



Hops

 @hopshadoop

 <http://github.com/hopshadoop>

 <http://www.hops.io>

Multi-tenant Streaming and TensorFlow as a Service with Hops

Theofilos Kakantousis
Researcher @ RISE SICS
Co-founder @ Logical Clocks AB

Slides by Jim Dowling, Theofilos Kakantousis

Big Data Conference Vilnius, Nov 2017

Hadoop-as-a-Service in Sweden

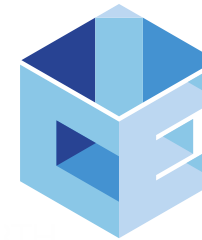
- **HopsWorks**

- Spark/Flink/Kafka/TensorFlow/Hadoop-as-a-service
- Built on Hops Hadoop (www.hops.io)



- **RISE SICS ICE**

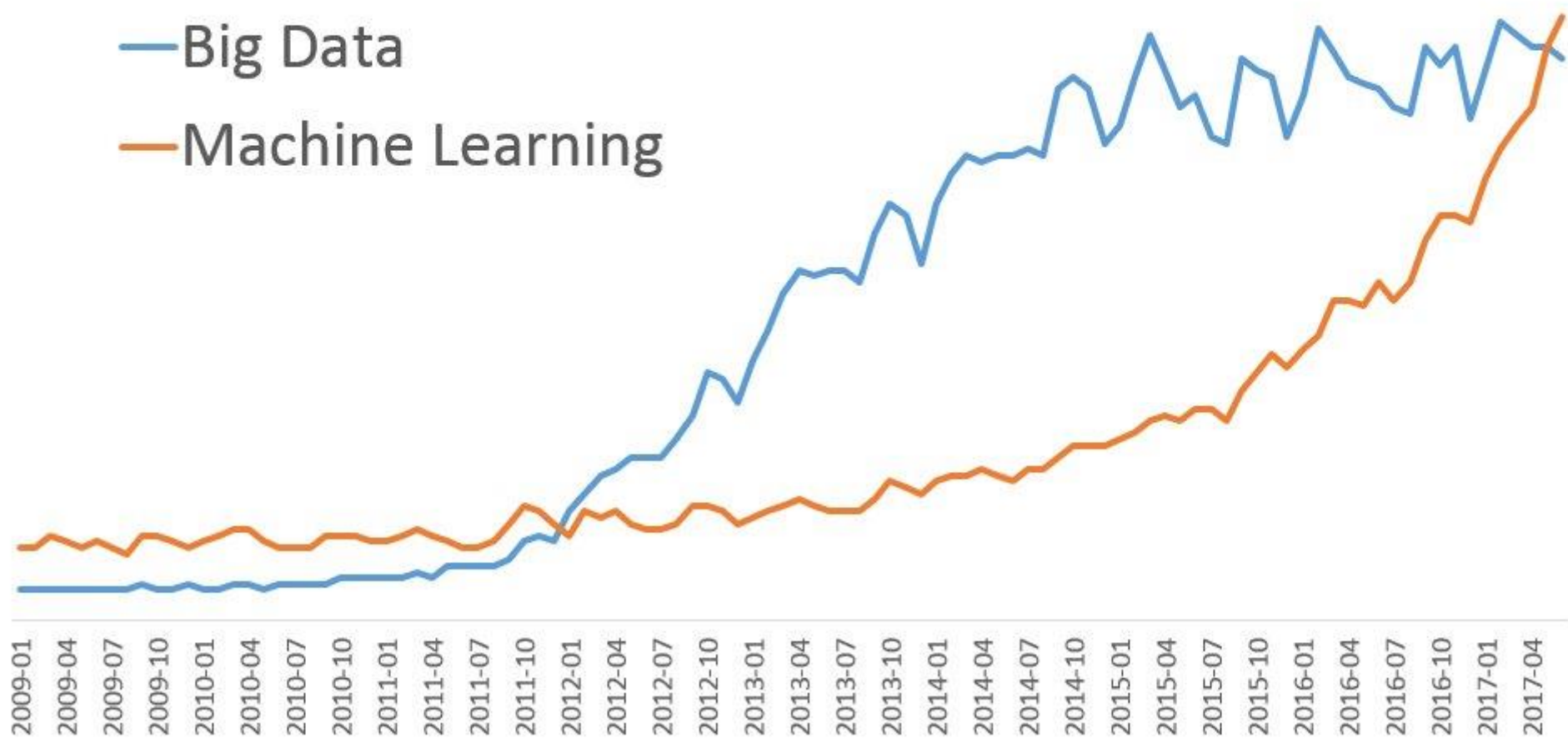
- 250 kW Datacenter, ~1000 servers
- Research and test environment



