使用 Analytics-Zoo 构建统一的大数据 AI 应用的架构实践

史栋杰

资深软件架构师

自我介绍

史栋杰，英特尔资深软件架构师。多年从事企业级计算、风控、大数据分析、云计算容器编排、数据分析与人工智能领域的研发，英特尔开源框架 BigDL 与 Analytics-Zoo 的贡献者之一。

目录

1. 大规模人工智能应用面临的挑战
2. 统一的大数据分析及人工智能
3. 跨行业的端到端客户案例实践

以数据为中心的世界

全球超过 OVER

一半 HALF 数据 OF THE WORLD'S DATA

创建于过去

WAS CREATED IN THE LAST

两年 2 YEARS

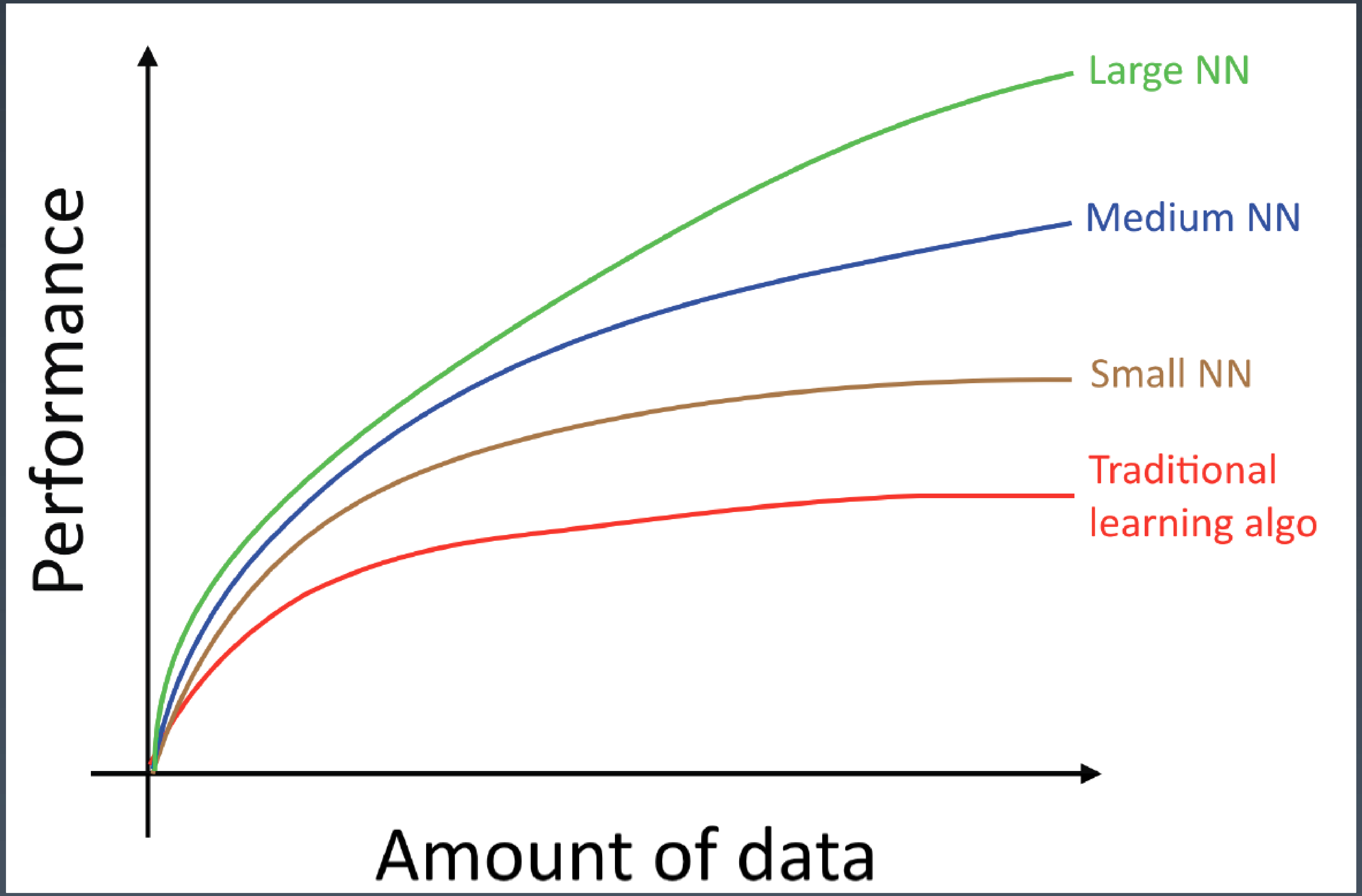
其中只有不到 LESS THAN

2% 的数据 HAS BEEN ANALYZED

经过了分析

大规模人工智能应用

数据驱动深度学习和人工智能应用



"Machine Learning Yearning",
Andrew Ng, 2016

大规模人工智能应用

正面临巨大的挑战

大规模人工智能应用

正面临巨大的挑战

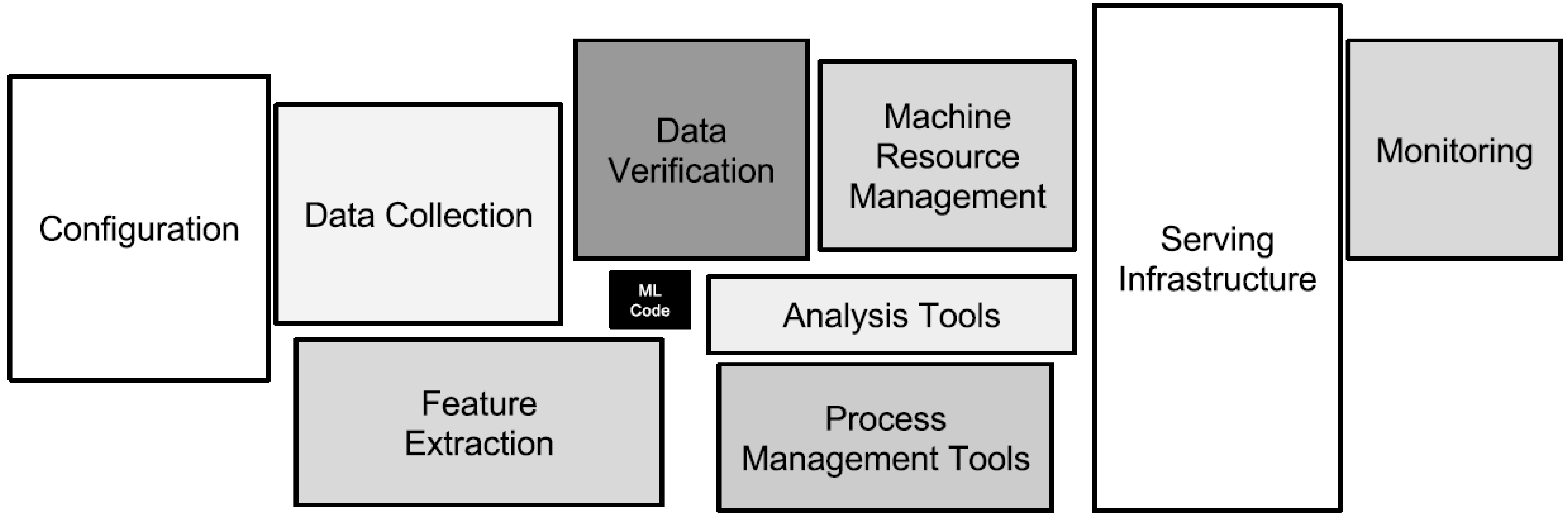


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

“Hidden Technical Debt in Machine Learning Systems”, Sculley et al., Google, NIPS 2015

统一的大数据分析及人工智能

获取 / 存储
Source/Store

清洗 / 准备
Clean/Prepare

分析 / 建模
Analyze /Modeling

部署 / 可视化
Deploy /Visualize

集成的数据流水线
Unified Data Pipeline

Spark Streaming
hadoop HDFS
APACHE HBASE
APACHE kafka
A distributed streaming platform
APACHE KUDU
APACHE Spark
RAY
Spark SQL
TensorFlow
Spark MLlib
BigDL
ANALYTICS ZOO

