

March 26, 2019

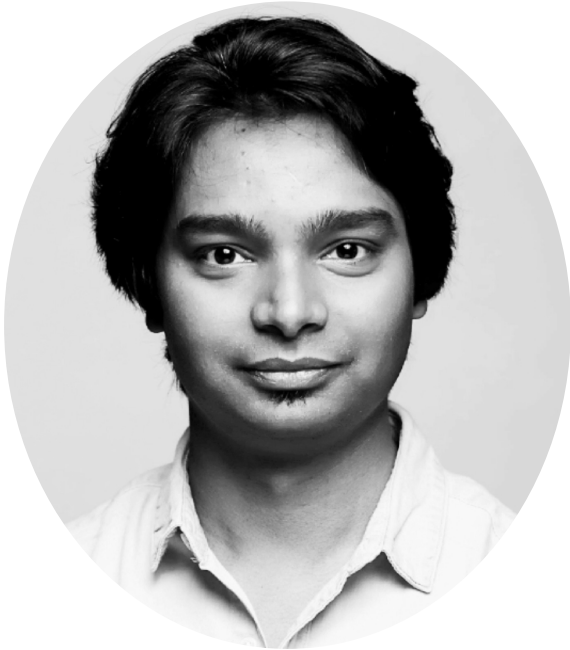
The Hitchhiker's Guide to Deep Learning-based Recommenders in Production

Abhishek Kumar
Pramod Singh

publicis
sapient

Strata
DATA CONFERENCE

About the Speaker



Abhishek Kumar

1. Sr. Manager Data Science, Publicis Sapient
2. Masters from University of California, Berkeley
3. Speaker @ O'Reilly Strata conference
4. Pluralsight Author

About the Speaker



Pramod Singh

1. Manager : Data Science, Publicis Sapient
2. Certified Data Scientist , IIM –Calcutta
3. MBA from Symbiosis University ,Pune
4. Published Author
 - Machine Learning using PySpark (Apress)

Why you should attend this session ?

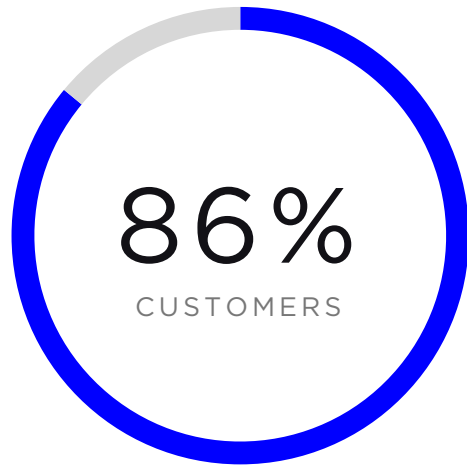
Deep Learning-based Recommenders in Production

Why Recommender Systems ?

Recommender Systems are Everywhere !



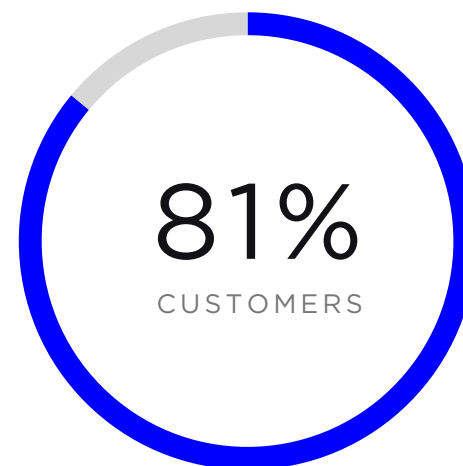
Research proves that consumer experience does matter



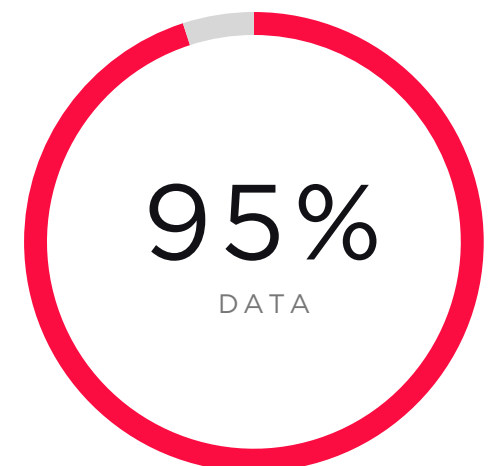
SAID THAT
PERSONALIZATION HAS HAD
SOME IMPACT ON
PURCHASING DECISION



IS SPENT ON PRODUCT
DISCOVERY & RESEARCH
ONLINE BY 50% CUSTOMERS



DEMAND IMPROVED
RESPONSE TIME

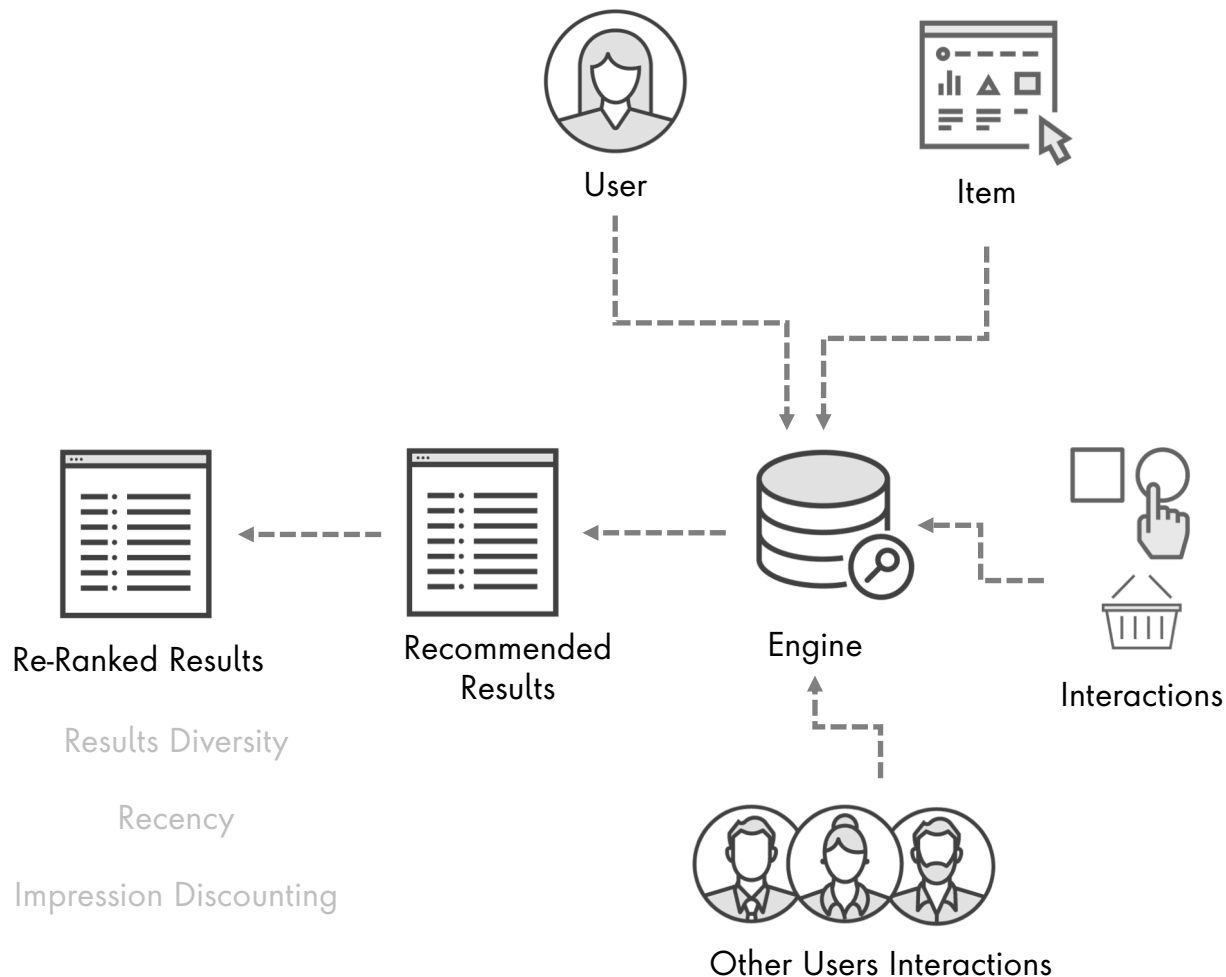


WITHIN ORGANIZATION
REMAINS UNTAPPED

Source : <http://www.nextopia.com/wp-content/uploads/2015/01/personalization-ecommerce-infographic.png>
<https://blog.hubspot.com/blog/tabid/6307/bid/23996/Half-of-Shoppers-Spend-75-of-Time-Conducting-Online-Research-Data.aspx>
<http://possible.mindtree.com/rs/574-LHH-431/images/Mindtree%20Shopper%20Survey%20Report.pdf>
<http://www.getelastic.com/using-big-data-for-big-personalization-infographic/>

Problem Space : Recommendation

Recommendation Engines



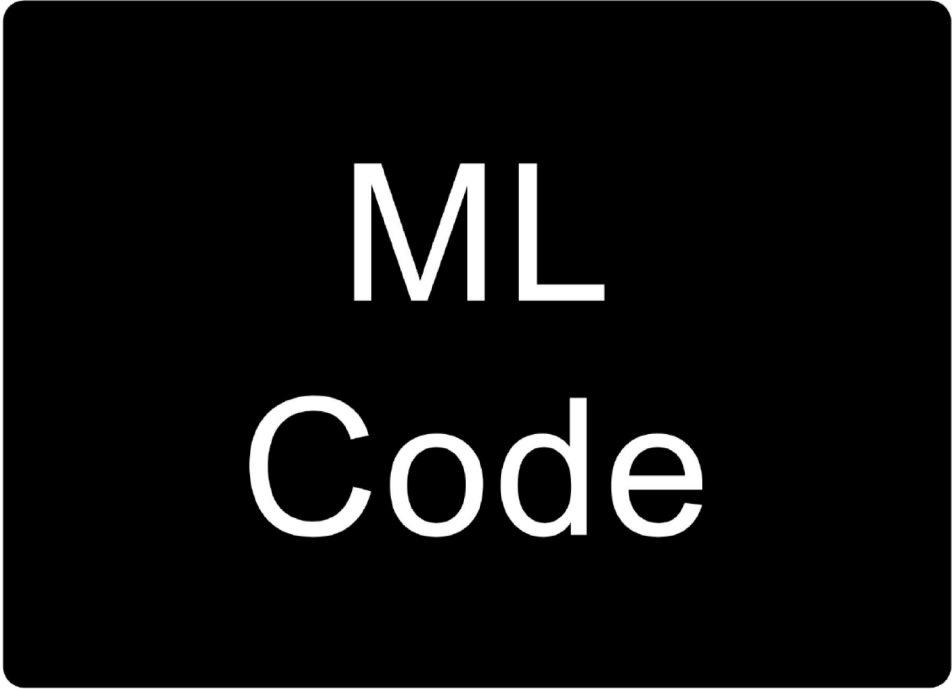
Challenges

- How to represent users and items?
- How to build hybrid systems with both interactions (collaborative) and user/item metadata ?
- How to use dynamic user behaviors?
- How to use implicit (view, share) feedback ?

Why Deep Learning based Recommenders ?

- Direct content Feature extraction instead of metadata
 - Text, Image, Audio
- Better representation of users and items for Recsys
- Hybrid algorithms and heterogeneous data can be used
- Better suited to model dynamic behavioral patterns and complex feature interactions

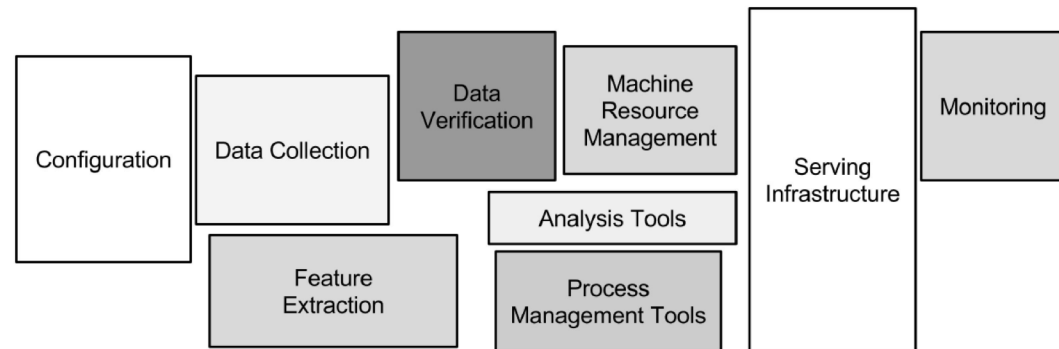
Deep Learning-based Recommenders in Production