<https://www.sics.se/projects/sics-ice-data-center-in-lulea>

Big Data and AI

Google Trends Worldwide



2017-11-30

www.hops.io

Becoming a BigData-AI Company

THE DATA SCIENCE **HIERARCHY OF NEEDS**

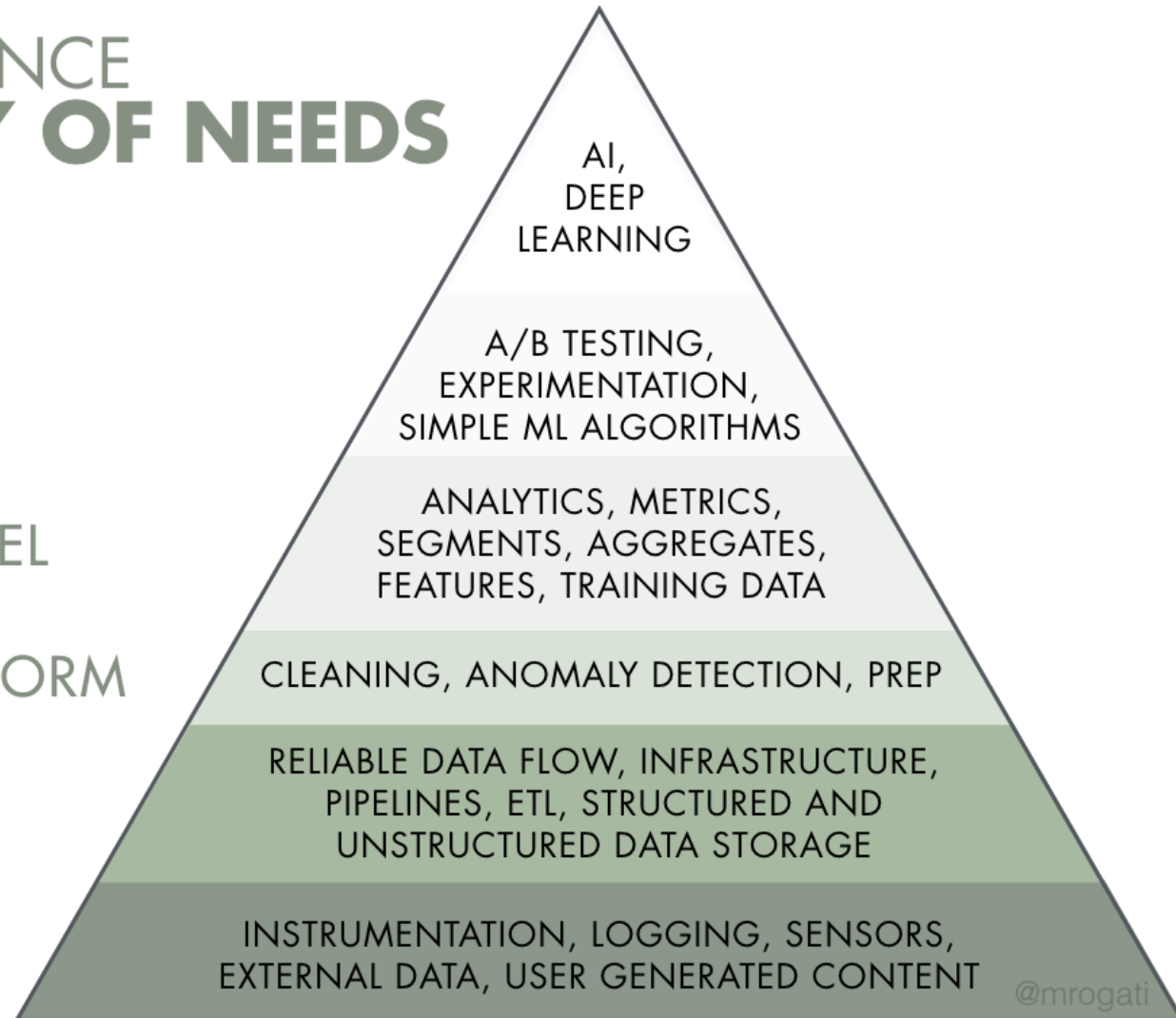
LEARN/OPTIMIZE

AGGREGATE/LABEL

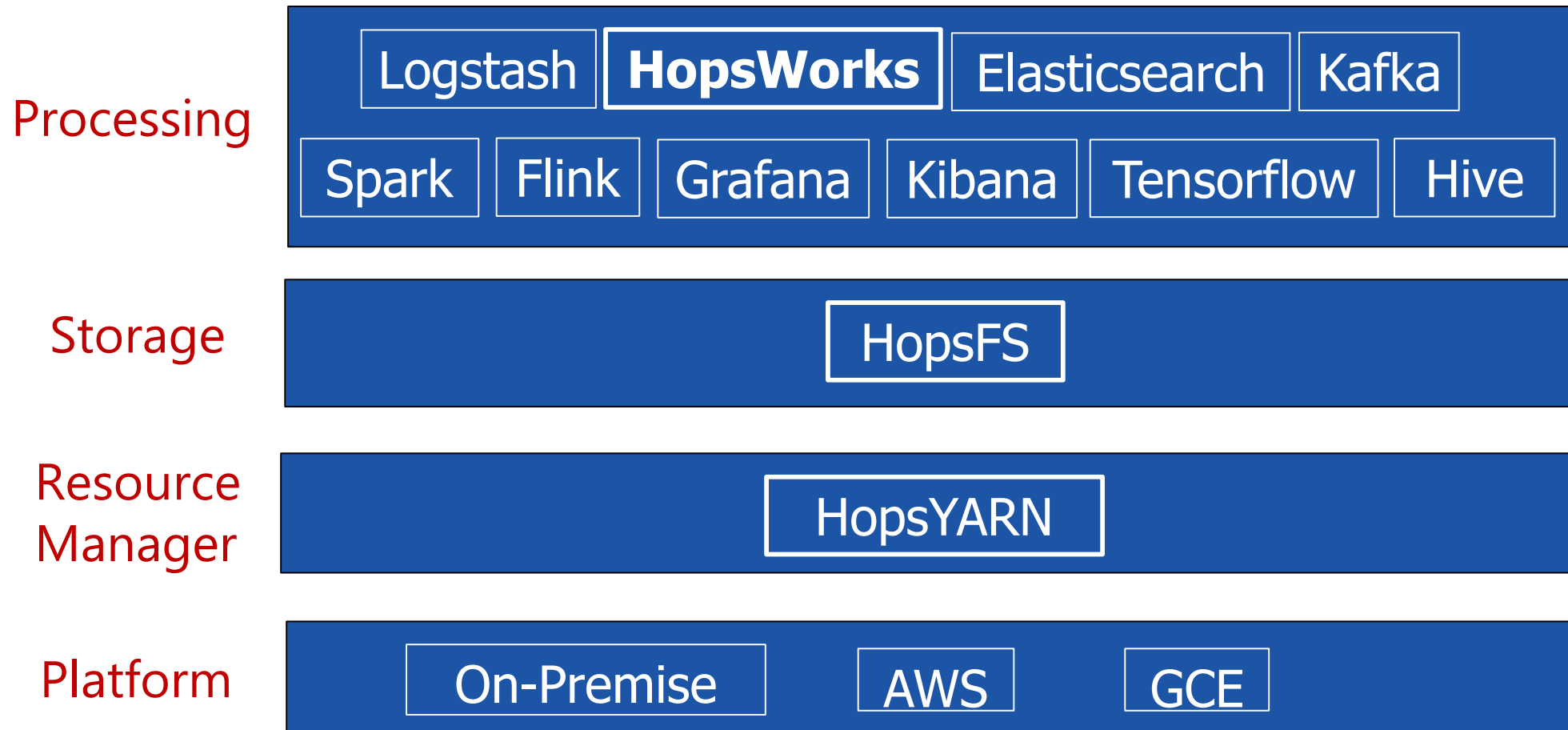
EXPLORE/TRANSFORM

MOVE/STORE

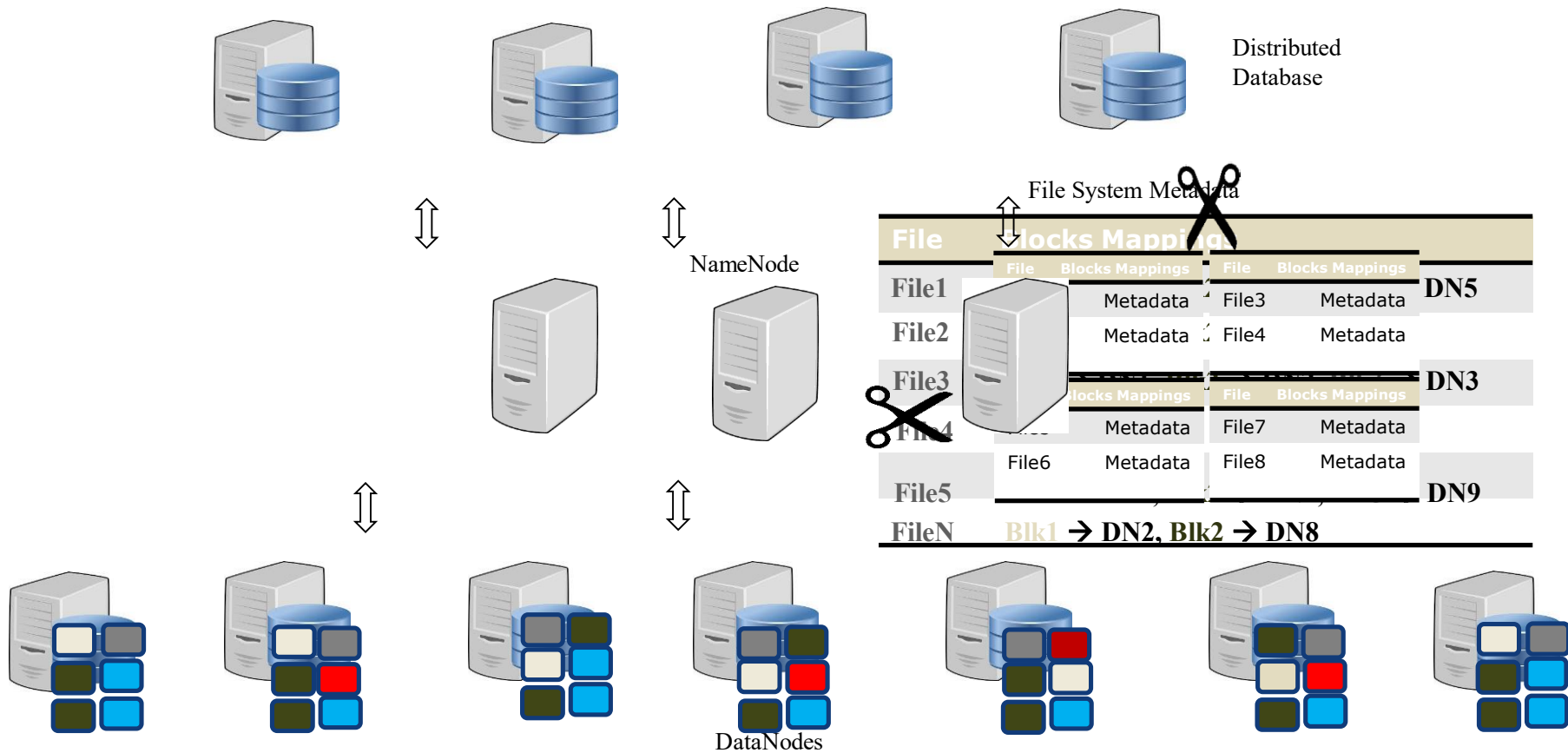
COLLECT



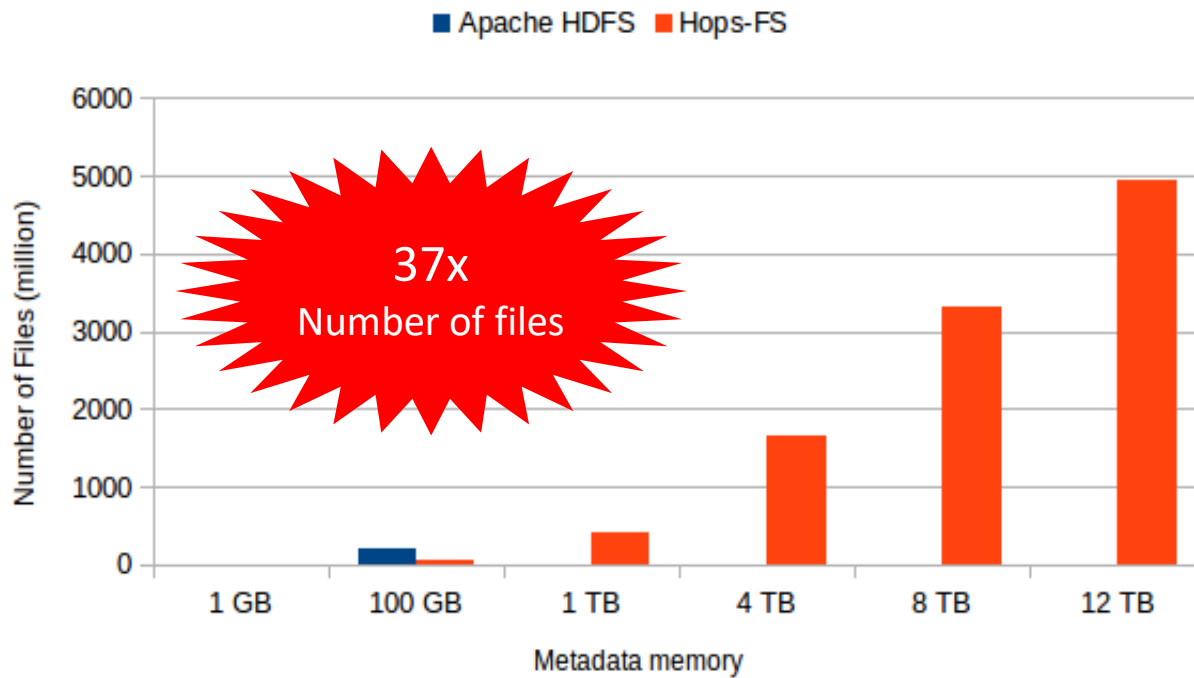
Hops Hadoop



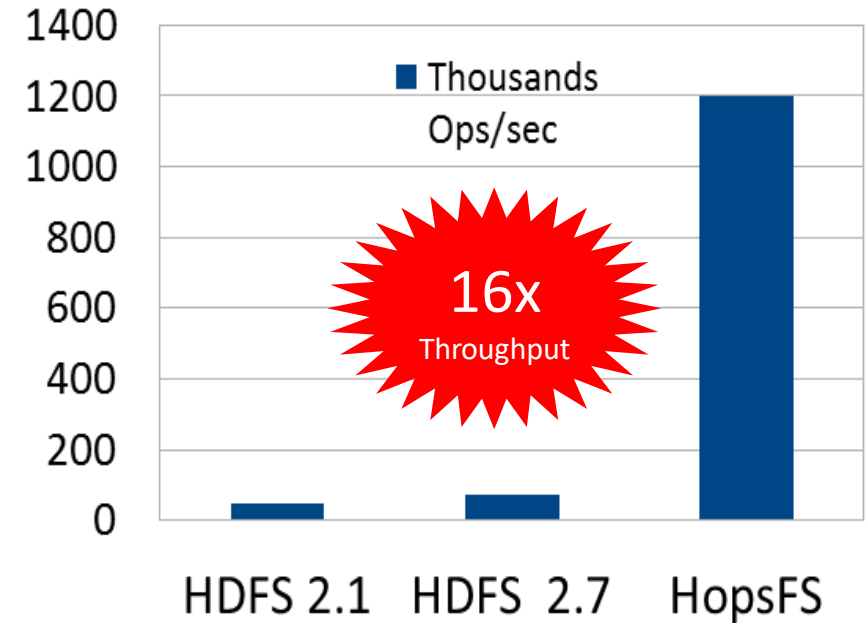
HopsFS Architecture



HopsFS: Next Generation HDFS*



Bigger

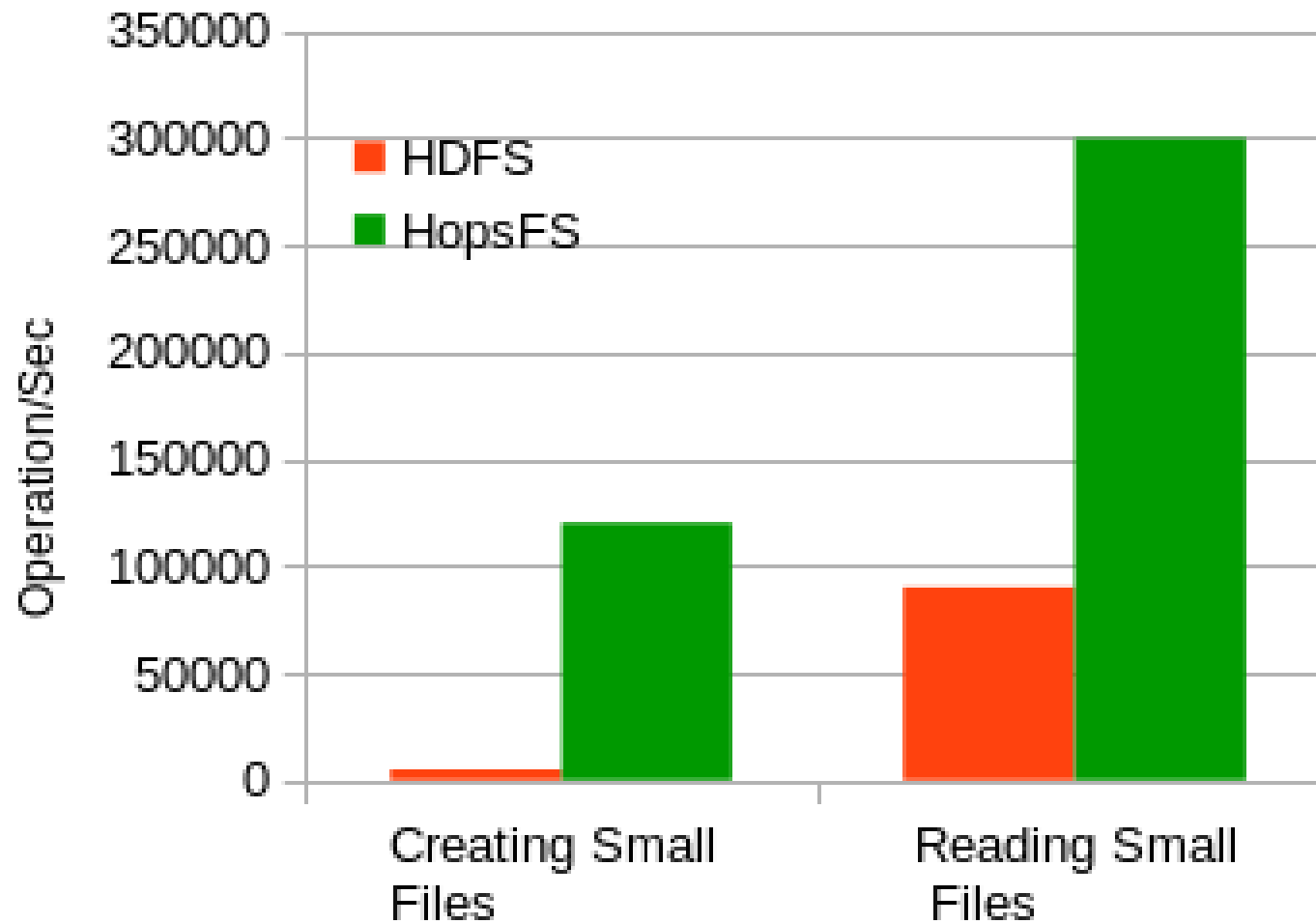


Faster

 **IEEE** Scale Challenge Winner (2017)

HopsFS now stores Small Files in the DB

Small Files Performance



Hops Concepts

Hadoop

- Clusters
- Users
- Jobs/Applications
- Files
- ACLs
- Sys Admins
- Kerberos



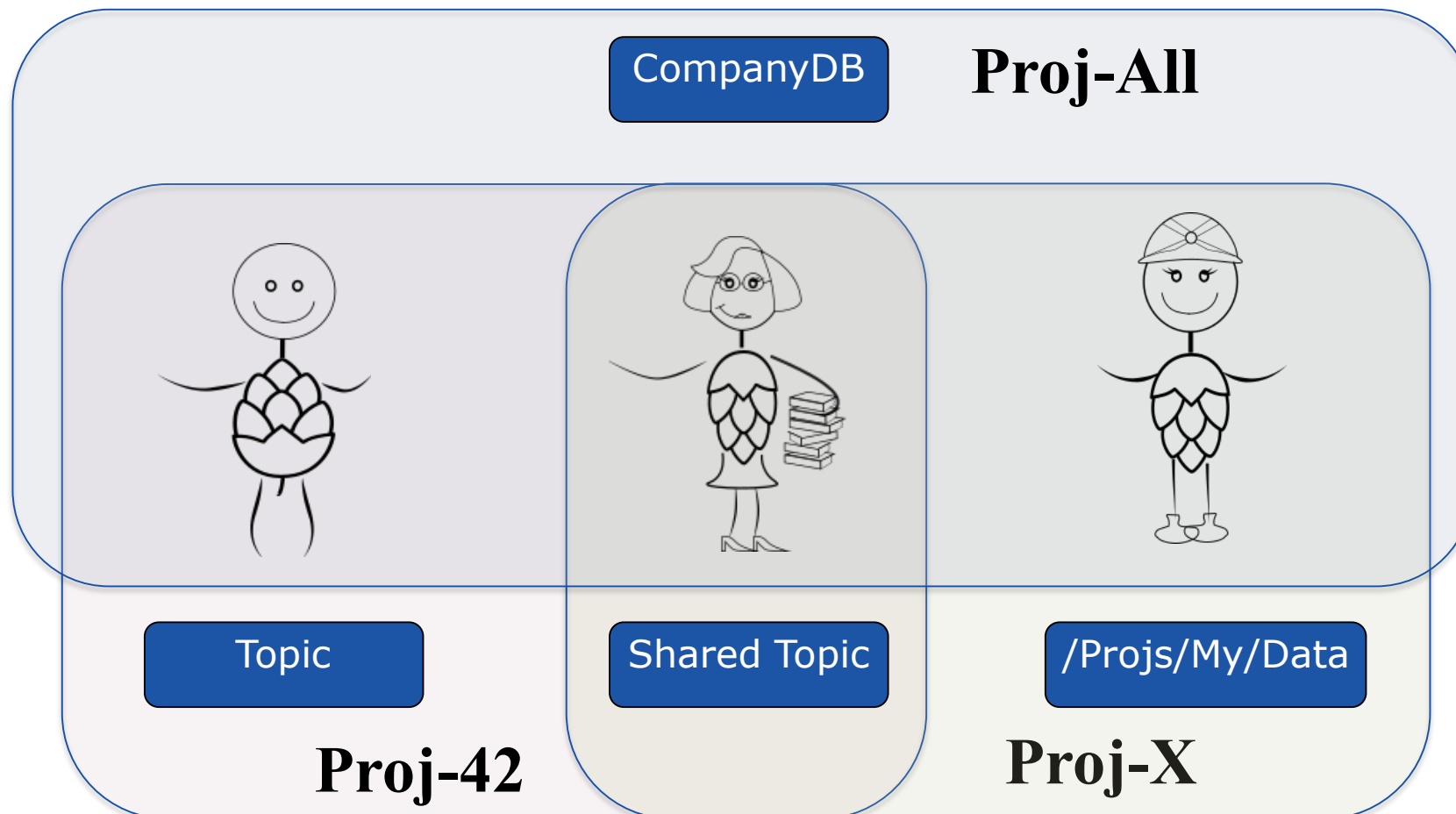
Hops

- Projects
 - Datasets/Topics
 - Per project users
 - Jobs/Notebooks
 - SSL/TLS

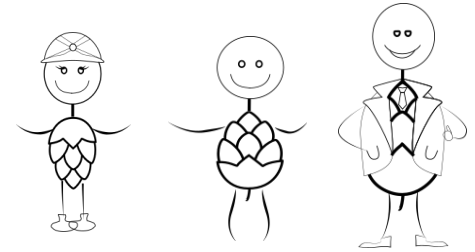
Privacy-by-Design with Projects, Data, Users

Projects for Software-as-a-Service

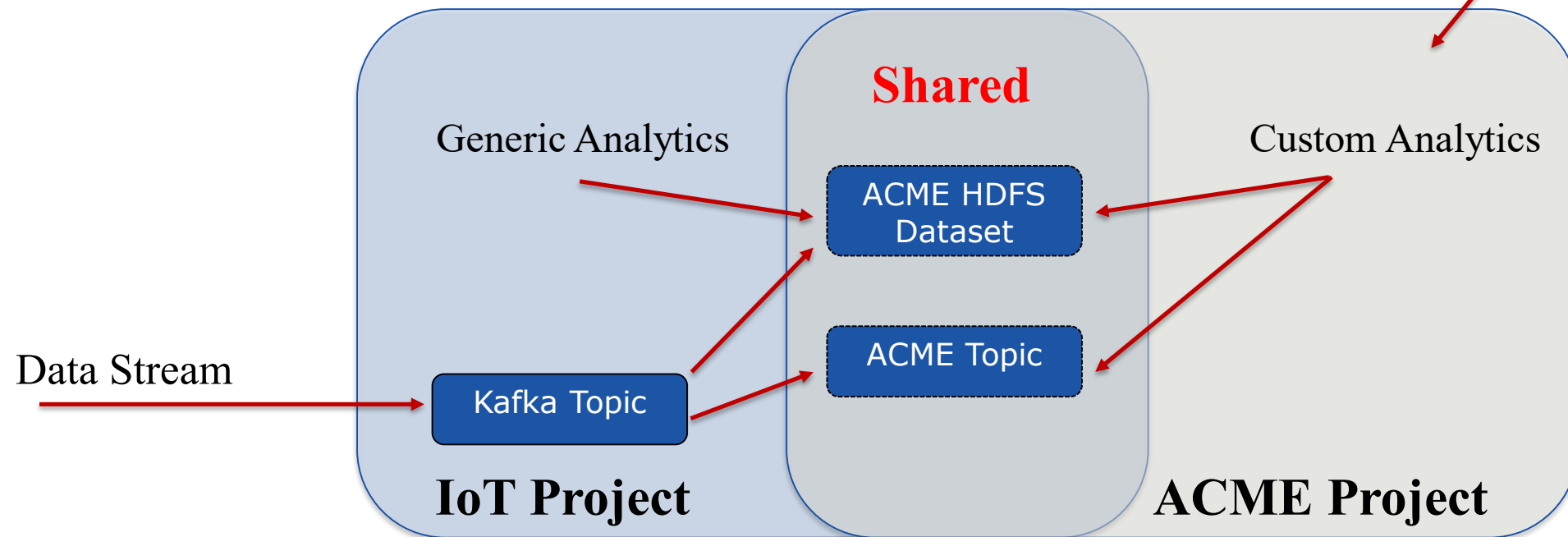
A **Project** is a Grouping of **Users** and **Data**