数据管理
Data Management

数据分析
Data Analysis

数据科学及人工智能
Data Science and AI

大数据上的人工智能

BigDL

基于 Apache Spark 的高性能
深度学习框架

High-Performance Deep Learning
Framework for Apache Spark

software.intel.com/bigdl

ANALYTICS ZOO

统一的分析 + 人工智能平台
Integrated Analytics + AI Toolkit

基于 Apache Spark 的分布式
TensorFlow、PyTorch、Keras 和 BigDL

高级流水线、参考用例、人工智能模型、特征工程等

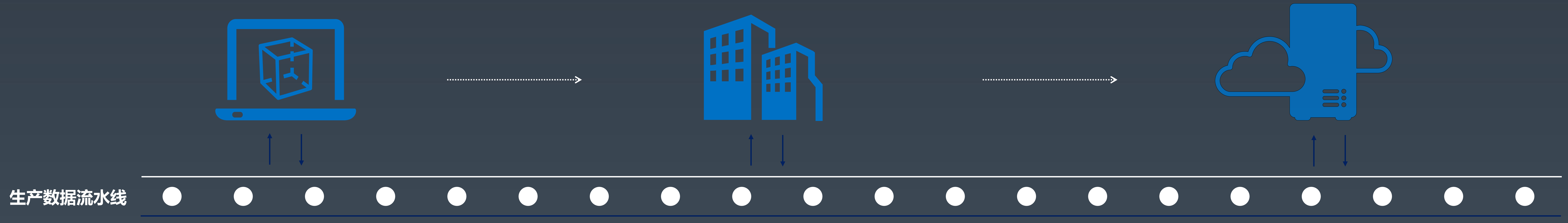
https://github.com/intel-analytics/analytics-zoo

加快数据分析及人工智能大规模应用
Accelerating DATA Analytics + AI Solutions DEPLOYMENT At SCALE

统一的数据分析和AI流水线

端到端、从原型到生产化部署的无缝扩展
Seamless Scaling from Laptop to Production

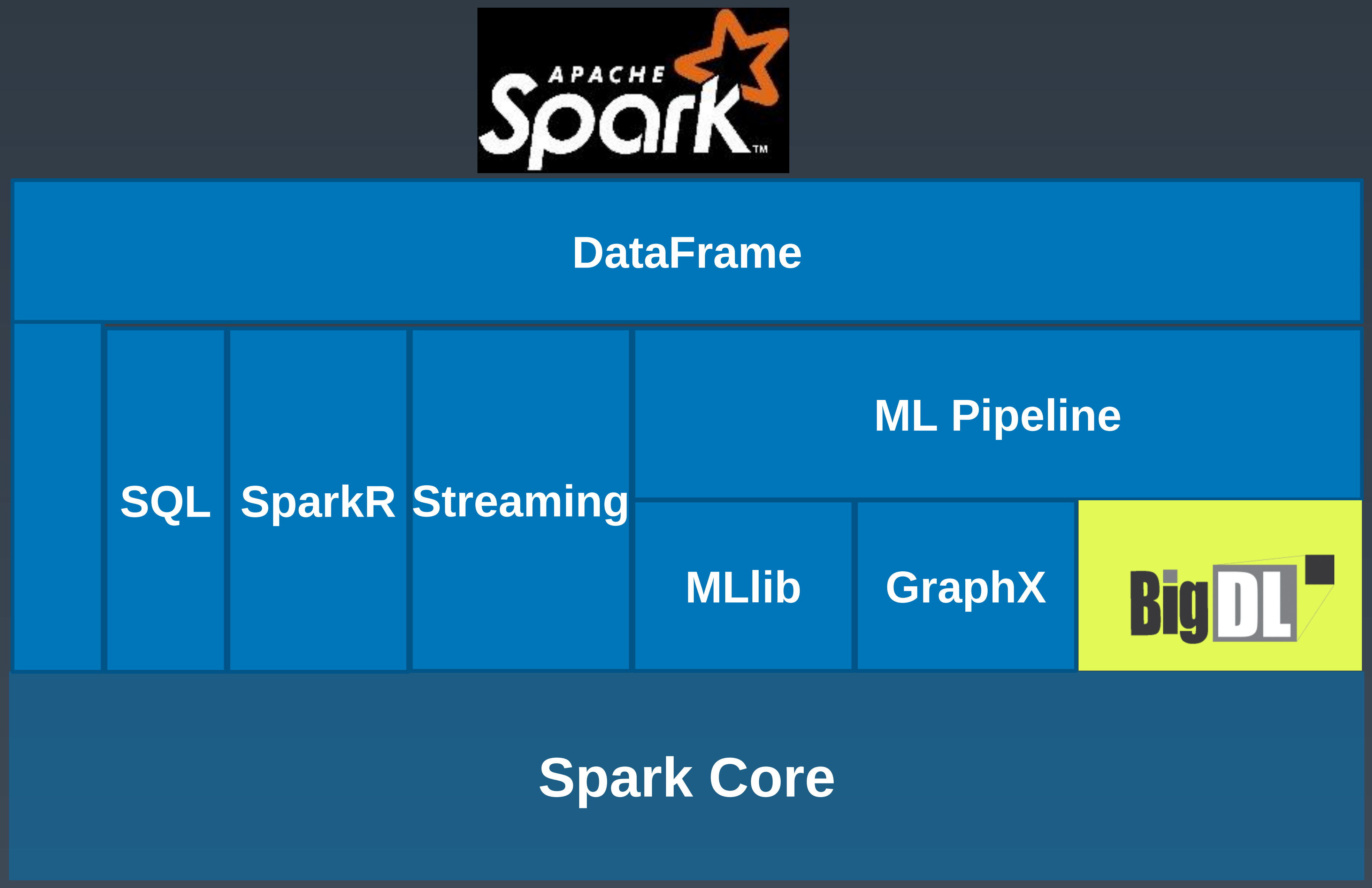
在笔记本电脑上使用样本数据构建原型	在集群上使用历史数据运行模型试验	在分布式生产环境中部署
Prototype on laptop using sample data	Experiment on clusters with history data	Production deployment w/ distributed data pipeline



- ❑ 从笔记本电脑到分布式集群**几乎无需任何代码更改**
- ❑ 无需数据拷贝，**直接访问生产大数据系统**
- ❑ 高效构建**端到端**的数据分析+ AI 流水线原型
- ❑ 无缝扩展部署到**大数据集群及生产环境**

BigDL

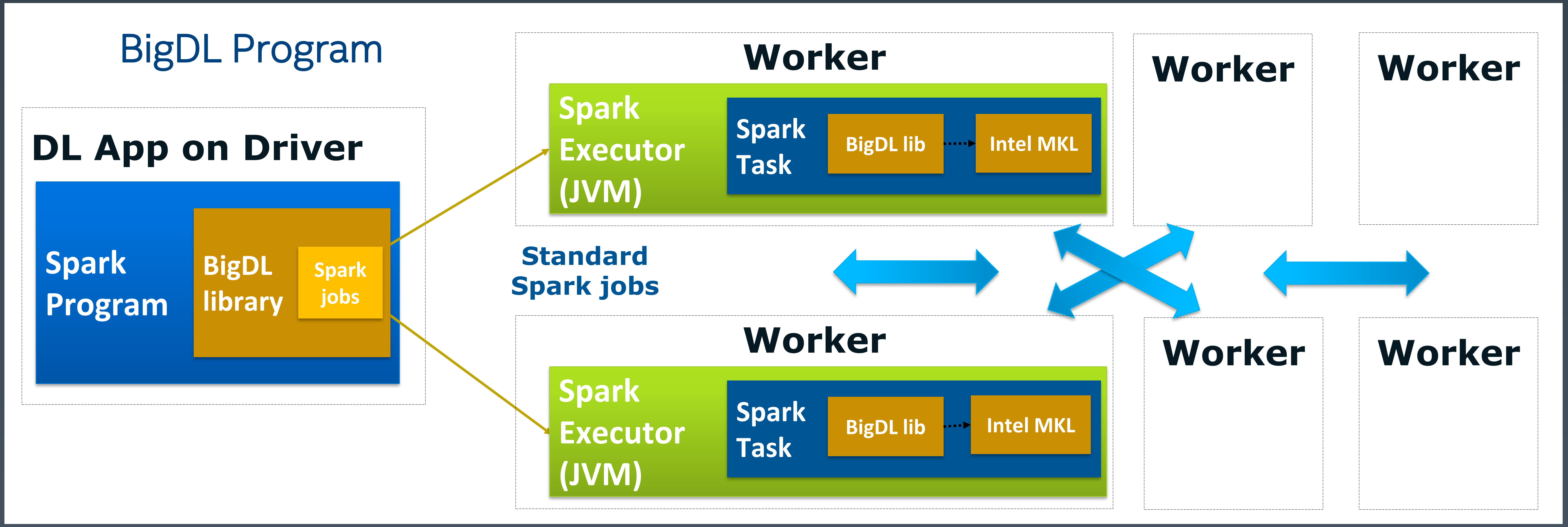
- ❑ 构建于Apache Spark*之上的深度学习框架
- ❑ 将深度学习带给广大的大数据用户和大数据数据科学家
  - 以标准Spark*程序方式创建深度学习应用
  - 运行于现有的YARN*/Spark*/K8S*集群
- ❑ 与流行的深度学习框架具有同等功能
  - Caffe*, Torch*, Tensorflow*等
- ❑ 高性能
  - 使用Intel MKL, MKLDNN加速
- ❑ 有效的扩展性
  - 使用Spark*作分布式训练及推理



