MC)

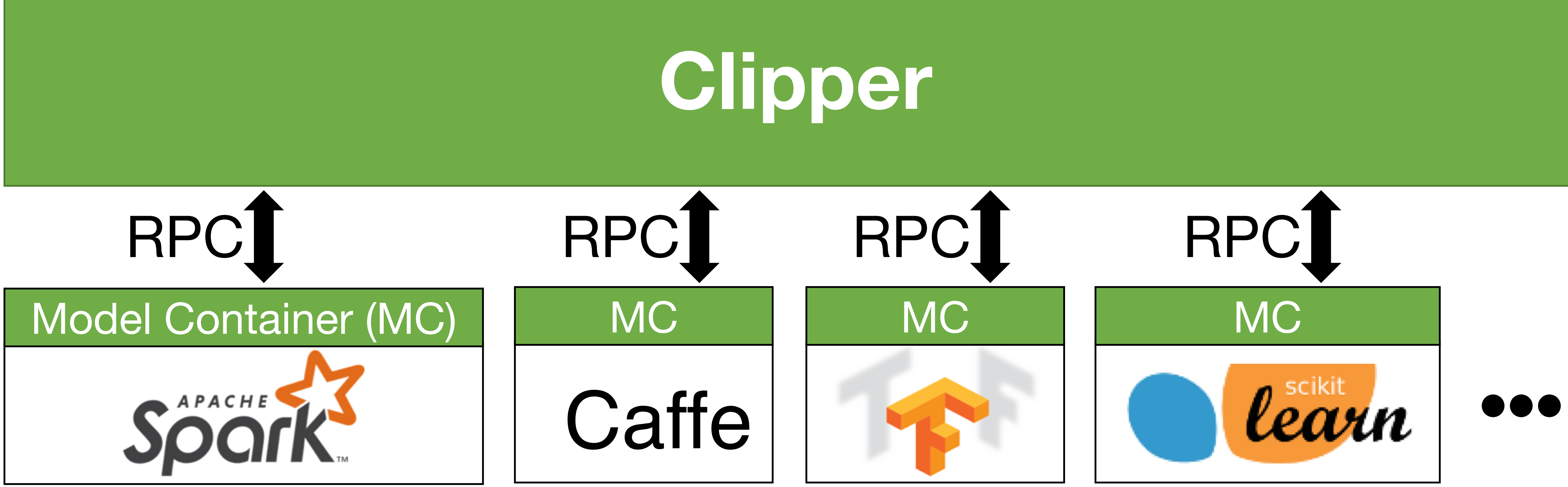
MC

MC

MC

Caffe

•••



Clipper

RPC RPC RPC RPC

Model Container (MC)

APACHE Spark

MC

Caffe

MC

MC

scikit learn

•••

Common Interface → Simplifies Deployment:

- Evaluate models using original code & systems

Container-based Model Deployment

Implement Model API:

```
class ModelContainer:
    def __init__(model_data)
    def predict_batch(inputs)
```

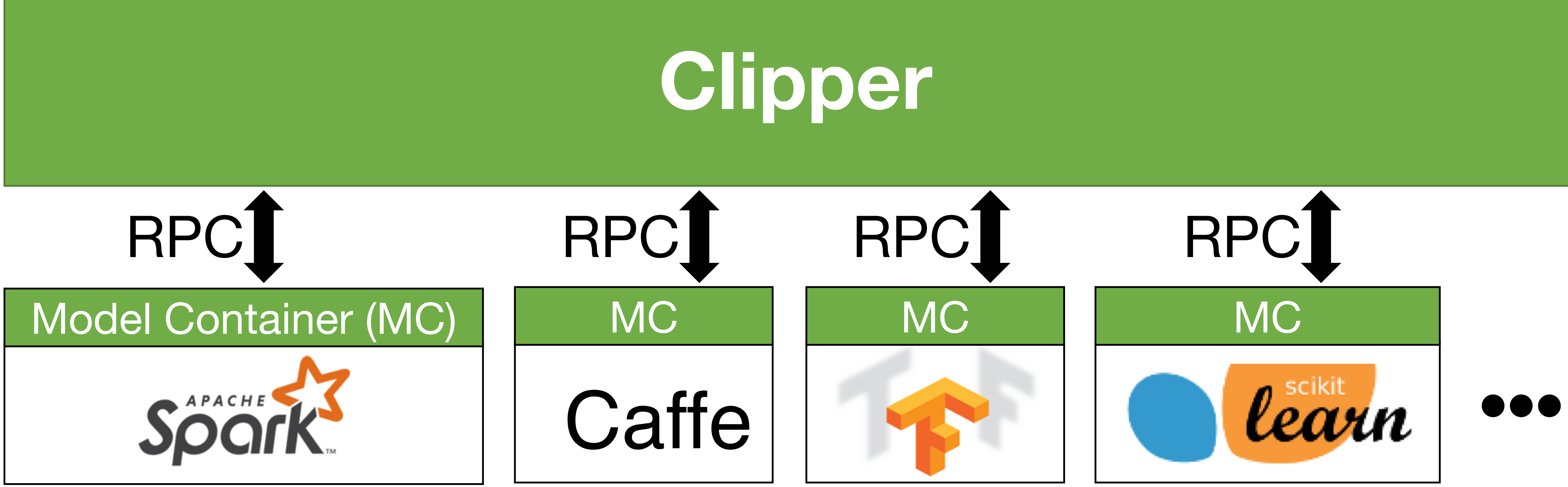
- API support for many programming languages
    - Python
    - Java
    - C/C++
    - R

Container-based Model Deployment

Package model implementation and dependencies

Model Container (MC)

```
class ModelContainer:
    def __init__(model_data)
    def predict_batch(inputs)
```



Clipper

RPC RPC RPC RPC

Model Container (MC)

MC

Caffe

MC

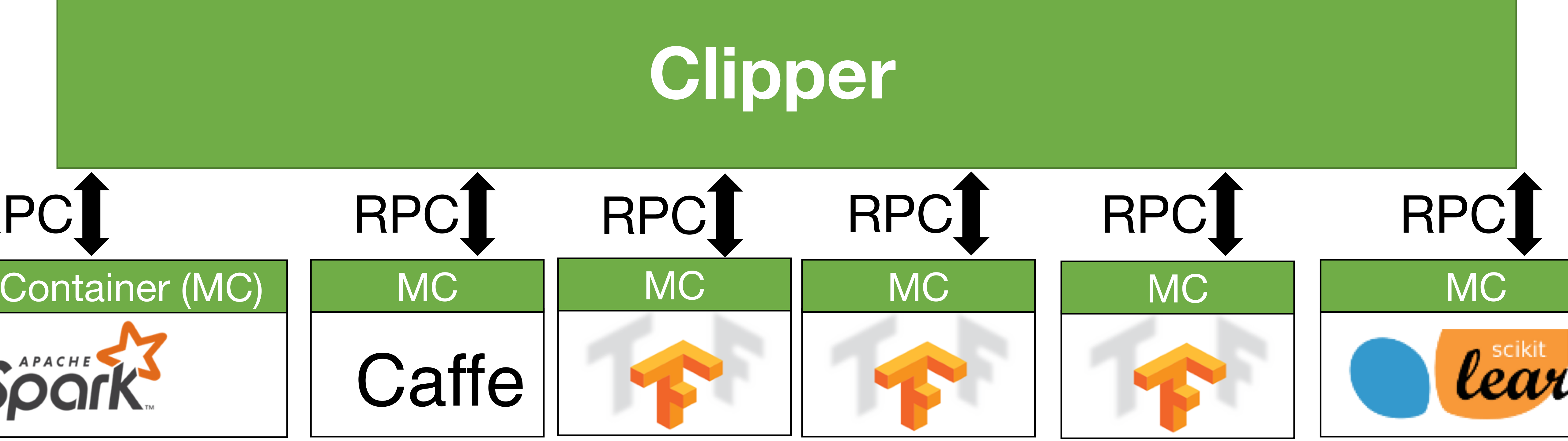
MC

scikit learn

•••

Common Interface → Simplifies Deployment:

- Evaluate models using original code & systems
- Models run in separate processes as Docker containers
    - **Resource isolation:** ML frameworks can be buggy



Clipper

PC RPC RPC RPC RPC RPC

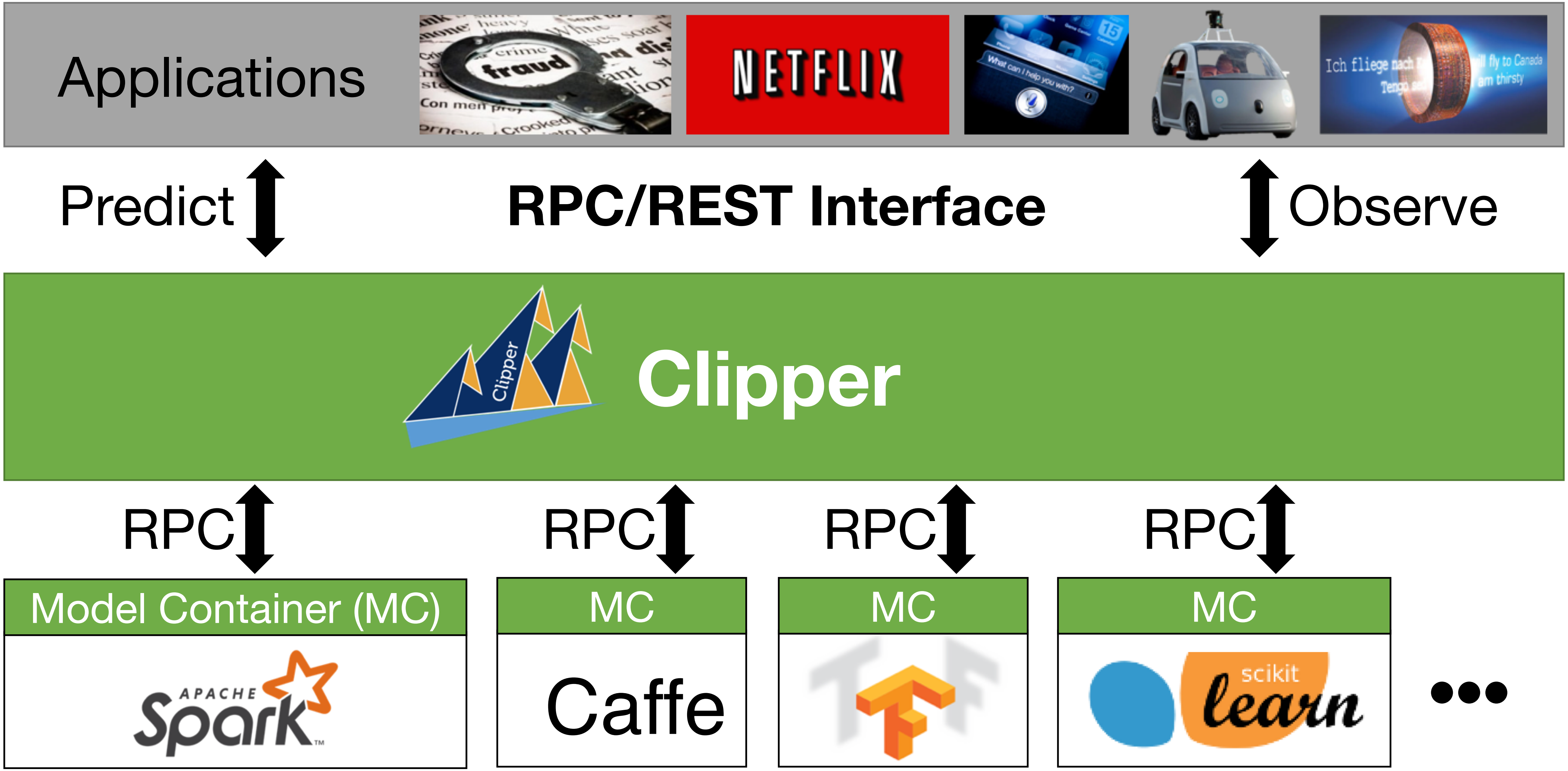
Container (MC) MC MC MC MC MC

Caffe

Common Interface → Simplifies Deployment:

- Evaluate models using original code & systems
- Models run in separate processes as Docker containers
  - **Resource isolation:** ML frameworks can be buggy
  - **Scale-out** at the level of individual models

Clipper Architecture



Applications

Predict

RPC/REST Interface

Observe

Clipper

RPC

RPC

RPC

RPC

Model Container (MC)

