Hopsworks – Self-Service Spark/Flink/Kafka/Hadoop

Jim Dowling
Associate Prof @ KTH
Senior Researcher @ SICS

Apache BigData Europe, 2016

Gwen (Chen) Shapira @gwenshap · 2 tim

The Cloudera dude at the meetup shows his cloud architecture - Kafka, Spark, Kudu, Kafka, S3, Impala... Hadoop has no Hadoop left in it.

8 20

Hadoop is not a cool kid anymore!

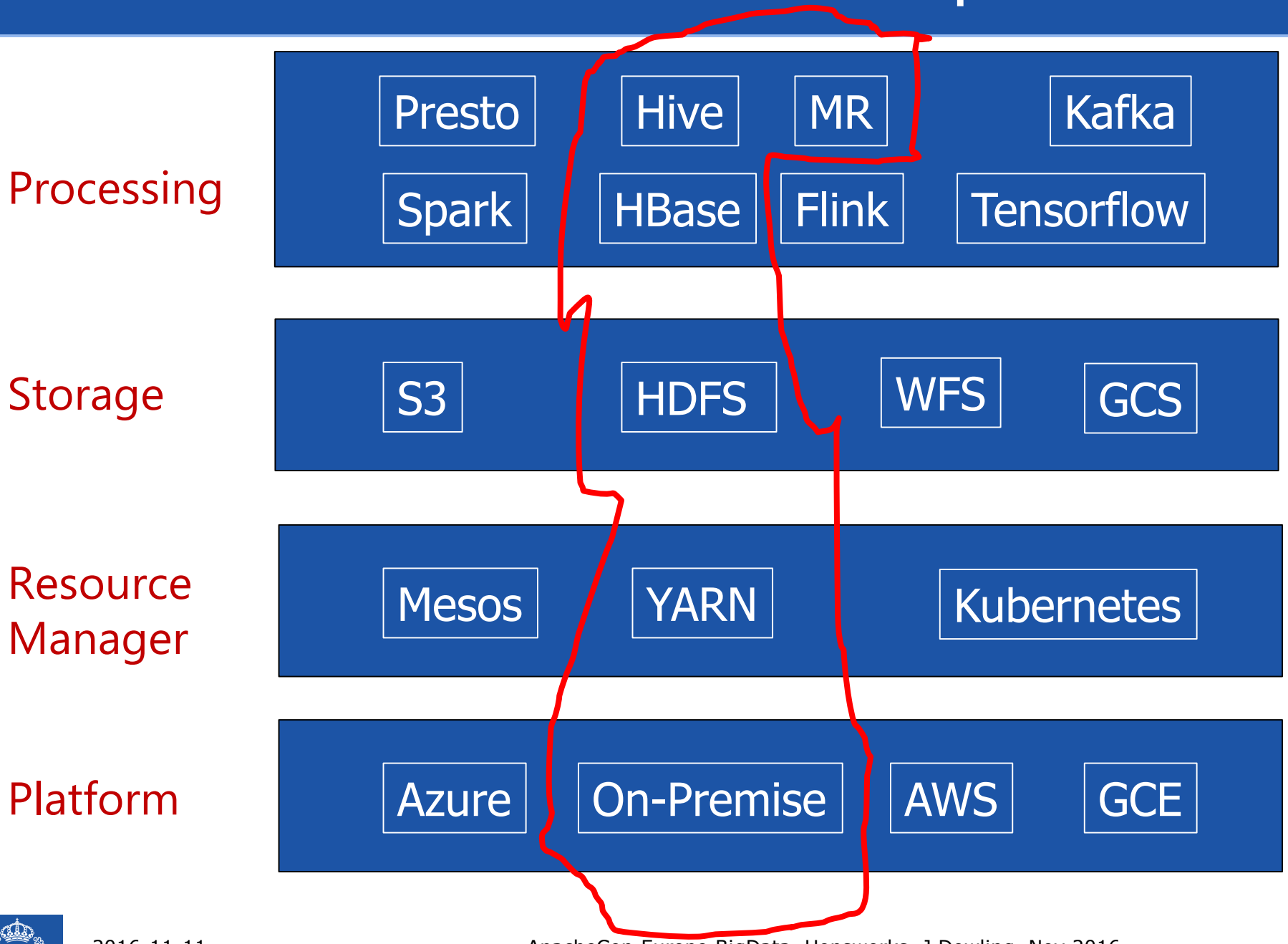
Where did it go wrong for Hadoop?

- Data Engineers/Scientists
  - Where is the User-Friendly tooling and Self-Service?
  - How do I install/operate anything other than a sandbox VM?

- Operations Folks
  - Security model has become incomprehensible (Sentry/Ranger)
  - Major distributions not open enough for patching
  - Sensitive data still requires its own cluster
  - Why not just use AWS EMR/GCE/Databricks/etc ?!?