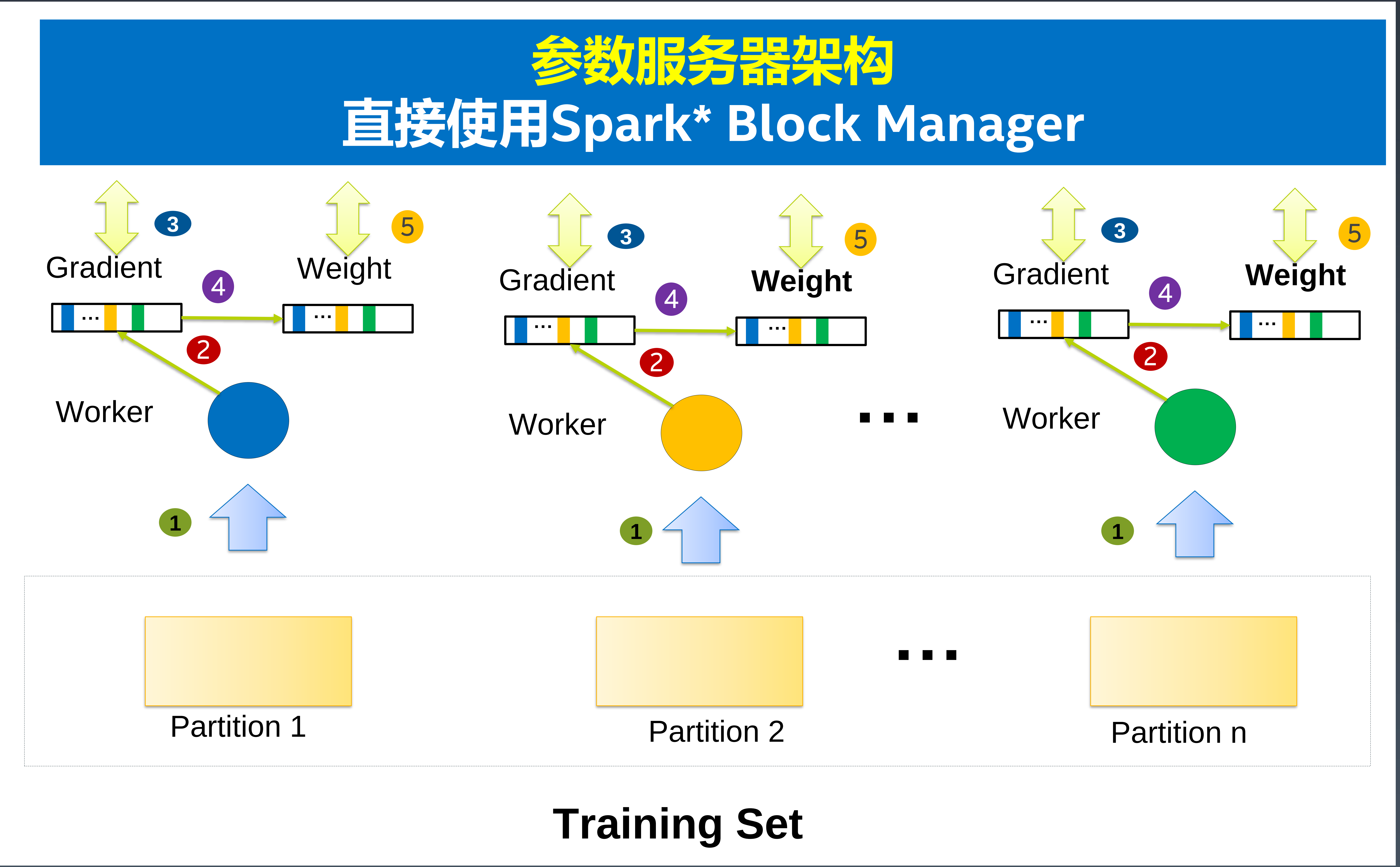
https://github.com/intel-analytics/BigDL
https://bigdl-project.github.io/

BigDL 以标准Spark*程序方式创建深度学习应用

- ❑ 标准的Spark* Job
  - 无需改变YARN*/Spark*/K8S*集群
- ❑ 数据并发
  - 每一个Spark Task在全量数据集的一个子集（Batch）上运行统一的模型



BigDL 分布式训练


参数服务器架构
直接使用Spark* Block Manager
3
Gradient
5
Weight
4
2
Worker
1
3
Gradient
5
Weight
4
2
Worker
1
...
3
Gradient
5
Weight
4
2
Worker
1
Partition 1
Partition 2
...
Partition n
Training Set

Analytics Zoo

统一的大数据分析+人工智能平台
Integrated Big Data Analytics and AI platform

用户案例 Use Case
- Recommendation
- Anomaly Detection
- Text Classification
- Text Matching

模型 Model
- Image Classification
- Object Detection
- Seq2Seq
- Transformer
- BERT

特征工程 Feature Engineering
- image
- 3D image
- text
- time series

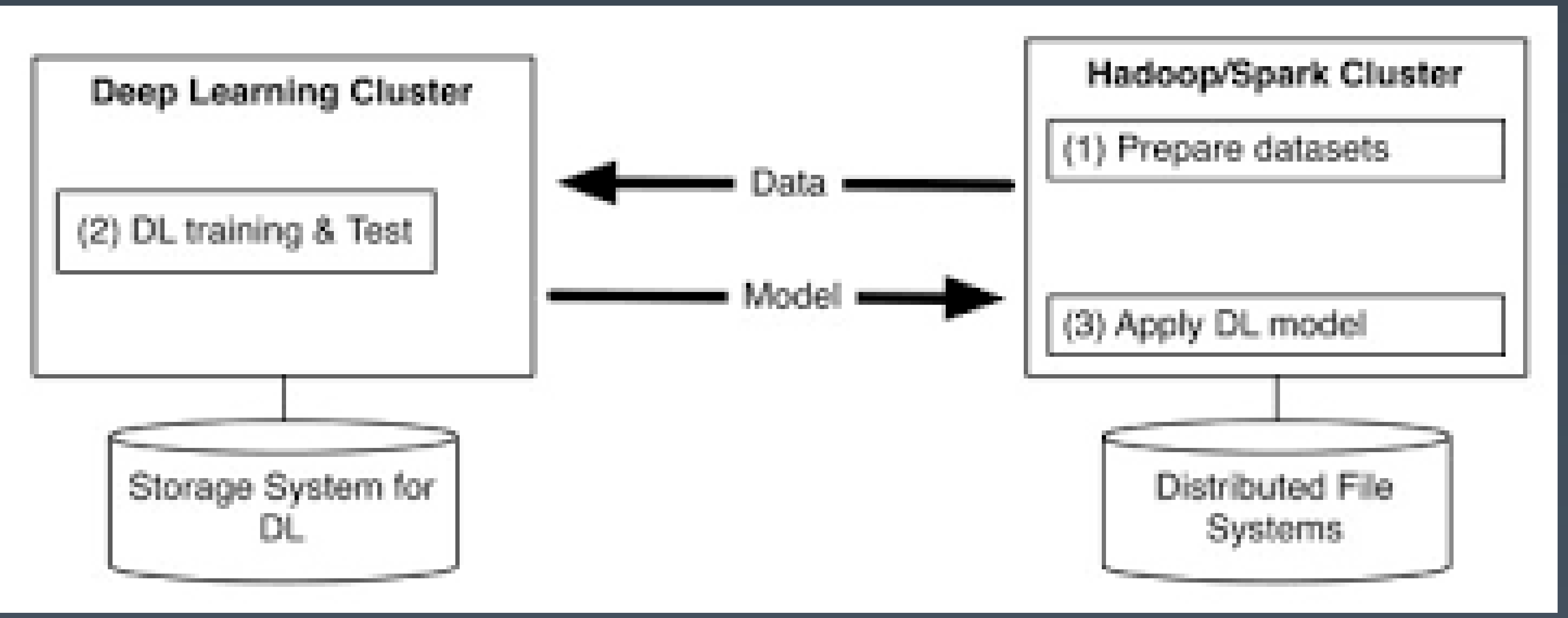
高级流水线 Advanced Pipeline
- tfpark: Distributed TensorFlow on Big Data
- Distributed Keras w/ autograd on Big Data
- nnframes: Spark Dataframes & ML Pipelines for Deep Learning
- Distributed Model Inference (batch, streaming & online)