MC

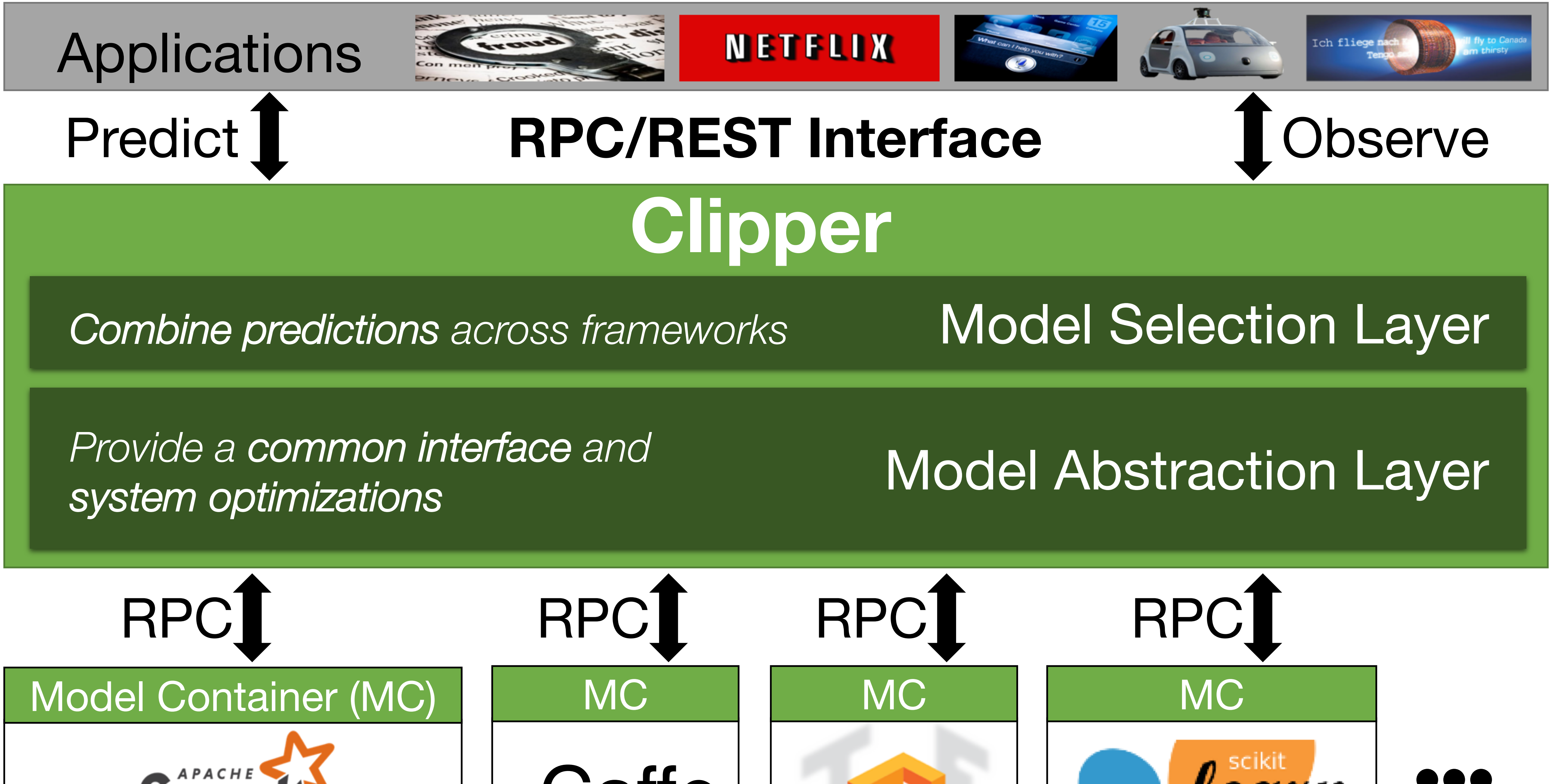
MC

MC

Caffe

•••

Clipper Architecture



Applications

Predict

RPC/REST Interface

Observe

Clipper

Combine predictions across frameworks

Model Selection Layer

Provide a common interface and system optimizations

Model Abstraction Layer

RPC RPC RPC RPC

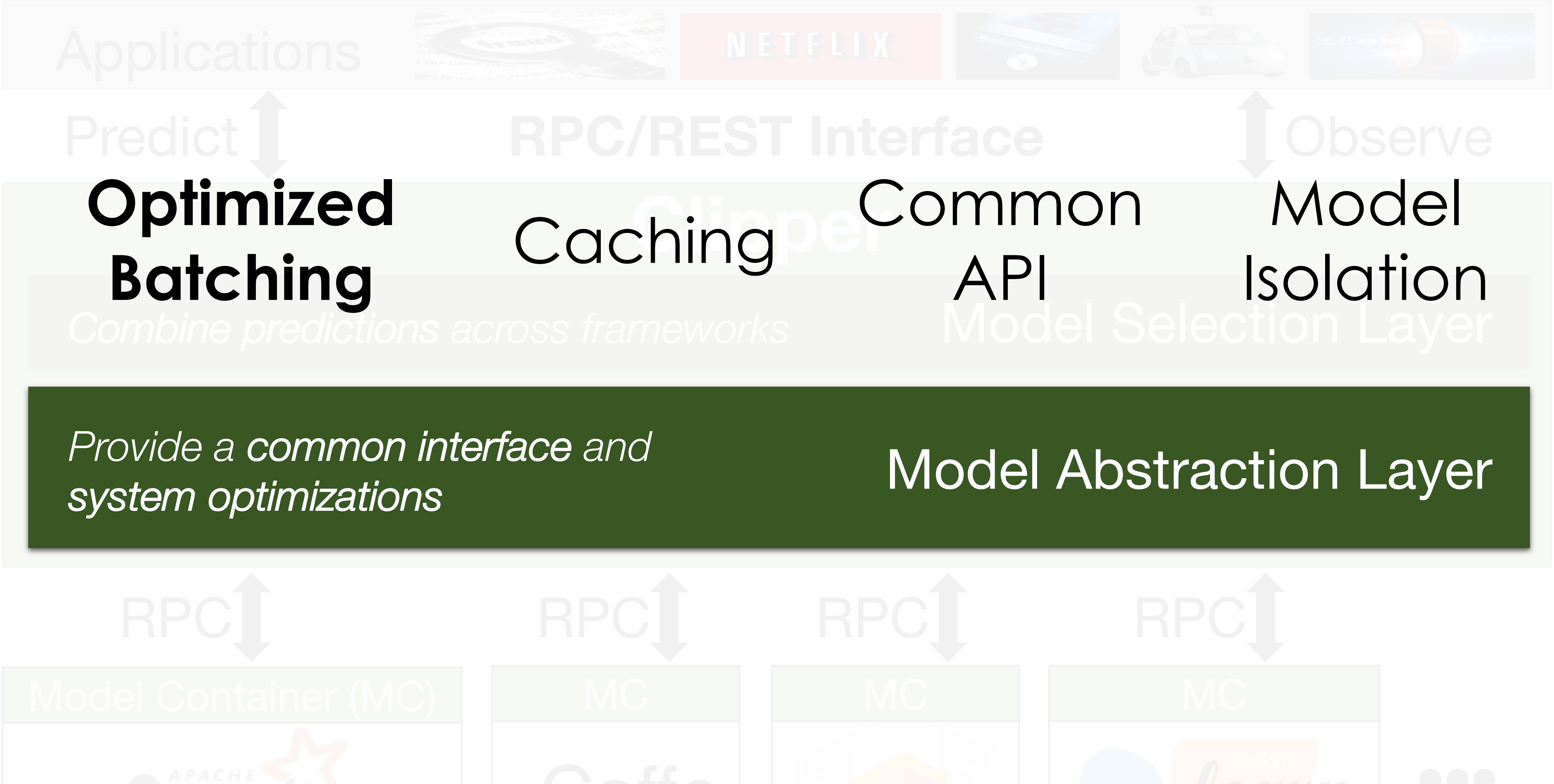
Model Container (MC)

MC

MC

MC

Clipper Architecture



Applications

Predict

RPC/REST Interface

Observe

Optimized Batching

Caching

Clipper

Common API

Model Isolation

Combine predictions across frameworks

Model Selection Layer

Provide a common interface and system optimizations

Model Abstraction Layer

RPC

RPC

RPC

RPC

Model Container (MC)

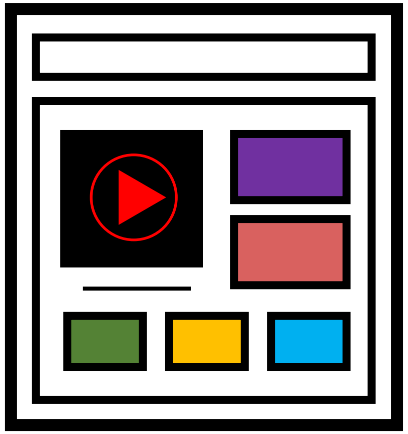
MC

MC

MC

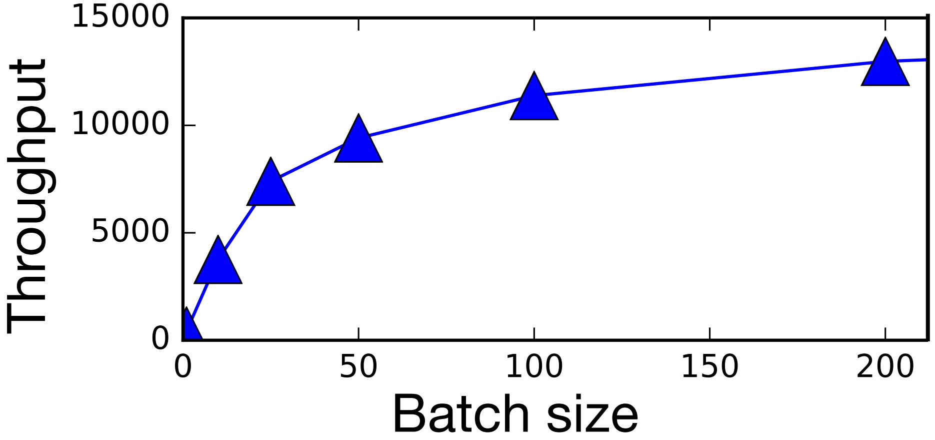
Batching to Improve Throughput

- Why batching helps:



A single page load may generate many queries

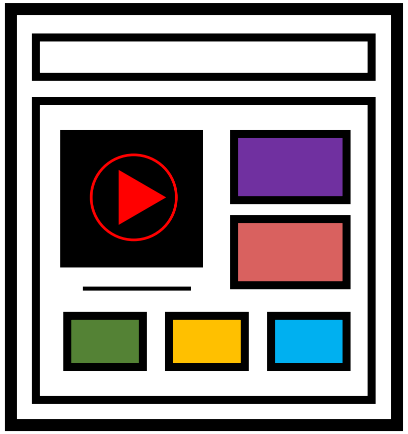
Throughput-optimized frameworks



- Optimal batch depends on:
  - hardware configuration
  - model and framework
  - system load

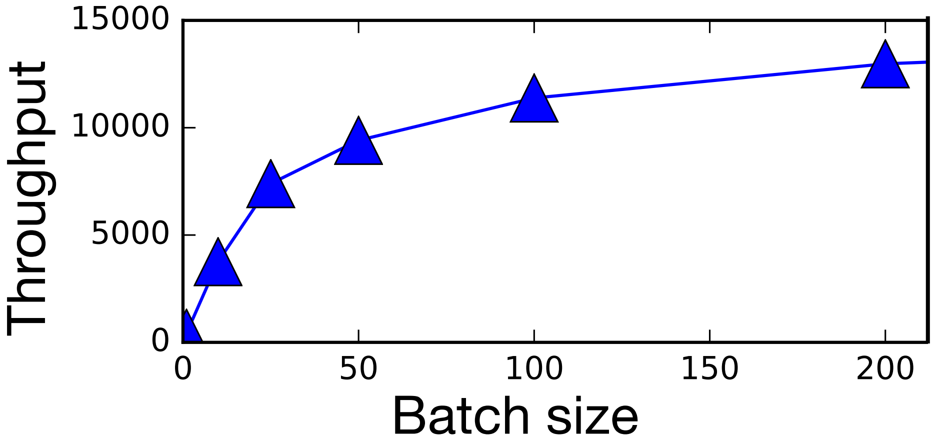
Latency-aware Batching to Improve Throughput

- Why batching helps:



A single page load may generate many queries

Throughput-optimized frameworks



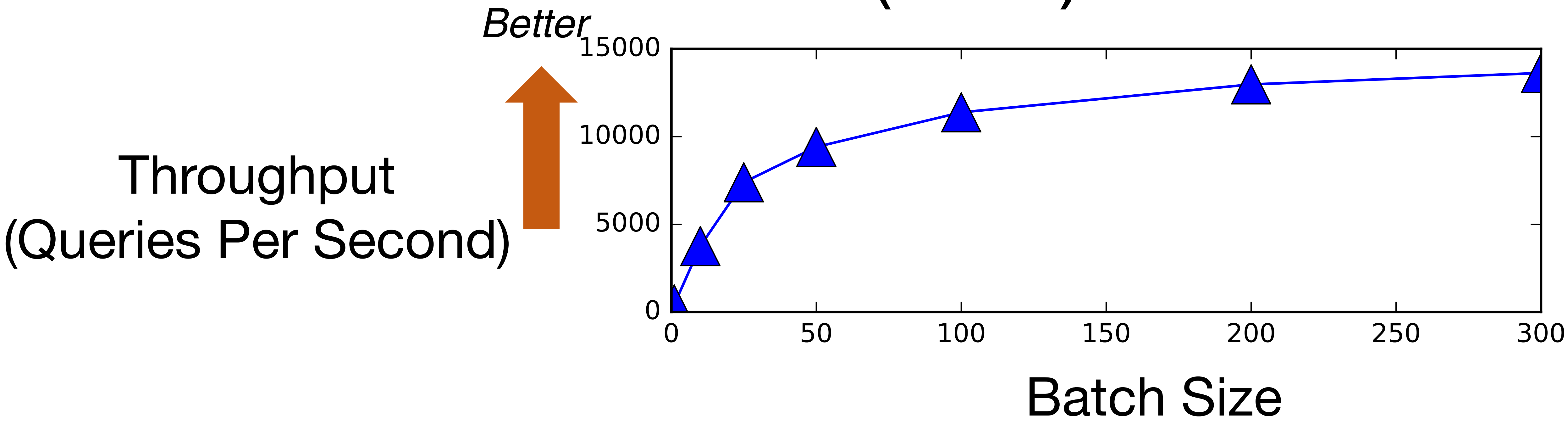
- Optimal batch depends on:
  - hardware configuration
  - model and framework
  - system load

Clipper Solution:

Adaptively tradeoff latency and throughput…

- Inc. batch size *until the latency objective is exceeded* (**Additive Increase**)
- If latency exceeds SLO cut batch size by a fraction (**Multiplicative Decrease**)