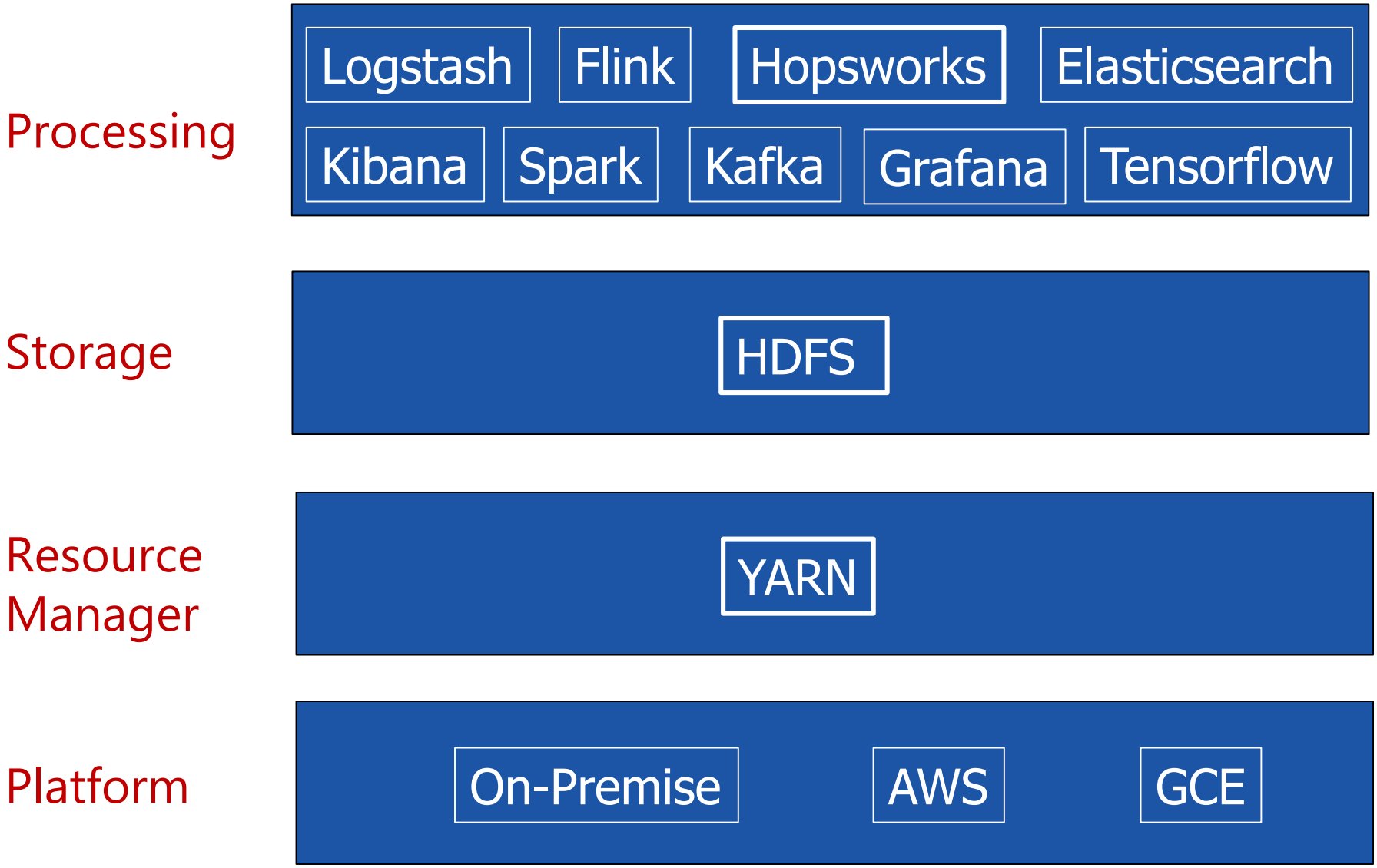

MAKE HADOOP GREAT AGAIN!
NOBODY HAS BIGGER DATA THAN ME. MY DATA IS THIS BIG!

Is this Hadoop?