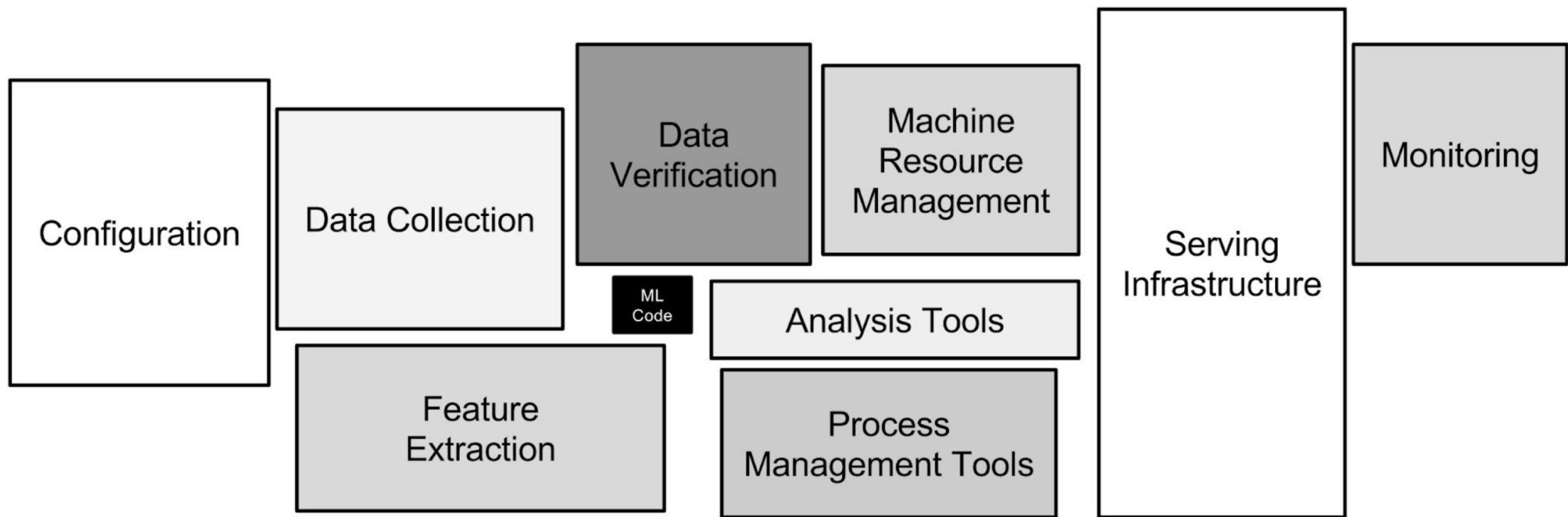
ML
Code

*Image Credits : "Kubeflow Explained" at Strata 2018 by Michelle Casbon

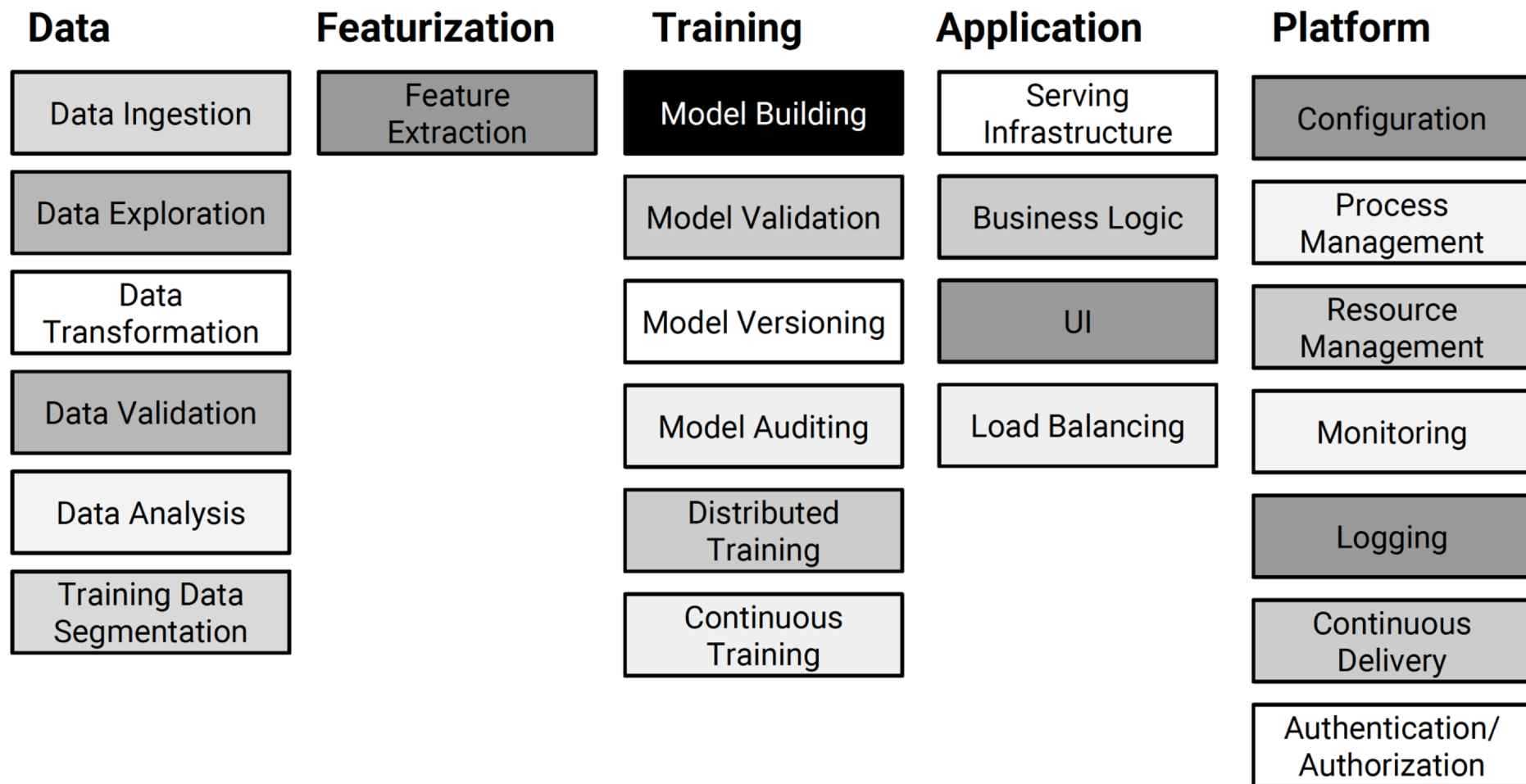
ML Code



*Image Credits : "Kubeflow Explained" at Strata 2018 by Michelle Casbon



*Image Credits : "Kubeflow Explained" at Strata 2018 by Michelle Casbon



*Image Credits : "Kubeflow Explained" at Strata 2018 by Michelle Casbon

What you will learn today ?

Experimentation/
Development

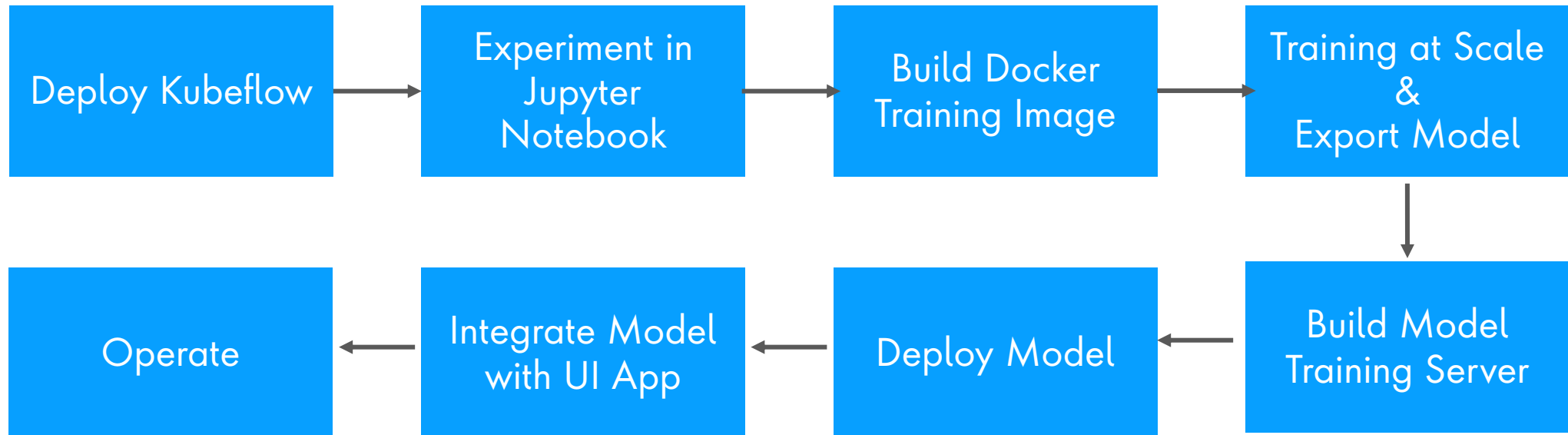
Training @Scale

Monitoring

Serving

Kubeflow

What you will learn today ?



What you will build today ?

Demo

- Add screen shot of
 - Ambassador
 - Jupyter hub
 - Training @scale (Pods)
 - Serving
 - Final UI

You can apply the learning of this session in any machine learning / deep learning based projects / products.



End-To-End ML/DL Workflow

How to make best out of today's session

- Fully hands on session where we will build end to end DL based recommender systems
- For environment access
 - Local
 - GCP (Recommended)
- Code labs are created for you as a step by step guide
- <http://bit.ly/strata19-sf>

- 1 Overview of the tutorial
- 2 Getting Started
- 3 Development on Jupyter Notebook
- 4 Training At Scale
- 5 Serving
- 6 Deploy UI
- 7 Clean up

← Strata Conference SF 2019 : The hitchhiker's guide to deep learning-based recommenders in production

1. Overview of the tutorial

In this tutorial, You will learn to setup and use [Kubeflow](#) for building and deploying deep learning based recommendation engine. Kubeflow an open-source project which aims to make running ML workloads on Kubernetes simple, portable and scalable. Kubeflow adds some resources to your cluster to assist with a variety of tasks, including training and serving models and running Jupyter Notebooks. It also extends the Kubernetes API by adding new Custom Resource Definitions (CRDs) to your cluster, so machine learning workloads can be treated as first-class citizens by Kubernetes.

What You'll Learn

- How to set up a Kubeflow cluster on GCP
- How to use Jupyter Hub on Kubeflow to run jupyter notebook
- How to use Tensorflow to build deep learning based recommendation engine

What You'll Need

- An active [GCP project](#)
- Access to the Google Cloud Shell, available in the [Google Cloud Console](#)
- If you'd prefer to complete the codelab on a local machine, you'll need to have [gcloud](#), [kubecti](#), and [docker](#) installed

You can create a free GCP account with \$300 credit. It would be sufficient for the purpose of this tutorial.

Section 1

RecSys 101

RecSys 101 : What is RecSys?

“Serve the **relevant** items to users in an **automated** fashion to optimize **short and long term business objectives**”

RELEVANT (WHAT)

1. Novelty
2. Serendipity
3. Diversity

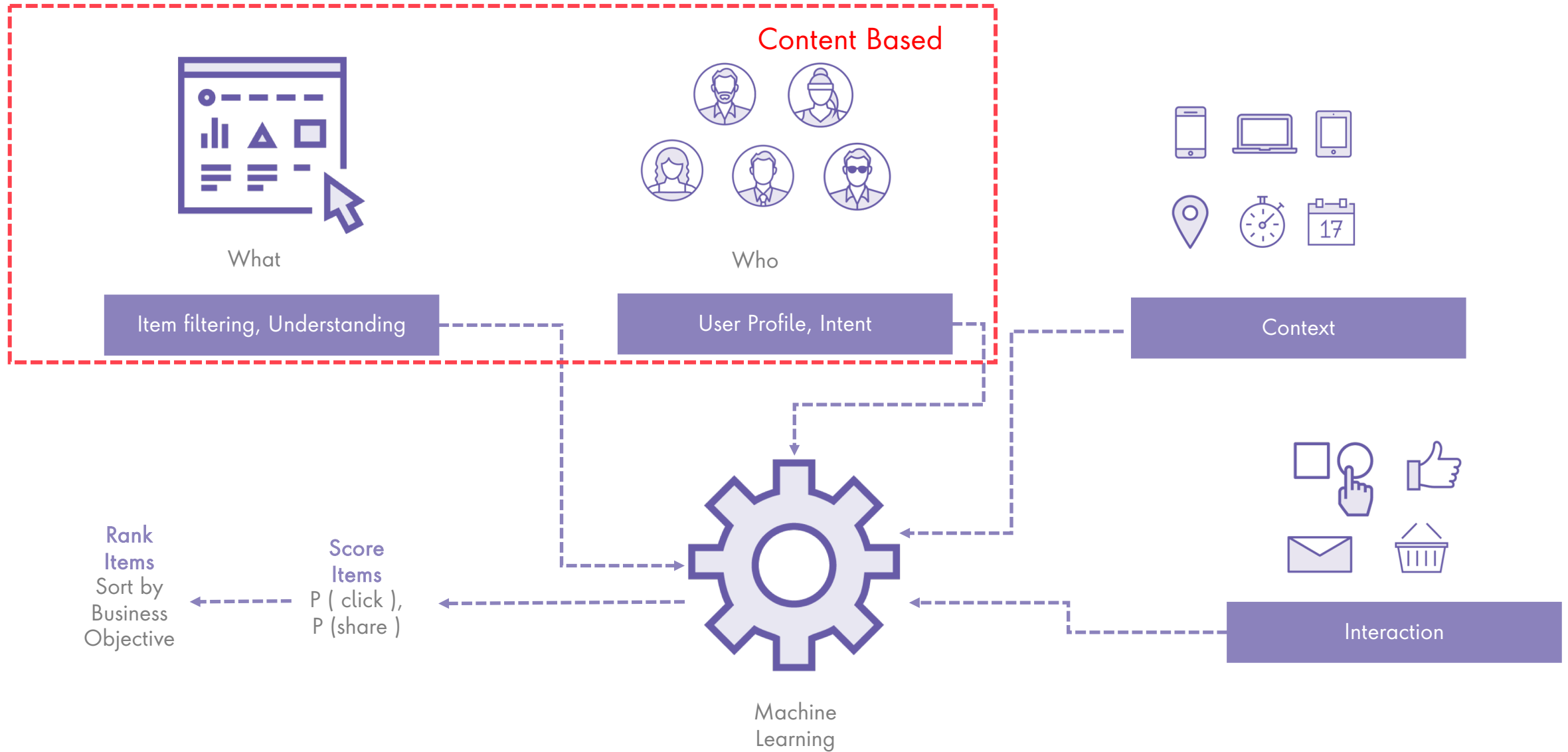
AUTOMATED (HOW)