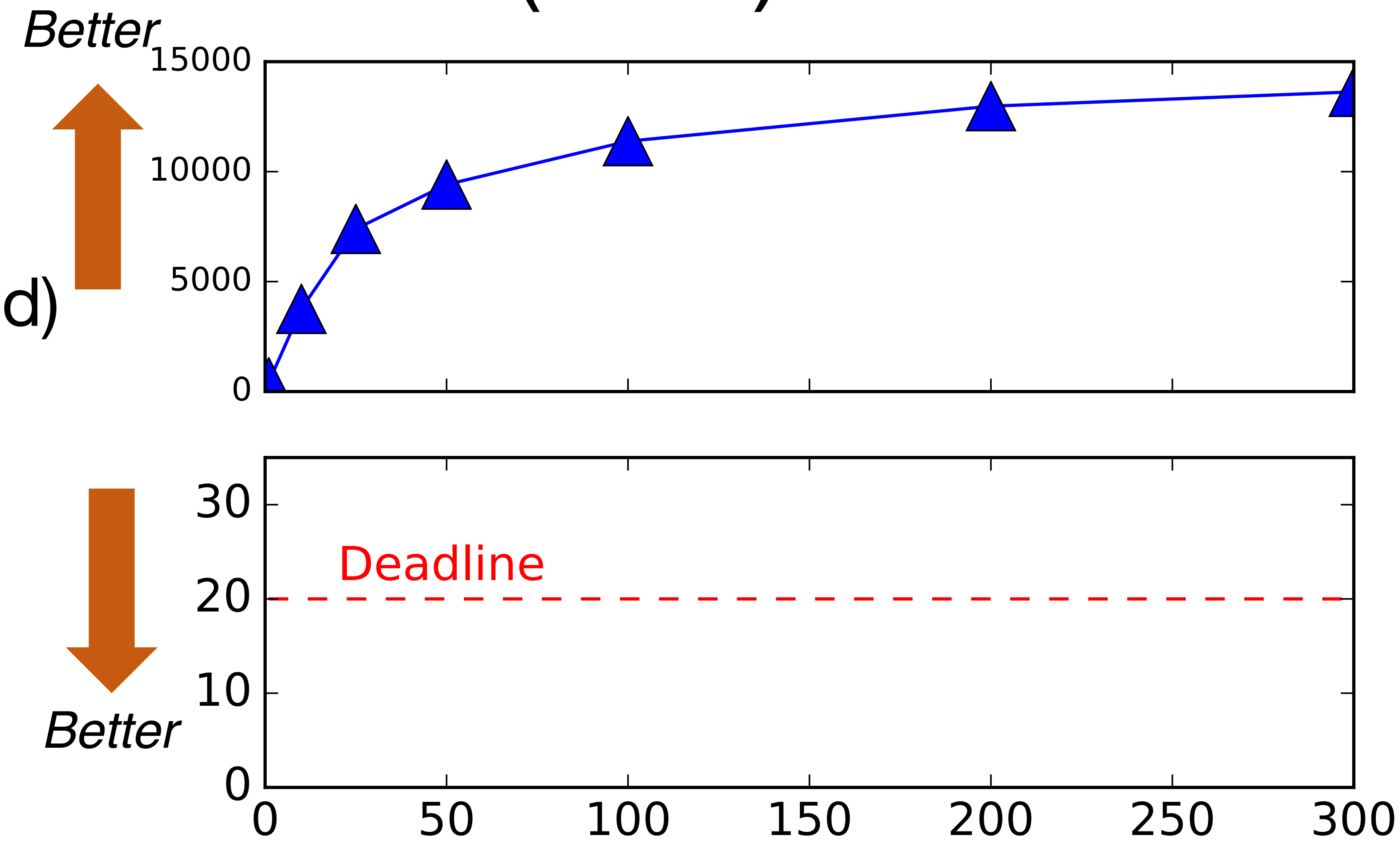
Tensor Flow Conv. Net (GPU)



Tensor Flow Conv. Net (GPU)

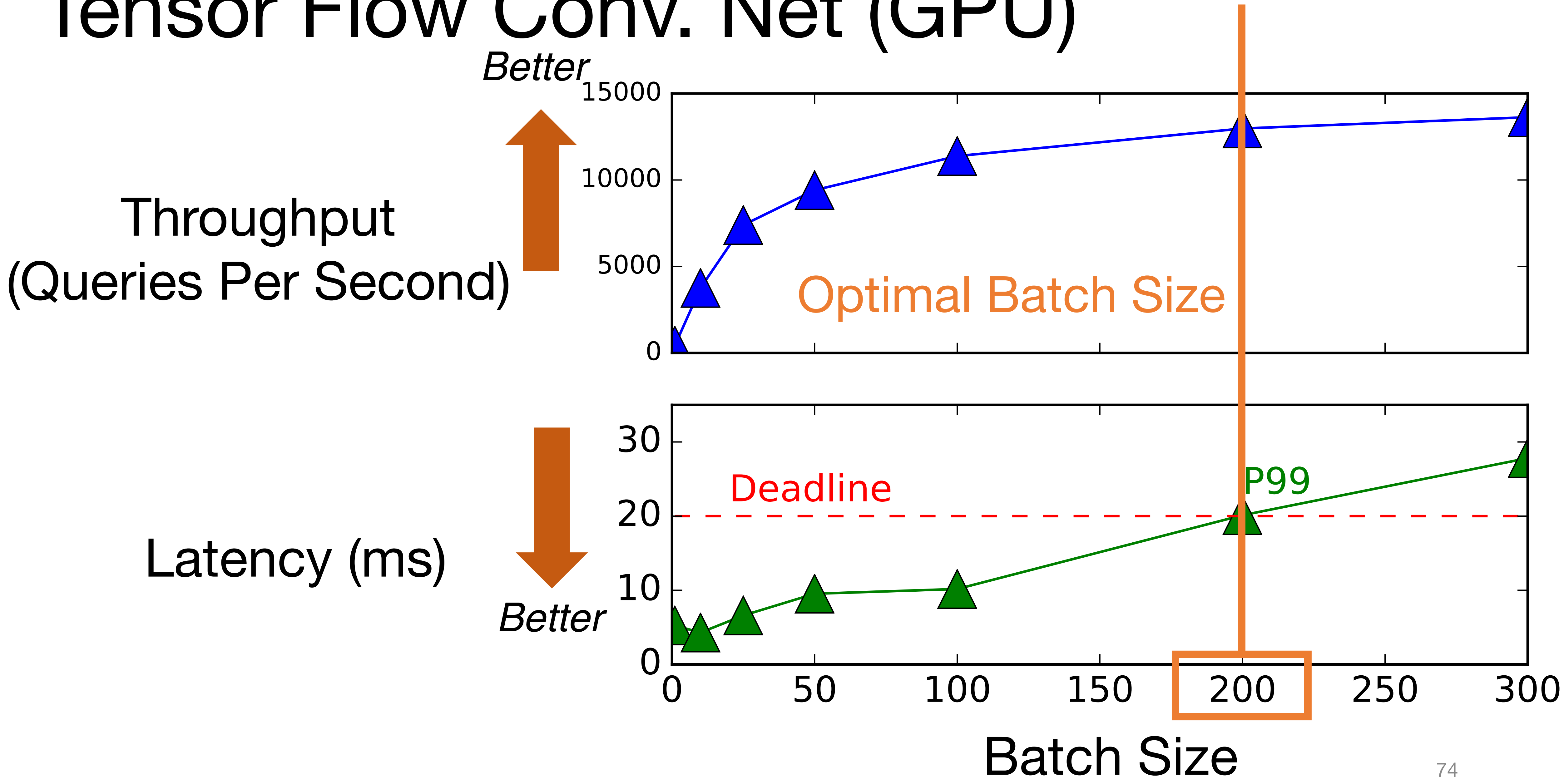
Throughput
(Queries Per Second)

Latency (ms)

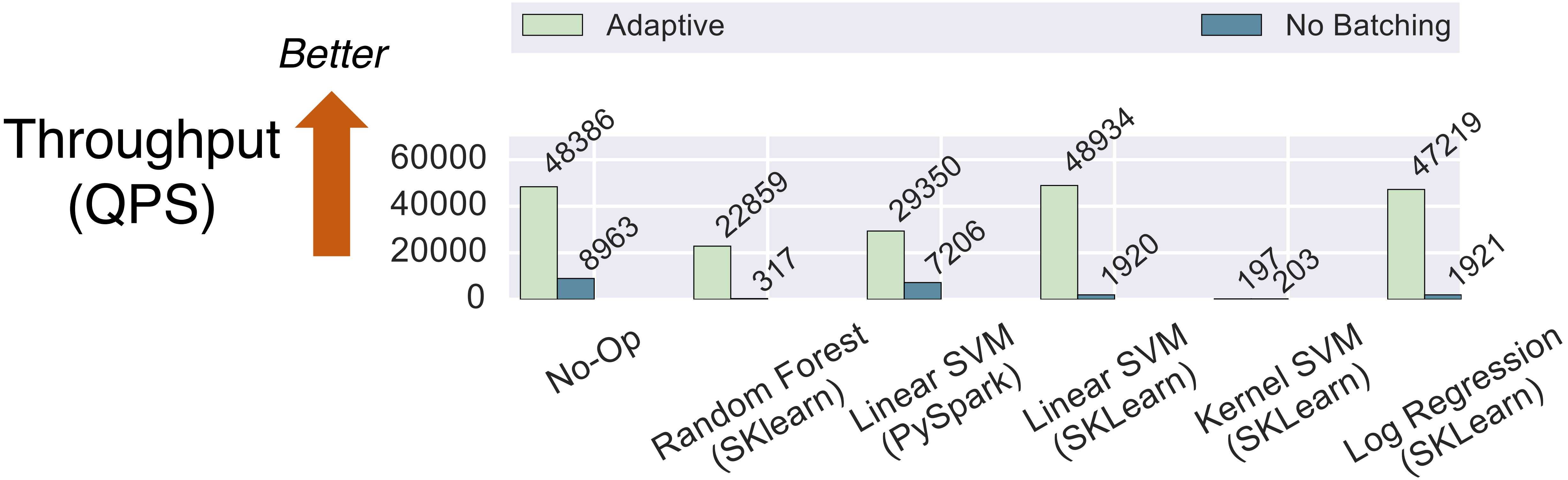


Batch Size

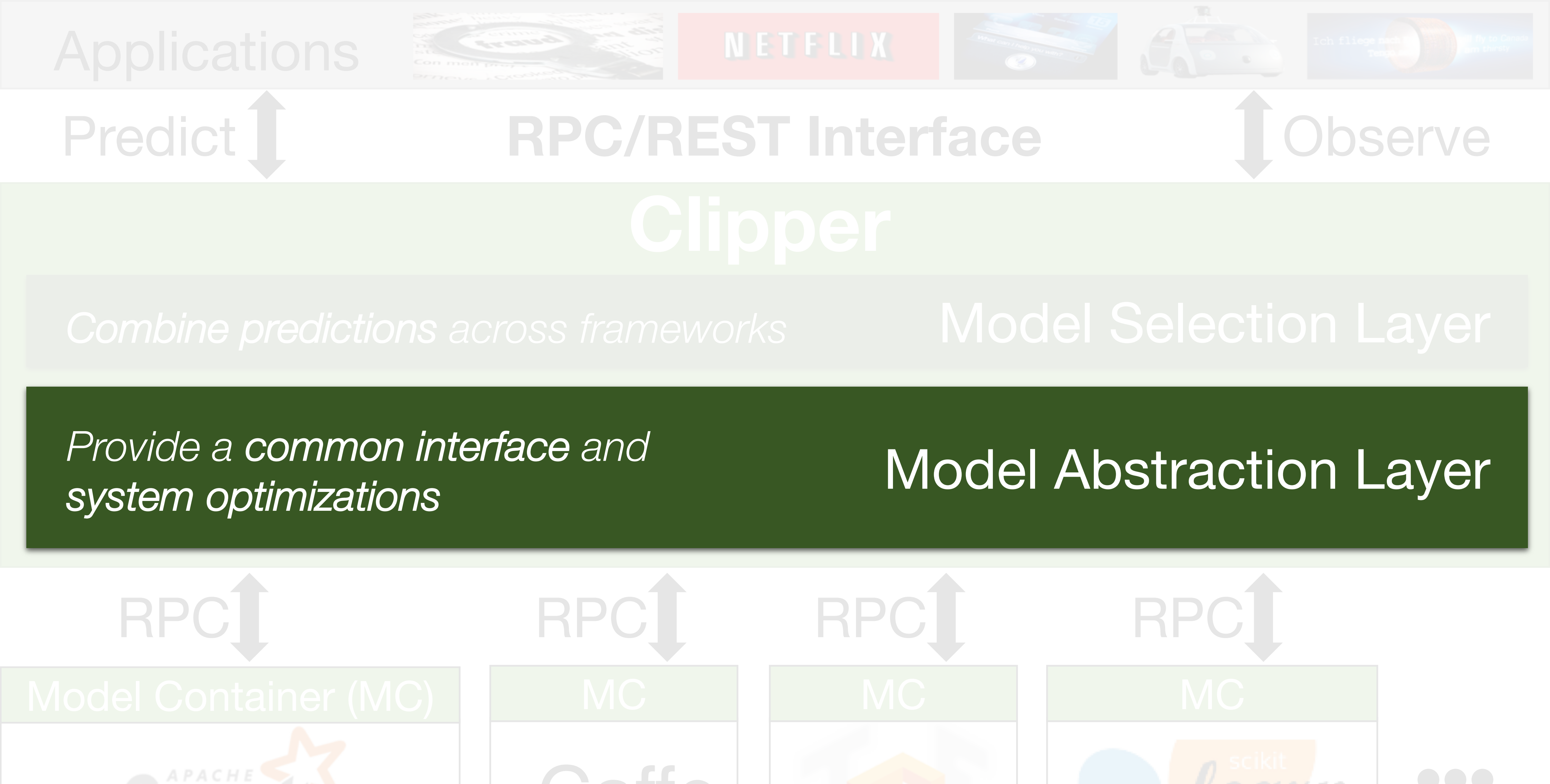
Tensor Flow Conv. Net (GPU)