后端 Backend
- TensorFlow*
- Keras*
- PyTorch*
- BigDL
- NLP Architect
- Apache Spark*
- Apache Flink*
- Ray*
- MKLDNN
- OpenVINO
- Intel® Optane™ DCPMM
- DL Boost (VNNI)

https://github.com/intel-analytics/analytics-zoo

基于 Spark* 的分布式 TensorFlow* 流水线

- ❑ 无缝集成 TensorFlow* 与Spark* & BigDL*
- ❑ 比 TensorFlow * built with MKL 更好的性能
- ❑ 易于使用的分布式API，很少的代码改动



在两个集群中迁移数据和模型的不便

```
#pyspark code
train_rdd = spark.hadoopFile(…).map(…)
dataset = TFDataset.from_rdd(train_rdd,…)

#tensorflow code
import tensorflow as tf
slim = tf.contrib.slim
images, labels = dataset.tensors
with slim.arg_scope(lenet.lenet_arg_scope()):
  logits, end_points = lenet.lenet(images, …)
loss = tf.reduce_mean( \
  tf.losses.sparse_softmax_cross_entropy( \
  logits=logits, labels=labels))

#distributed training on Spark
optimizer = TFOptimizer.from_loss(loss, Adam(…))
optimizer.optimize(end_trigger=MaxEpoch(5))
```

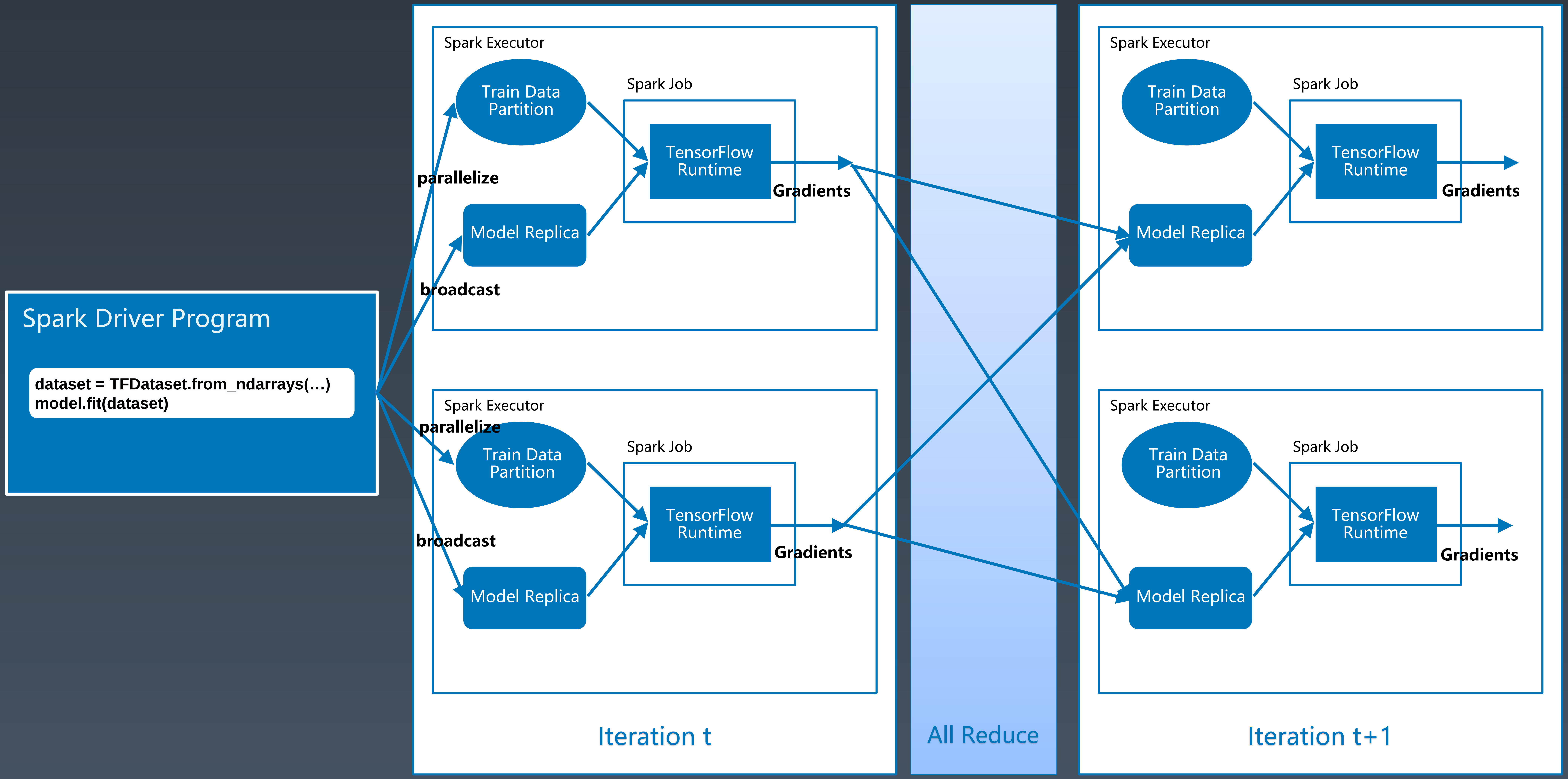
用 PySpark* 载入数据以及预处理数据或特征工程

用 TensorFlow*或Keras*定义深度学习模型

在Spark*上分布式训练或者推理

在 PySpark 程序中内嵌 TensorFlow 代码

基于 Spark* 的分布式 TensorFlow* 流水线



基于Spark* Dataframe & ML 流水线的深度学习

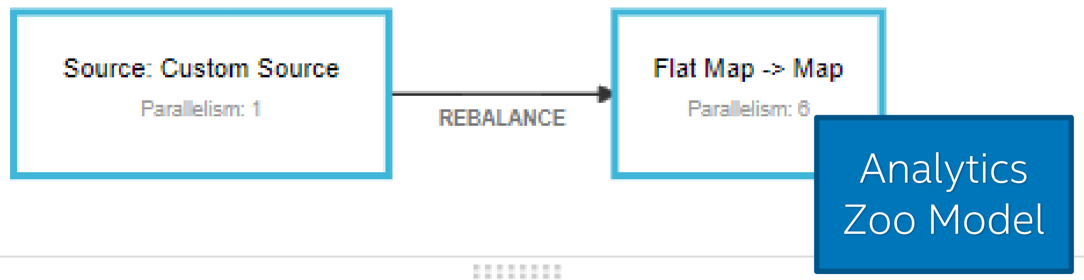
```
#Spark dataframe transformations
parquetfile = spark.read.parquet(...)
train_df = parquetfile.withColumn(...)

#Keras API
model = Sequential()
        .add(Convolution2D(32, 3, 3, activation='relu', input_shape=...)) \
        .add(MaxPooling2D(pool_size=(2, 2))) \
        .add(Flatten()).add(Dense(10, activation='softmax')))

#Spark ML pipeline
Estimater = NNEstimater(model, CrossEntropyCriterion()) \
             .setLearningRate(0.003).setBatchSize(40).setMaxEpoch(5) \
             .setFeaturesCol("image")
nnModel = estimater.fit(train_df)
```

在 Spark* Dataframe 和 ML 流水线中，直接支持深度神经网络模型

分布式、实时 (流式) 模型推理流水线

- ❑ 纯Java或Python API
- ❑ 支持Spark* Streaming, Flink*, Storm*, Kaka*等
- ❑ 支持Web Services
- ❑ 使用OpenVINO和DL Boost(VNNI) 加速