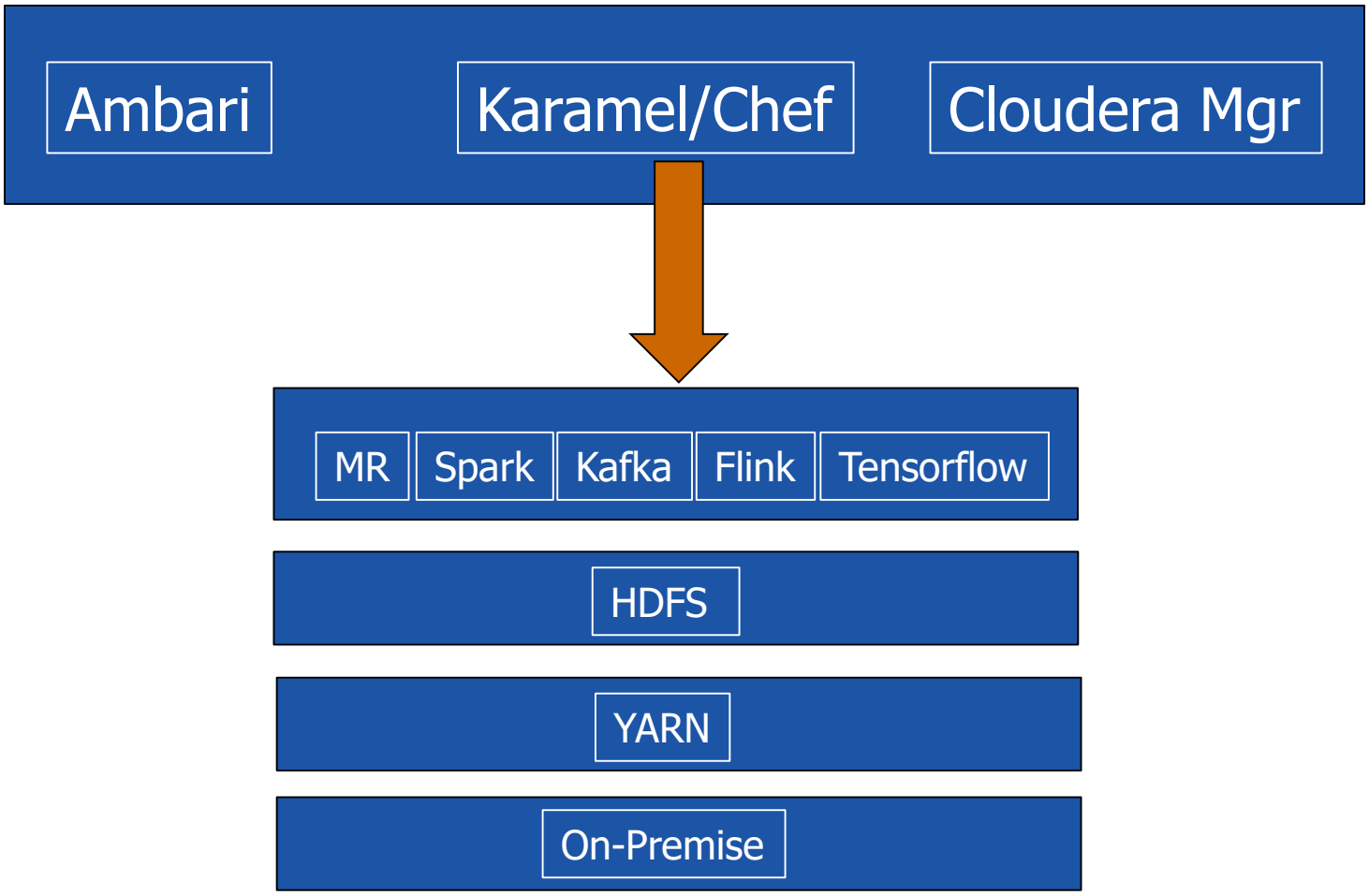
How about this?

Here's Hops Hadoop Distribution



Hadoop Distributions Simplify Things

Install / Upgrade

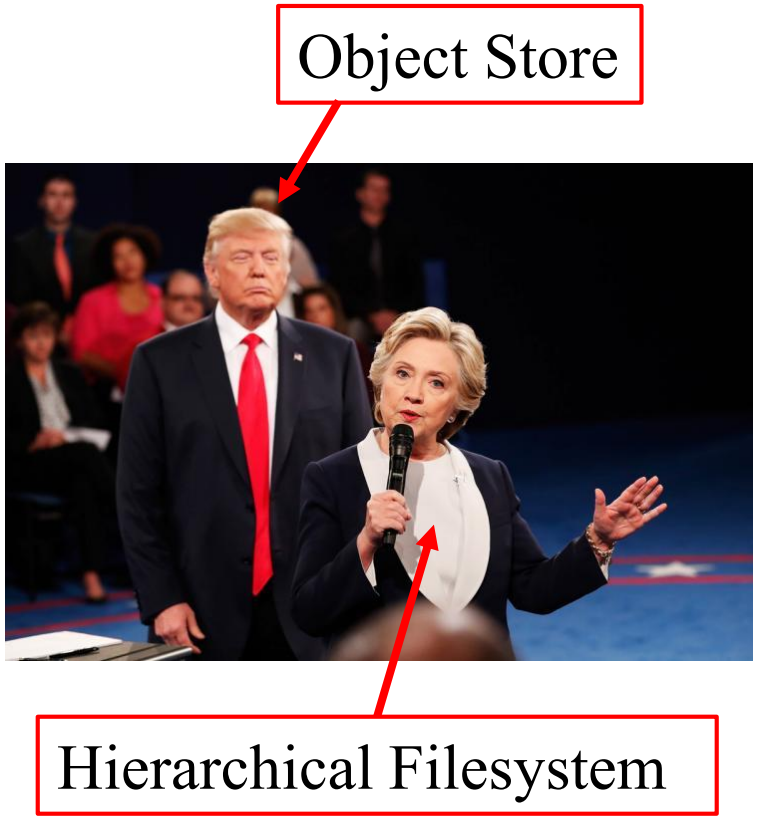


Cloud-Native means Object Stores, not HDFS

Object Stores and False Equivalence*

- Object Stores are inferior to hierarchical filesystems
  - False equivalence in the tech media

- Object stores can scale better, but at what cost?
  - Read-your-writes existing objs#
  - Write, then list
  - Hierarchical namespace properties
    - Quotas, permissions
  - Other eventual consistency probs