1. No manual intervention
2. Scale Up

BUSINESS OBJECTIVES (WHY)

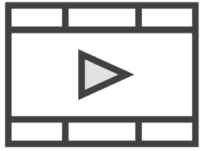
1. Short Term Business Objectives
 - a. High clicks
 - b. Revenue
 - c. Positive explicit ratings
2. Long Term Business Objectives
 - a. Increased engagement
 - b. Increase in social action
 - c. Increase in Subscriptions

RecSys 101 : Internals



RecSys 101 : Content Based Recommendation

Recommends an item to a user based upon a description of the item and a profile of the user's interests



Representing Items using Features

Drama	Arty	Comedy	Action	Commercial
0.7	0	0.2	0				0.8



User Profile

Creating a user profile that describes the types of items the user likes/dislikes

RecSys 101 : Content Based Recommendation



- More than 100 million monthly active users
- Over 30 million songs



Track: May 16

Artist: Lagwagon

Album: Let's Talk About Feelings

Release: 1998

```
{  
  "danceability" : 0.560,  
  "energy" : 0.527,  
  "key" : 2,  
  "loudness" : -9.783,  
  "mode" : 1,  
  "speechiness" : 0.0374,  
  "acousticness" : 0.516,  
  "instrumentalness" : 0.0000240,  
  "liveness" : 0.156,  
  "valence" : 0.336,  
  "tempo" : 93.441,  
  "type" : "audio_features",  
  "id" : "2z7D7kbpRcTvEdT71tdiNQ",  
  "uri" : "spotify:track:2z7D7kbpRcTvEdT71tdiNQ",  
  "track_href" : "https://api.spotify.com/v1/tracks/2z7D7kbpRcTvEdT71tdiNQ",  
  "analysis_url" : "http://echonest.com/analysis/2z7D7kbpRcTvEdT71tdiNQ",  
  "duration_ms" : 168720,  
  "time_signature" : 4  
}
```

RecSys 101 : Content Based Recommendation

Pros

No need of other users data

Easy to understand reason behind recommendation

Capable of recommending new and unknown items

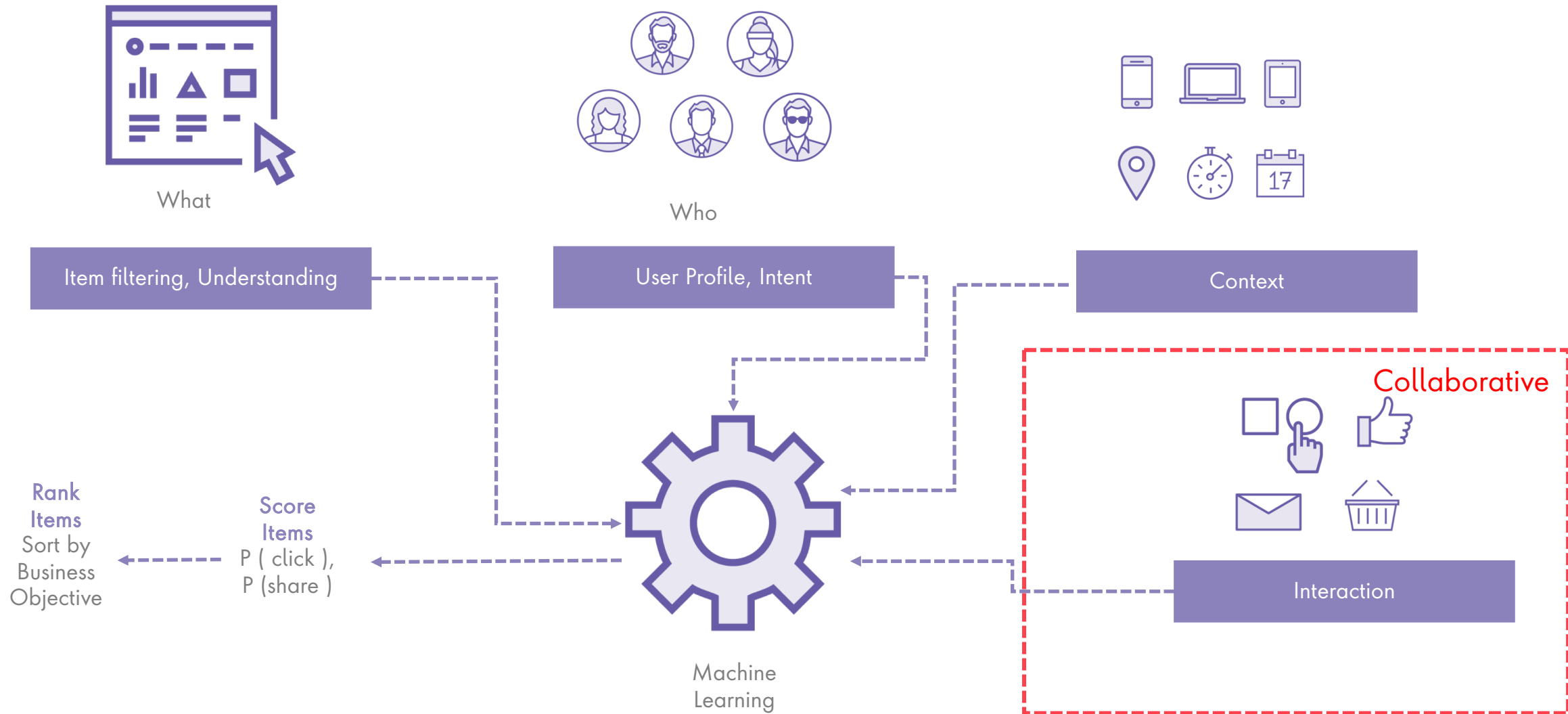
Cons

Can only be effective in limited circumstances

No suitable suggestions if content doesn't have enough information

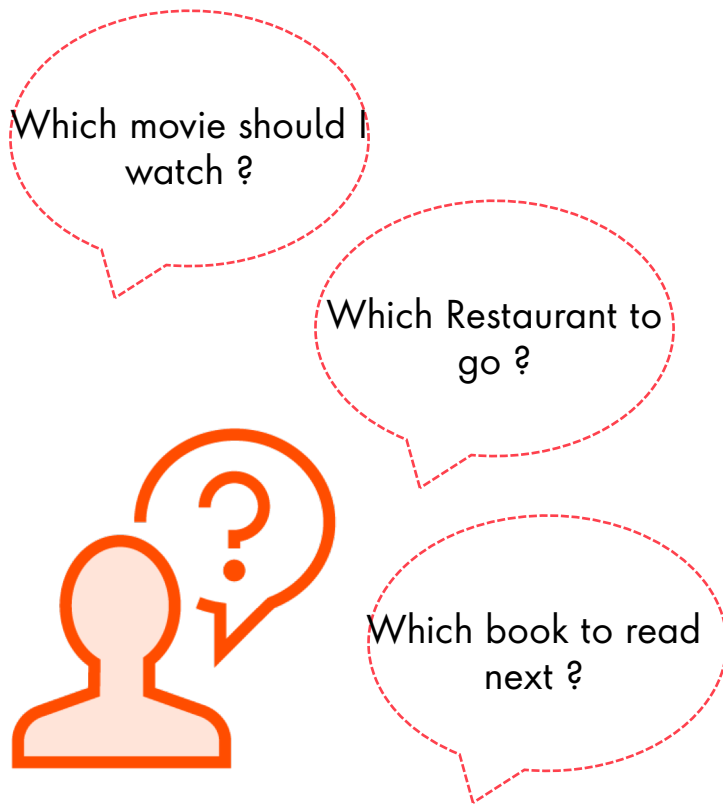
Depend entirely on previous selected items and therefore cannot make predictions about future interests of users

RecSys 101 : Internals



RecSys 101 : Collaborative Filtering

Unlike Content based filtering , Collaborative Filtering doesn't require any product description at all



RecSys 101 : Collaborative Filtering : Interactions / Feedback



Explicit



Ratings

Implicit



Purchased



Add to cart

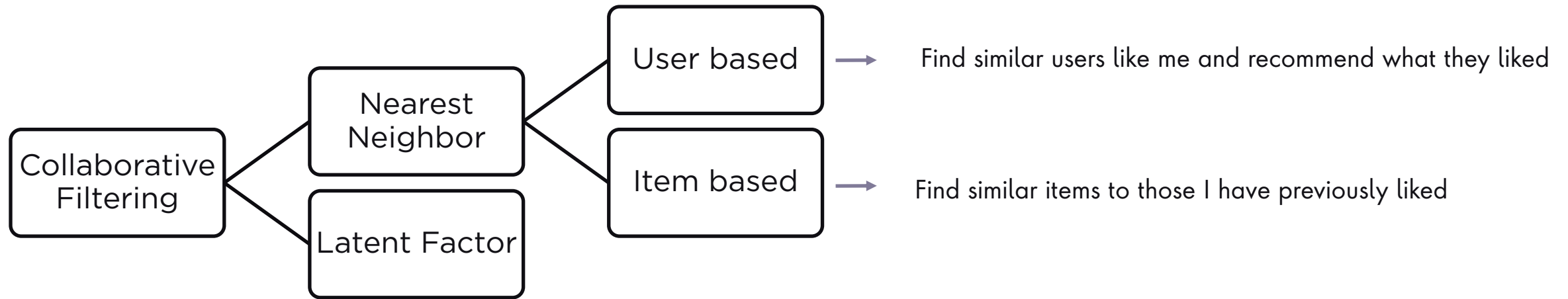


Viewed



Shared

RecSys 101 : Collaborative Filtering



Factor based techniques (Matrix Factorization, Factorization Machine)

- Scalability
- Predictive accuracy
- Can model real-life situations (e.g. Biases, Additional Input sources , Temporal Dynamics)

\$ 1 Million Netflix Challenge

RecSys 101 : Collaborative Filtering : Latent Factor

Take the users and their feedback for different items and identify hidden factors that influence the user feedback

The idea is to factorize or decompose the user item matrix into two matrices

- Users are mapped on to hidden factors
- Items are mapped on to hidden factors

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	X		X		X	
User 2		X	X			
User 3				X		X
User 4					X	
User 5	X	X		X		X
User 6			X	X		
User 7	X	X	X		X	X
User 8		X		X		
User 9			X			

R

\approx

	UF1	UF2
User 1		
User 2		
User 3		
User 4		
User 5		
User 6		
User 7		
User 8		
User 9		

U

X

V

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
IF1						
IF2						

RecSys 101 : Collaborative Filtering

Pros

Content information not required either of users or items

Personalized recommendations using other user's experience

No domain experience required

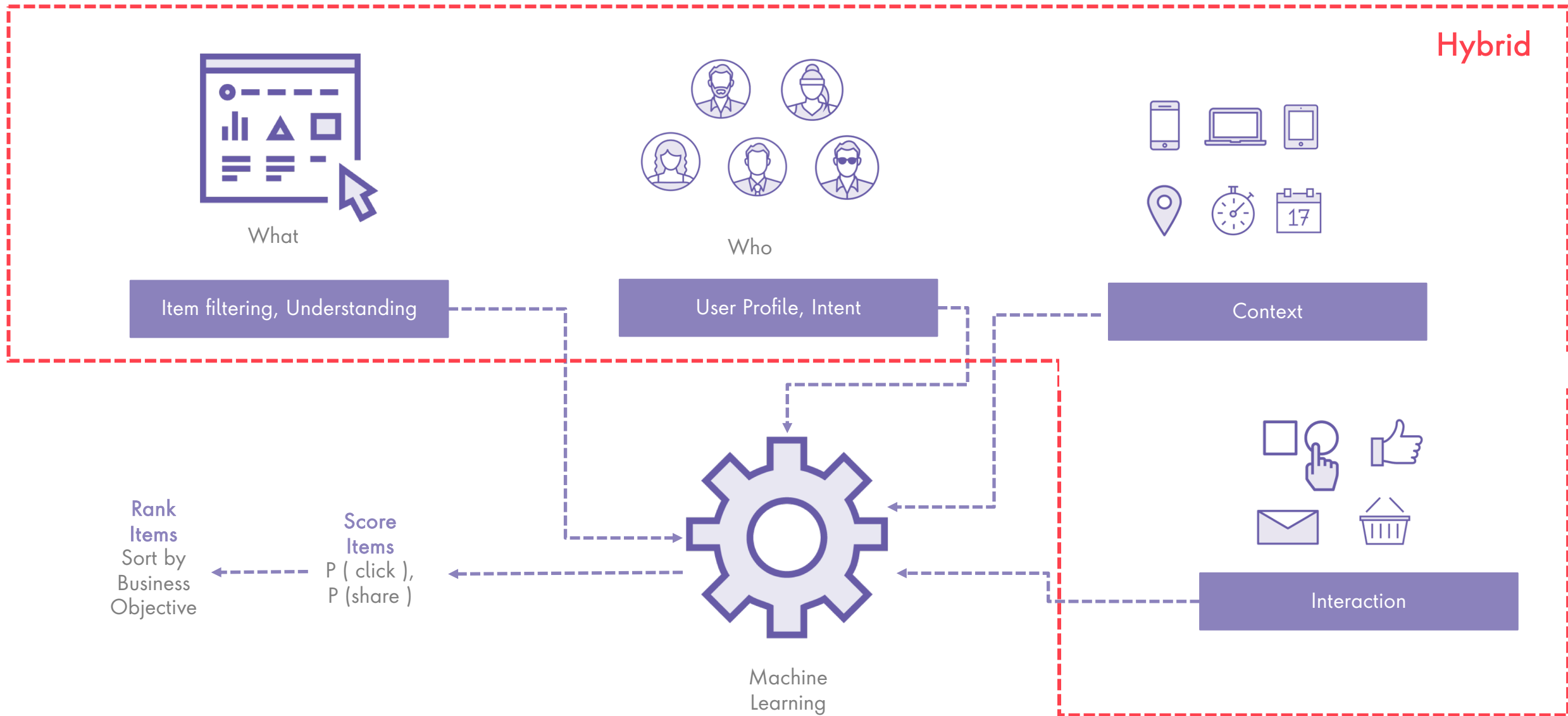
Cons

Cannot produce recommendations if there is no interaction data available (**Cold Start Problem**)