Overview Timeline Exceptions Configuration



Aggregate task statistics by TaskManager

Start Time	End Time	Duration	Name	Bytes received	Records received	Bytes sent	Records sent	Parallelism	Tasks	Status
2019-08-01, 10:09:36	2019-08-01, 10:13:20	3m 44s	Source: Custom Source	0 B	0	537 MB	330	1	0 0 1 0 0 0 0	RUNNING
2019-08-01, 10:09:36	2019-08-01, 10:13:20	3m 44s	Flat Map -> Map	536 MB	326	0 B	320	6	0 0 6 0 0 0 0	RUNNING

Start Time	End Time	Duration	Bytes received	Records received	Bytes sent	Records sent	Attempt	Host	Status	
2019-08-01, 10:09:40		3m 40s	83.4 MB	53	0 B	52	1	Almaren-Node-018:41072	RUNNING	Analytics Zoo Model
2019-08-01, 10:09:40		3m 40s	80.4 MB	54	0 B	53	1	Almaren-Node-018:41072	RUNNING	Analytics Zoo Model
2019-08-01, 10:09:36		3m 44s	76.2 MB	54	0 B	53	1	Almaren-Node-019:39744	RUNNING	Analytics Zoo Model
2019-08-01, 10:09:40		3m 40s	96.9 MB	55	0 B	54	1	Almaren-Node-019:39744	RUNNING	Analytics Zoo Model
2019-08-01, 10:09:40		3m 40s	87.8 MB	55	0 B	54	1	Almaren-Node-020:35343	RUNNING	Analytics Zoo Model

POJO风格的模型推理API

- ❑ 支持加载各种框架的模型
  - BigDL
  - Caffe*
  - TensorFlow*
  - OpenVINO
- ❑ 线程安全及自动扩展
  - 多工作模型共享参数

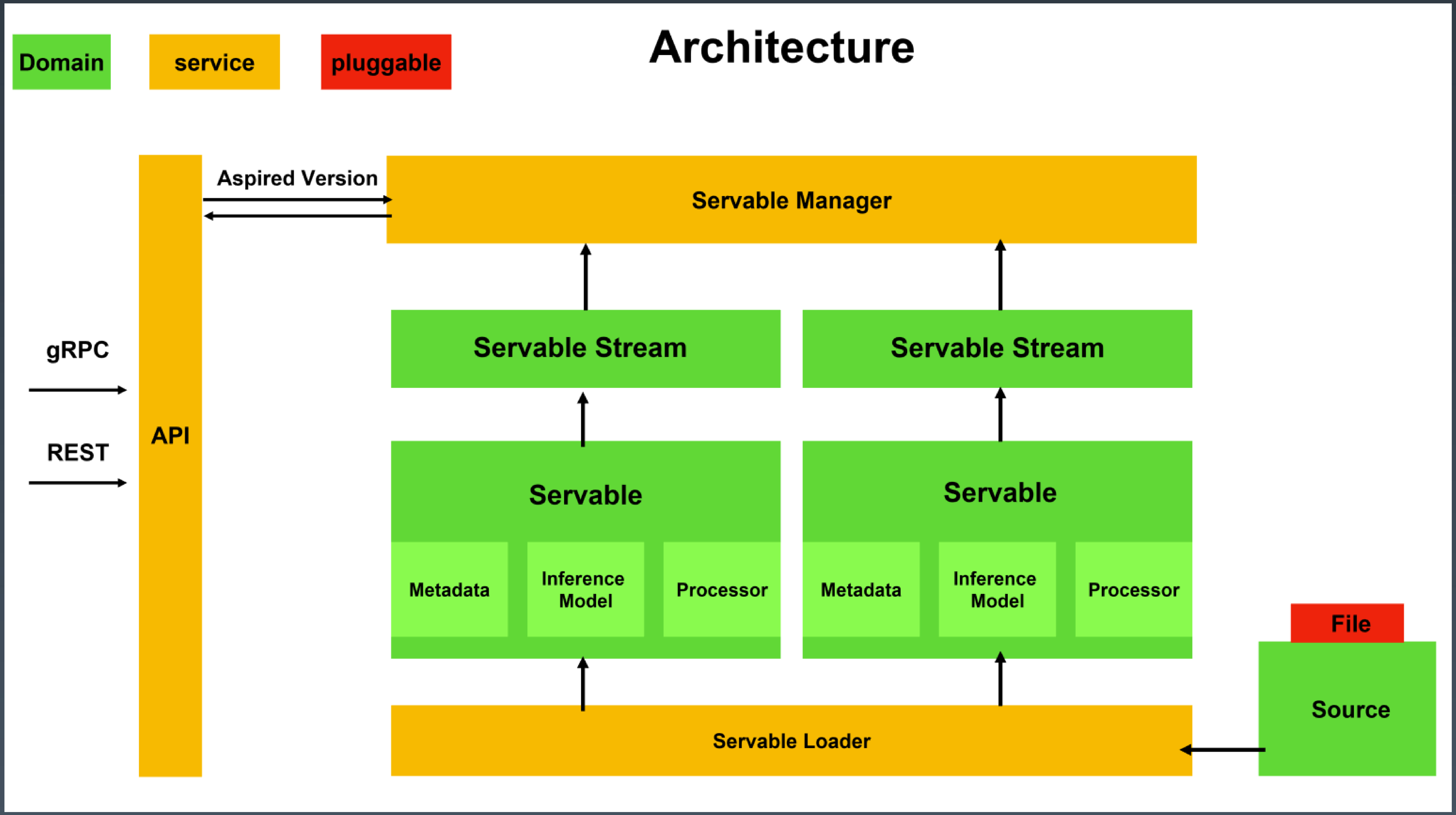
```java
import com.intel.analytics.zoo.pipeline.inference.AbstractInferenceModel;

public class TextClassification extends AbstractInferenceModel {
  public RankerInferenceModel(int concurrentNum) {
    super(concurrentNum);
  }
  ...
}

public class ServingExample {
  public static void main(String[] args) throws IOException {
    TextClassification model = new TextClassification();
    model.load(modelPath, weightPath);

    texts = …
    List<JTensor> inputs = preprocess(texts);
    for (JTensor input : inputs) {
      List<Float> result =
          model.predict(input.getData(), input.getShape());
      ...
  }
}
```

使用OpenVINO加速模型推理

- ❑ 支持Image Classification 和 Object Detection等
- ❑ 支持加载TensorFlow*模型
- ❑ 支持直接加载OpenVINO IR
- ❑ 支持模型动态Optimize及 Calibrate

```python
from zoo.common.nncontext import init_nncontext
from zoo.feature.image import ImageSet
from zoo.pipeline.inference import InferenceModel

sc = init_nncontext("OpenVINO Object Detection Inference Example")
images = ImageSet.read(options.img_path, sc,
                       resize_height=600,
resize_width=600).get_image().collect()
input_data = np.concatenate([image.reshape((1, 1) + image.shape) for
image in images], axis=0)

model = InferenceModel()
model.load_tf(options.model_path, backend="openvino",
model_type=options.model_type)
predictions = model.predict(input_data)

# Print the detection result of the first image.
print(predictions[0])
```