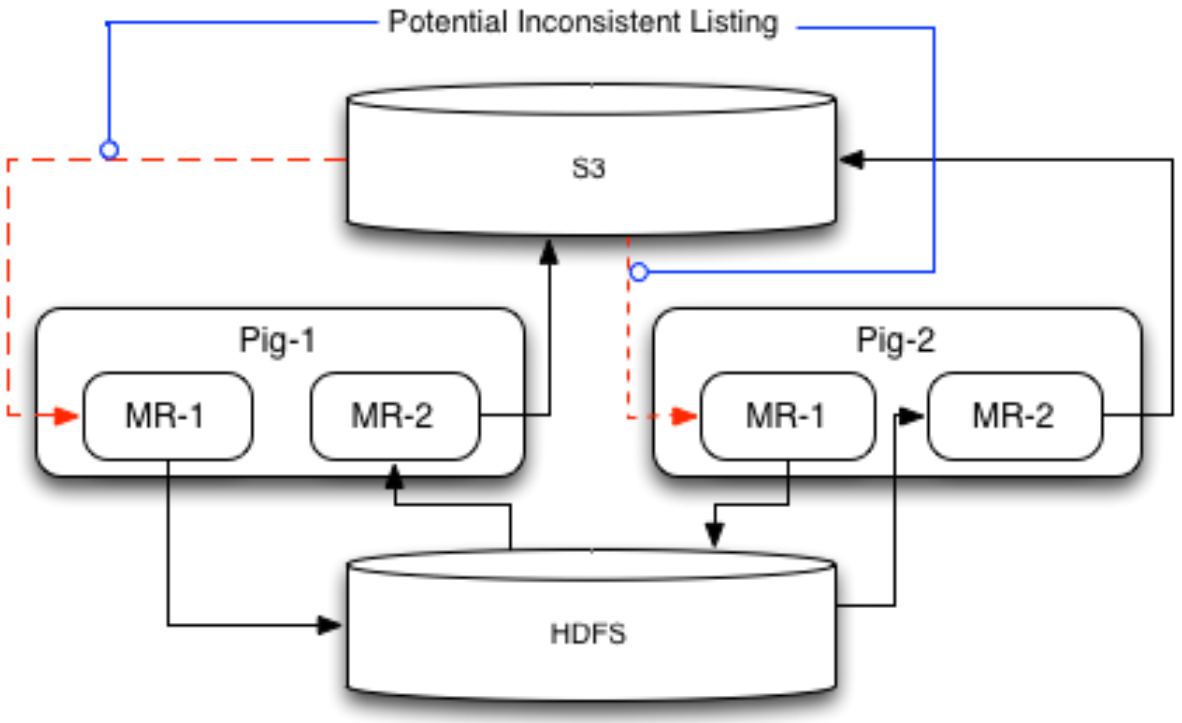


*False Equivalence**: "unfairly equating a minor failing of Hillary Clinton's to a major failing of Donald Trump's."

Eventual consistency in Object Stores

Implement your own eventually consistent metadata store to mirror the (unobservable) metadata

- Netflix for AWS*
  - s3mpr
- Spotify for GCS
  - Tried porting s3mper, now own solution.



http://techblog.netflix.com/2014/01/s3mper-consistency-in-cloud.html

Can we open up the ObjectStore's metadata?