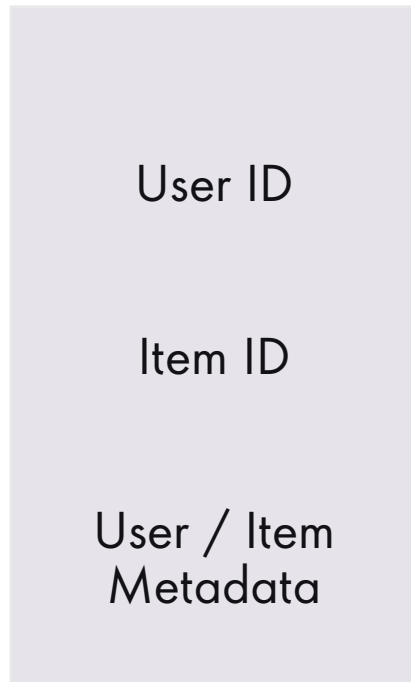
Often demonstrate poor accuracy when there is little data about users' ratings (**Sparsity**)

Popular items get more feedback (**Popularity bias**)

RecSys 101 : Internals

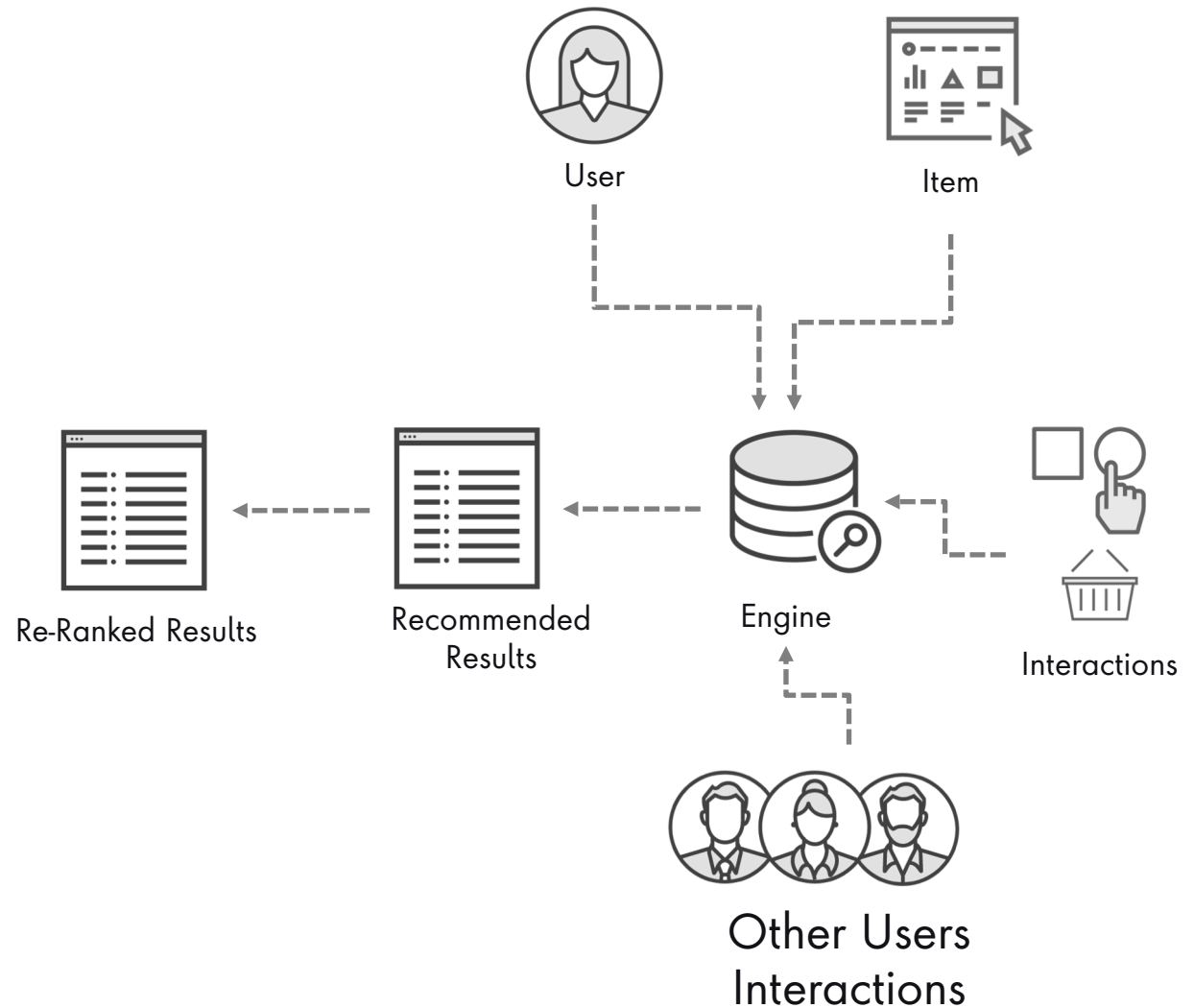


Representation : A Key Aspect



Representation

Recommendation Engines



Matrix Factorization

Items

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	X		X		X	
User 2		X	X			
User 3				X		X
User 4					X	
User 5	X	X		X		X
User 6			X	X		
User 7	X	X	X		X	X
User 8		X		X		
User 9			X			

Users

R

\approx

	UF1	UF2
User 1		
User 2		
User 3		
User 4		
User 5		
User 6		
User 7		
User 8		
User 9		

U

X V

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
IF1						
IF2						

How to better represent users and items ?

What about item and user metadata ?

RecSys 101 : Hybrid Recommendation Engine

Pros

Solve the issue of Cold Start by leverage both content and collaboration

Use of Implicit feedback reduces the sparsity issues to a large extent

Can include higher order feature interactions as well

Cons

Difficult to implement

Matrix Factorization

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	X		X		X	
User 2		X	X			
User 3				X		X
User 4					X	
User 5	X	X		X		X
User 6			X	X		
User 7	X	X	X		X	X
User 8		X		X		
User 9			X			

$$R$$

211

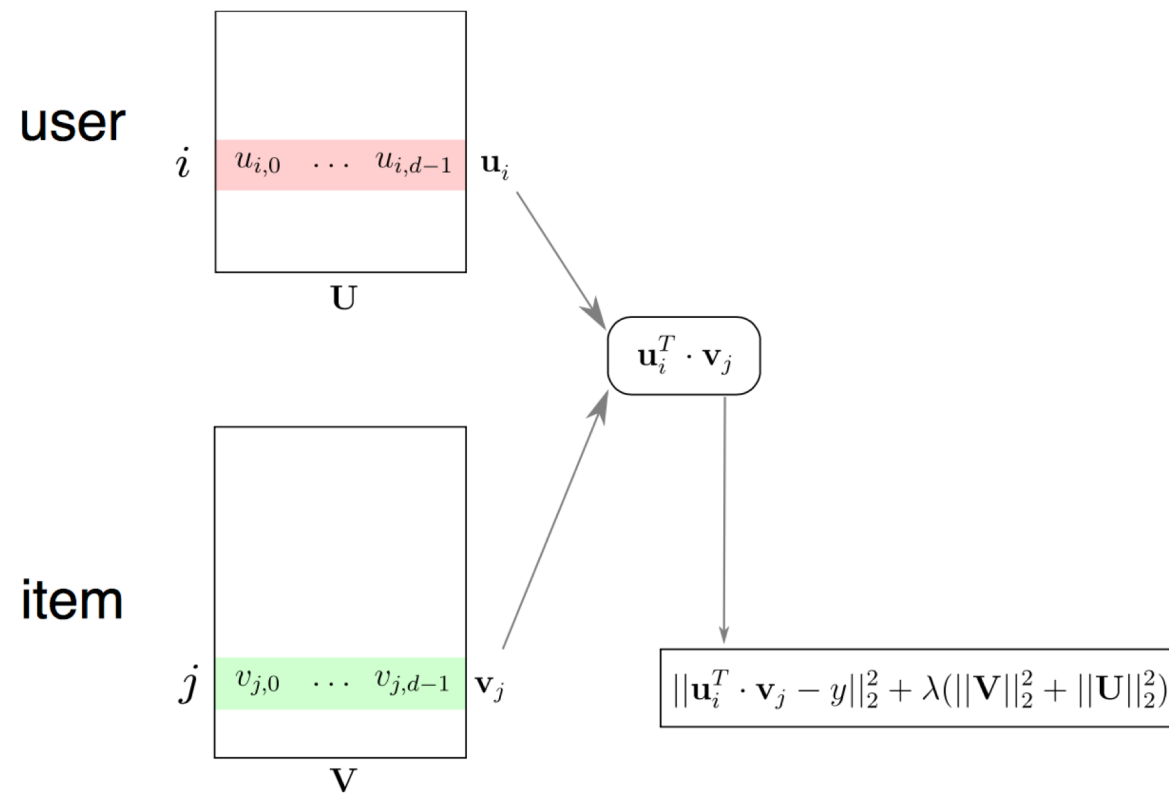
	UF1	UF2
User 1		
User 2		
User 3		
User 4		
User 5		
User 6		
User 7		
User 8		
User 9		