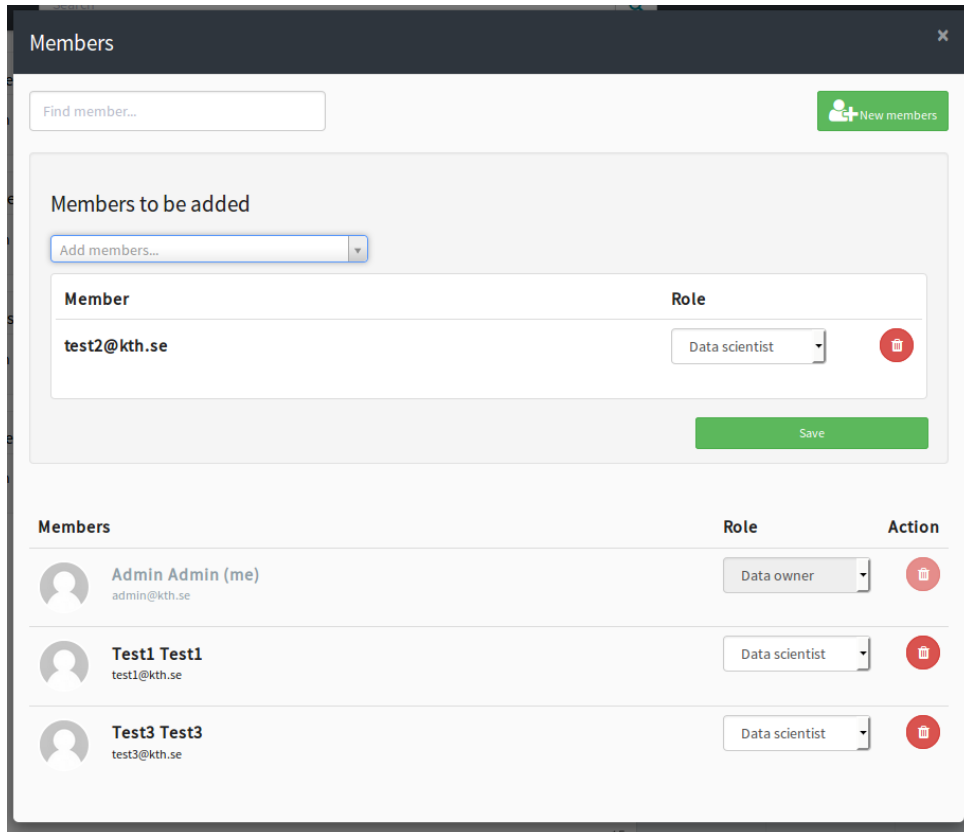
SaaS IoT Platform: Project per Customer



ACME manage membership



Project Roles



- Data Owner Privileges
 - Import/Export data
 - Manage Membership
 - Share DataSets, Topics
- Data Scientist Privileges
 - Write and Run code

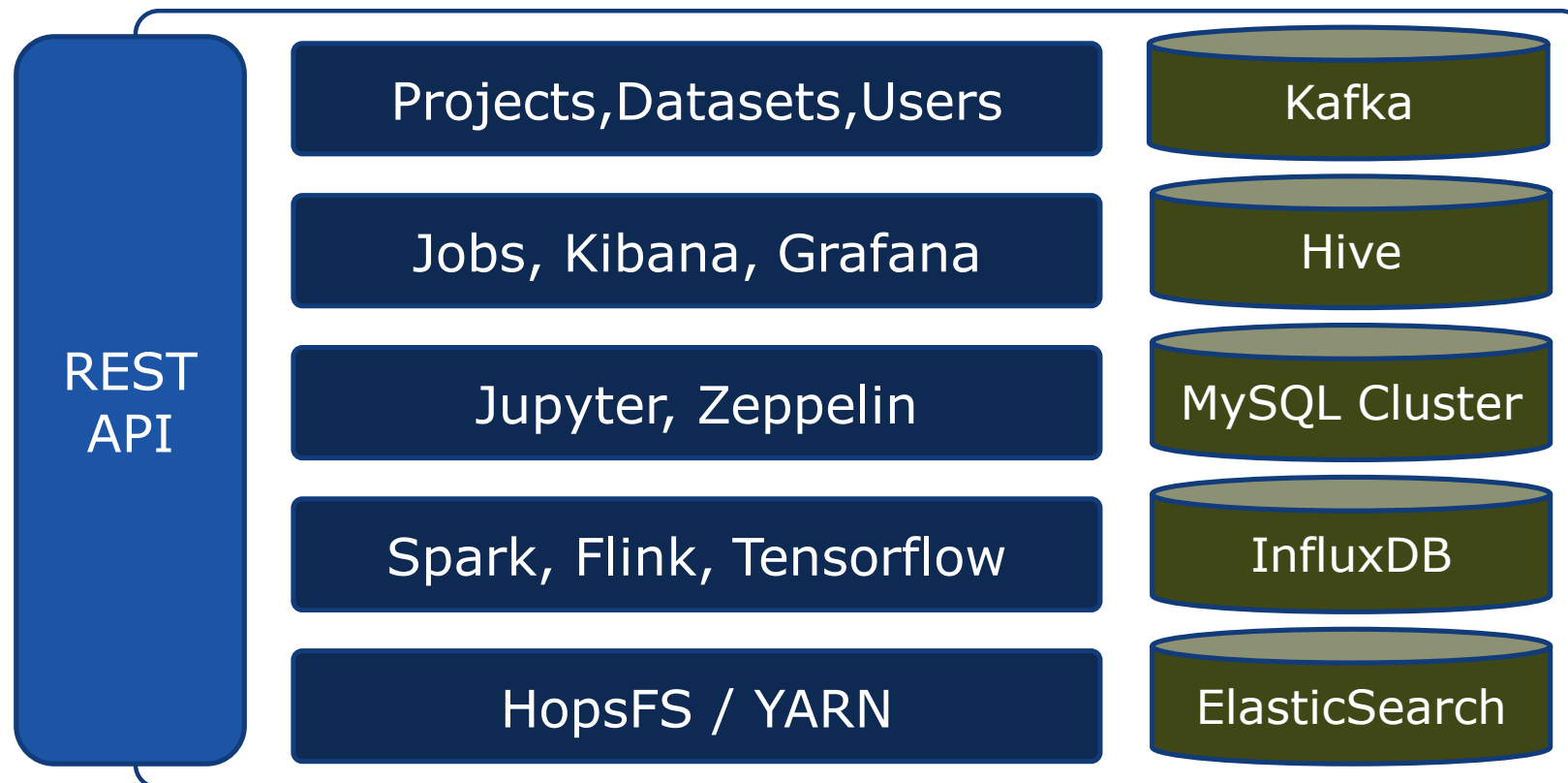
We delegate administration of privileges to users