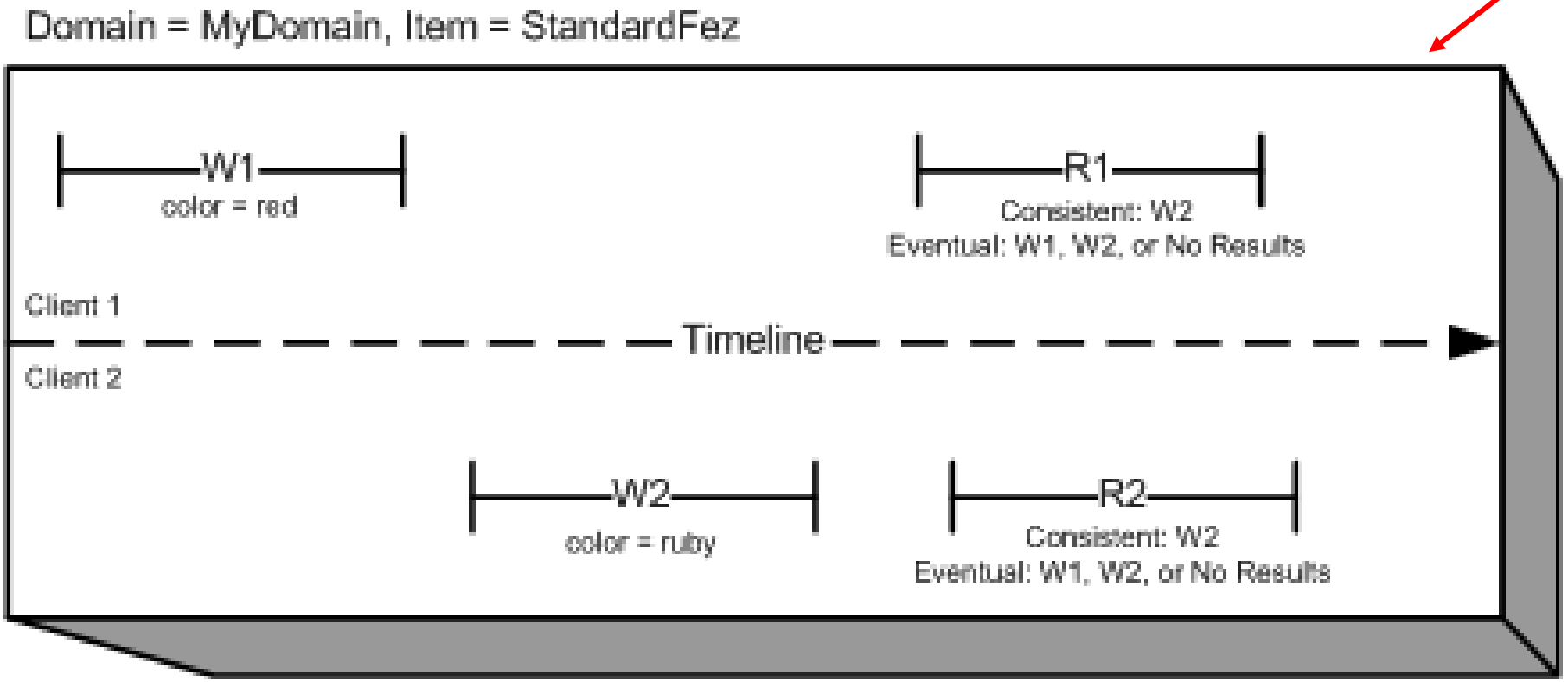
Object Store metadata is a Pandora's Box. Best keep it closed.

Eventual Consistency is Hard

```
r1 = read();
```

```
if (r1 == null) {
// Loop until read completes
} else if (r1 == W1){
// do Y
} else if (r1 == W2) {
// do Z
}
```

S3 guarantees



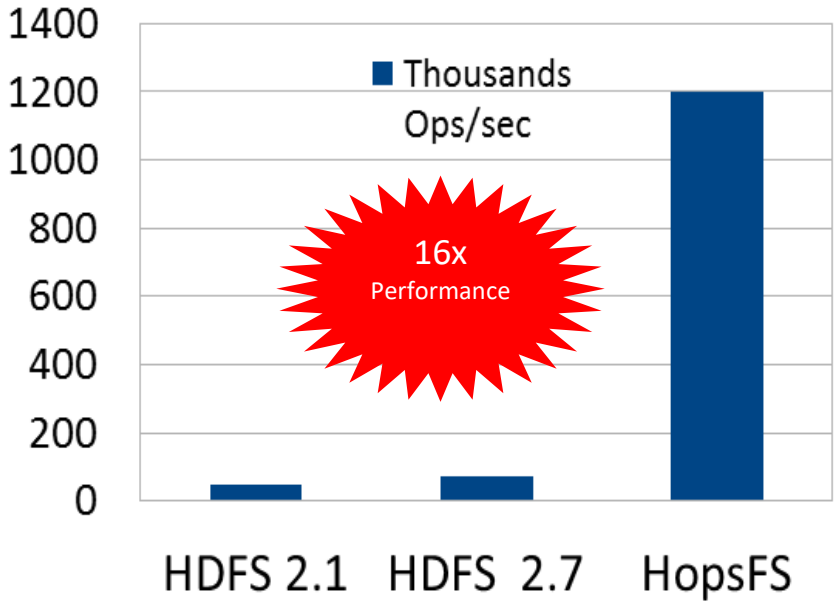
Prediction

NoSQL systems grew from the need to scale-out relational databases.
NewSQL systems bring both scalability and strong consistency,
and companies moving back to strong consistency.*

Object stores systems grew from the need to scale-out filesystems.
A new generation of hierarchical filesystem will appear that bring both scalability and hierarchy, and companies will *eventually* move back to scalable hierarchical filesystems.

*http://www.cubrid.org/blog/dev-platform/spanner-globally-distributed-database-by-google/