Latency-aware Batching to Improve Throughput



Clipper Architecture



Applications

Predict

RPC/REST Interface

Observe

Clipper

Combine predictions *across frameworks*

Model Selection Layer

Provide a ***common interface*** *and* ***system optimizations***

Model Abstraction Layer

RPC

RPC

RPC

RPC

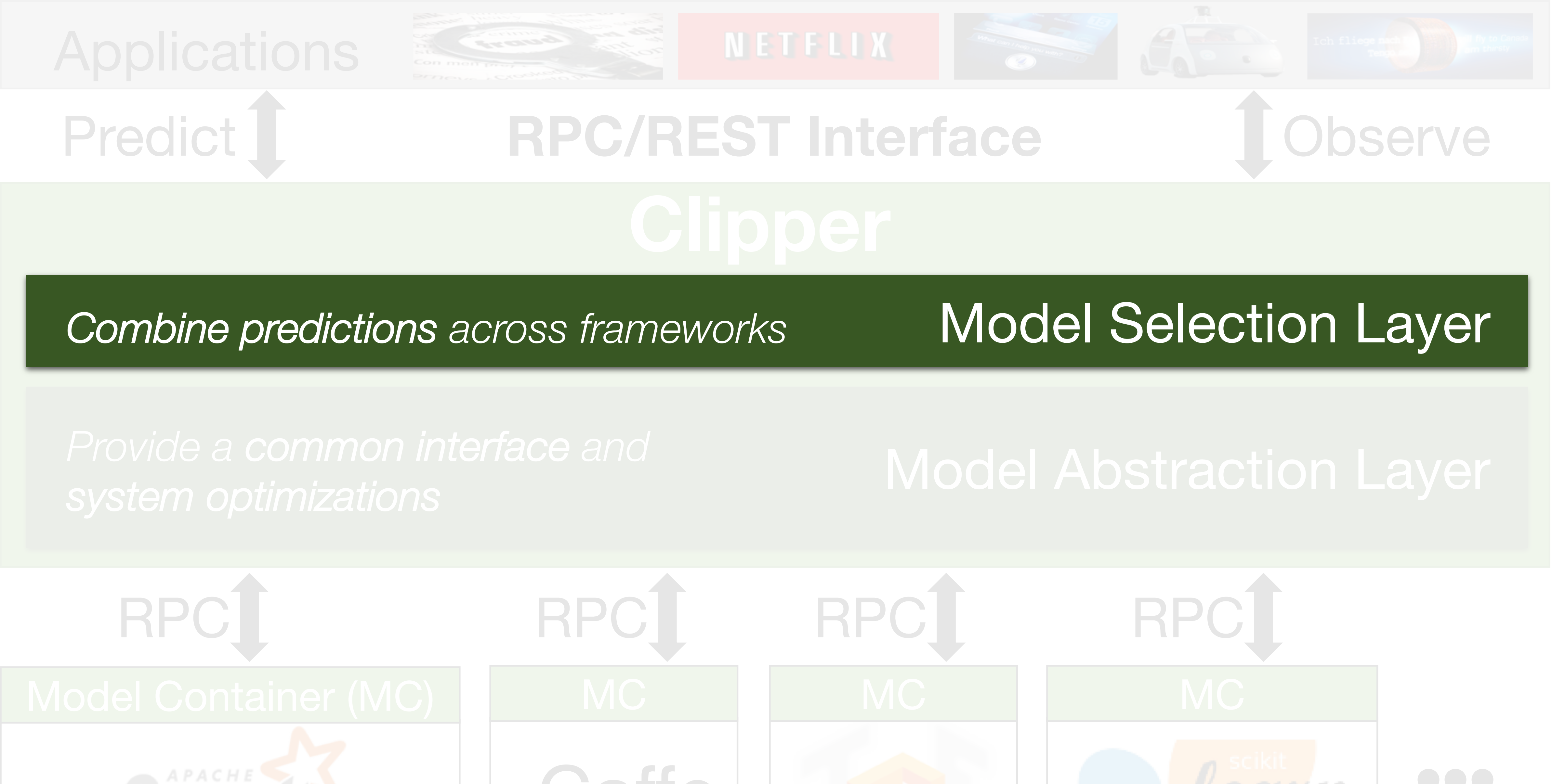
Model Container (MC)

MC

MC

MC

Clipper Architecture



Applications

Predict

RPC/REST Interface

Observe

Clipper

Combine predictions across frameworks

Model Selection Layer

Provide a common interface and system optimizations

Model Abstraction Layer

RPC

RPC

RPC

RPC

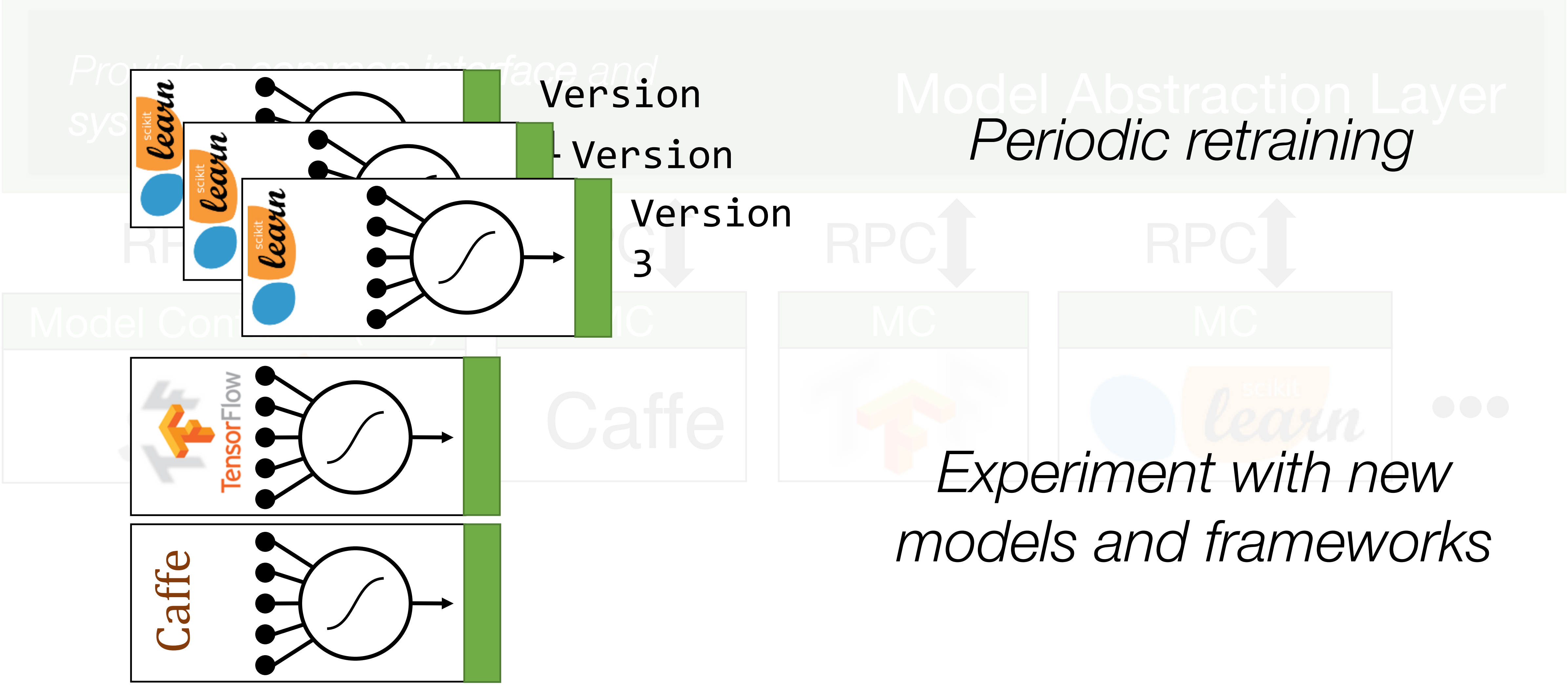
Model Container (MC)

MC

MC

MC

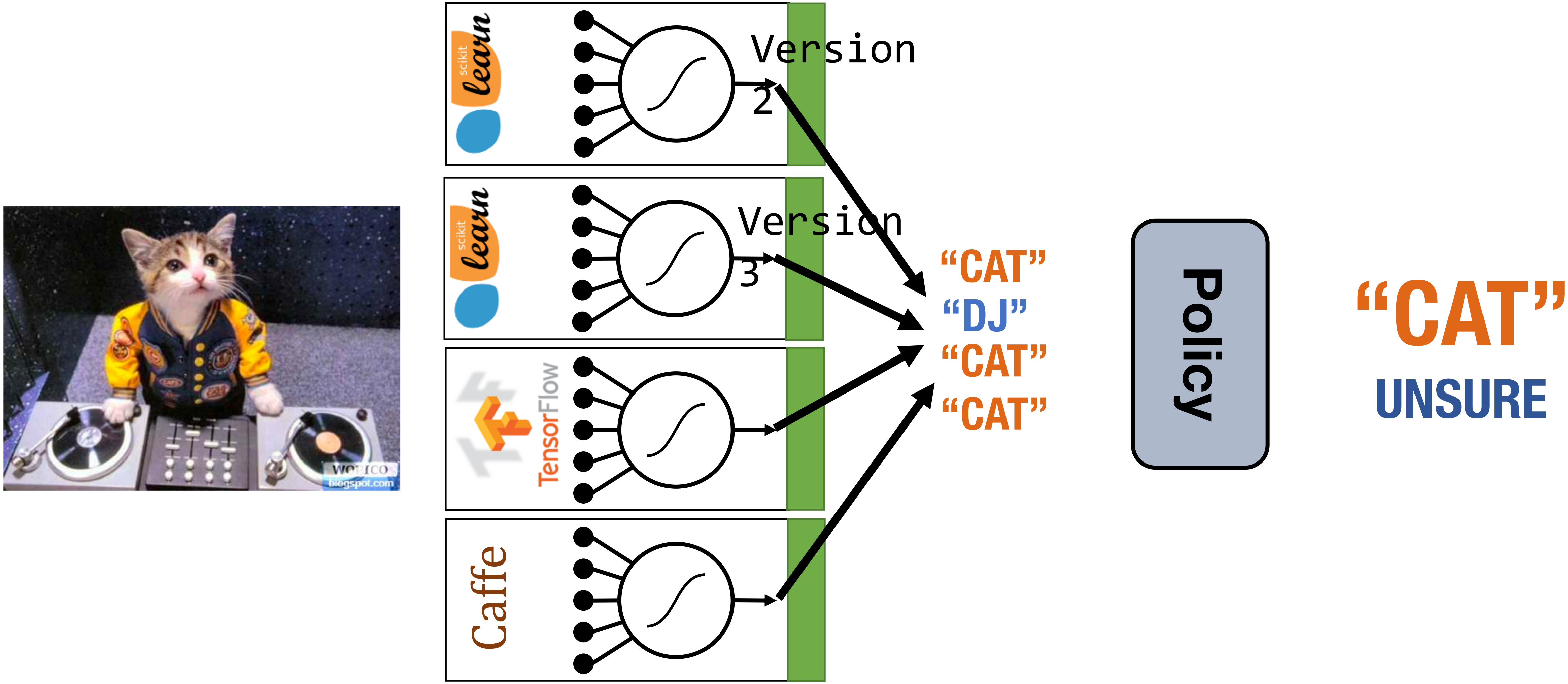
Combine predictions across frameworks Model Selection Layer



Periodic retraining

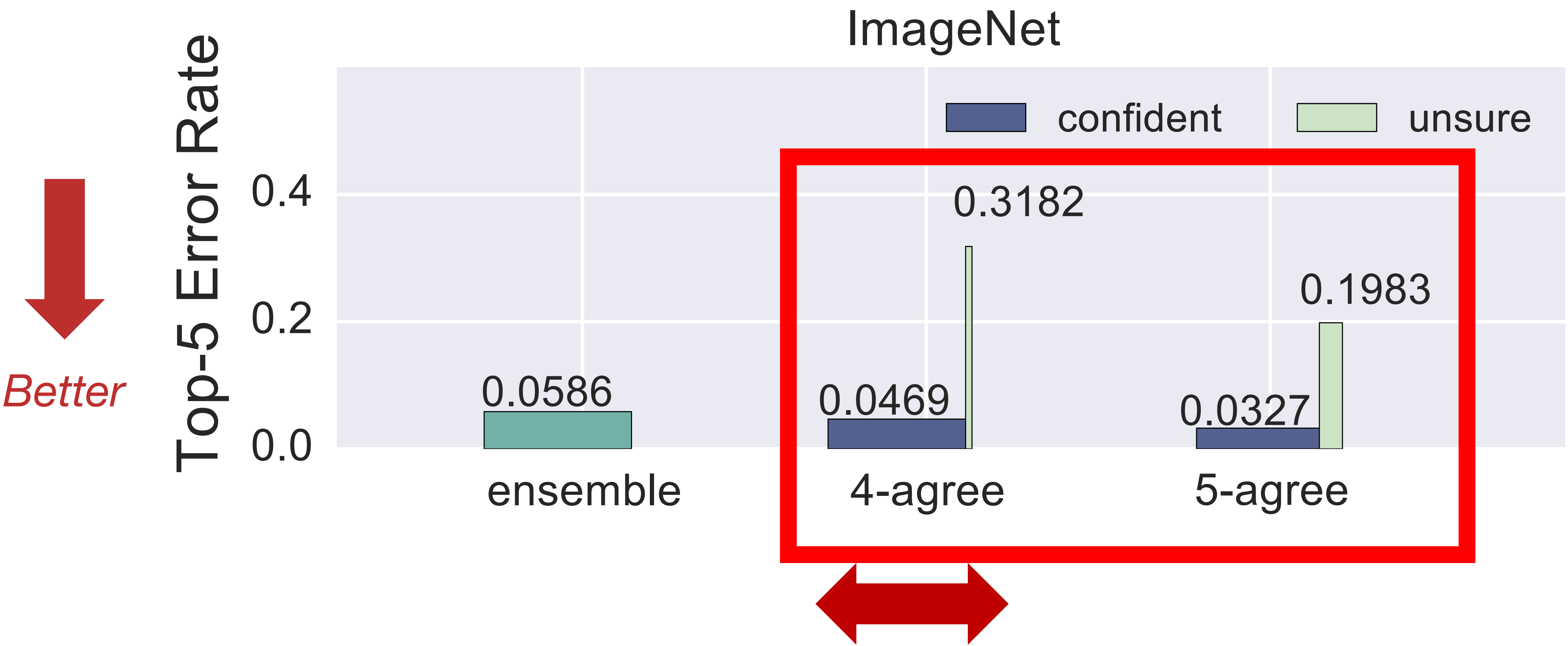
Experiment with new models and frameworks