Analytics Zoo *Web Serving*

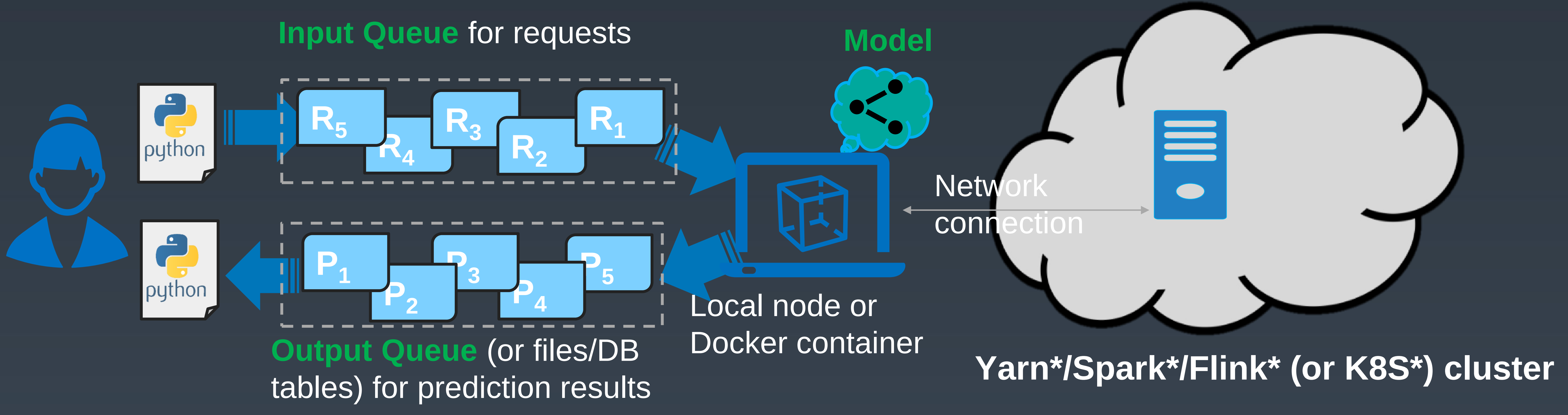


Analytics Zoo *Cluster Serving* 使分布式推理更加简单

部署

√ 一个本地节点或者一个Docker容器

√ 已有的 Yarn/Spark/Flink (or K8s) 集群



使用

1 一条命令:
- 启动Docker容器以及*Zoo Cluster Serving*
  - ❑ 此命令指定:
    - 输入 和 输出 的队列名字
    - 模型 的文件路径
    - 预/后处理 的文件路径
    - 集群 的访问路径

2 一个简单的Python脚本:
- 将请求数据发送到 Input Queue
- 从 Output Queue (或文件/数据库)获得推理结果

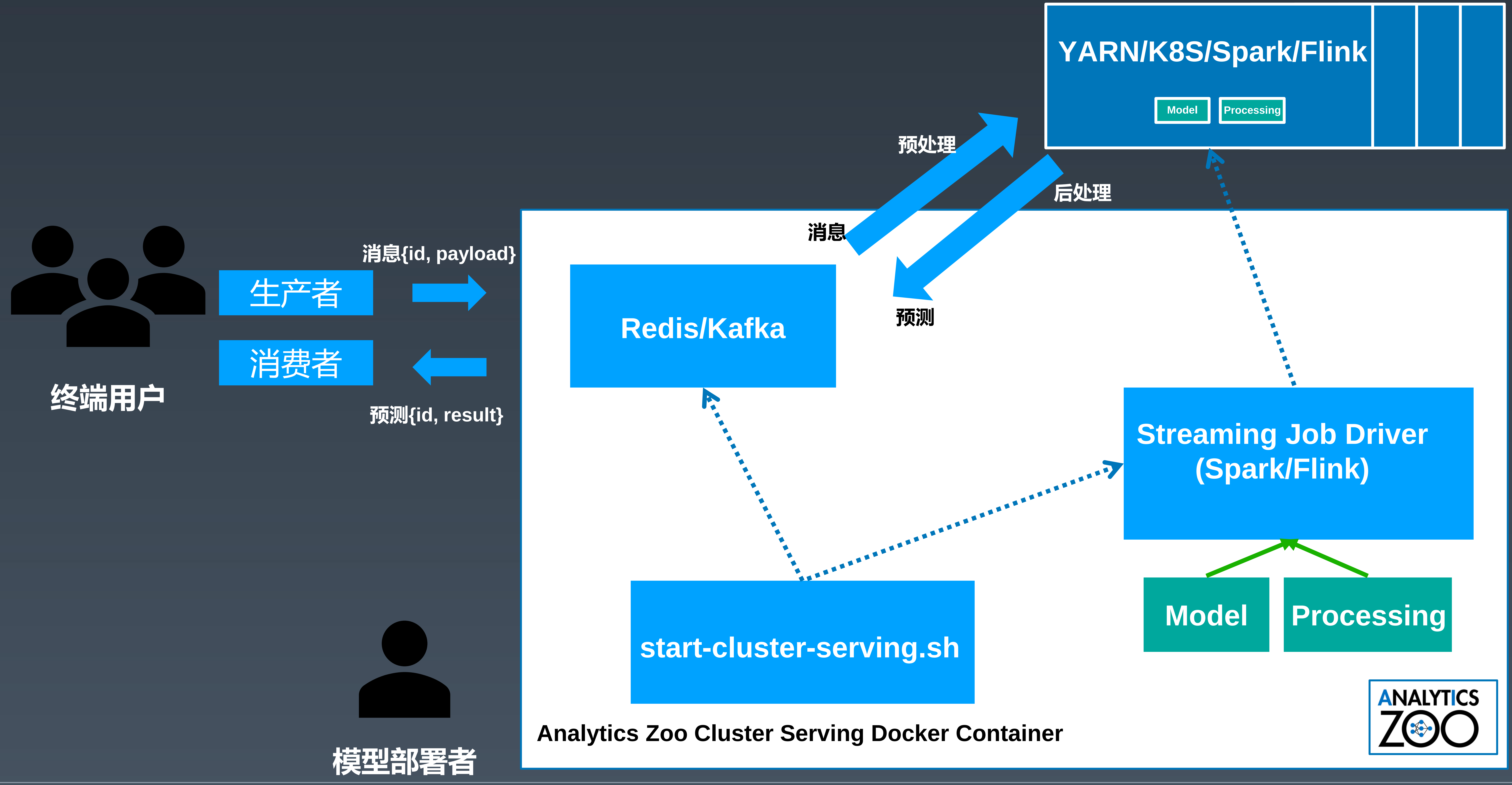
3 Analytics Zoo 在集群上自动执行分布式、实时（流式）模型推理
- 支持 TensorFlow*, Keras*, PyTorch*, Caffe*, BigDL 和 OpenVINO 的模型, 可使用 Int8 加速
- 通过Spark* Streaming 或 Flink* 线性扩展