HopsFS and Hops YARN



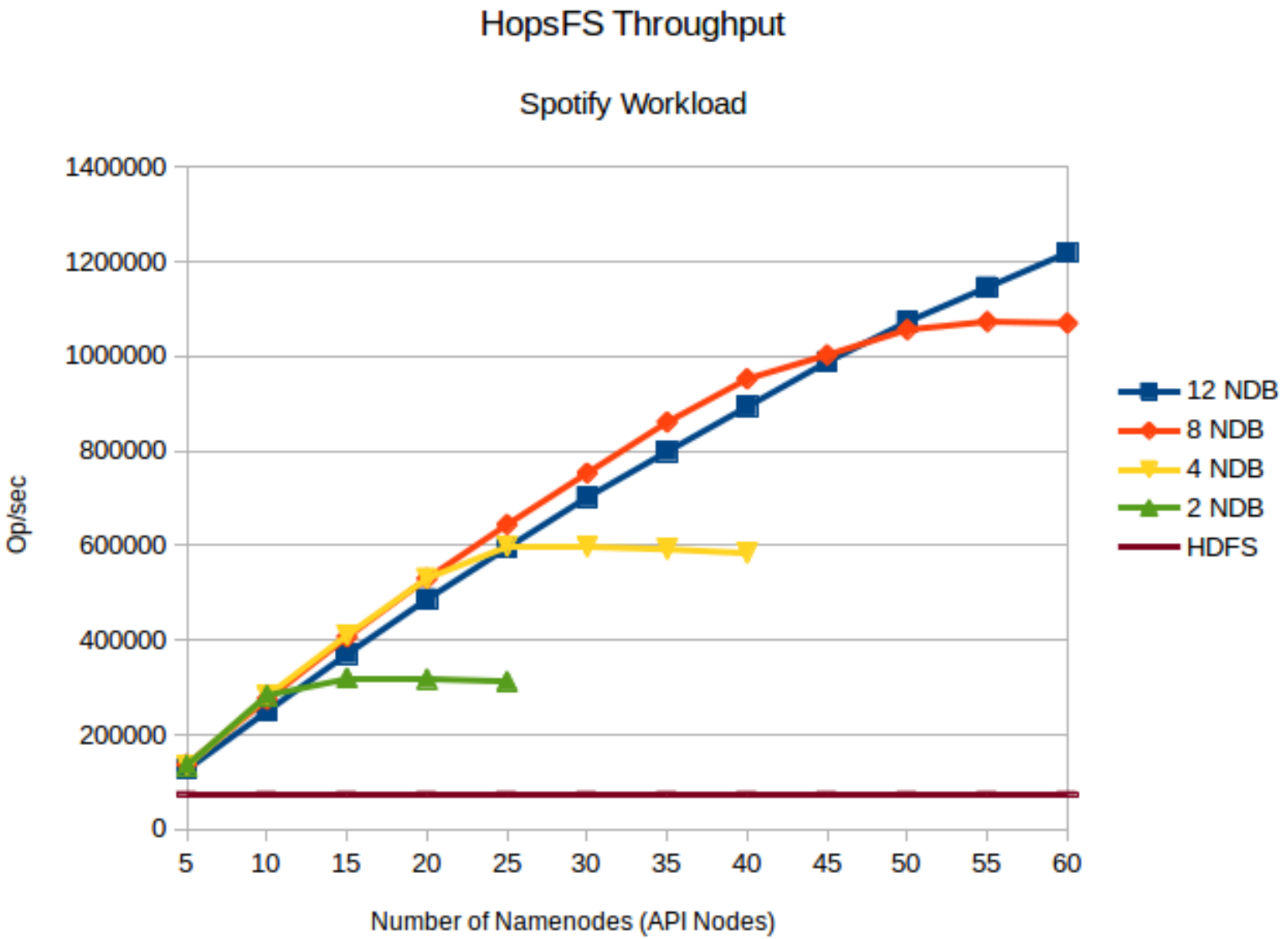
Externalizing the NameNode/YARN State

- Problem: Metadata is preventing Hadoop from becoming the great platform we know she can be!