Hopsworks: Hadoop for Humans

Hopsworks Data Platform

Develop → Train → Test → Deploy

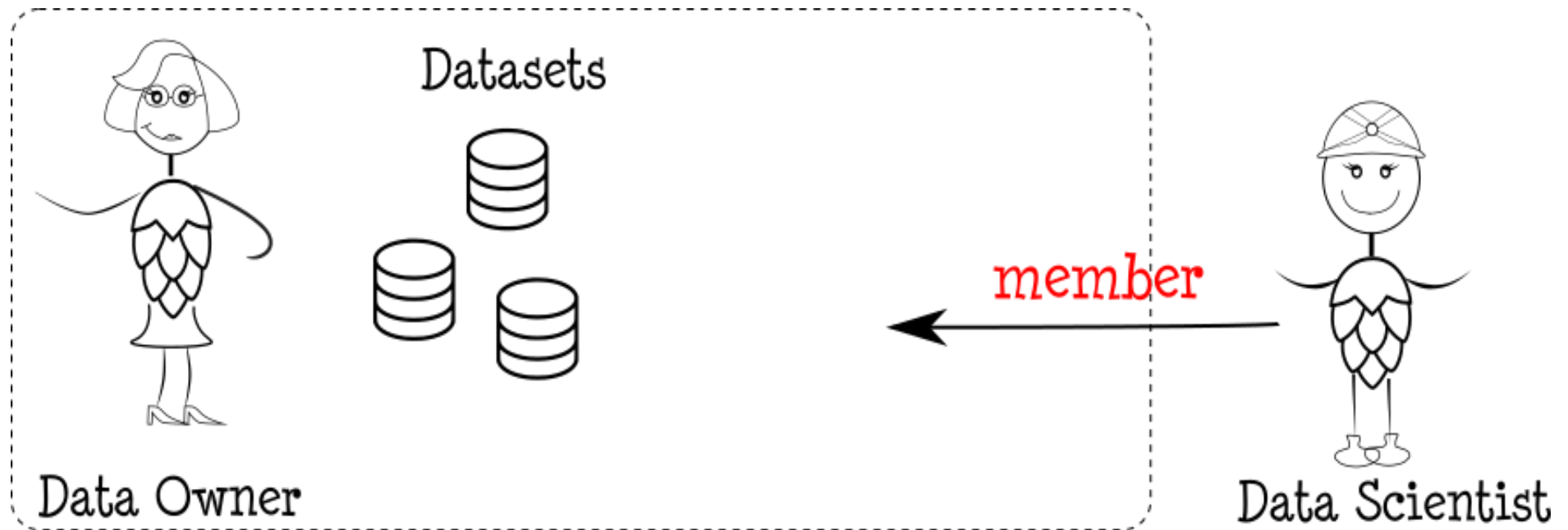
Hopsworks



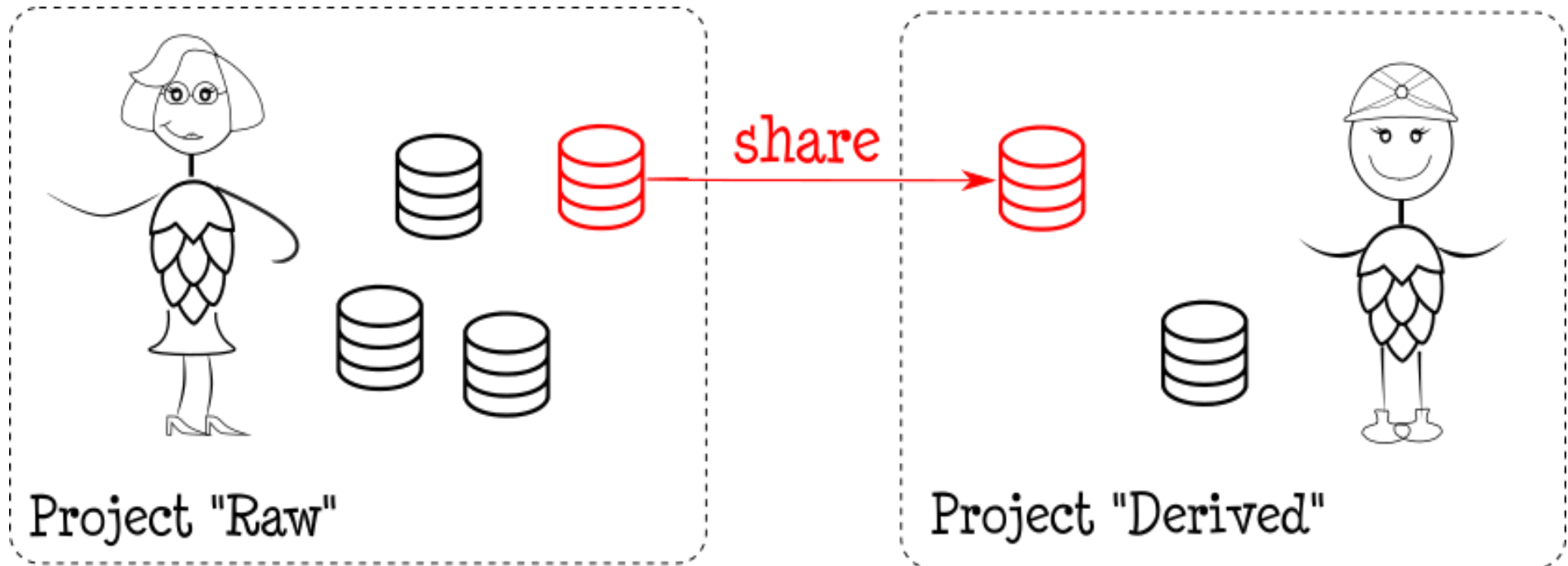
Hopsworks



Manage Projects like GitHub

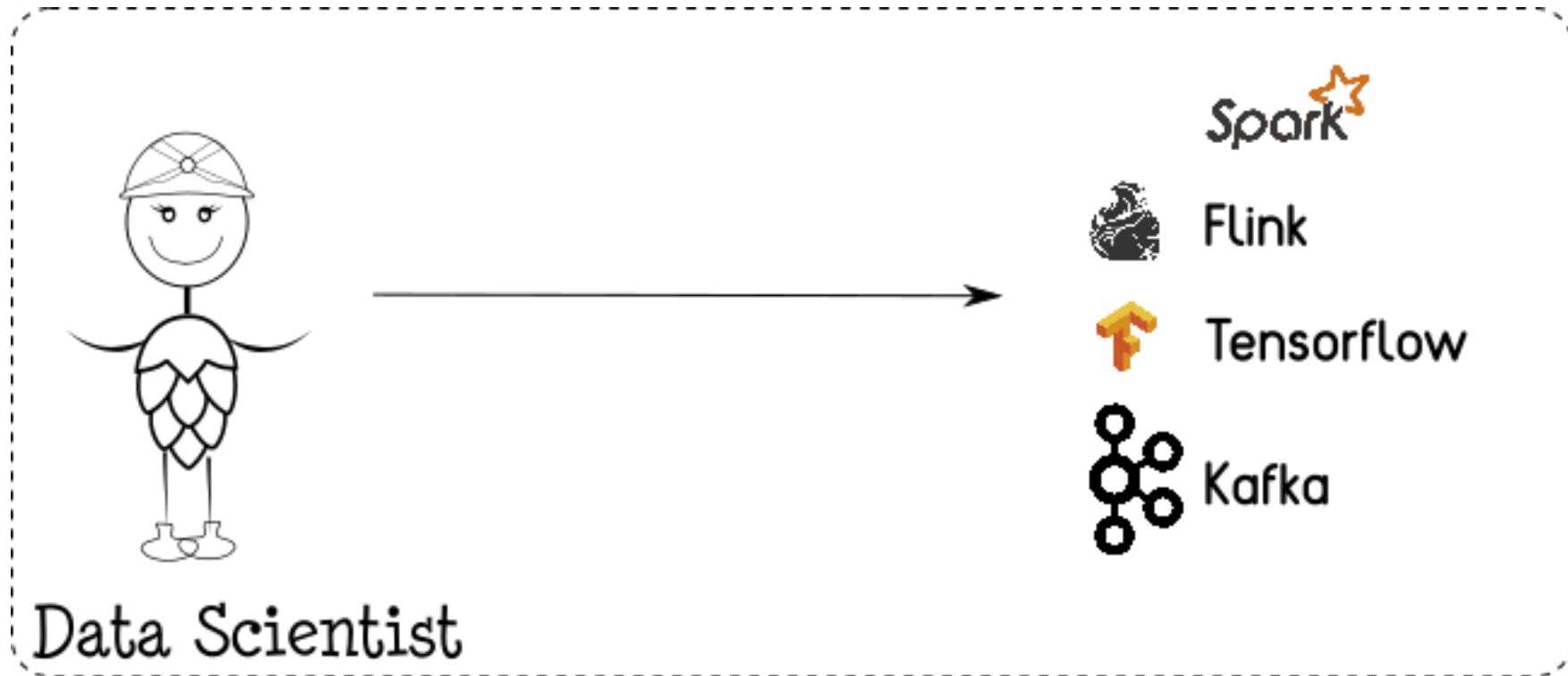


Share like in Dropbox

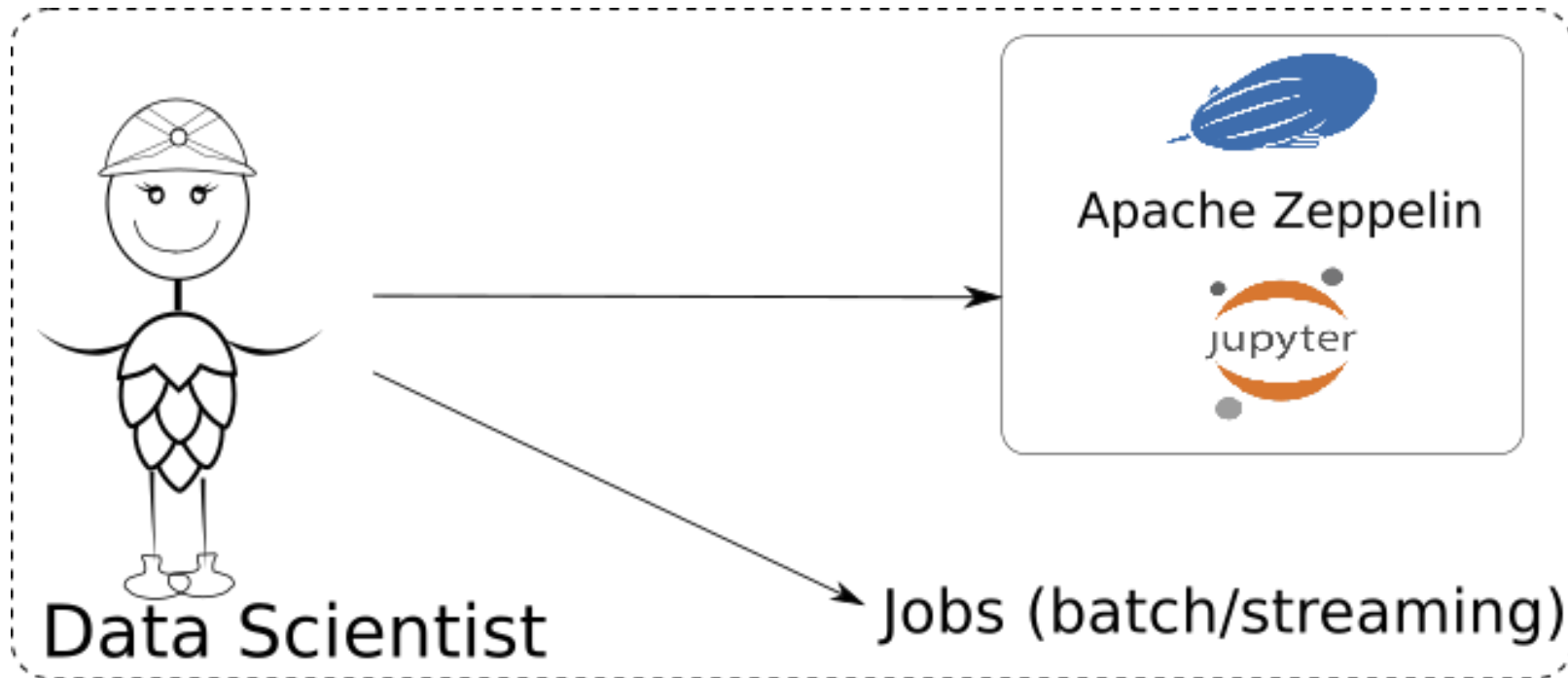


Share any Data Source/Sink: HDFS Datasets, Kafka Topics, etc

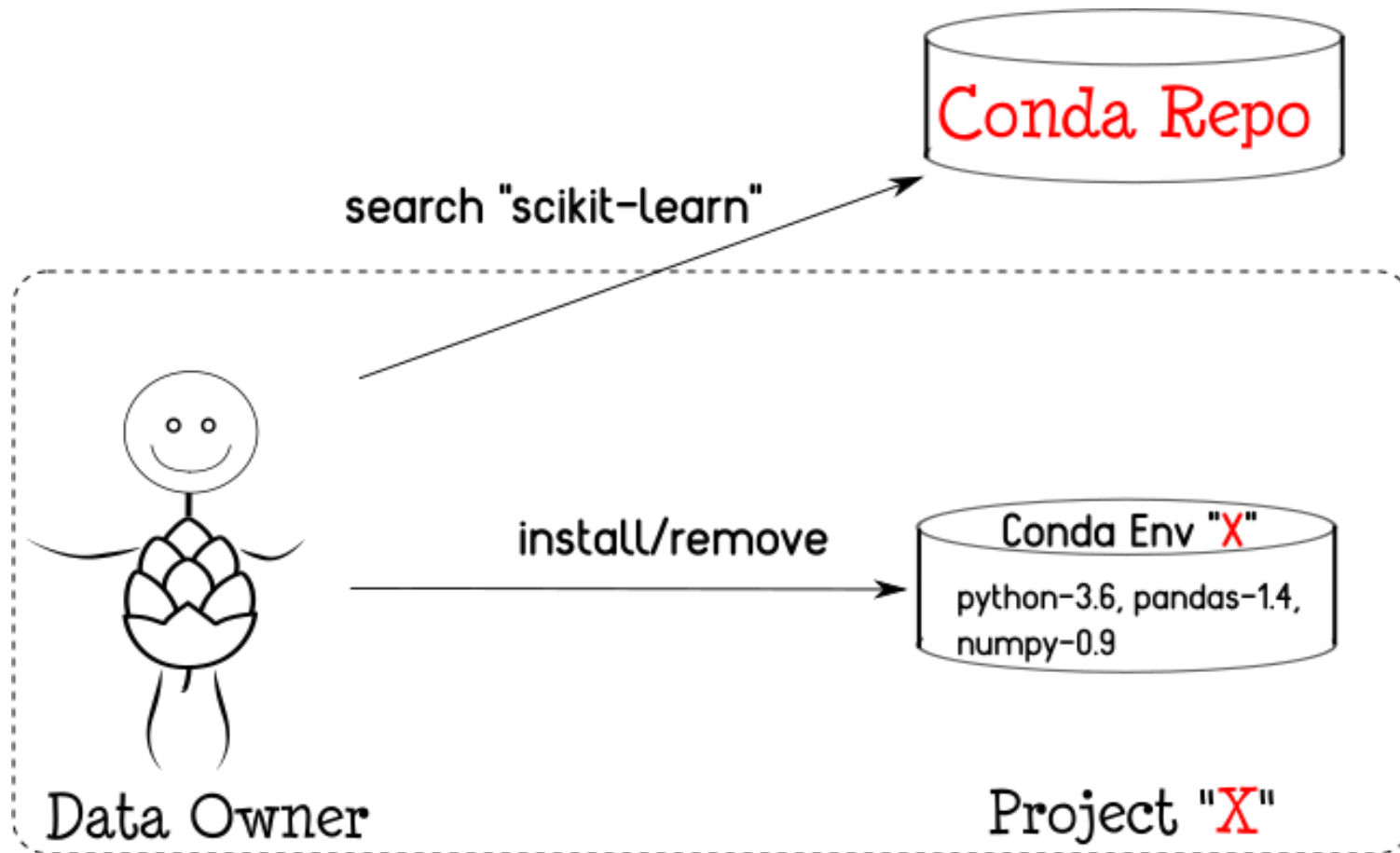
Modern Data Parallel Platforms



Workflow/Jobs and Notebook Support

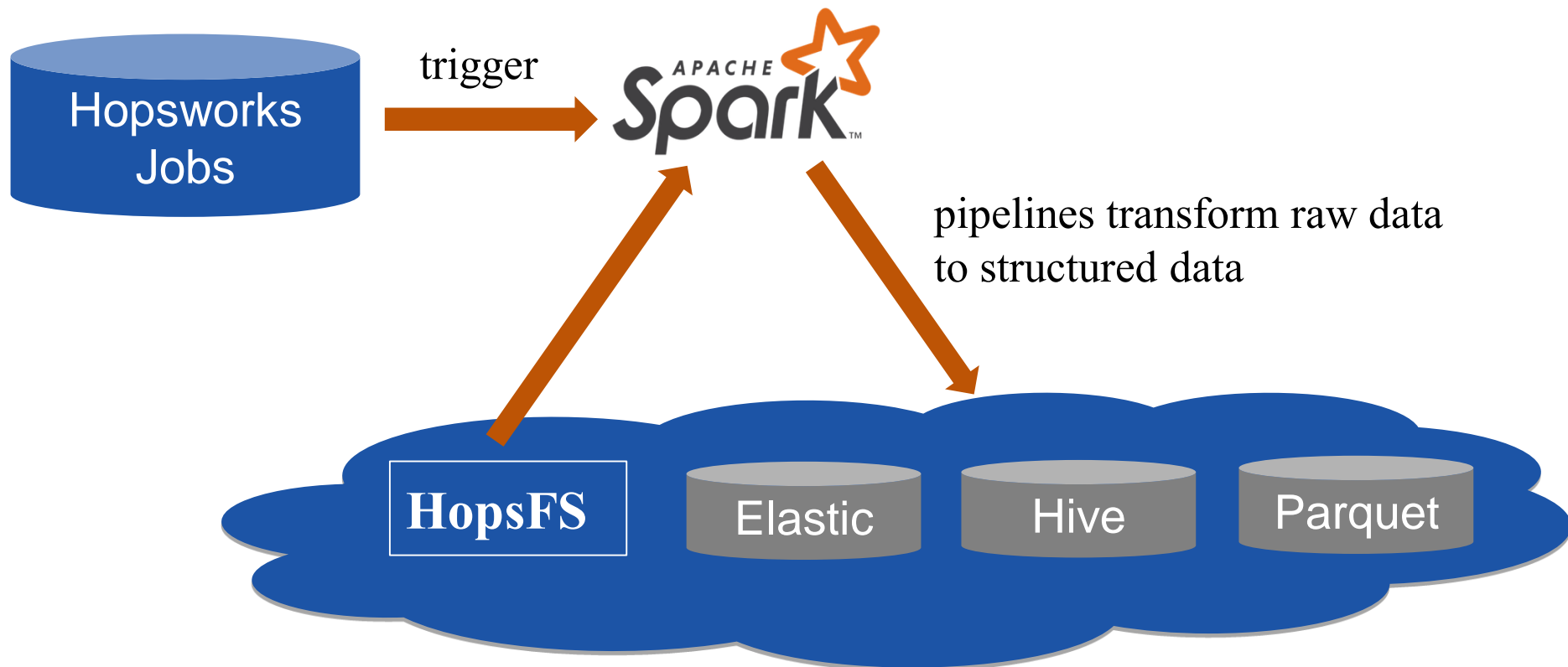


Custom Python Environments with Conda

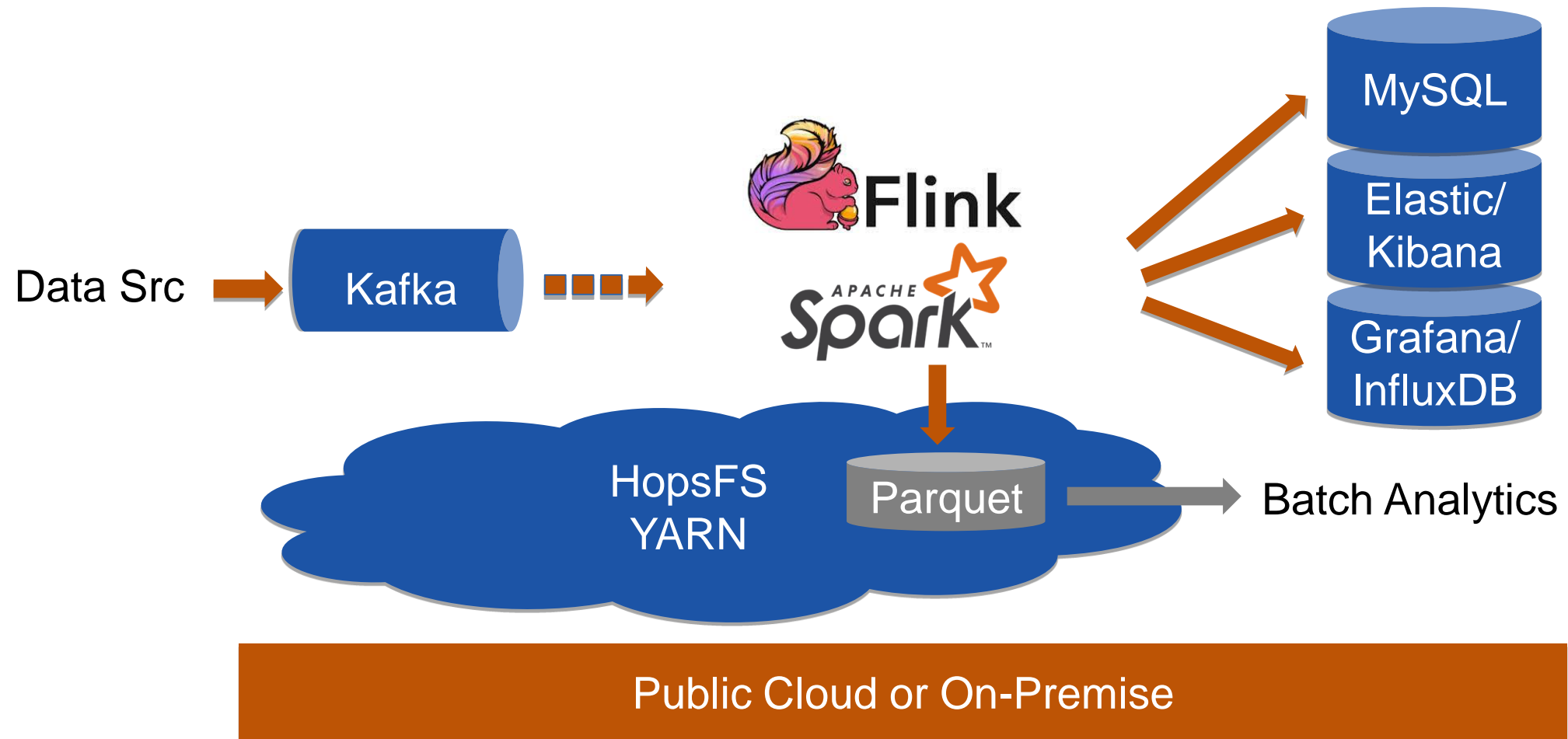


Python libraries are usable by Spark/Tensorflow

ETL Workloads (Jobs UI)



Streaming Analytics in Hopsworks




Look Ma, No Kerberos!