√ 可扩展的分布式推理由Analytics Zoo托管

√ 用户无需为开发和部署复杂的分布式推理方案而费心

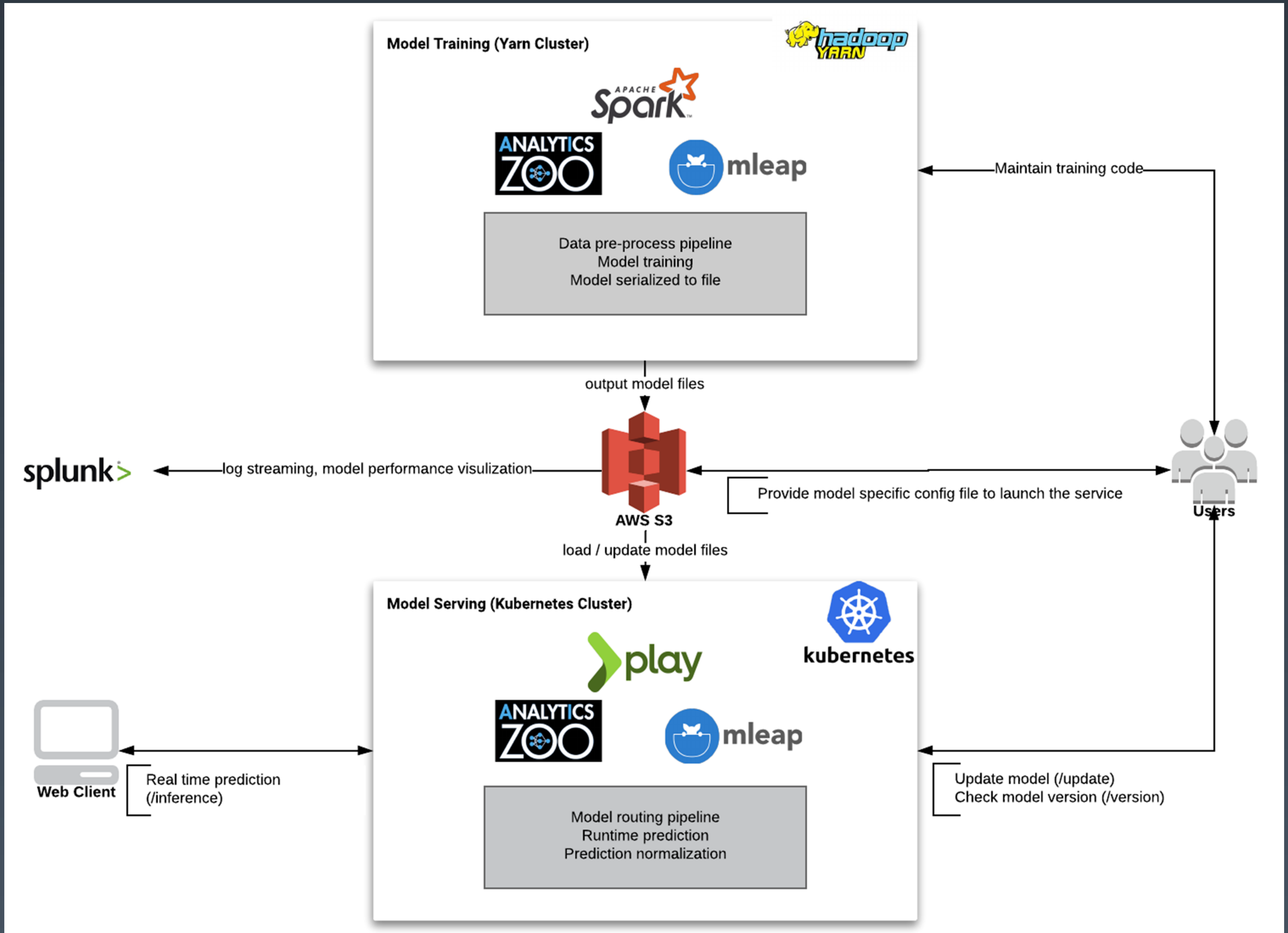
** 文中涉及的其它名称及商标属于各自所有者资产。*

Analytics Zoo *Cluster Serving* 架构



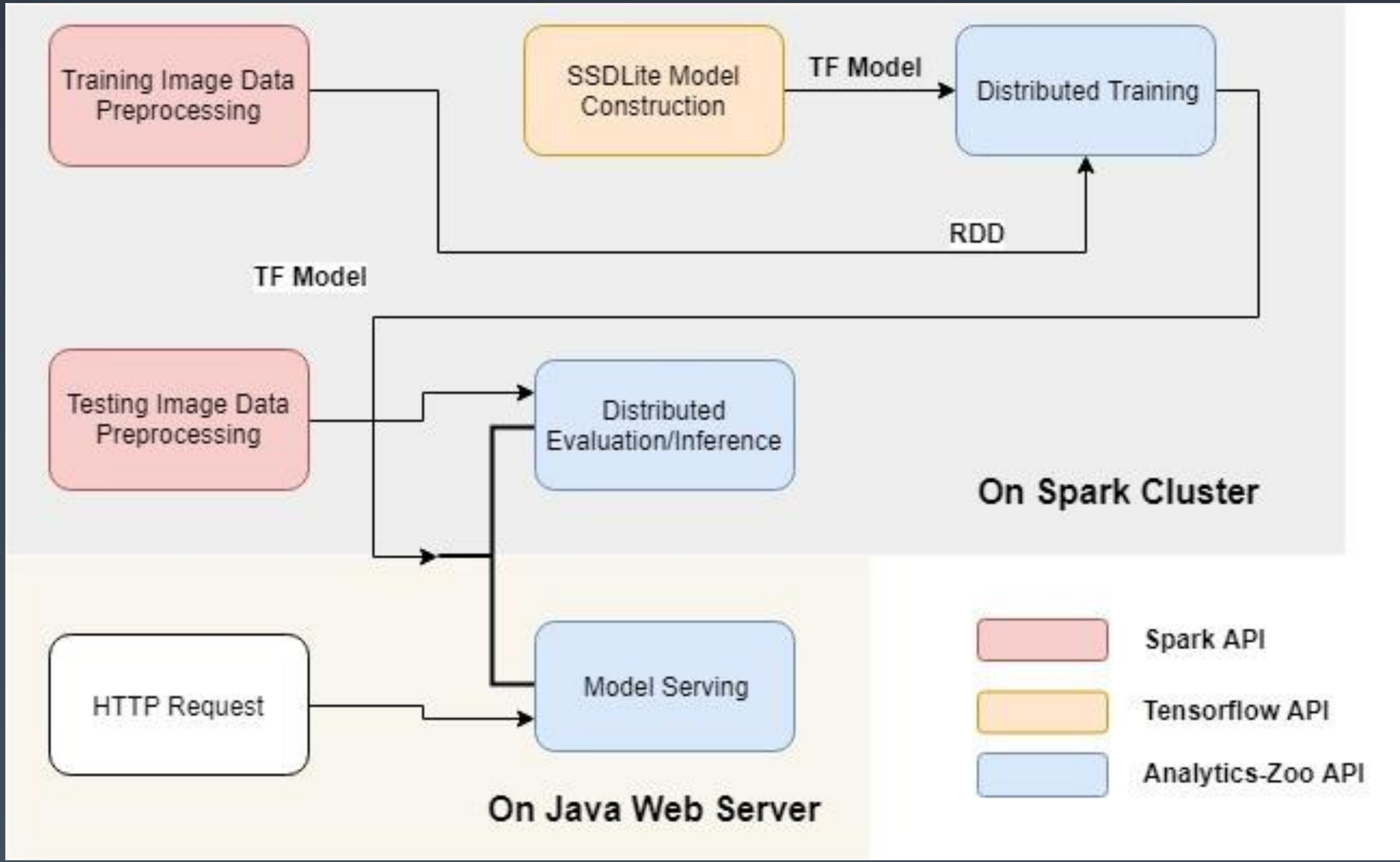
跨行业的端到端客户案例实践

Office Depot*：基于用户 Session 行为的产品推荐



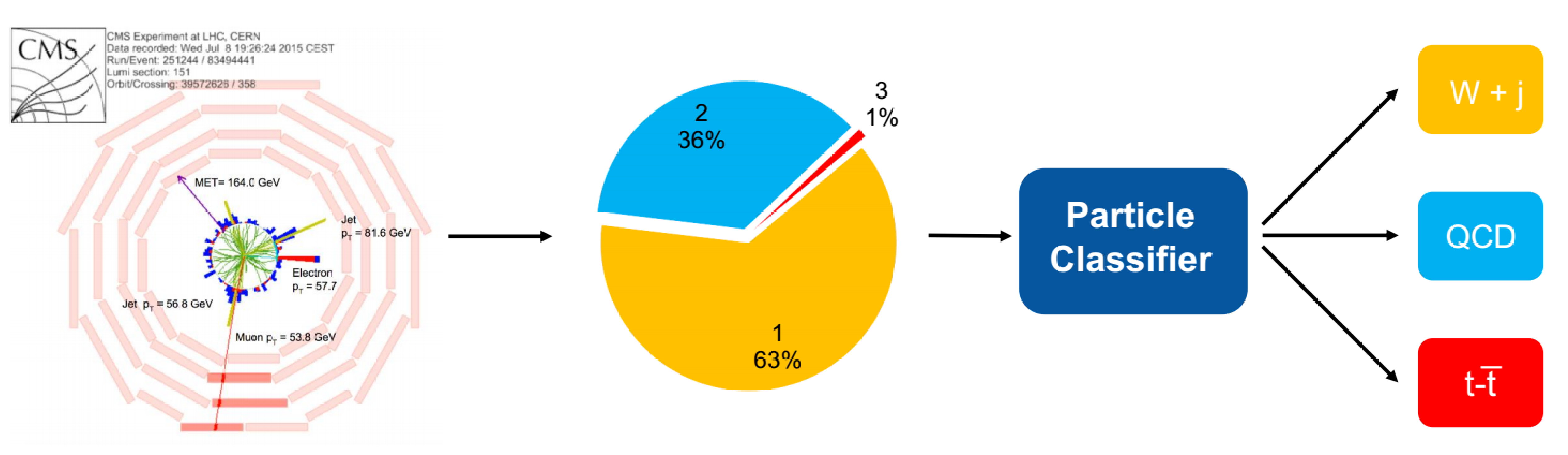
https://software.intel.com/en-us/articles/real-time-product-recommendations-for-office-depot-using-apache-spark-and-analytics-zoo-on
https://conferences.oreilly.com/strata/strata-ca-2019/public/schedule/detail/73079