Selection Policy can Calibrate Confidence



Selection Policy: Estimate confidence

Selection Policy: Estimate confidence



Better

width is percentage of query workloads

Project Status and Development

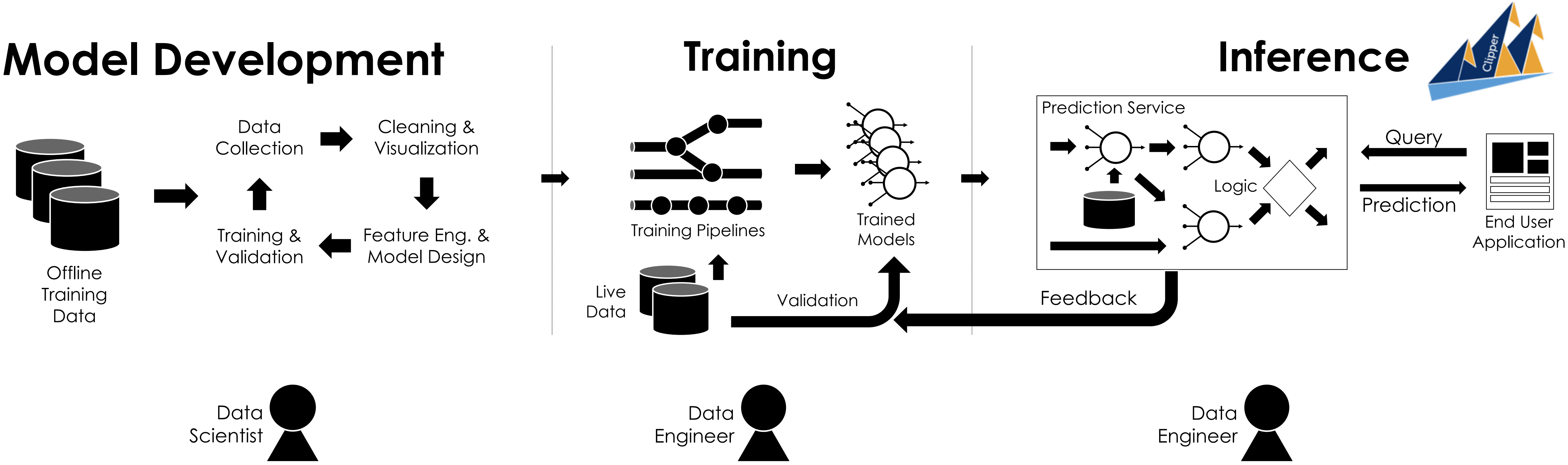
- Current development focus:
  - **stability** and **performance** improvements
  - **easy model deployment** for common ML frameworks: *Pytorch, caffe2 (via onnx), tensorflow, xgboost, mxnet, pyspark, scikit learn*
  - **metrics** and **monitoring** infrastructure using Prometheus
- **Development Team:** *22 active contributors*
  - Including 8 from outside Berkeley
- Working with several organizations on **production deployments**
  - SAP, Scotia Bank, Pacific AI, ARM…

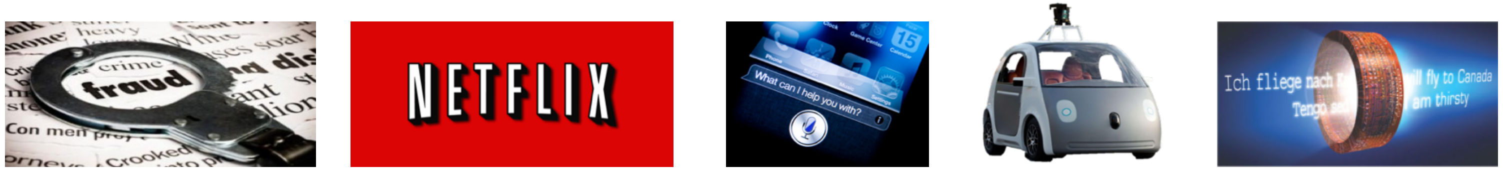
I made a case for

Model & Data Engineering

and outlined some of the
opportunities & **challenges**

Machine Learning Lifecycle





Middle layer for prediction serving.

Common Abstraction **System Optimizations**

---

theano
scikit learn
TensorFlow
PYTORCH
VW
Spark
Caffe
dmlc mxnet
KALDI

Clipper

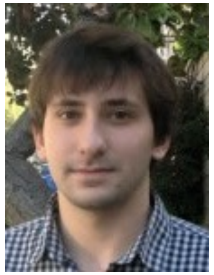
http://clipper.ai

Open-source prediction serving system for low-latency, high-throughput predictions across machine learning models and frameworks.

Thank you!

jegonzal@berkeley.edu

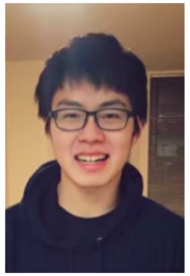
Collaborators


Daniel Crankshaw


Rolando Garcia


Joe Hellerstein


Yika Luo


Simon Mo


Vikram Sreekanti


Ion Stoica


Alexey Tumanov


Xin Wang


Neeraja Yadwadkar


Corey Zumar

Research Sponsors

Alibaba Group 阿里巴巴集团
amazon webservices™
蚂蚁金服 ANT FINANCIAL
Capital One
ERICSSON
facebook
Google

HUAWEI
intel Leap ahead™
Microsoft
Scotiabank
splunk>
vmware
U.S. DEPARTMENT OF HOMELAND SECURITY

Bonus!

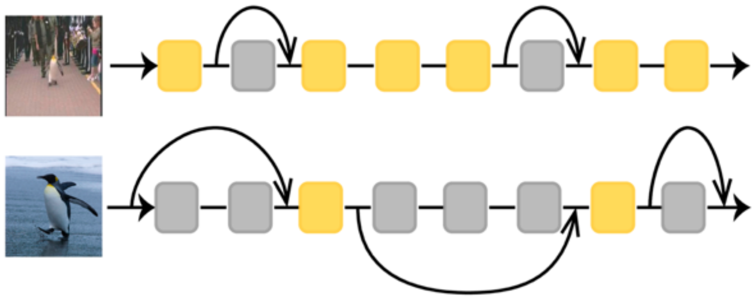
A few of the RISE Lab projects …

Real-time Inference

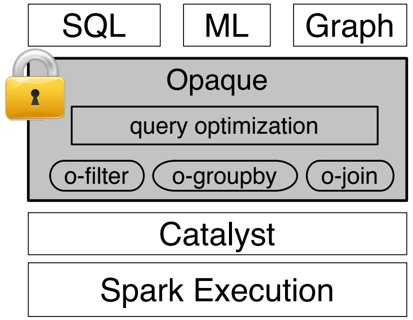
Clipper

IDK Prediction Cascades

Teaching AI to think fast
by
learning not to overthinking



SkipNets: RL for Dynamic Network Design



SQL | ML | Graph

Opaque

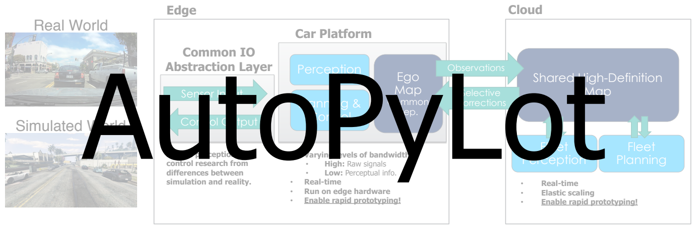
query optimization

o-filter | o-groupby | o-join

Catalyst

Spark Execution

Hardware Security for Apache Spark



AutoPyLot

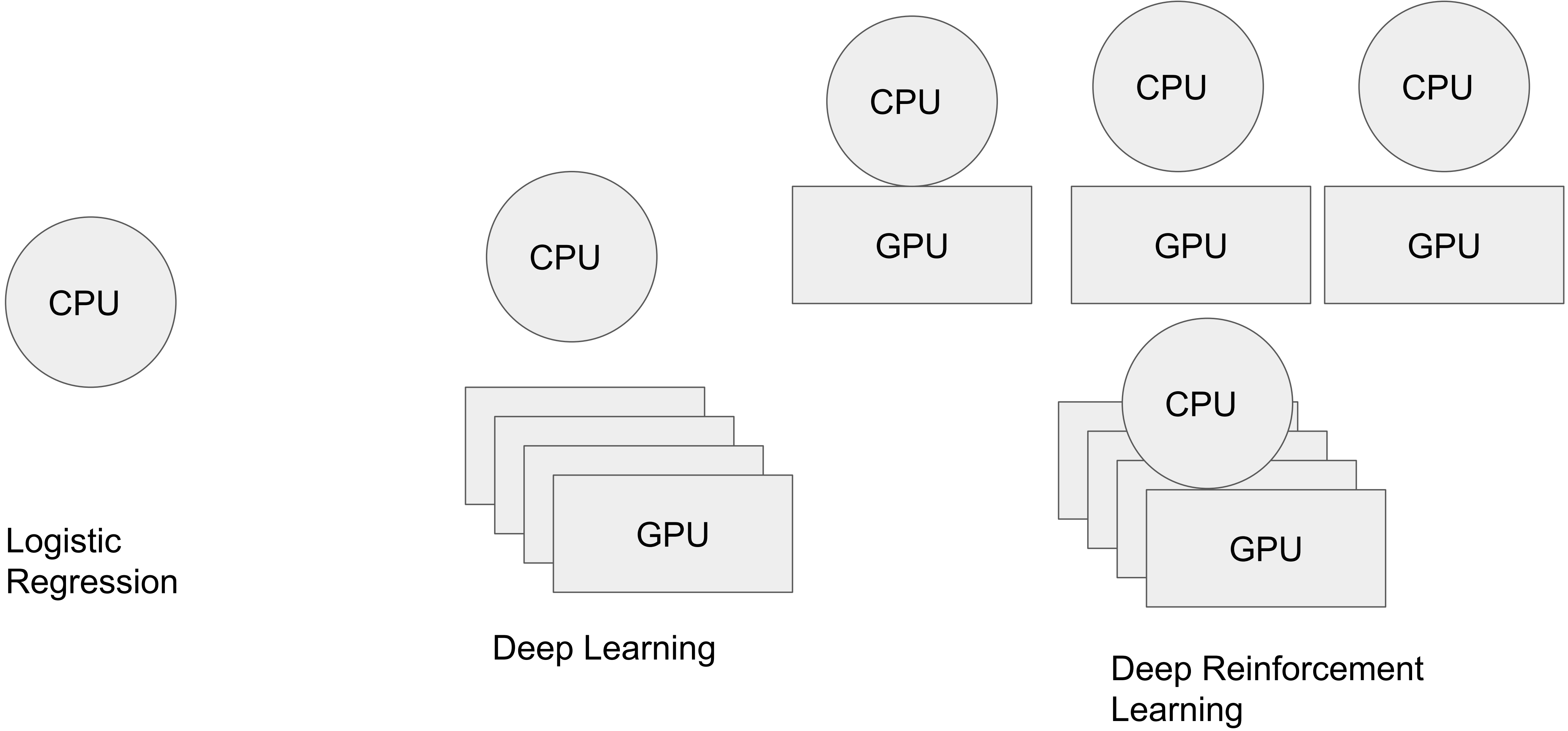
An open platform for **Autonomous Vehicles**

RAY

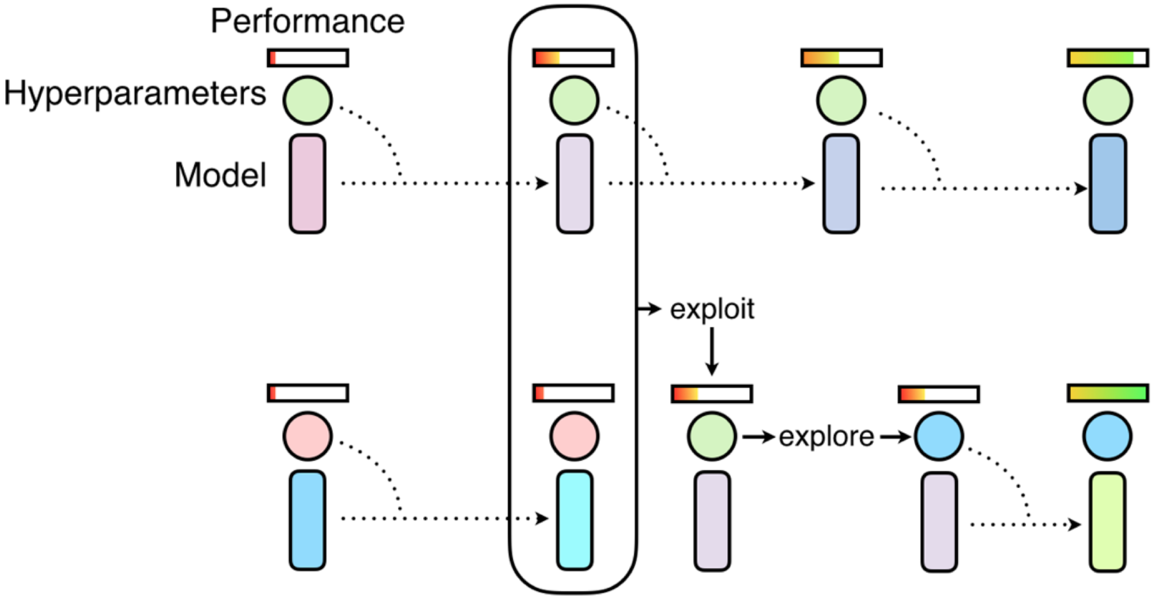
Parallel Python for Reinforcement Learning