- For each project, a user is issued with a SSL/TLS(X.509) certificate for both authentication and encryption.
 - Project based access on Kafka resources.
 - Custom Authorizer
- Services are also issued with SSL/TLS certificates.
 - Both user and service certs are signed with the same CA.
 - Services extract the userID from RPCs to identify the caller.
 - HADOOP-13836
- Draws on ideas from Netflix' BLESS system

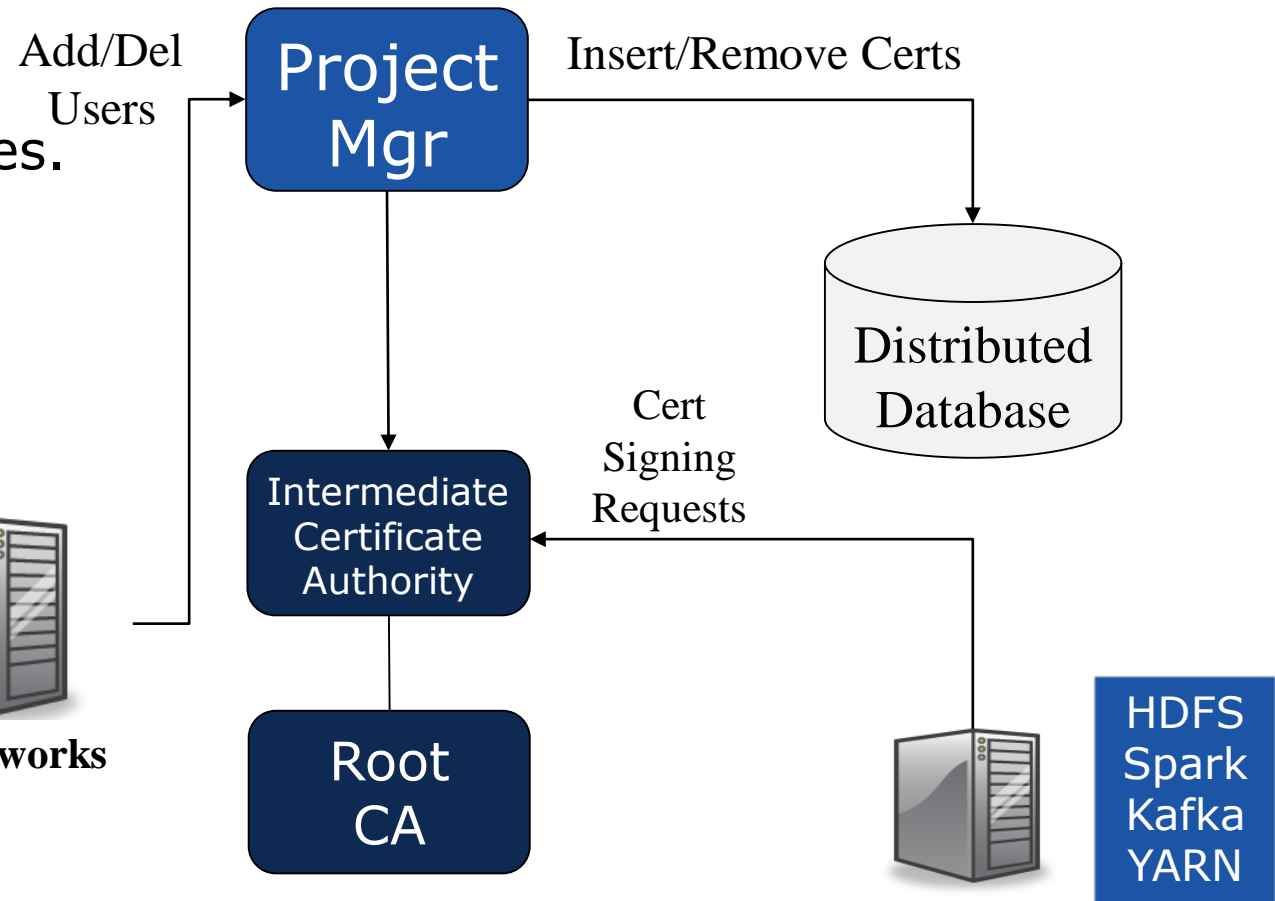
SSL/TLS Certificate Generation

Users don't see the certificates.
Users authenticate using:

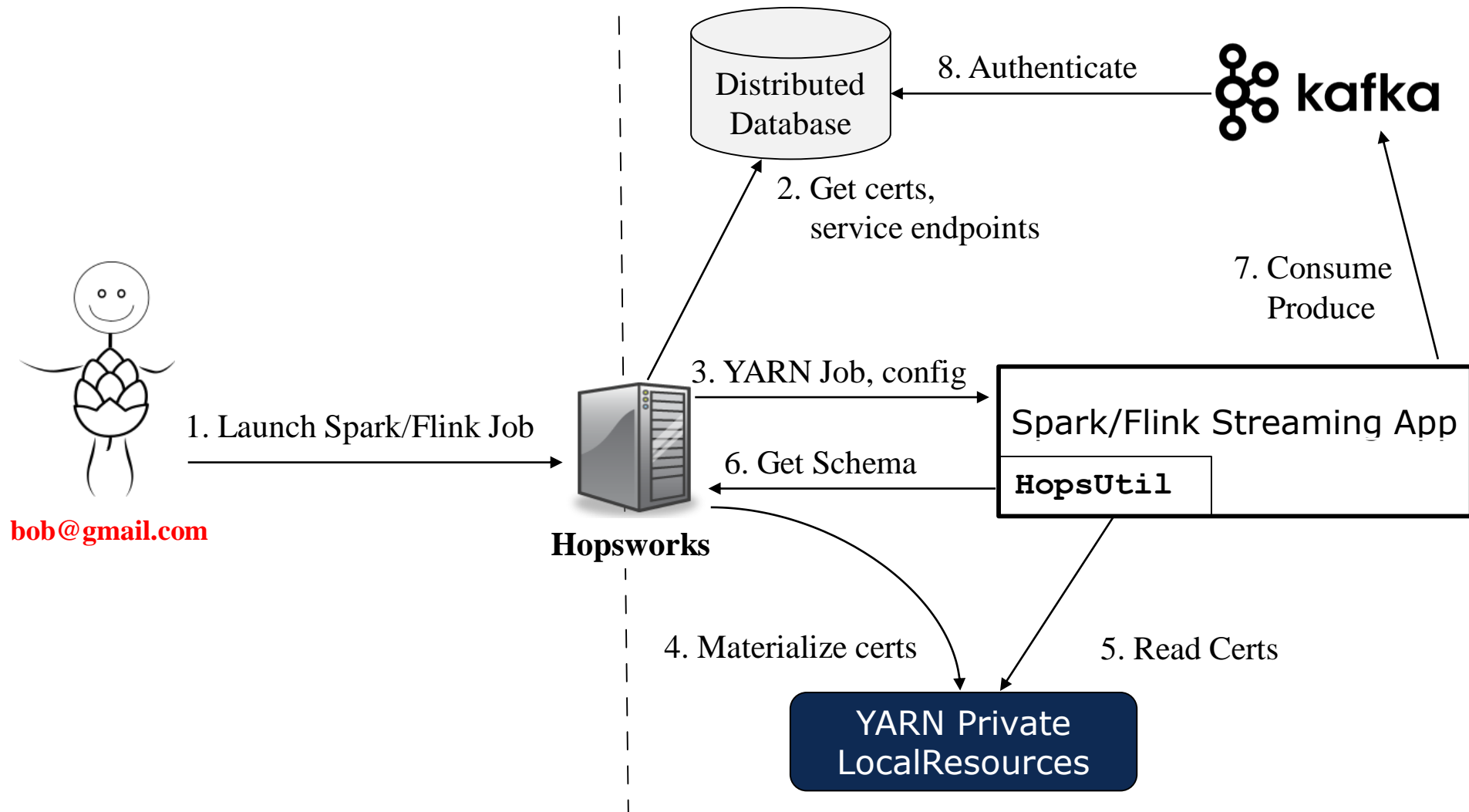
- LDAP
- password
- 2-Factor Authentication


bob@gmail.com


Hopworks



Distributing Certs for Spark/Flink Streaming



Making Big Data Easier to Program

HopsUtil

Secure Structured Spark Streaming App

- 1. Discover:** Schema Registry and Kafka Broker Endpoints
 - 2. Create:** Kafka Properties file with certs and broker details
 - 3. Create:** Producer/Consumer using Kafka Properties
- Developer

- 4. Download:** the Schema for the Topic from the Schema Registry
 - 5. Distribute:** X.509 certs to all hosts on the cluster
 - 6. Cleanup securely**
- Operations

HopsUtil simplifies Secure Spark/Kafka

```
Properties props = new Properties();
props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG,
brokerList);
props.put(SCHEMA_REGISTRY_URL, restApp.restConnect);
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
org.apache.kafka.common.serialization.StringSerializer.class);
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
io.confluent.kafka.serializers.KafkaAvroSerializer.class);
props.put("producer.type", "sync");
props.put("serializer.class", "kafka.serializer.StringEncoder")
;
props.put("request.required.acks", "1");
props.put("ssl.keystore.location", "/var/ssl/kafka.client.keyst
ore.jks")
props.put("ssl.keystore.password", "test1234")
props.put("ssl.key.password", "test1234")
ProducerConfig config = new ProducerConfig(props);
String userSchema = "{\"namespace\": \"example.avro\",
\"type\": \"record\", \"name\": \"User\", \" +
                \"fields\": [{\"name\": \"name\",
\"type\": \"string\"}]}}";
Schema.Parser parser = new Schema.Parser();
Schema schema = parser.parse(userSchema);
GenericRecord avroRecord = new GenericData.Record(schema);
avroRecord.put("name", "testUser");
Producer<String, String> producer = new Producer<String,
String>(config);
ProducerRecord<String, Object> message = new
ProducerRecord<>("topicName", avroRecord );
producer.send(data);
```

```
SparkProducer producer =
HopsUtil.getSparkProducer();
Map<String, String> message = ...
sparkProducer.produce(message);
```

Lots of Hard-Coded Endpoints Here!

<https://github.com/hopshadoop/hops-util>

Services for Streaming Analytics

Kafka Self-Service UI

The screenshot displays the HopsWorks interface for managing Kafka topics. The left sidebar shows navigation options like 'producer', 'Zepelin', 'Jobs', 'Jobs History', 'Kafka', 'Data Sets', 'Settings', 'Members', and 'Metadata Designer'. The main content area shows the 'Topics' tab with a search bar and a 'New Topic +' button. Below, the 'hellotopic' is detailed with its schema (kafkaschema (1)), ACLs, and partitions.

Project	UserEmail	Permission	Operation	Host	Role	Remove	Edit
producer	admin@kth.se	allow	*	*	*		
consumer	tkak@kth.se	allow	*	*	*		

Partition id	Partition leader	Partition replicas	Insync replicas
1	10.0.2.15	["10.0.2.15"]	["10.0.2.15"]
0	10.0.2.15	["10.0.2.15"]	["10.0.2.15"]

Manage & Share

- Topics
- ACLs
- Avro Schemas

Realtime Logs

- YARN aggregates logs on job completion
 - No good to us for Streaming
- Collect logs and make them searchable in real-time using Logstash, Elasticsearch, and Kibana
 - Log4j auto-configured to write to Logstash

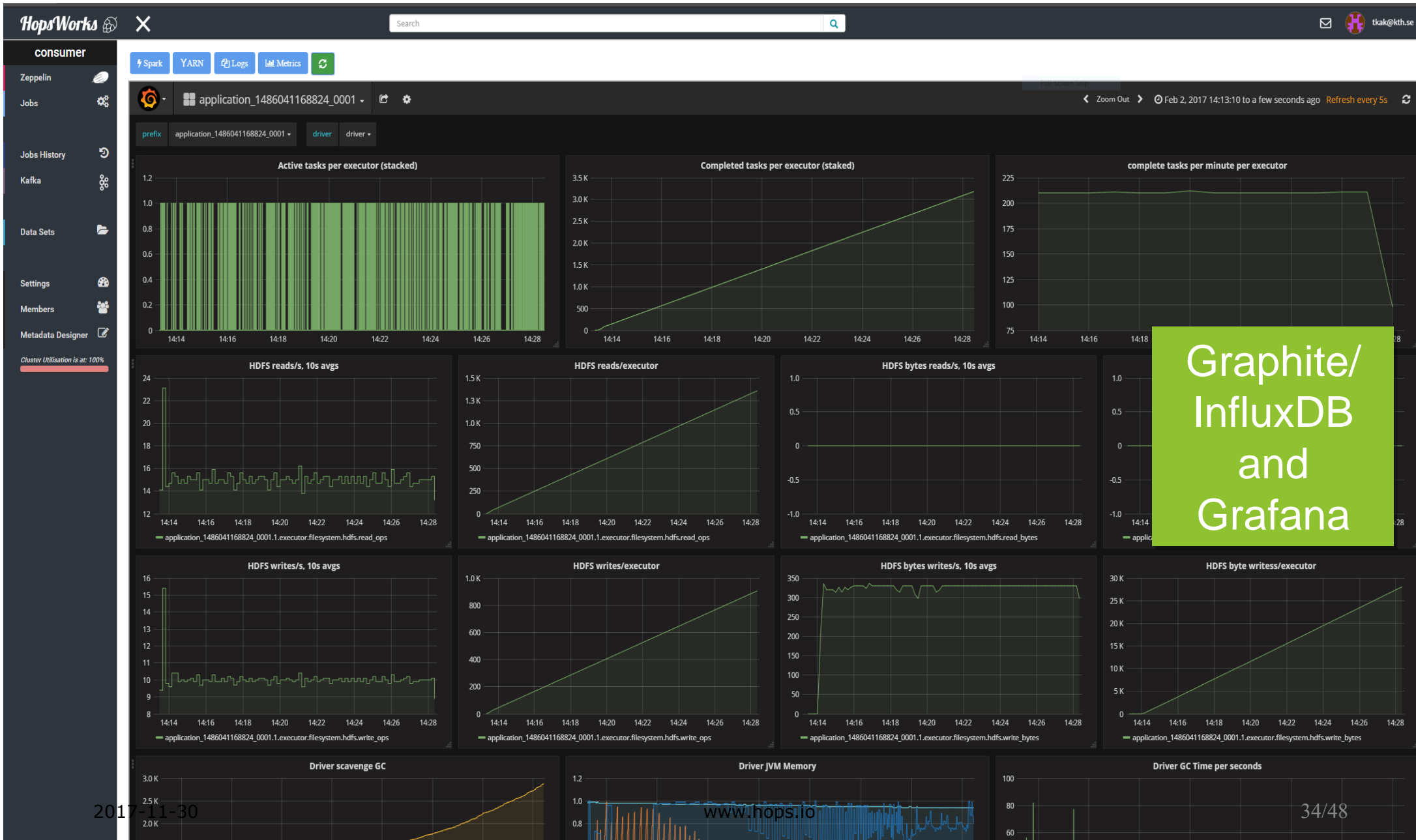
<http://mkuthan.github.io/blog/2016/09/30/spark-streaming-on-yarn/>

Realtime Logs

The screenshot displays the HopsWorks Kibana interface. At the top, there's a search bar and navigation tabs for Spark, YARN, Logs, and Metrics. The left sidebar contains navigation options like Zeppelin, Jobs, Jobs History, Kafka, Data Sets, Settings, Members, and Metadata Designer. The main area shows a log stream for a consumer application with columns for @timestamp, priority, application, logger_name, thread, and message. The logs show various events including signal handler registration, library loading, resource preparation, and security manager operations.