$$U$$

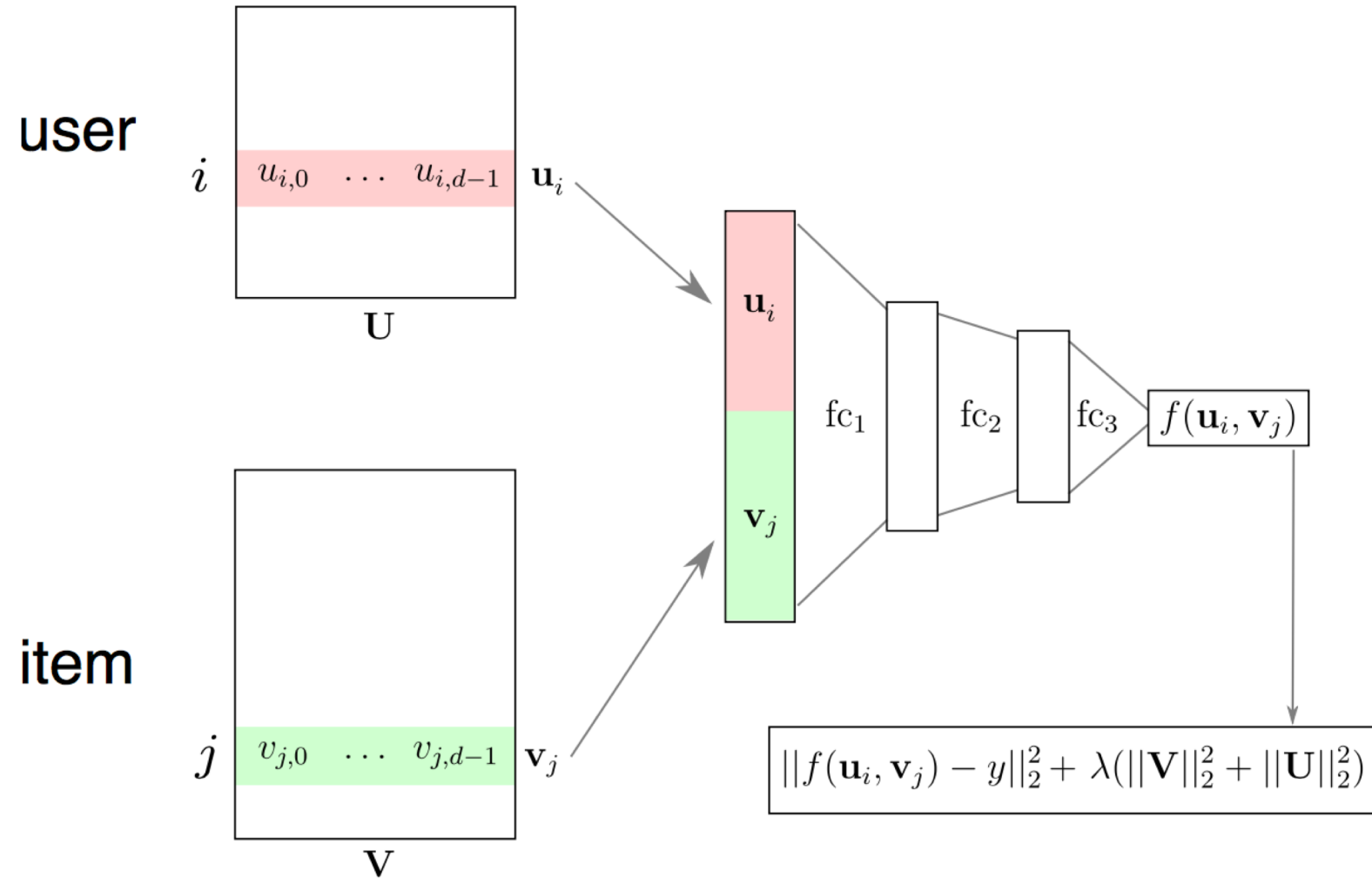
X

$$V$$

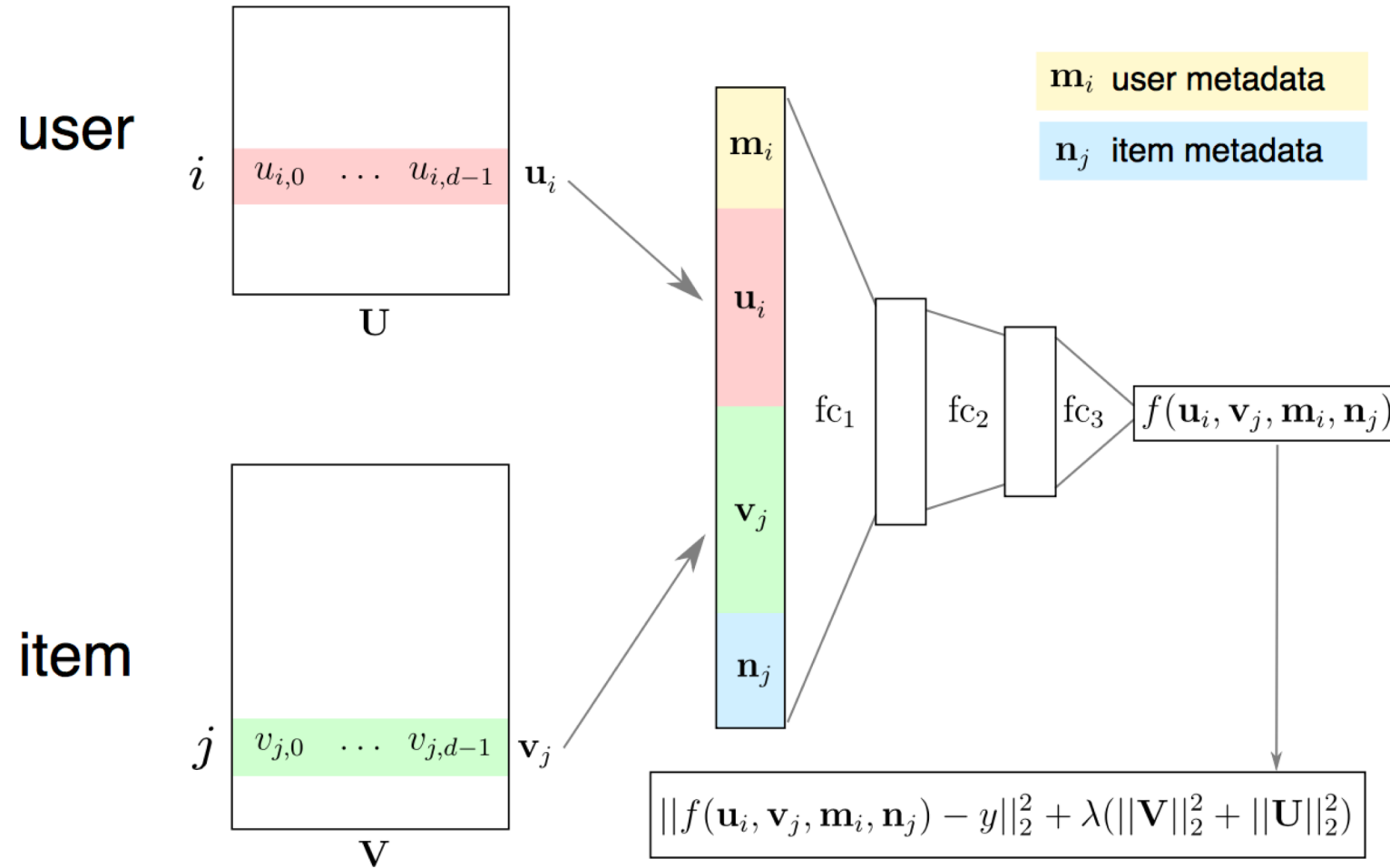
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
IF1						
IF2						



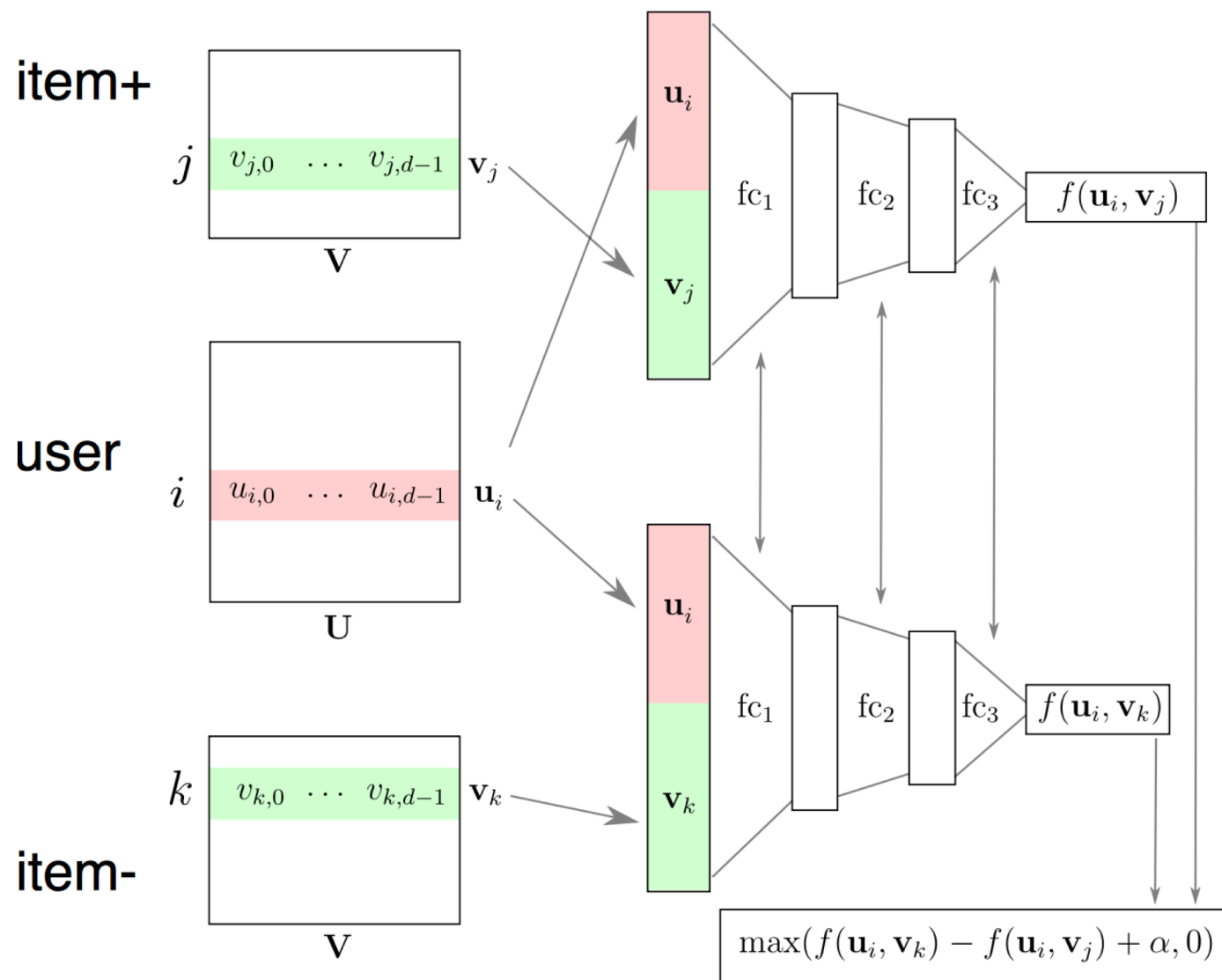
Matrix Factorization : Deep Neural Networks



Matrix Factorization : Deep Neural Networks with Metadata



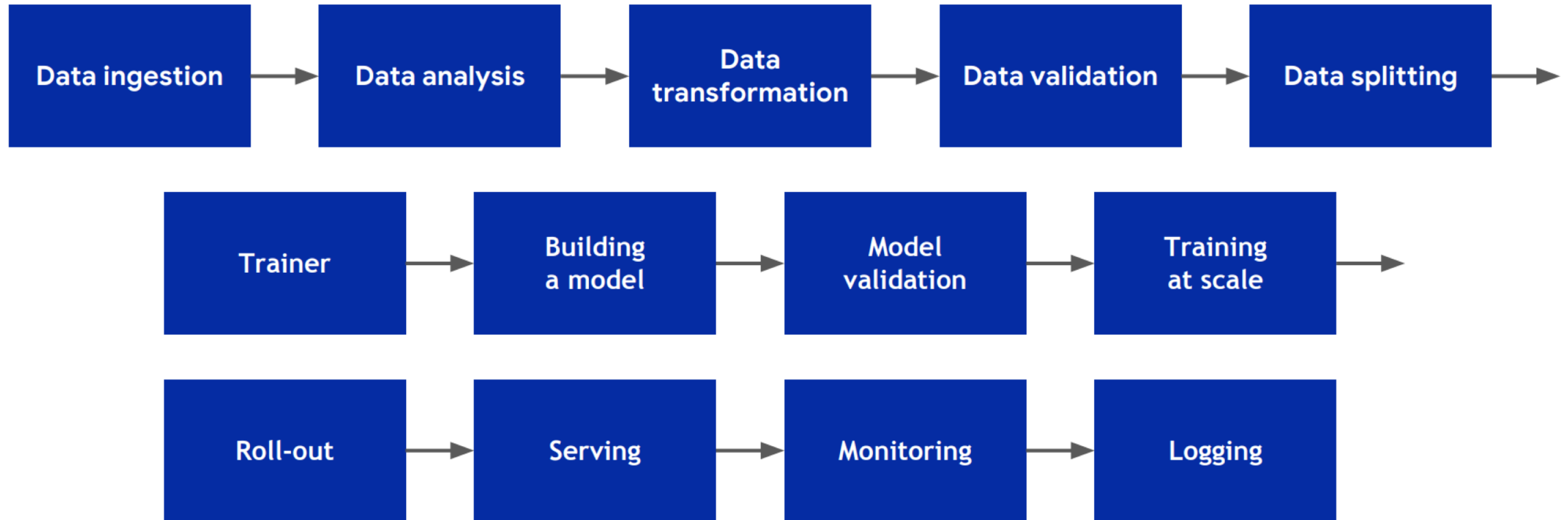
Deep Triplet Network with Implicit Feedback