Ray Tune

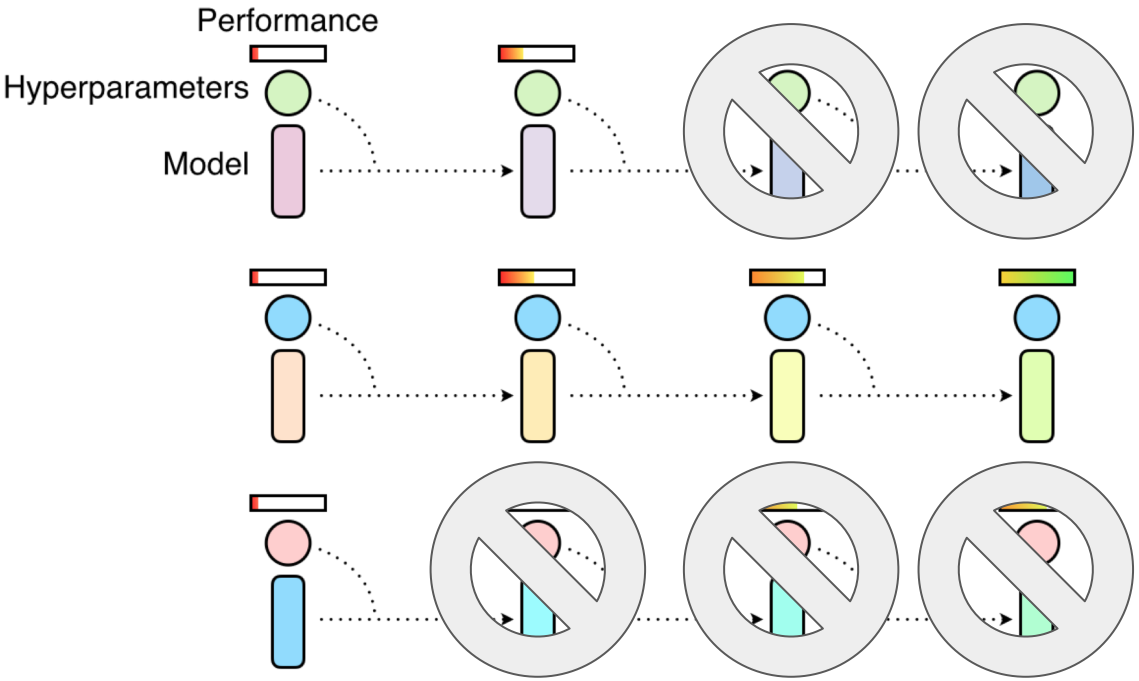
RL and Deep Learning workloads demand different resource requirements.



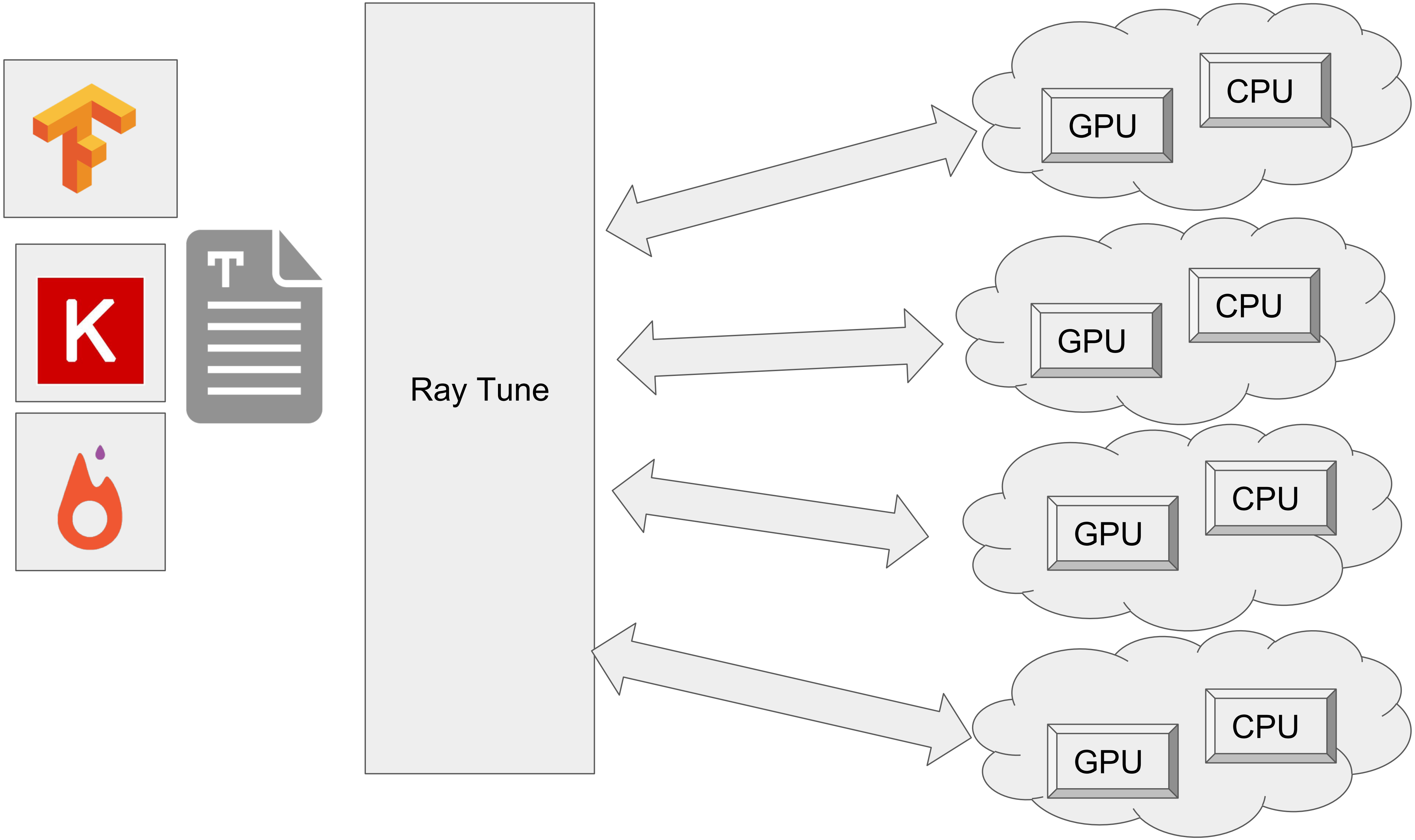
New Algorithms for hyperparameter tuning require more complicated control flows



Population Based Training



HyperBand



Ray Tune

GPU

CPU

GPU

CPU

GPU

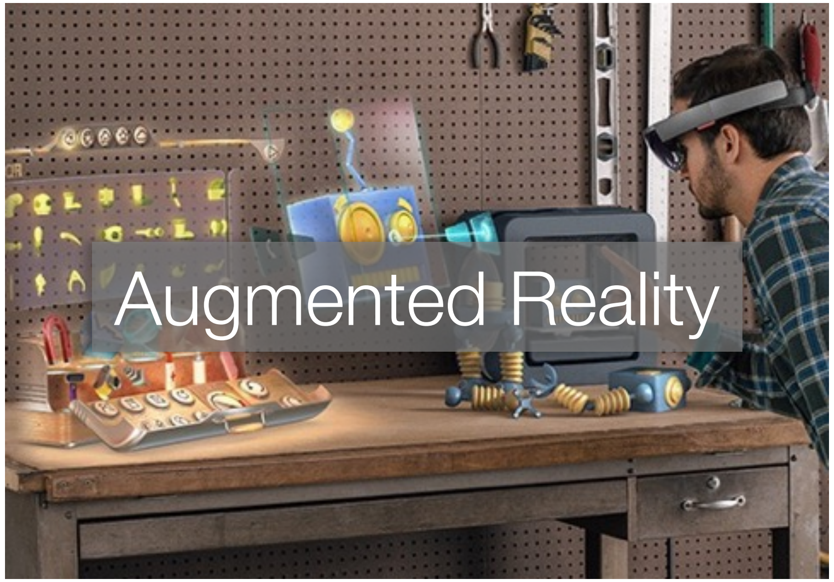
CPU

GPU

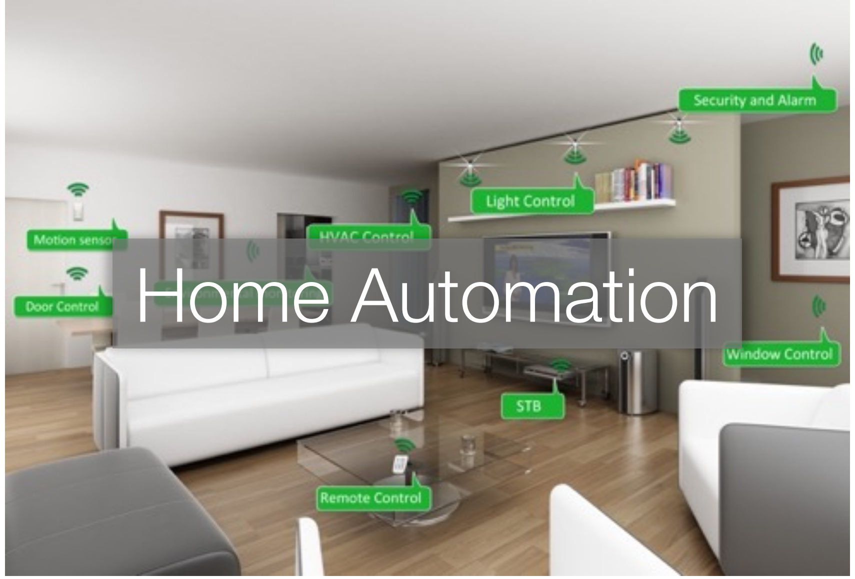
CPU

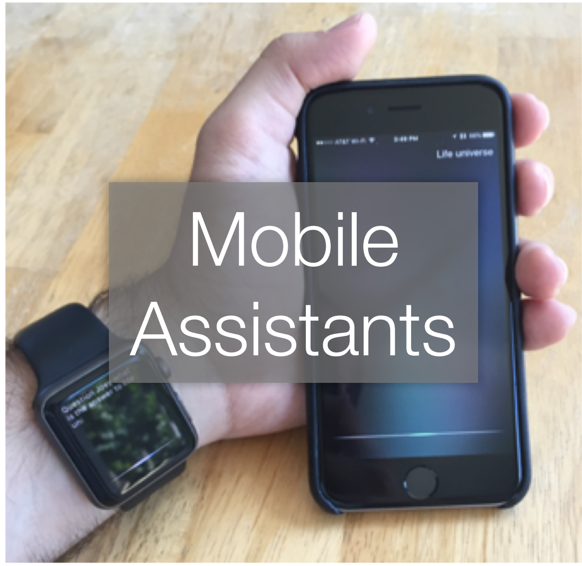
Security

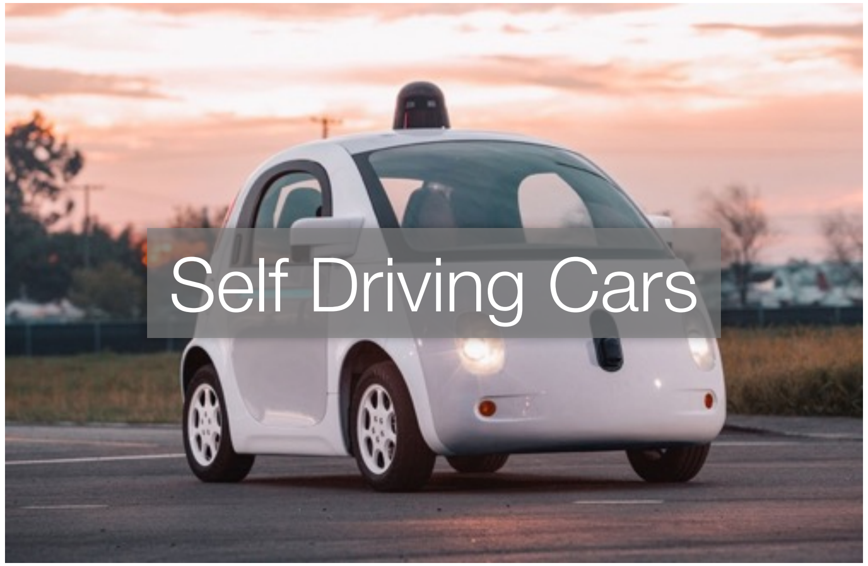
Machine Learning is on the critical path


Augmented Reality


Home Security


Home Automation


Mobile Assistants


Self Driving Cars

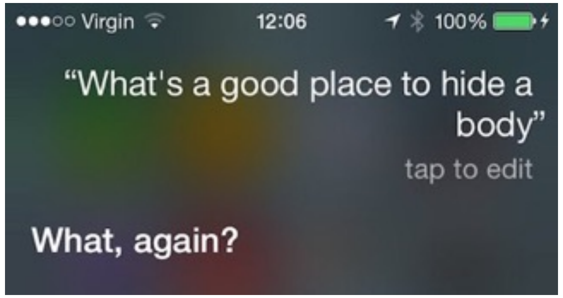

Personal Robotics

Machine Learning in **Sensitive Contexts**

AR/VR Systems

Home Monitoring

Voice Technologies

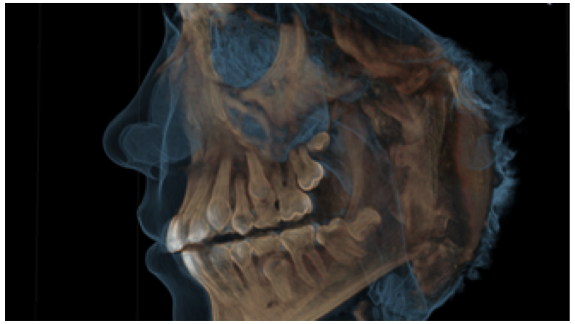

●●●○○ Virgin 12:06 100%

“What's a good place to hide a body”

tap to edit

What, again?