@timestamp	priority	application	logger_name	thread	message	host
February 2nd 2017, 14:08:28.059	INFO	application_1486036627012_0001	org.apache.spark.util.SignalUtils	main	Registered signal handler for TERM	10.0.2.1 5:47523
February 2nd 2017, 14:08:28.095	INFO	application_1486036627012_0001	org.apache.spark.util.SignalUtils	main	Registered signal handler for HUP	10.0.2.1 5:47523
February 2nd 2017, 14:08:28.101	INFO	application_1486036627012_0001	org.apache.spark.util.SignalUtils	main	Registered signal handler for INT	10.0.2.1 5:47523
February 2nd 2017, 14:08:28.518	WARN	application_1486036627012_0001	org.apache.hadoop.util.NativeCodeLoader	main	Loaded the native-hadoop library	10.0.2.1 5:47523
February 2nd 2017, 14:08:28.724	INFO	application_1486036627012_0001	org.apache.spark.deploy.yarn.ApplicationMaster	main	Preparing Local resources	10.0.2.1 5:47523
February 2nd 2017, 14:08:28.987	INFO	application_1486036627012_0001	org.apache.spark.deploy.yarn.ApplicationMaster	main	Prepared Local resources Map(log4j.properties -> resource { scheme: "hdfs" host: "10.0.2.15" port: 8020 file: "/user/glassfish/log4j.properties" } size: 627 timestamp: 1486036278839 type: ARCHIVE visibility: PRIVATE, __app__jar -> resource { scheme: "hdfs" host: "10.0.2.15" port: 8020 file: "/Projects/producer/jars/hops-spark-0.1.jar" } size: 17027 timestamp: 1486040155175 type: FILE visibility: APPLICATION, kafka_t_certificate -> resource { scheme: "hdfs" host: "10.0.2.15" port: 8020 file: "/user/glassfish/kafkacerts/consumer__tkak0000/consumer__tkak0000_tstore.jks" } size: 1478 timestamp: 1486040903923 type: FILE visibility: APPLICATION, hops-util-0.1.jar -> resource { scheme: "hdfs" host: "10.0.2.15" port: 8020 file: "/Projects/producer/jars/hops-util-0.1.jar" } size: 99768946 timestamp: 1486040164008 type: FILE visibility: PRIVATE, kafka_k_certificate -> resource { scheme: "hdfs" host: "10.0.2.15" port: 8020 file: "/user/glassfish/kafkacerts/consumer__tkak0000/consumer__tkak0000_kstore.jks" } size: 4082 timestamp: 1486040903990 type: FILE visibility: APPLICATION,	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.072	INFO	application_1486036627012_0001	org.apache.spark.deploy.yarn.ApplicationMaster	main	ApplicationAttemptId: appattemp_1486036627012_0001_000001	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.519	INFO	application_1486036627012_0001	org.apache.spark.SecurityManager	main	Changing view acls to: glassfish,consumer__tkak0000	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.521	INFO	application_1486036627012_0001	org.apache.spark.SecurityManager	main	Changing modify acls to: glassfish,consumer__tkak0000	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.535	INFO	application_1486036627012_0001	org.apache.spark.SecurityManager	main	Changing view acls groups to:	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.538	INFO	application_1486036627012_0001	org.apache.spark.SecurityManager	main	Changing modify acls groups to:	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.540	INFO	application_1486036627012_0001	org.apache.spark.SecurityManager	main	SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(glassfish, consumer__tkak0000); groups with view permissions: Set(); users with modify permissions: Set(glassfish, consumer__tkak0000); groups with modify permissions: Set()	10.0.2.1 5:47523
February 2nd 2017, 14:08:29.545	INFO	application_1486036627012_0001	org.apache.spark.deploy.yarn.ApplicationMaster	main	Starting the user application in a separate Thread	10.0.2.1 5:47523

Resource Monitoring/Alerting



Zeppelin Notebooks

The screenshot displays the HopsWorks Zeppelin interface. At the top, there's a navigation bar with the HopsWorks logo, a search bar, and a user profile for 'admin@kth.se'. Below this is a sidebar with navigation options: 'ff', 'Zeppelin', 'Jobs', 'Jobs History', 'Kafka', 'Data Sets', 'Settings', 'Members', and 'Metadata Designer'. The main area shows a 'Create New Notebook' button and a 'Goto Zeppelin' button. A central panel displays a Zeppelin notebook with three SQL queries and their visualizations:



- Query 1:** `select age, count(1) value from bank group by age order by age`. Visualization: Histogram of age with a peak around 48. Y-axis ranges from 0.0 to 2,085.0.
- Query 2:** `select age, job, count(1) value from bank where age < ${maxAge=30} group by age, job order by age`. Visualization: Histogram of job for ages < 30. Y-axis ranges from 0.0 to 12.0.
- Query 3:** `select age, count(1) value from bank where marital=${marital=single, single|divorced|married} group by age`. Visualization: Histogram of age for a specific marital status. Y-axis ranges from 0.0 to 1,017.0.


On the right side, there's a 'Connected' status indicator and a list of interpreters:

INTERPRETERS	ADVANCED
flink Interpreter	running
angular Interpreter	stopped
livy Interpreter	stopped
spark Interpreter	stopped
md Interpreter	stopped

i Running a paragraph in a notebook will automatically start the necessary interpreters for that job.

Jupyter Notebooks

HopsWorks X Search  

Demo  **jupyter Demo** Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help PySpark

In [5]:

```
df = spark.read.csv('hdfs://192.168.56.101:8020/Projects/Demo/DemoData/mcs_201611.csv', \
                    sep=';', schema=schema, ignoreLeadingWhiteSpace=True, \
                    ignoreTrailingWhiteSpace=True, \
                    timestampFormat='yyyy-MM-dd HH:mm:ss.SSS')
```

In [6]:

```
df.count()
```


85275962

In [7]:

```
df.createOrReplaceTempView("FlowData")
```

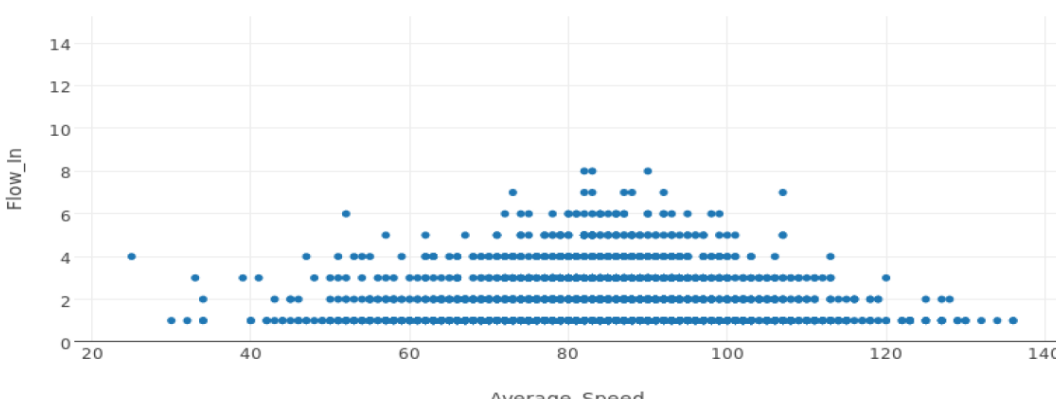
In [9]:

```
%%sql -o query1
SELECT Timestamp, Ds_Reference, Detector_Number, Flow_In, Average_Speed FROM FlowData WHERE Status == 3
```

Type: **Table** Pie Scatter Line Area Bar

Encoding:
X: Average_Speed
Y: Flow_In
Func: -

Log scale X
Log scale Y



The scatter plot displays the relationship between Average_Speed (X-axis, ranging from 20 to 140) and Flow_In (Y-axis, ranging from 0 to 14). The data points are blue dots, showing a dense distribution of points across the range of Average_Speed, with Flow_In values generally between 1 and 8. There is a slight increase in Flow_In values as Average_Speed increases, particularly between 80 and 100.

Per-project Conda Libs in Hopsworks

The screenshot shows the Hopsworks web interface. At the top, there's a search bar and a user profile for 'admin@kth.se'. The left sidebar contains navigation options: Jupyter, Zeppelin, Jobs, Kafka, Data Sets, Settings, Members, and Metadata Designer. The main content area is titled 'Install Python Libraries with Anaconda' and shows the 'Python Libraries' tab for the 'sparksummit' project. A search bar contains 'pandas'. Below it, a table lists the libraries to be installed:

Library Name	Version	Status	Action
pandas	0.20.3	Not Installed	Install
pandas-datareader	0.5.0	Not Installed	Install
pandas-profiling	1.4.0	Not Installed	Install
pandasql	0.7.3	Not Installed	Install

Dela* – A Global Ecosystem for Datasets

The screenshot shows the HopsWorks web interface. At the top left is the HopsWorks logo. A search bar contains the text 'milli' with a search icon to its right. Below the search bar, a light gray box contains the text 'Search results'. Underneath, it says '3 Results found for: milli'. Three dataset cards are displayed in a row. Each card has a title, owner information, date created, and size. The first card is for 'millimetre' (0.2KB), the second for 'millionsong' (0.4KB), and the third for 'millionthings' (0.4KB). Each card also has a download icon and a menu icon.

Dataset Name	Owner	Date Created	Size
millimetre	Admin Admin	2/23/17 7:44 AM	0.2KB
millionsong	Admin Admin	2/23/17 8:06 AM	0.4KB
millionthings	Admin Admin	2/23/17 7:43 AM	0.4KB

Peer-to-Peer Search and Download for Huge DataSets
(ImageNet, YouTube8M, MsCoCo, Reddit, etc)

[*http://ieeexplore.ieee.org/document/7980225/](http://ieeexplore.ieee.org/document/7980225/) (ICDCS 2017)