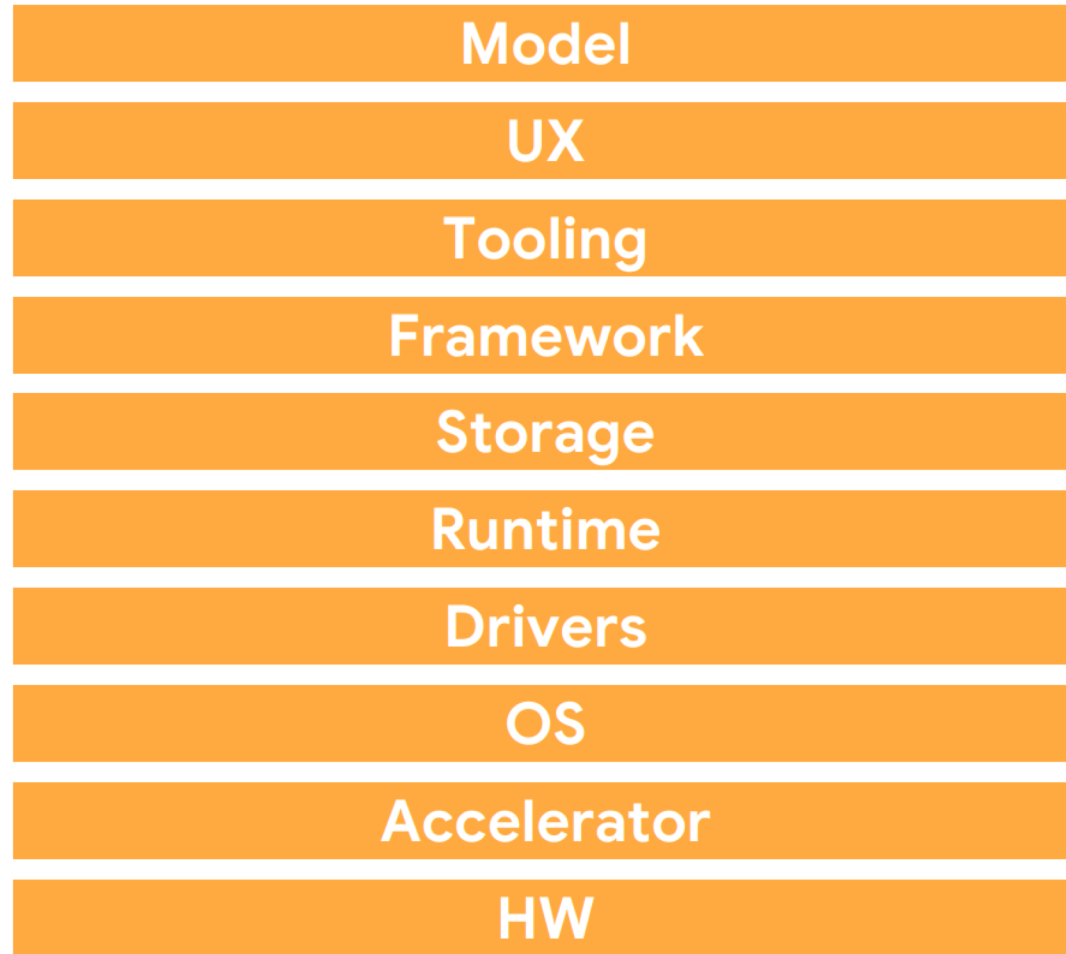
Section 2

Kubeflow

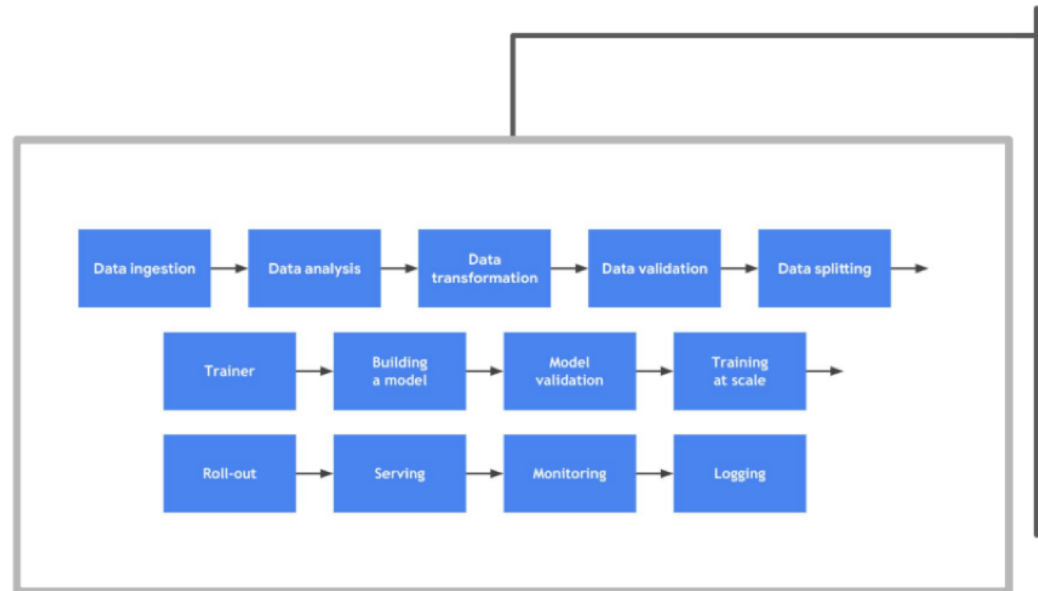
Platform



Experimentation

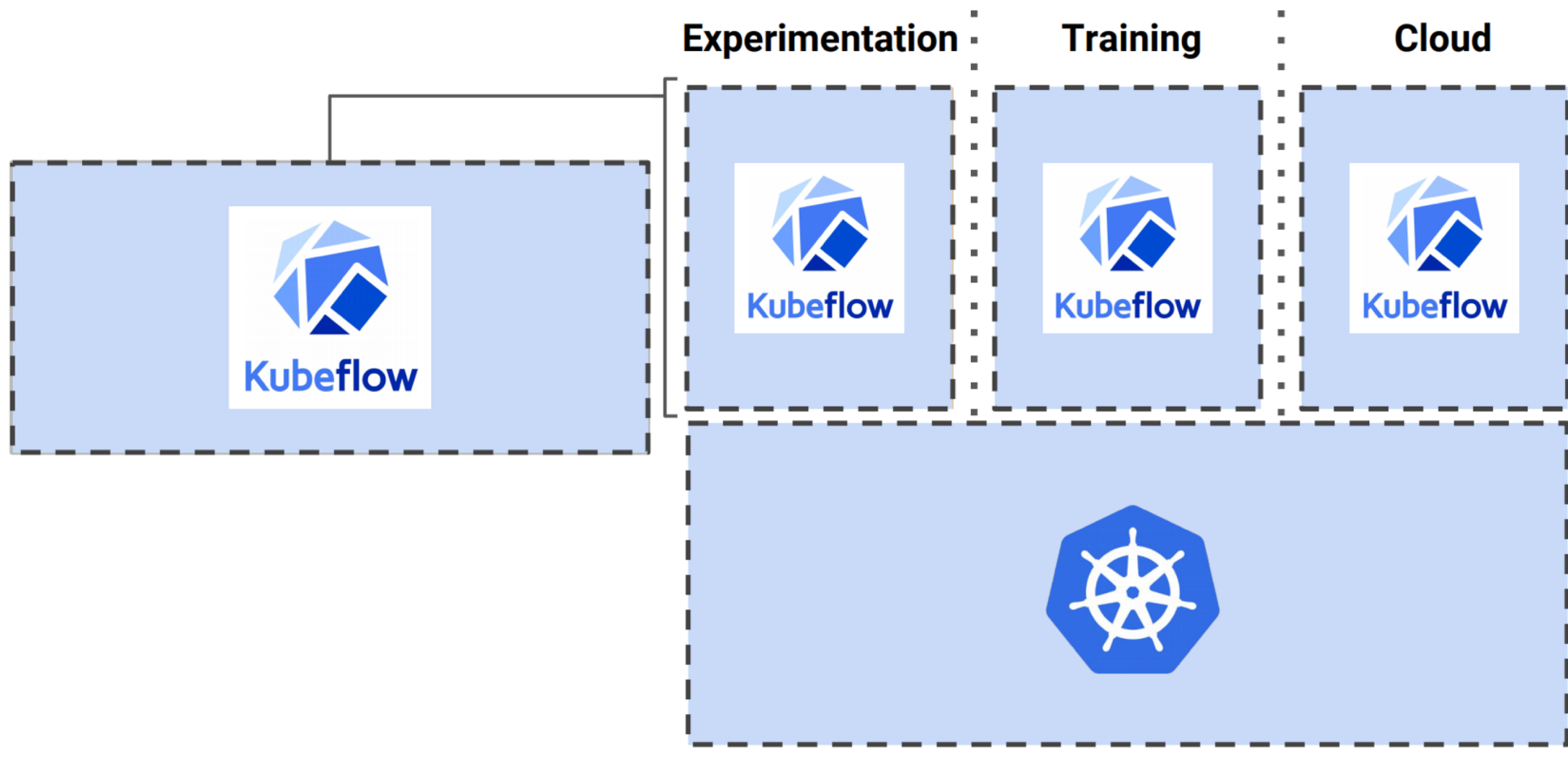


Experimentation





*Image Credits : "Machine Learning as code" at Kubecon 2018 by David Aronchick and Jason Smith



*Image Credits : "Machine Learning as code" at Kubecon 2018 by David Aronchick and Jason Smith

What is Kubeflow ?



“The Machine Learning Toolkit for Kubernetes”

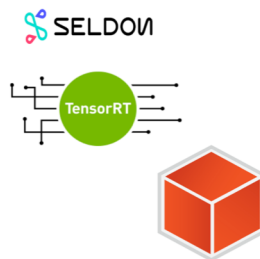
A curated set of compatible tools and artifacts that lays a foundation for running production ML apps



Notebook



TF Model Training



TF serving
Seldon,
TensorRT



Pipelines



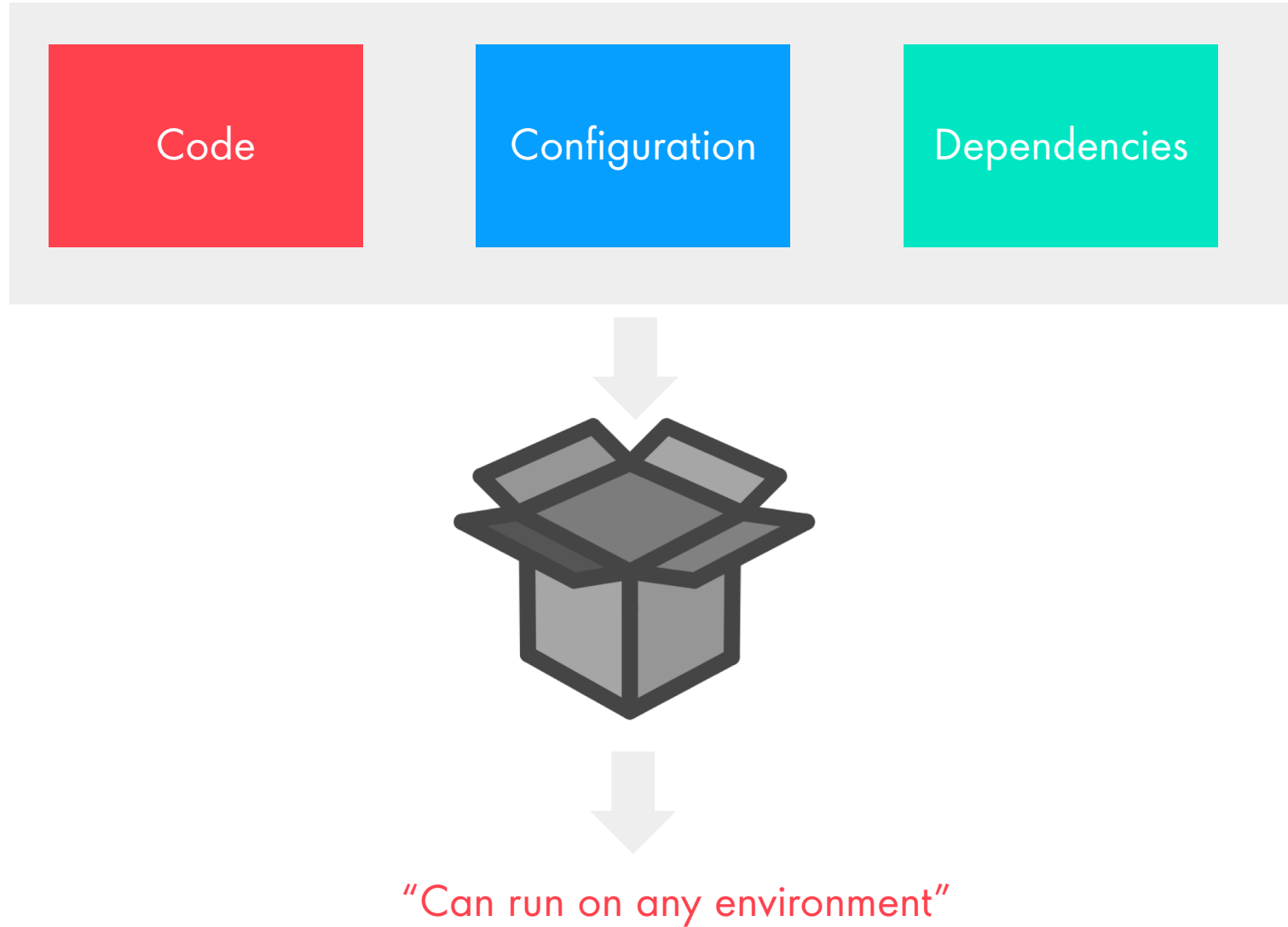
Multi-framework
Integration

Make it Easy for Everyone to Develop, Deploy and Manage
Portable, Distributed ML
on Kubernetes



Kubeflow Mission

What are Containers ?



What is Kubernetes ?