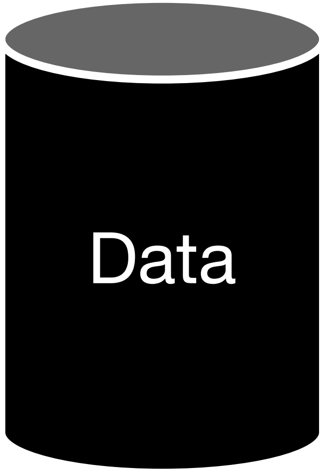
Medical Imaging



Protect the **data**, the **model**, and the **query**

Protect the **data**, the **model**, and the **query**

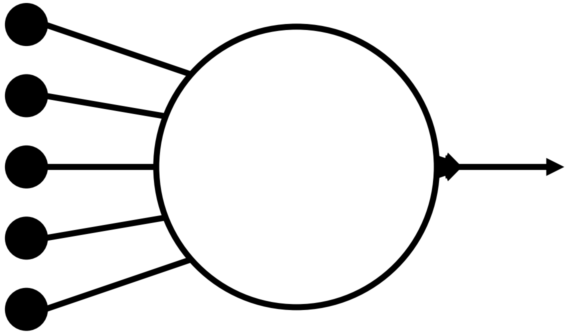
High-Value **Data** *is* ***Sensitive***



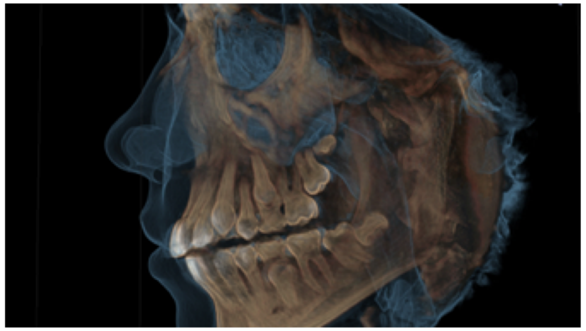
- Medical Info.
- Home video
- Finance

Models capture **value** in data

- Core Asset
- “Contain” the data

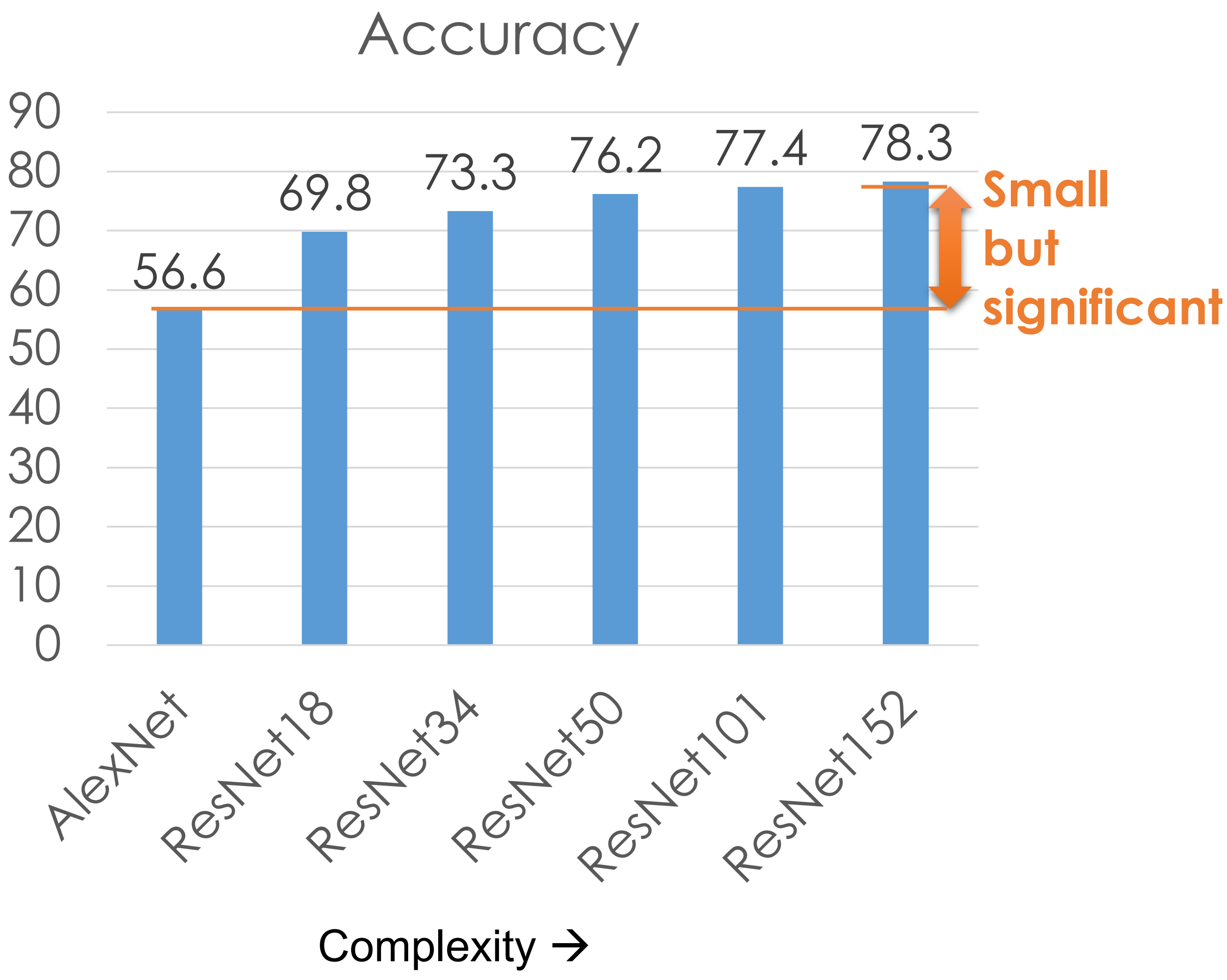


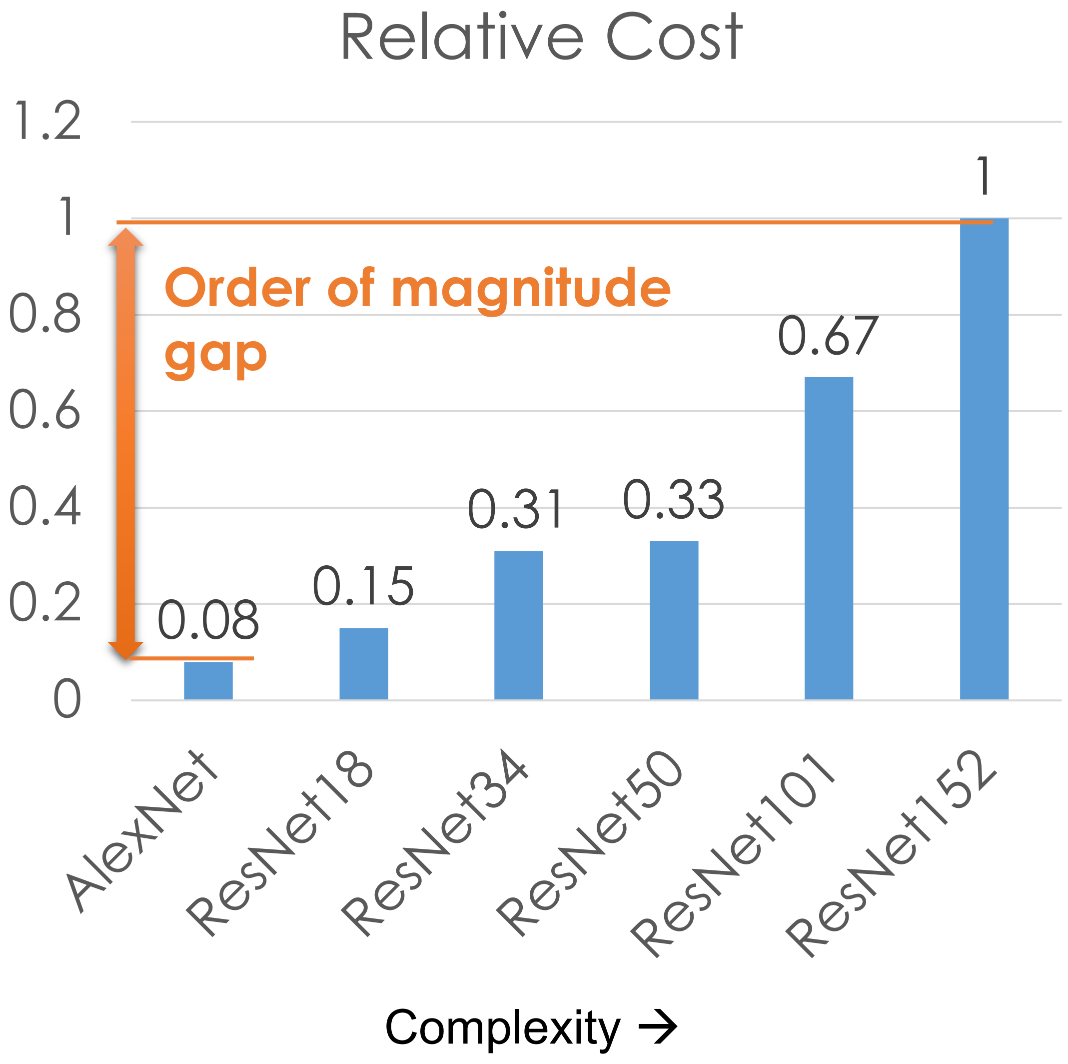
Queries can be as sensitive as the data



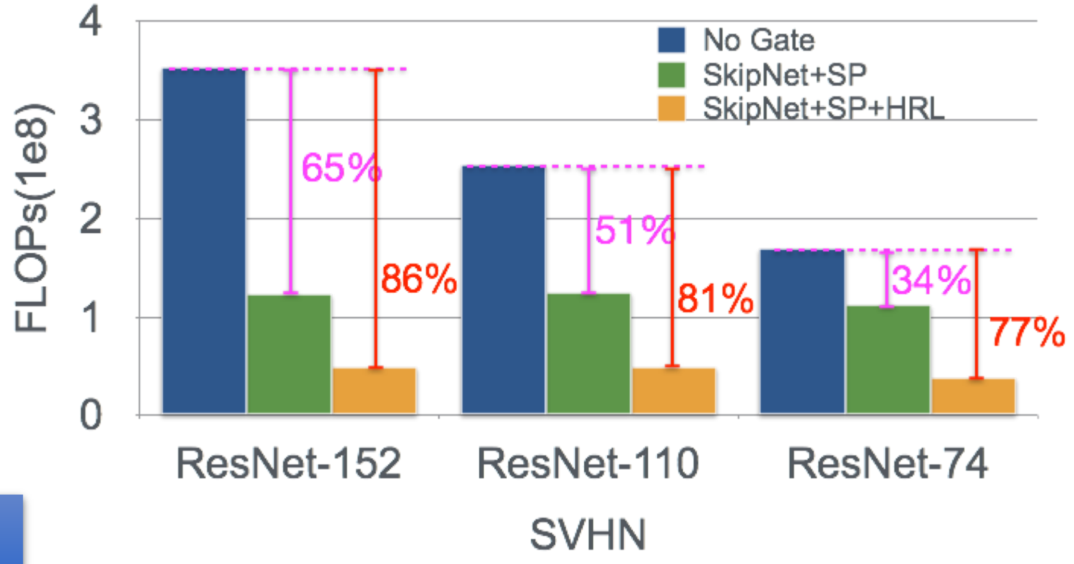
Dynamic Inference

Model **costs are increasing** much **faster than** gains in **accuracy**.