Becoming a BigData-AI Company

THE DATA SCIENCE **HIERARCHY OF NEEDS**

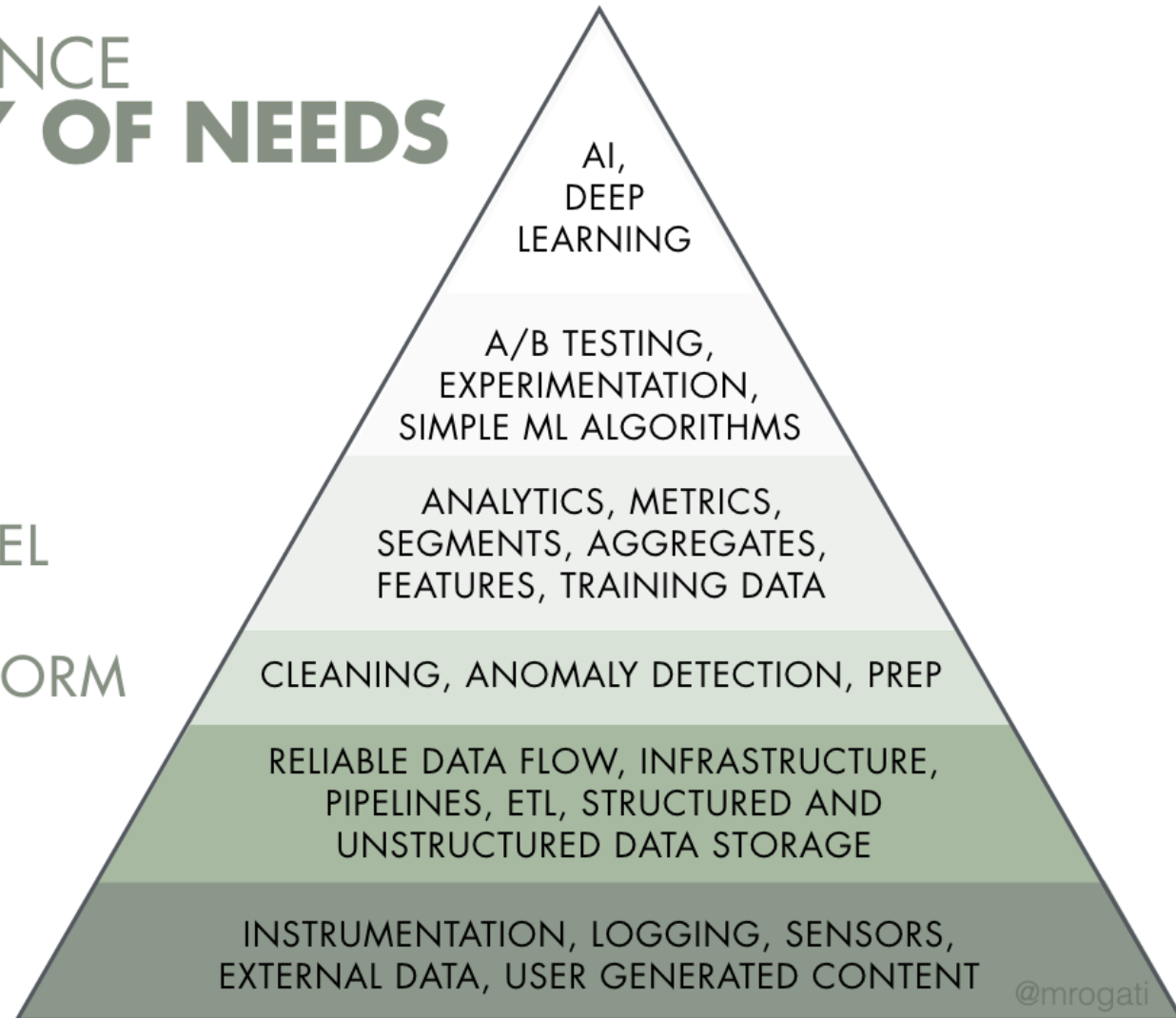
LEARN/OPTIMIZE

AGGREGATE/LABEL

EXPLORE/TRANSFORM

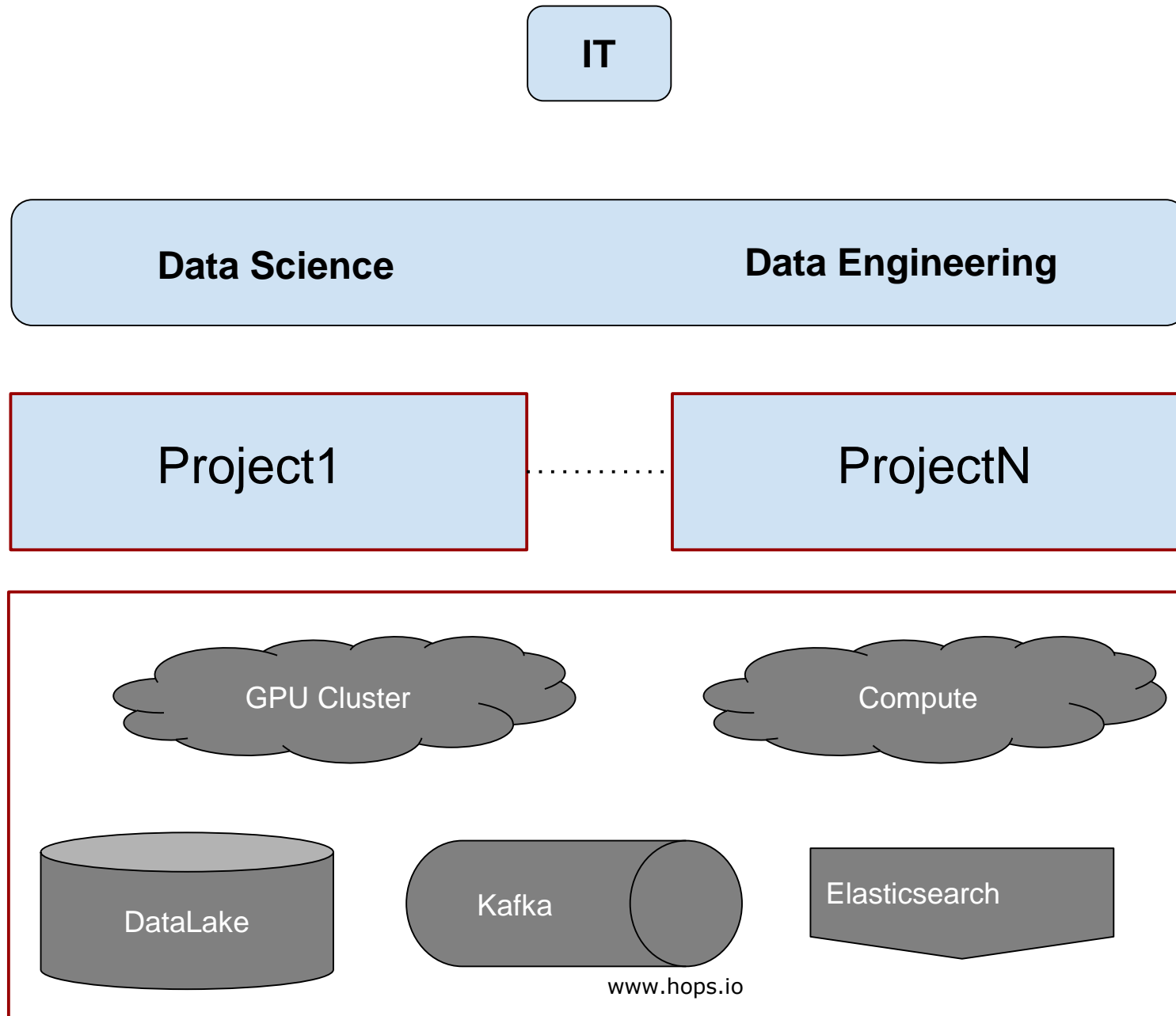
MOVE/STORE

COLLECT

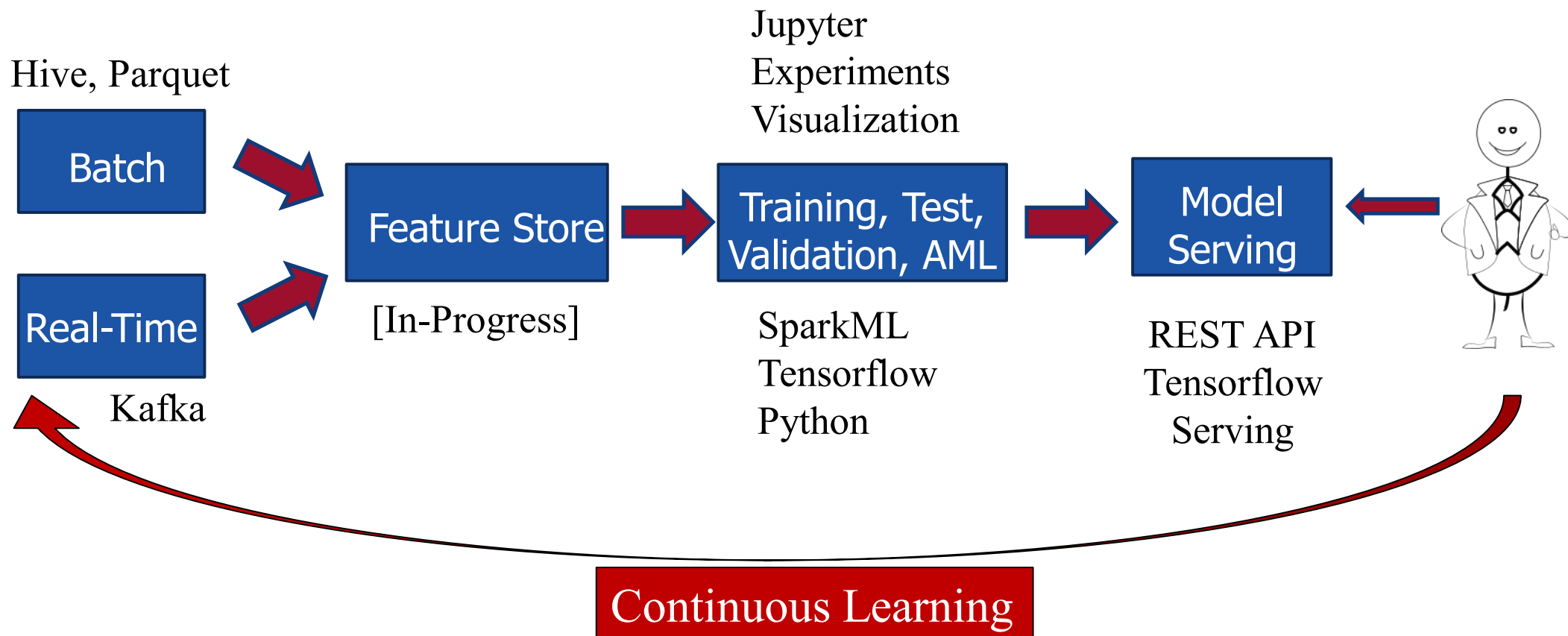


Hopsworks – TensorFlow As a Service

Hops Way: Unified AI and Big Data



Hopsworks Machine Learning-as-a-Service



Deep Learning Hierarchy of Scale

Training Time for ImageNet

Minutes

DDL
AllReduce
on GPU Servers

Hours

DDL with GPU Servers
and Parameter Servers

Days/Hours

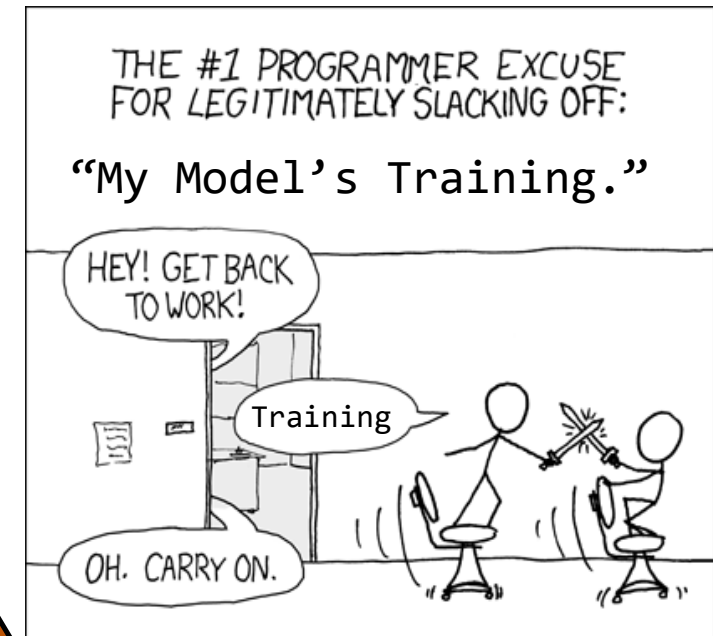
Parallel Experiments on GPU Servers

Days

Many GPUs on a Single GPU Server

Weeks

Single GPU



Lots of good GPUs > A few great GPUs

Hops



100 x Nvidia 1080Ti (DeepLearning11)

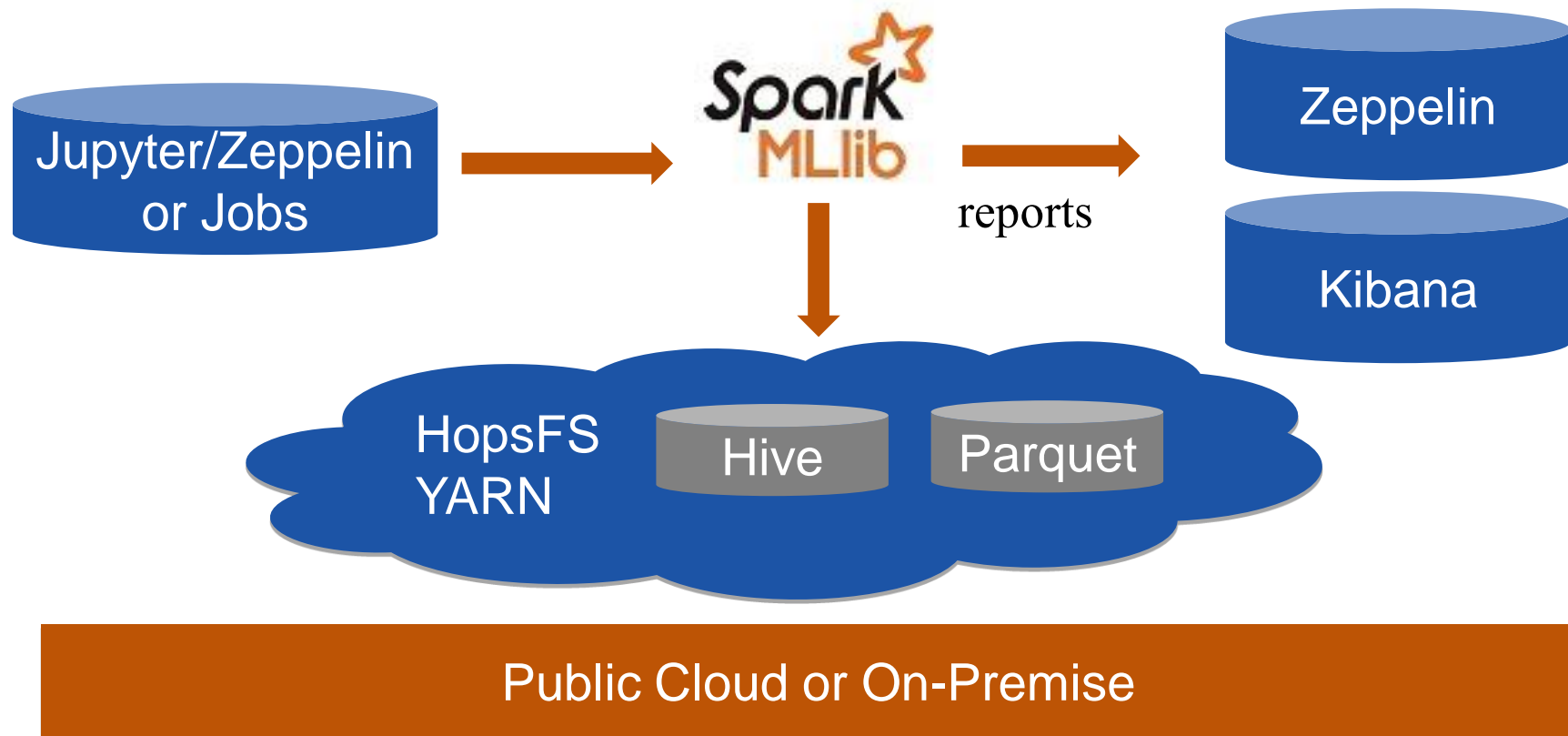
VS



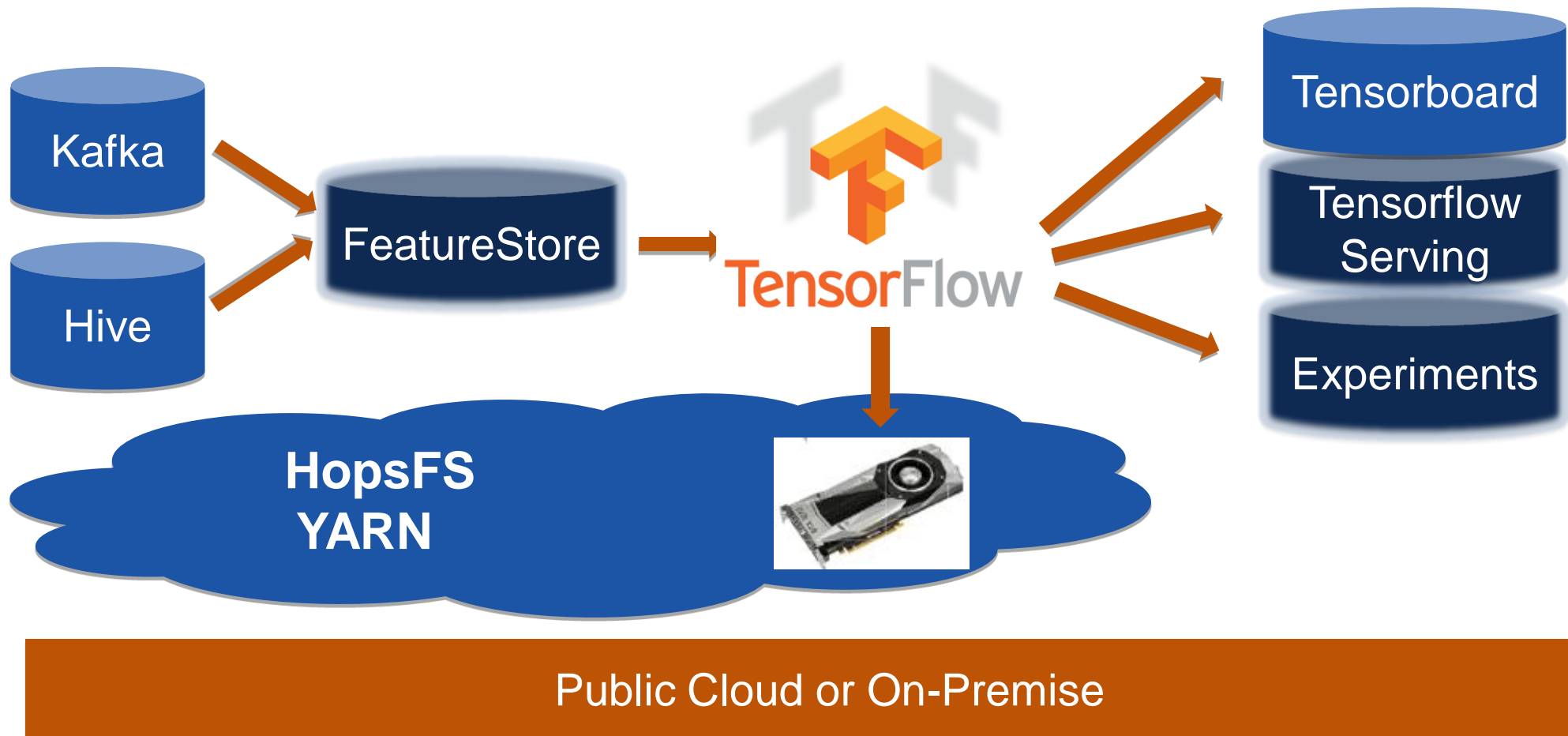
8 x Nvidia P/V100 (DGX-1)

Both top (100 GPUs) and bottom (8 GPUs) cost the same: 150K Euro.