- Provides runtime environment for Docker Containers (encapsulate application and its dependencies)
- Scale and load balance docker containers
- Abstract the infra for containers
- Declarative definition for running containers
- Perform update (or rolling update)
- Service discovery and exposure

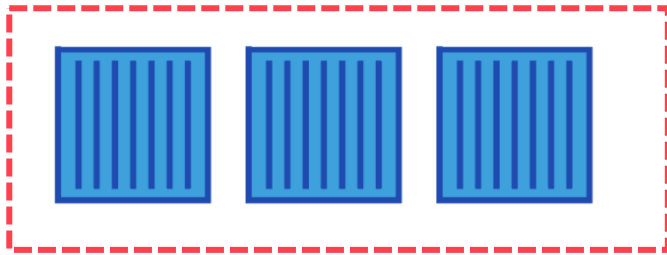
Make life easy while working with containers



kubernetes

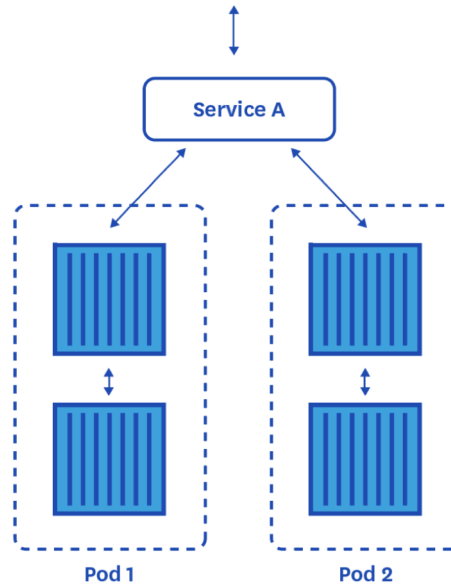
Just Enough K8S core pieces

POD



Group of one or More Containers

Service



Single Exposed Component for Multiple copies of Pods

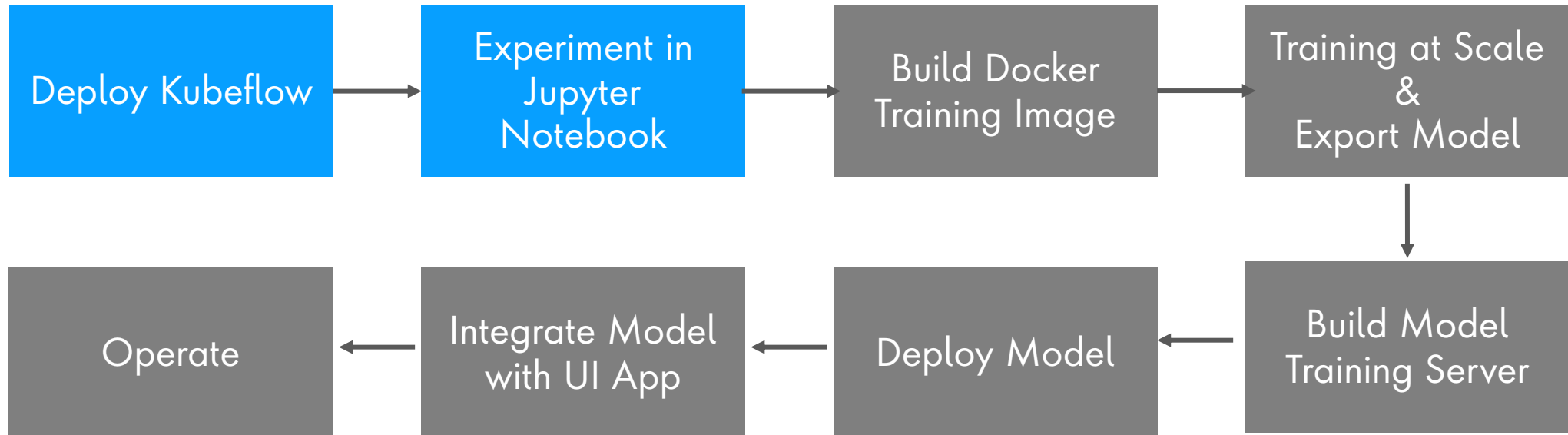
Manifest File

```
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: rss-site
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: front-end
          image: nginx
          ports:
            - containerPort: 80
        - name: rss-reader
          image: nickchase/rss-php-nginx:v1
          ports:
            - containerPort: 88
```

Declarative way for your required resources

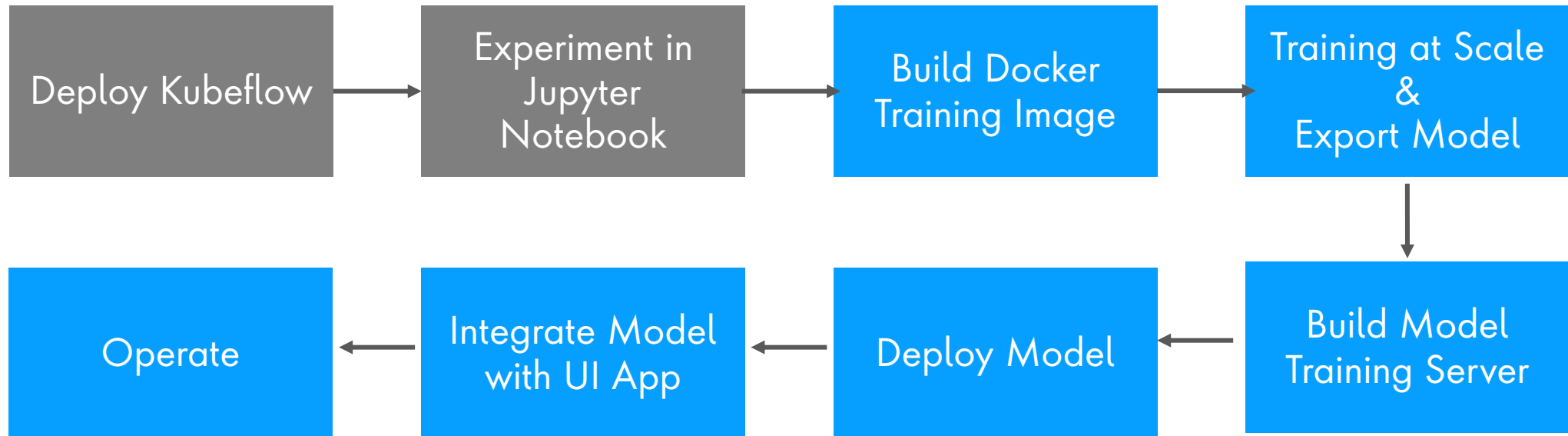
Lab : Building and Evaluating Deep Learning Based Recommenders

What we will do in this Codelab ?



Break

What we will cover after the break ?

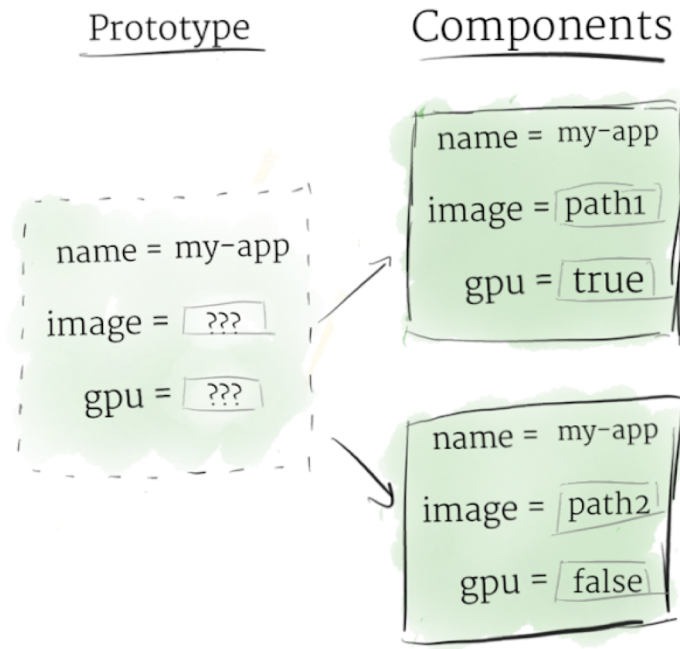


+
Reference Architectures
Hyper-Parameter Tuning (Katib)
CI / CD Pipeline (Argo)
Best Practices

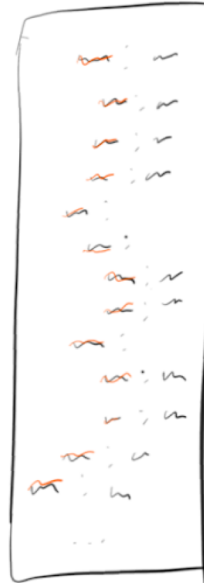
Lab : Building and Evaluating Deep Learning Based Recommenders - Continue