Dynamic Networks
Learning Not to Overthink



IDK Prediction Cascades


Query

Simple Model

I don't Know

Accurate Model

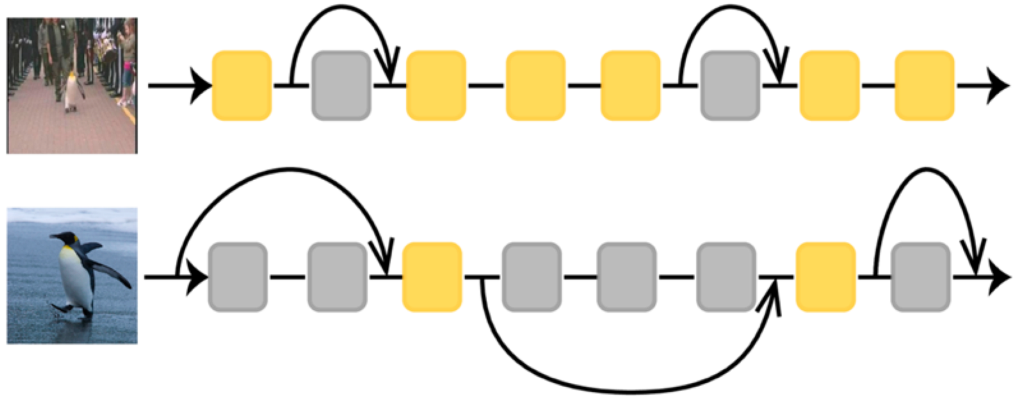
Fast

Slow

Prediction

Prediction

SkipNets: gated execution



Easy Images
Skip **Many** Layers

Hard Images
Skip **Few** Layers

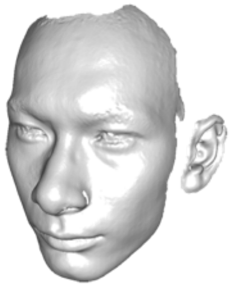
IDK Prediction Cascades

Simple models for simple tasks

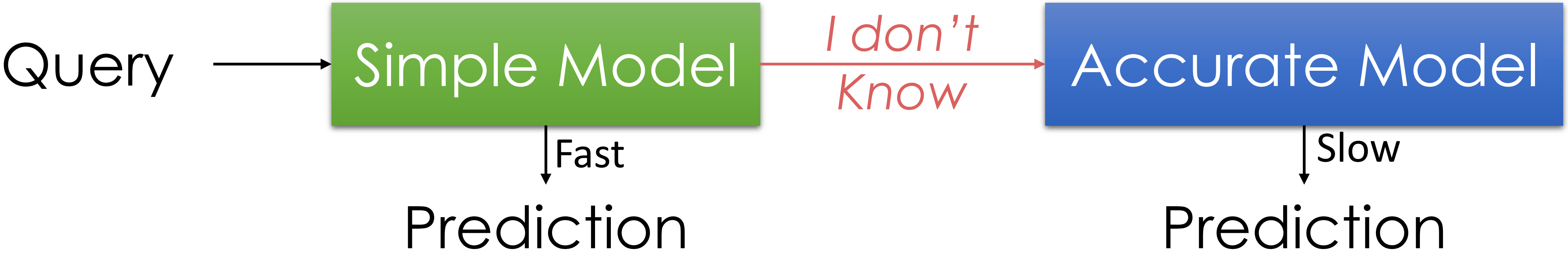
Xin
Wang

Yika
Luo

Zi-Yi
Duo



Fisher
Yu

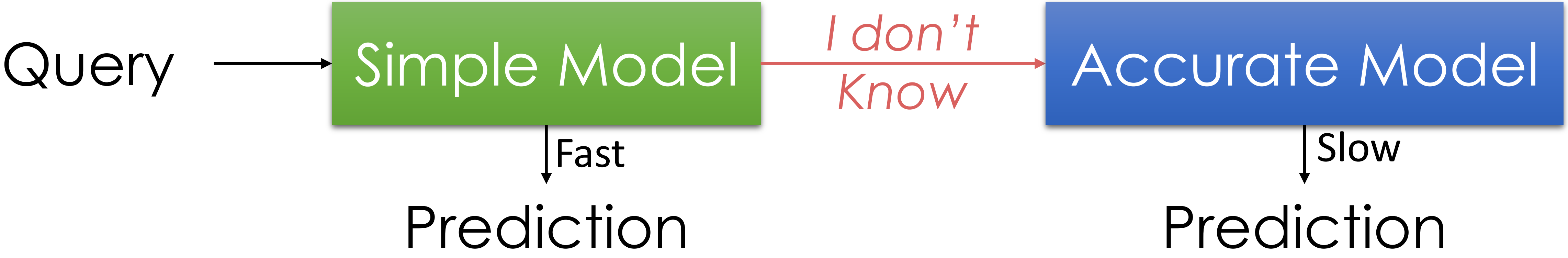


Query → Simple Model —*I don't Know*→ Accurate Model

Simple Model —Fast→ Prediction

Accurate Model —Slow→ Prediction

Learn to combine **fast (inaccurate) models** with **slow (accurate) models** to maximize accuracy while reducing computational costs.


Query

Simple Model

I don't Know

Accurate Model

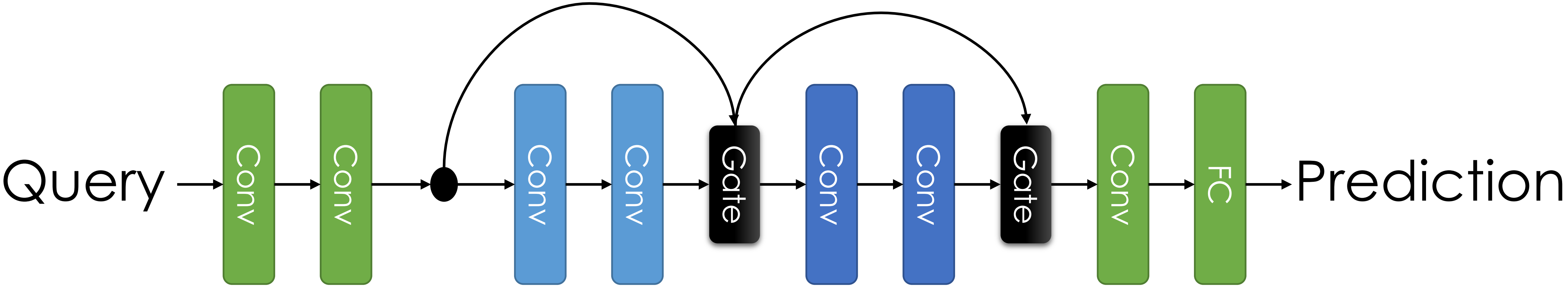
Fast

Prediction

Slow

Prediction

SkipNet: dynamic execution within a model


Query

Conv

Conv

Conv

Conv

Gate

Conv

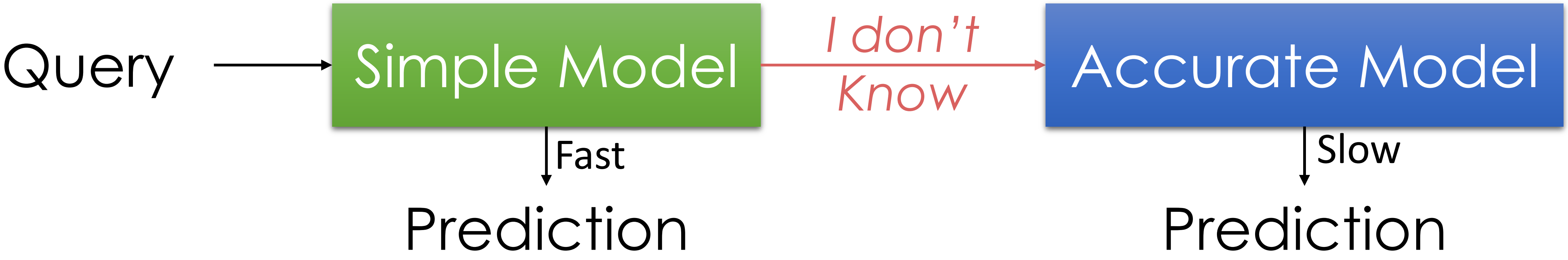
Conv

Gate

Conv

FC

Prediction



Query

Simple Model

I don't Know

Accurate Model

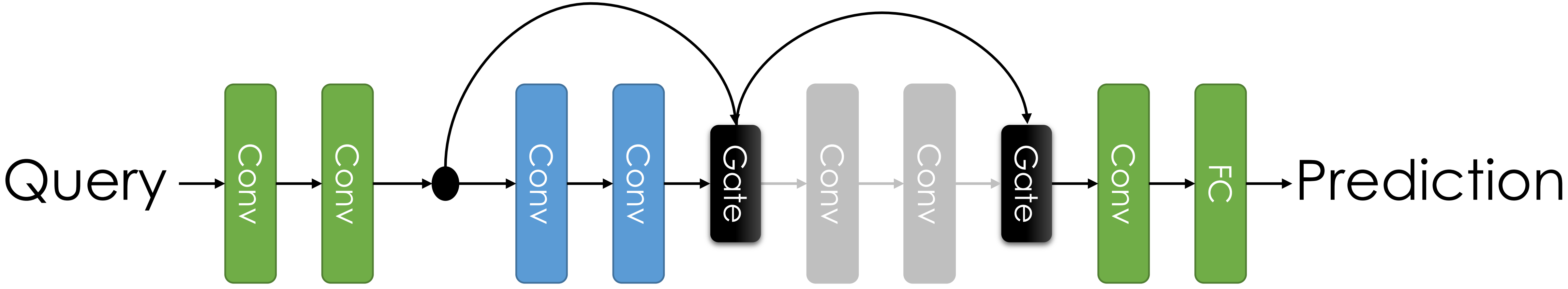
Fast

Prediction

Slow

Prediction

SkipNet: dynamic execution within a model



Query

Conv

Conv

Conv

Conv

Gate

Conv

Conv

Gate

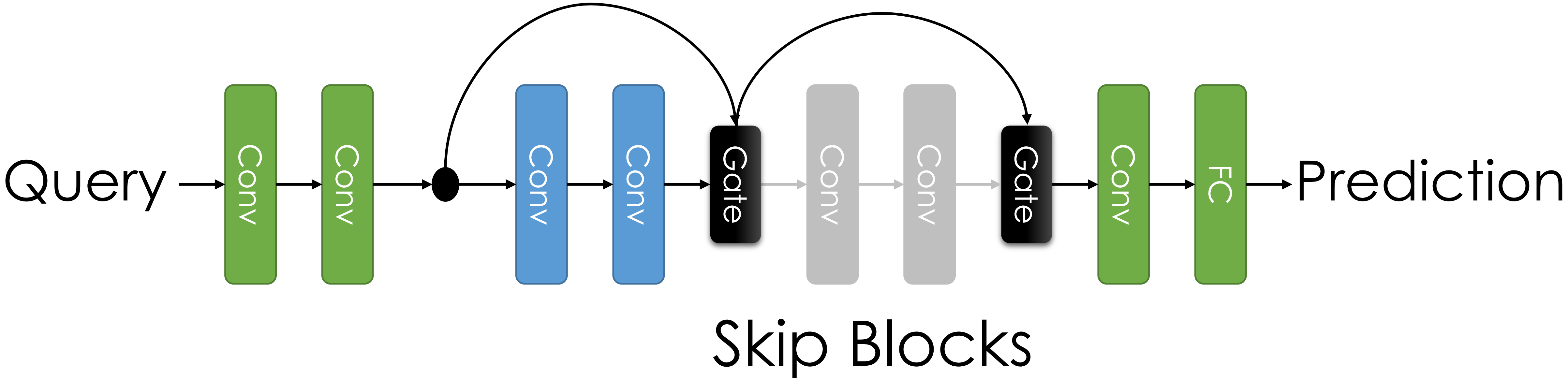
Conv

FC

Prediction

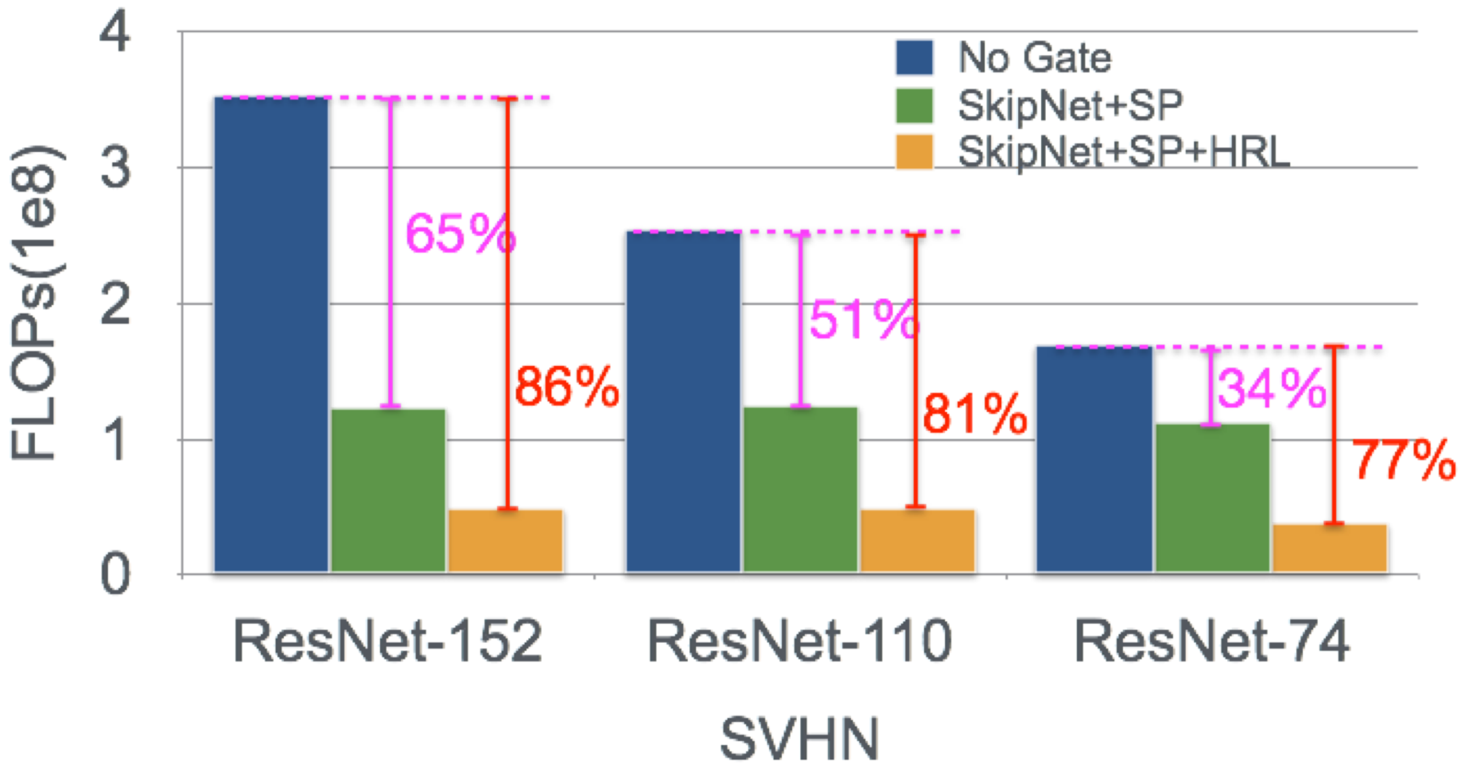
Skip Blocks

SkipNet: dynamic execution within a model



- Combine **reinforcement learning** with **supervised pre-training** to learn a gating policy

SkipNet Performance



Easy Images
Skip **Many** Layers

Hard Images
Skip **Few** Layers