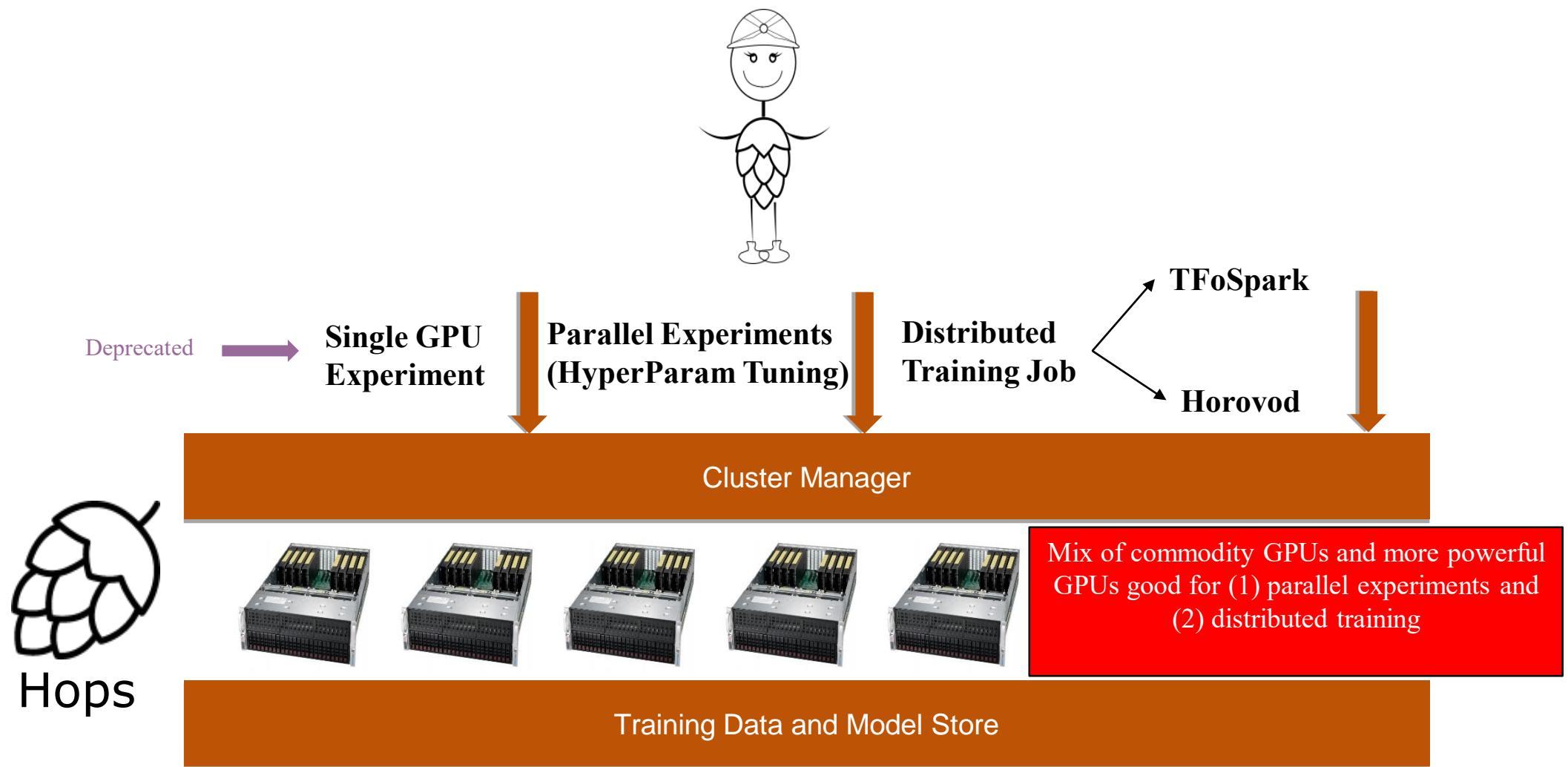
Business Intelligence Workloads



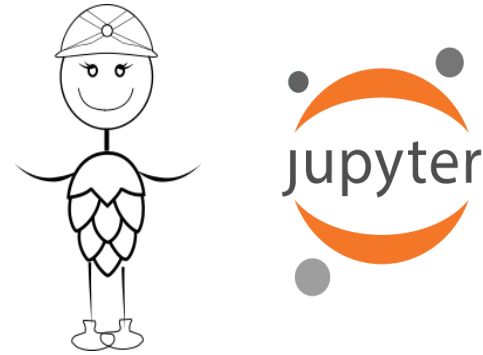
Deep Learning in Hopsworks



Spark & TensorFlow – Cluster Integration

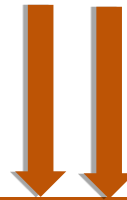


GPU Resource Requests in Hops



4 GPUs on any host
10 GPUs on 1 host

100 GPUs on 10 hosts with 'Infiniband'
20 GPUs on 2 hosts with 'Infiniband_P100'



HopsYARN (Supports GPUs-as-a-Resource)

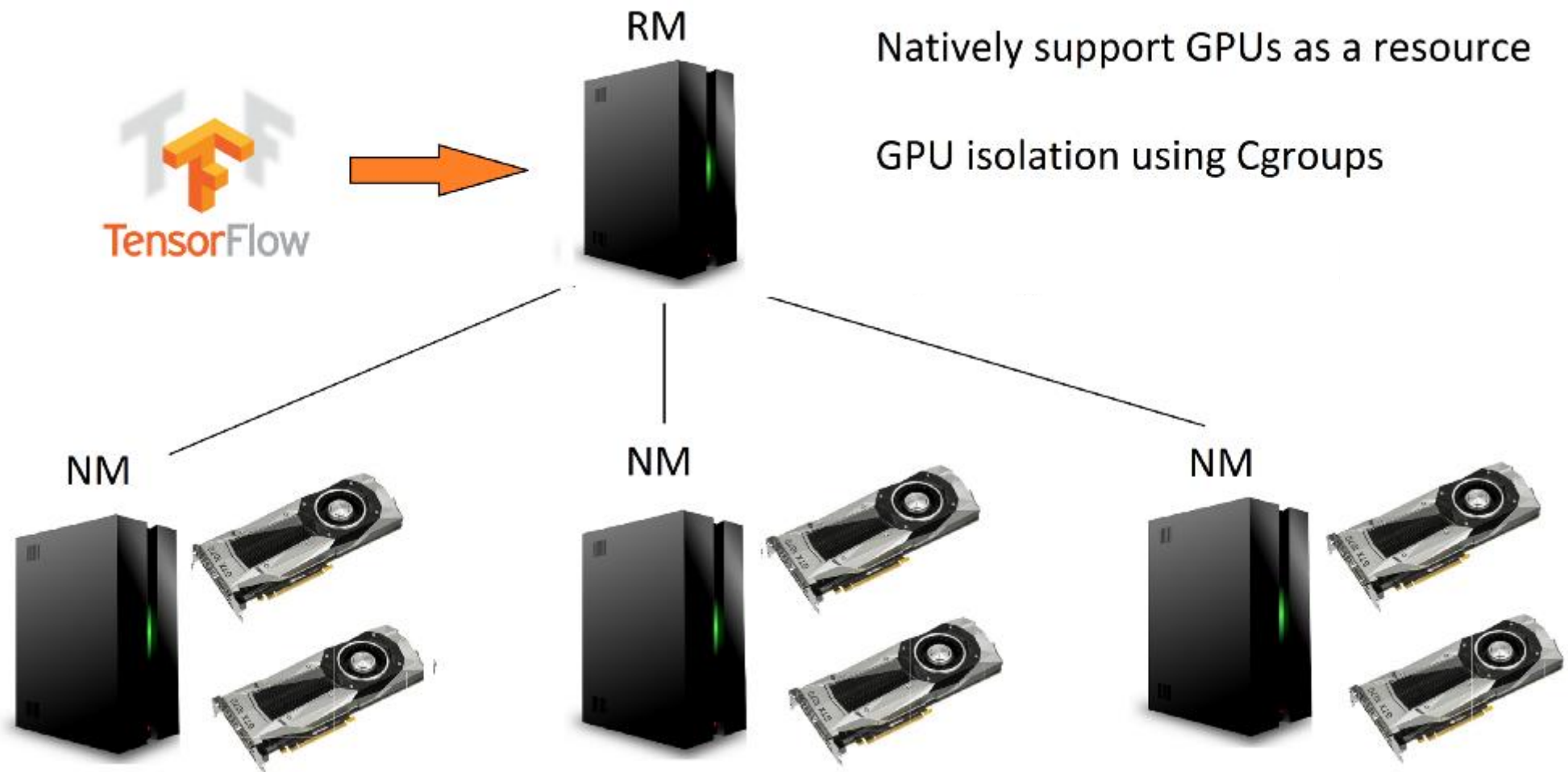


Hops



HopsFS

GPUs supported as a Resource in Hops 2.8.2*



*Robin Andersson, GPU Integration for Deep Learning on YARN, MSc Thesis, 2017

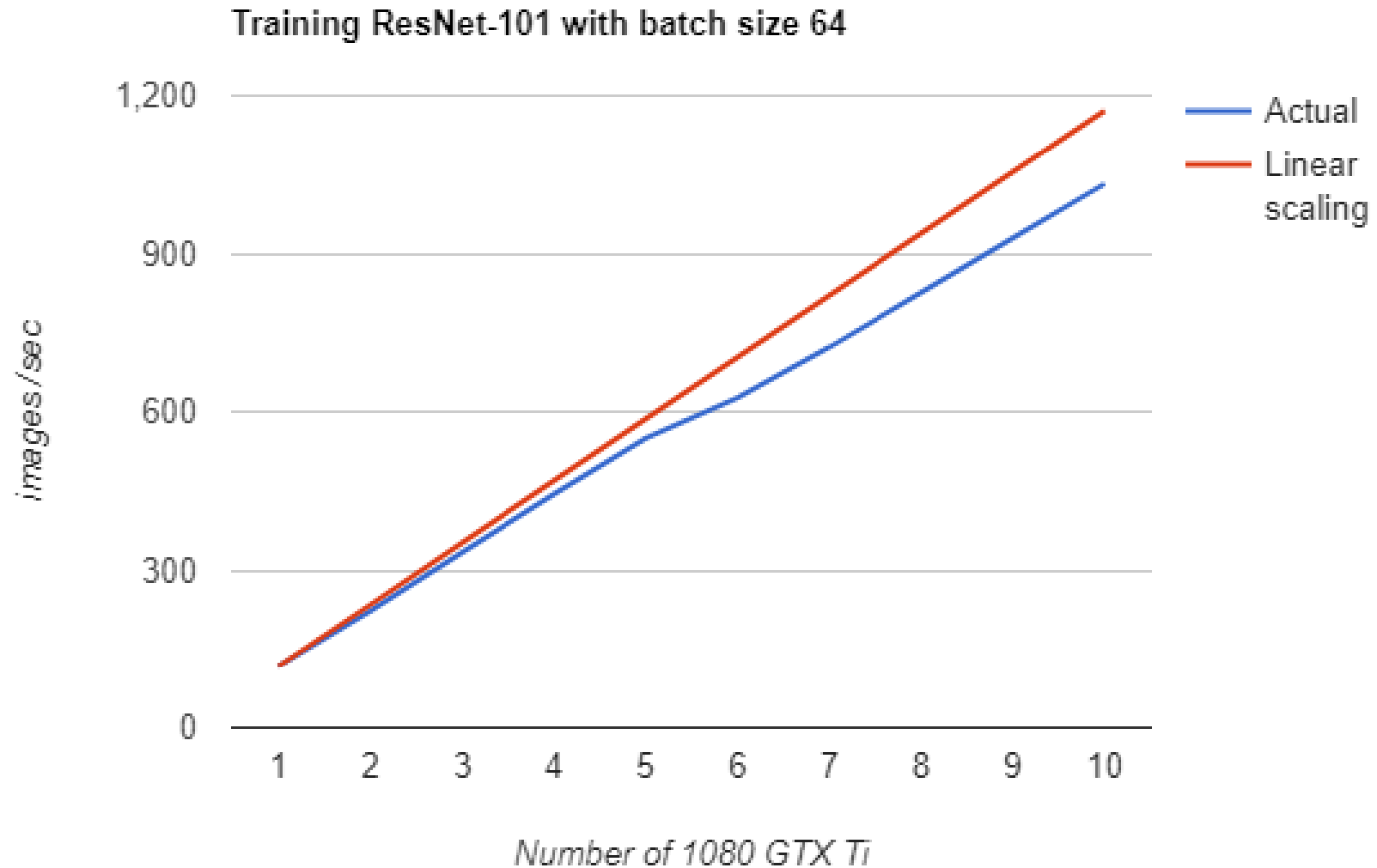
Hops is the only Hadoop distribution to support GPUs-as-a-Resource.

HopsUtil

- pip install hops
- Distributed TensorFlow in Python
 - Parameter sweeps for ML experiments
 - Managing tensorboard, Load/save models in HopsFS

```
from hops import tflauncher
args_dict = {'steps': [500, 2500, 5000, 7500, 5000]}
tflauncher.launch(spark, training, args_dict)
```

TensorFlow/Hops on 10 1080Ti GPUs



TensorBoard Integration

demo_tensorflow_a...

- Jupyter
- Zeppelin
- Jobs
- Data Sets
- Settings
- Members
- Metadata Designer
- Cluster Utilization: 25%

TensorBoard

SCALARS IMAGES GRAPHS PROJECTOR INACTIVE

Search: *.*

Tags matching *.* (all tags) 3

accuracy global_step/sec loss

Tag	Count
accuracy	1
global_step	1
loss	1

Left sidebar controls:

- Show data download links:
- Ignore outliers in chart scaling:
- Tooltip sorting method: default
- Smoothing: 0.6
- Horizontal Axis: STEP (selected), RELATIVE, WALL
- Runs filter: learning_rate=0.001.dropout=0.45, learning_rate=0.001.dropout=0.45/eval, learning_rate=0.001.dropout=0.45/eval_one_pass, learning_rate=0.001.dropout=0.7, learning_rate=0.001.dropout=0.7/eval
- TOGGLE ALL RUNS

Demo!



Hops

Summary

- Europe's Only Hadoop Distribution – Hops Hadoop
 - Fully Open-Source
 - Supports larger/faster Hadoop Clusters
- Hopsworks is a new Data Platform built on HopsFS with first-class support for Streaming and ML
 - Spark or Flink
 - TensorFlow
 - Support services for Streaming Analytics and ML-Deep Learning

The Team

Active:

Jim Dowling, Seif Haridi, Tor Björn Minde, Gautier Berthou, Salman Niazi, Mahmoud Ismail, Theofilos Kakantousis, Ermias Gebremeskel, Antonios Kouzoupis, Alex Ormenisan, Fabio Buso, Robin Andersson, August Bonds, Filotas Siskos, Mahmoud Hamed.



www.hops.io

 @hopshadoop

Alumni:

Vasileios Giannokostas, Johan Svedlund Nordström, Rizvi Hasan, Paul Mälzer, Bram Leenders, Juan Roca, Misganu Dessalegn, K "Sri" Srijevantham, Jude D'Souza, Alberto Lorente, Andre Moré, Ali Gholami, Davis Jaunzems, Stig Viaene, Hooman Peiro, Evangelos Savvidis, Steffen Grohsschmiedt, Qi Qi, Gayana Chandrasekara, Nikolaos Stanogias, Daniel Bali, Ioannis Kerkinos, Peter Buechler, Pushparaj Motamari, Hamid Afzali, Wasif Malik, Lalith Suresh, Mariano Valles, Ying Lieu, Fanti Machmount Al Samisti, Braulio Grana, Adam Alpire, Zahin Azher Rashid, Aruna Kumari Yedurupaka, Tobias Johansson, Roberto Bampi.



Karolinska
Institutet



SCANIA
yanzi
networks



Hops

Thank You.

Follow us:

@hopshadoop

Star us:

<http://github.com/hopshadoop/hopsworks>

Join us:

<http://www.hops.io>