Ksonnet

ksonnet

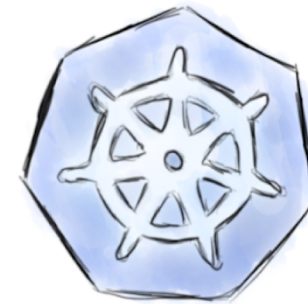


YAML

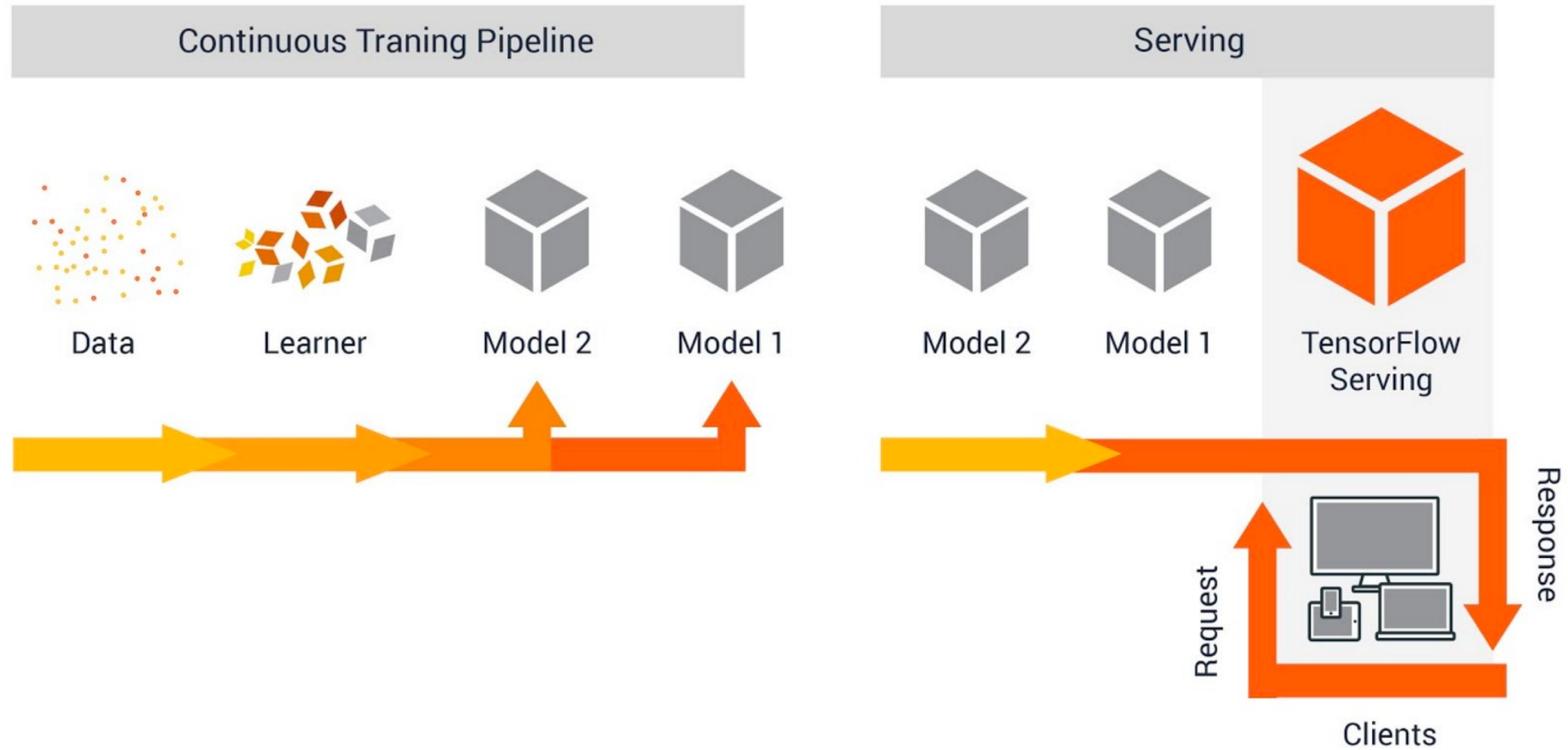


Kubernetes

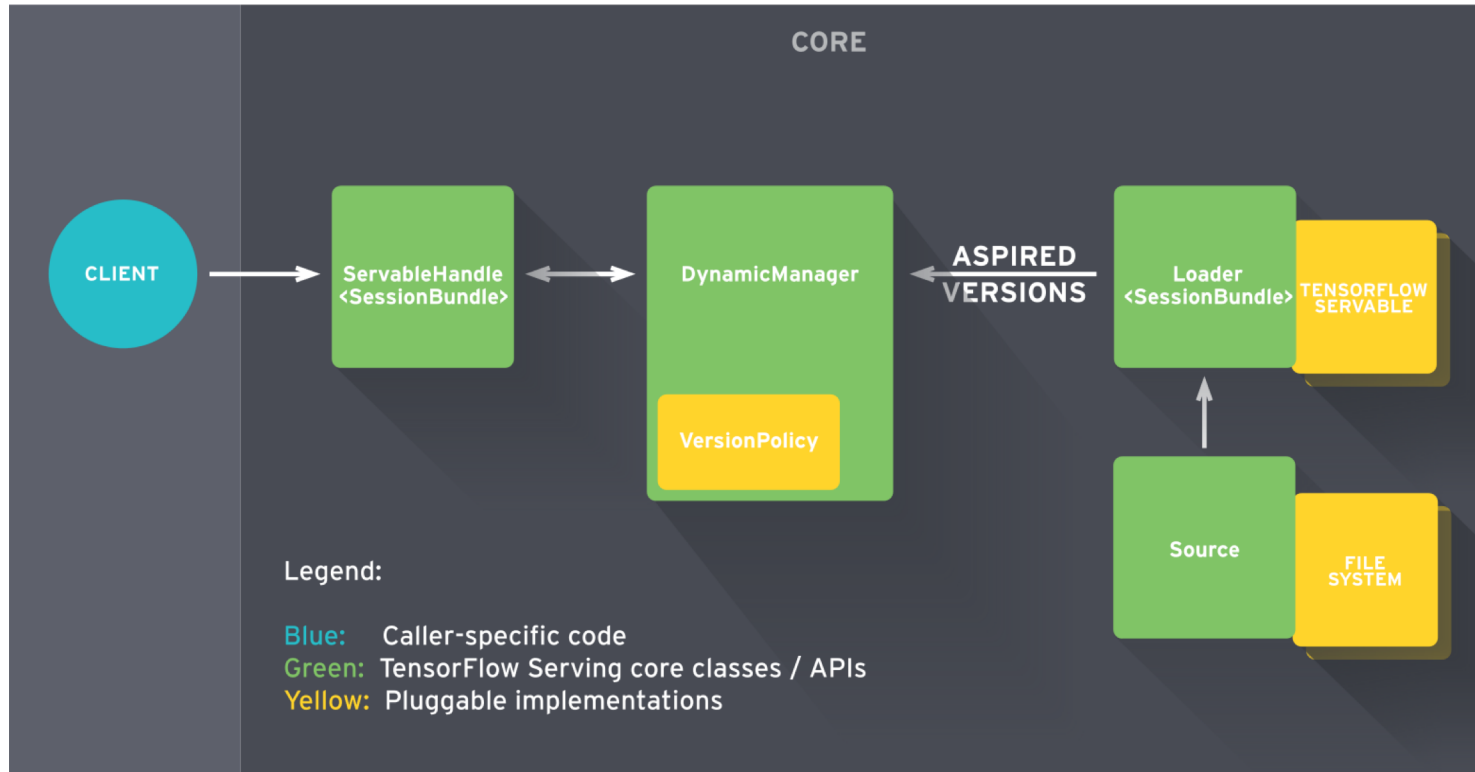
Kubernetes Resources



Tensorflow Serving

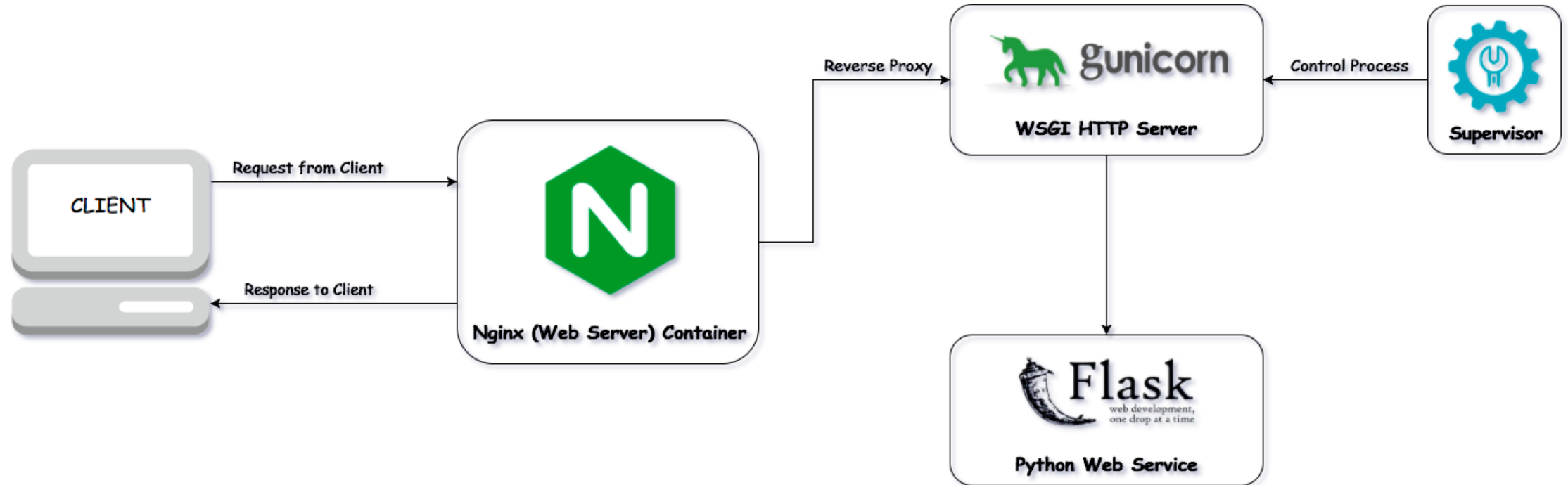


Tensorflow Serving



- Low latency inference
- Model versioning and rollback
- Custom version policy for A/B and Bandit tests
- Uses highly efficient gRPC and Protocol Buffers

API Layer



Reference Architecture

Data Ingestion and
Processing Pipeline

Model & Testing
Framework

Machine Learning
Server

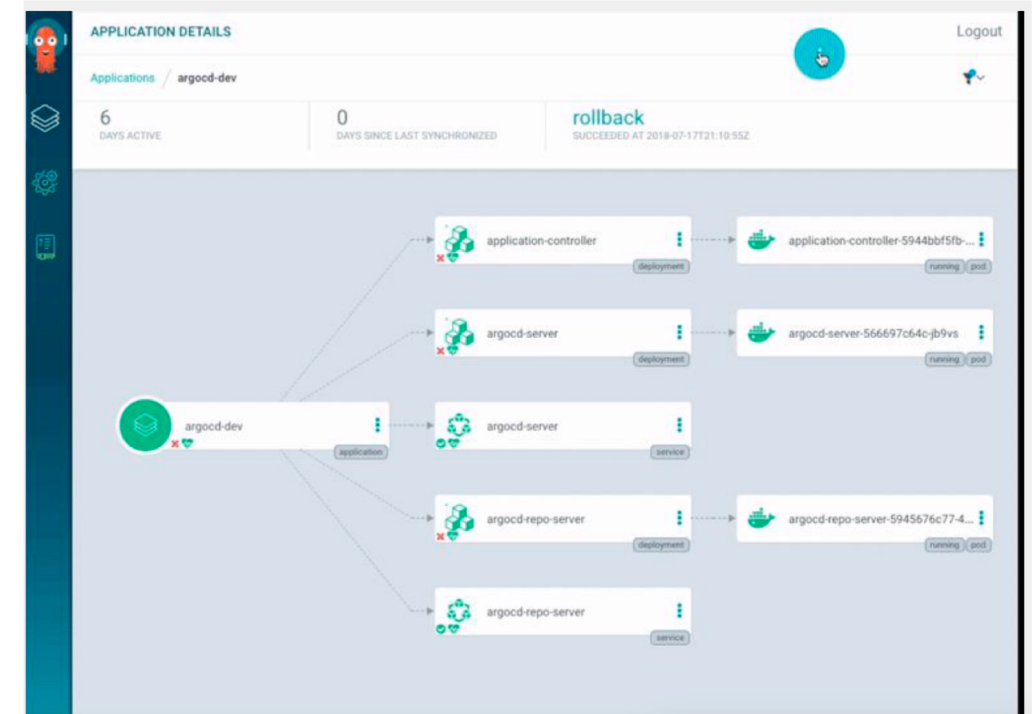
Hyper-Parameter Tuning

Katib (by NTT)

- Pluggable micro-service
- Multiple architecture for Hyper-Parameter tuning (Grid, Random, Bayesian)
- Different optimization algorithms Different frameworks

StudyJob (K8s CRD)

- Hides complexity from user
- No code needed to do hyper-parameter tuning

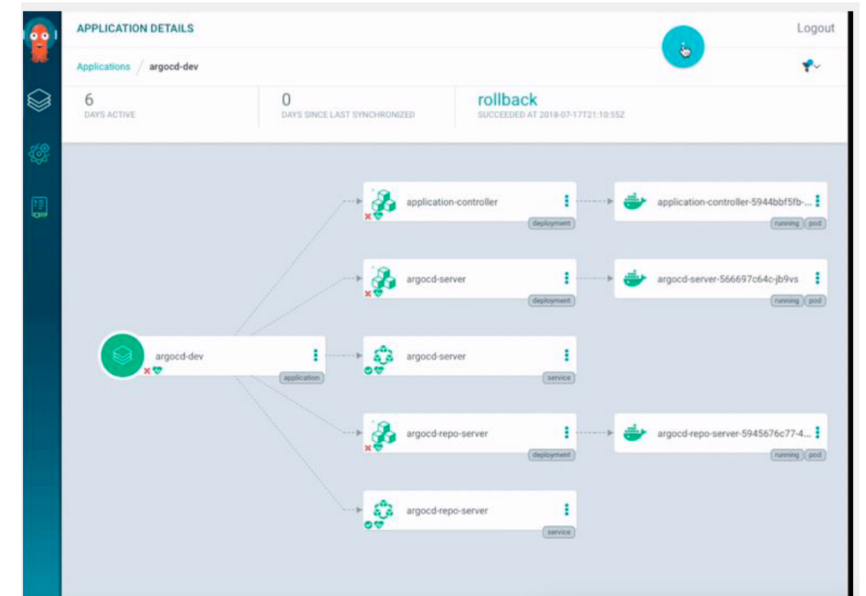


Synchronize Kubeflow development with Git

CI / CD Pipeline

[Argo CD](#) is a Kubernetes-native Declarative Continuous Delivery tool that follows the **GitOps methodology**

- Integrations with templating tools like Ksonnet, Helm, and Kustomize in addition to plain yaml files to define the desired state of an application
- Automated or manual syncing of applications to its desired state
- Intuitive UI to provide observability into the state of applications
- Extensive CLI to integrate Argo CD with any CI system
- Enterprise-ready features like auditability, compliance, security, RBAC, and SSO

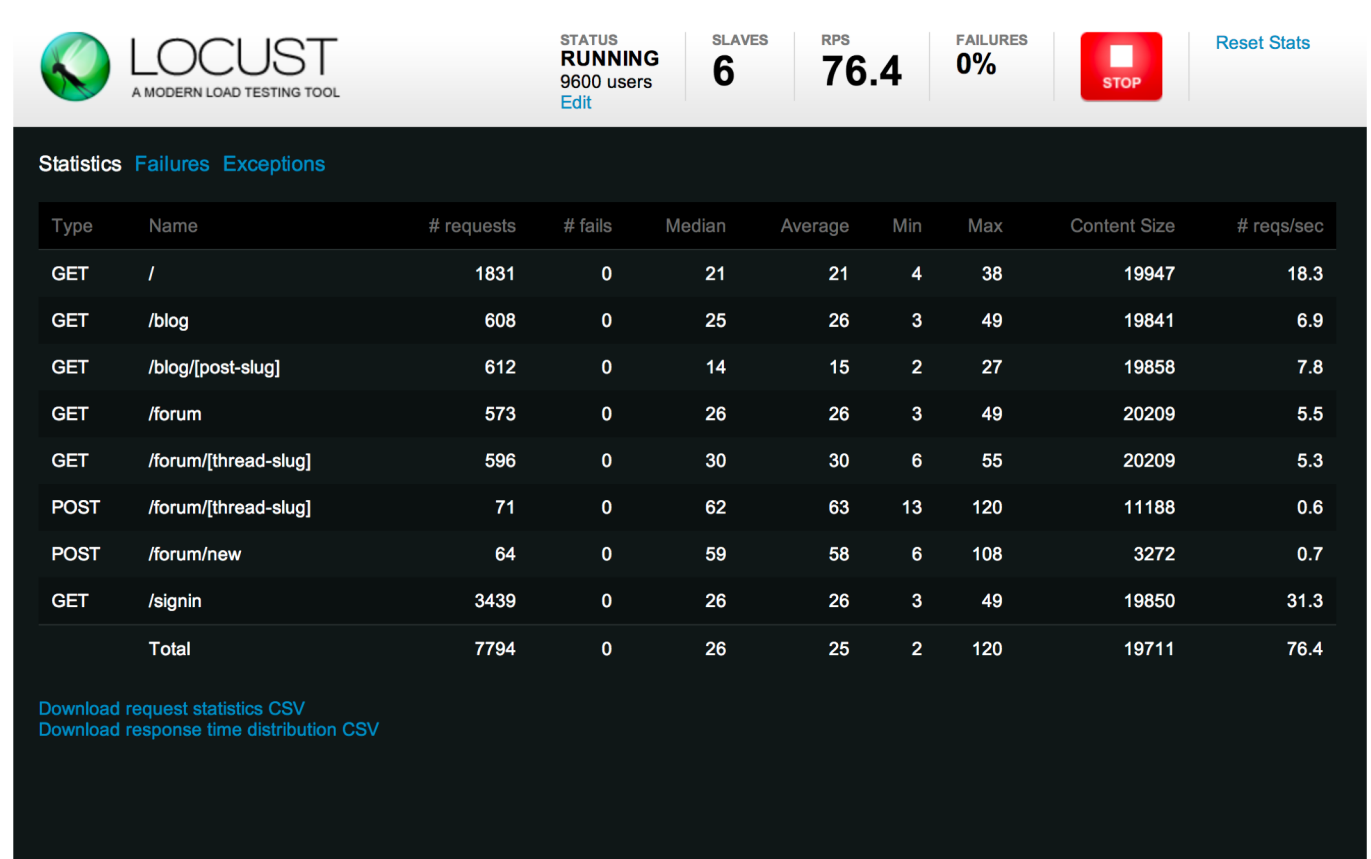


Synchronize Kubeflow development with Gi

Performance Load Testing

Locust : Open Source Load Testing Tool

- Define user behaviour in code (in Python)
- Distributed & scalable (can even run on Kubernetes Cluster)
- Proven & battle tested



Alternatives Open Source Frameworks

Open Source

MLFlow (Databricks)

Bighead (AirBnb)

Michelangelo (Uber)

Cloud Specific / Commercial Offerings

Azure ML Studio

AWS Sagemaker

Google Cloud-ML

H2O

Domino

Anaconda Enterprise

And Many More

Thanks

26-MAR-2019

Next Steps

- Provide feedback on the tutorial
- Download & review tutorial material
 - Concepts
 - Demos
- Share
 - Progress, Issues, Use-cases
 - Twitter
 - @meabhishekkumar
 - @pramodchahar