- Solution: Move the metadata off the JVM Heap

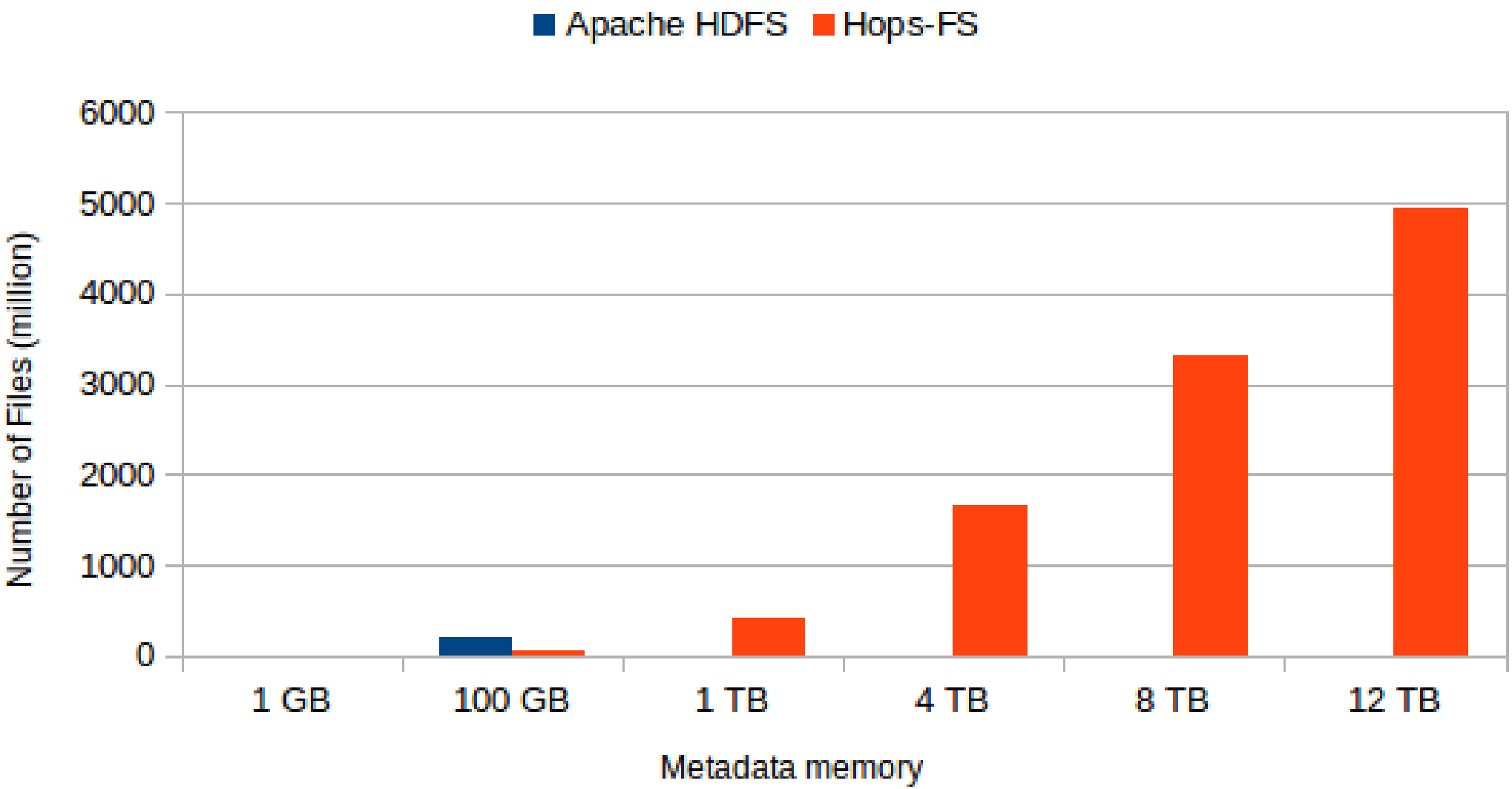
- Where?
  To an in-memory database that can be transactionally and efficiently queried and managed. The database should be Open-Source. We chose NDB – aka MySQL Cluster.

HopsFS – 1.2 million ops/sec (16X HDFS)



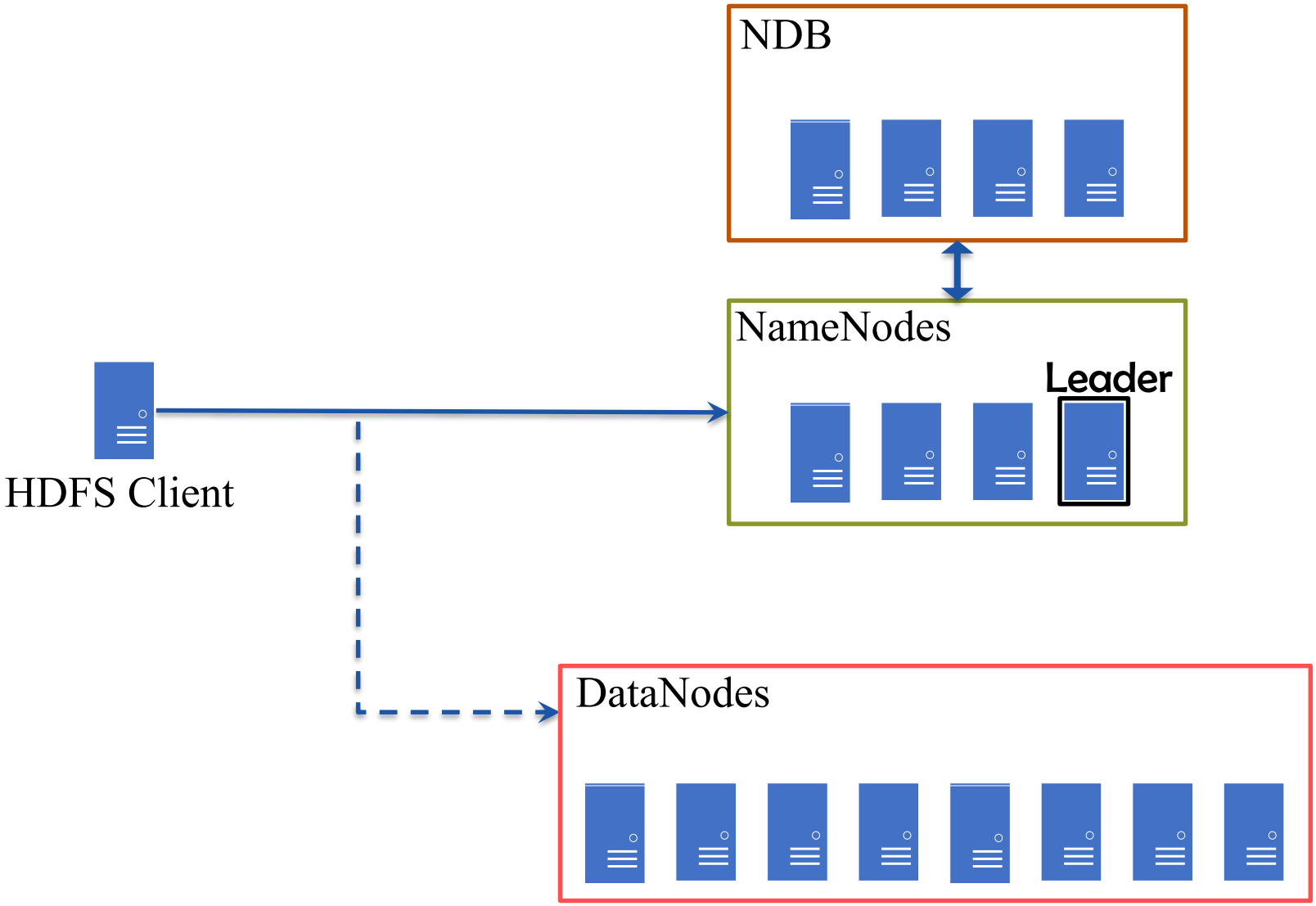
NDB Setup: Nodes using Xeon E5-2620 2.40GHz Processors and 10GbE.
NameNodes: Xeon E5-2620 2.40GHz Processors machines and 10GbE.