美的*：工业视觉检测云平台



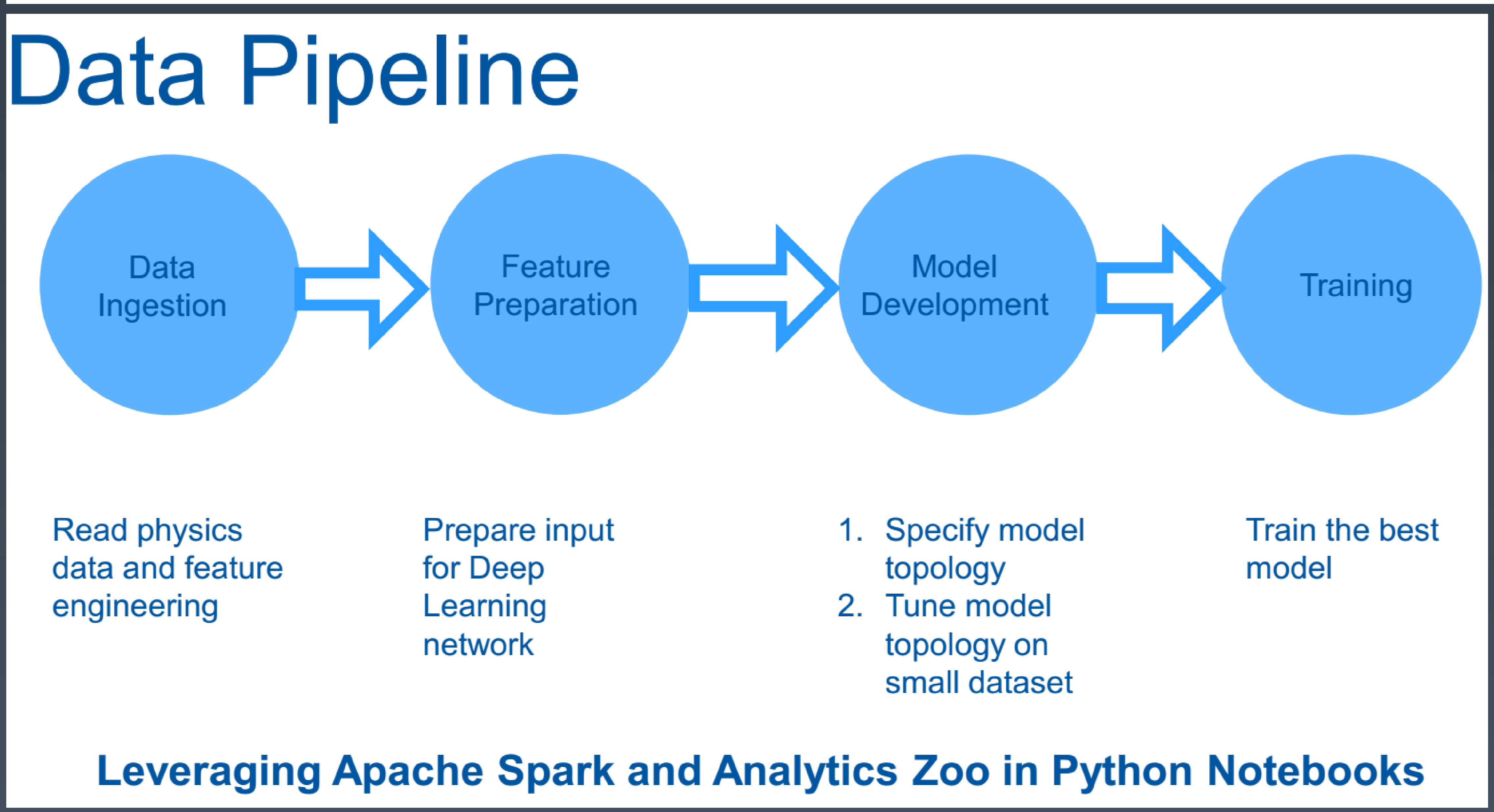
https://software.intel.com/en-us/articles/industrial-inspection-platform-in-midea-and-kuka-using-distributed-tensorflow-on-analytics
https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/midea-case-study.html

CERN*：基于深度学习的高能物理粒子事件分类



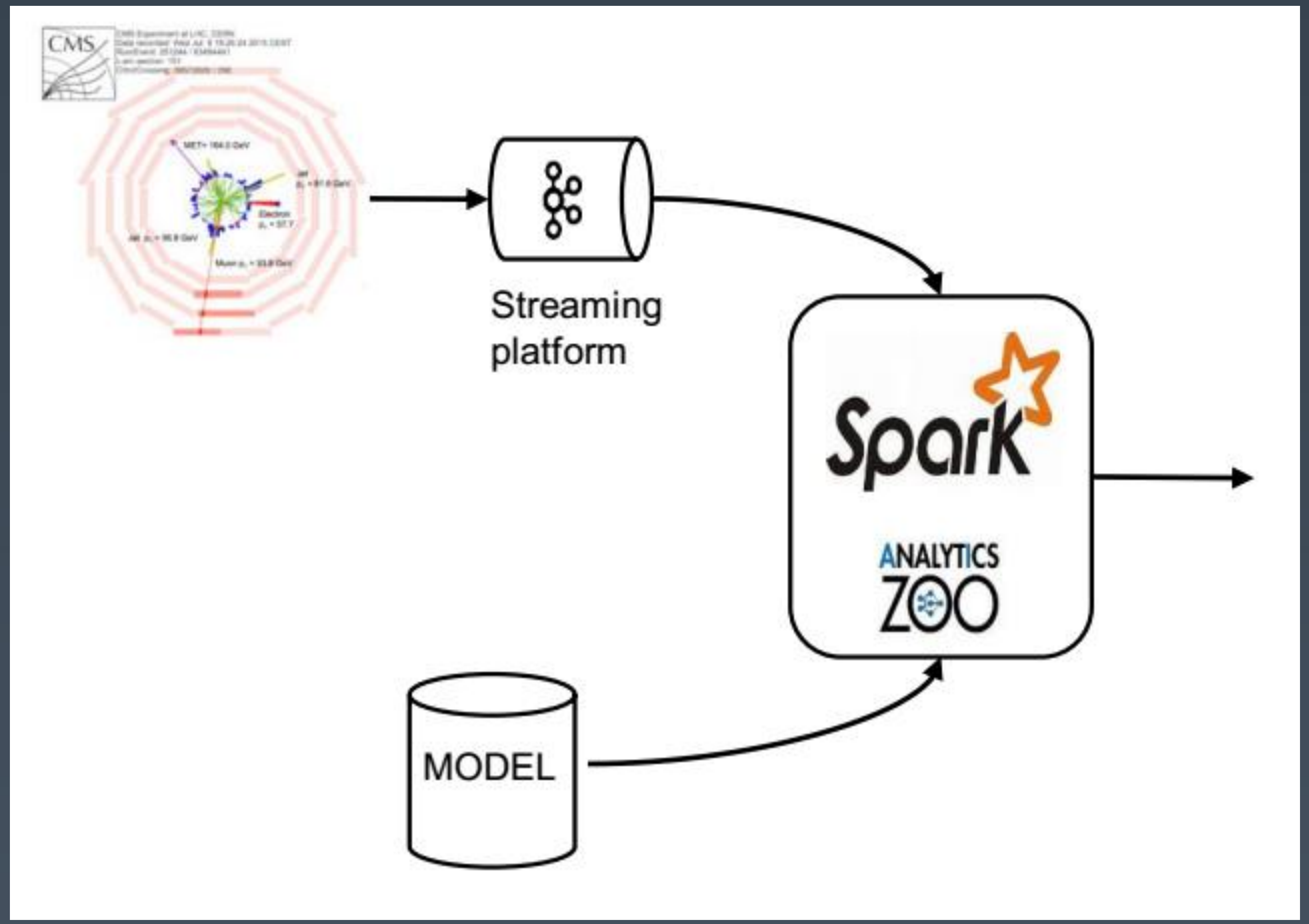
高能物理数据的深度学习流水线

Deep Learning Pipeline for High Energy Physics



使用 Apache Kafka* 和 Spark* 构建模型服务

Model serving using Apache kafka and Spark Streaming



https://db-blog.web.cern.ch/blog/luca-canali/machine-learning-pipelines-high-energy-physics-using-apache-spark-bigdl

https://databricks.com/session/deep-learning-on-apache-spark-at-cerns-large-hadron-collider-with-intel-technologies

[1] 该CERN的解决方案为原型实现，尚未投入生产部署。

THANKS!

LEGAL NOTICES AND DISCLAIMERS