HopsFS Metadata Scaleout

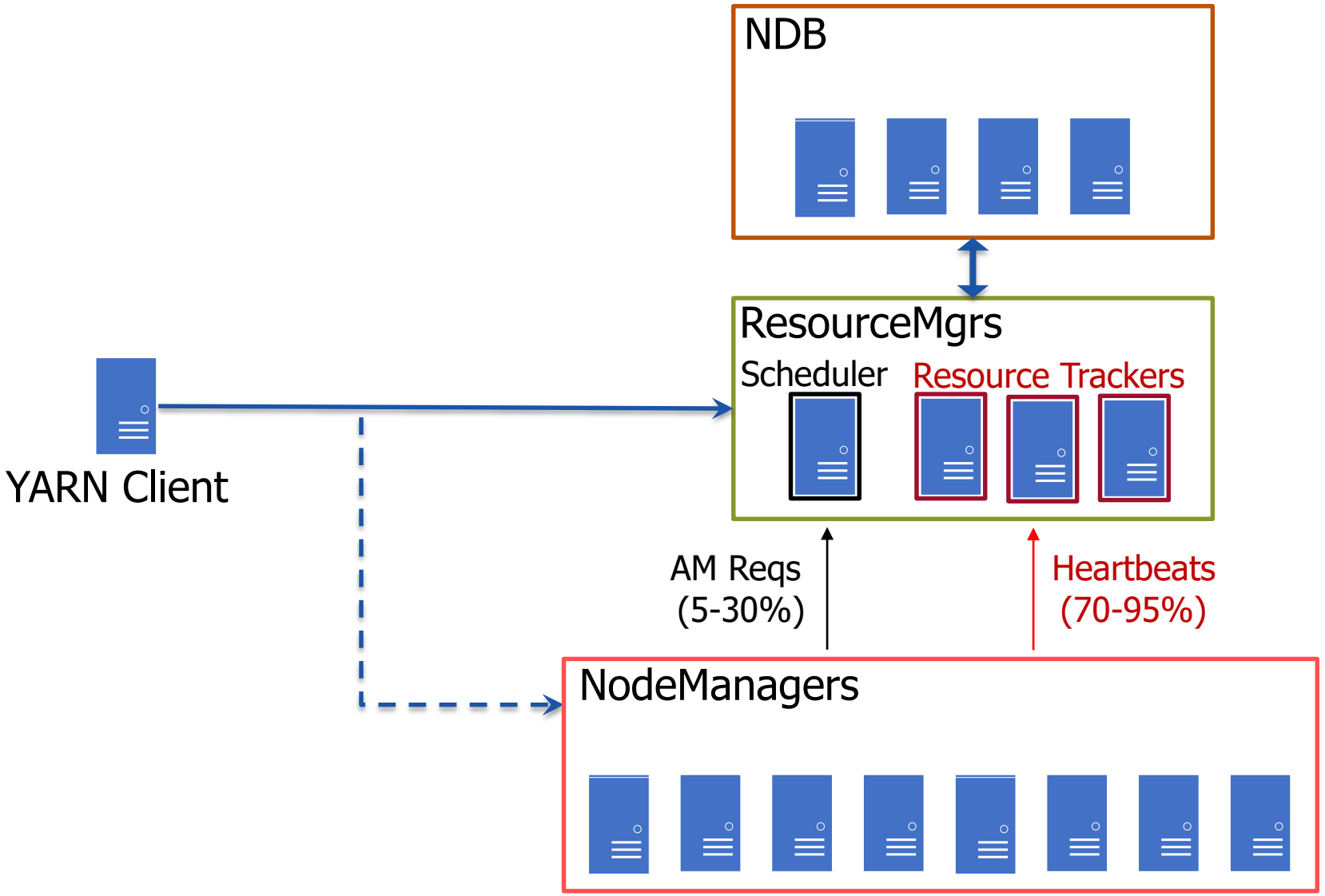


Assuming 256MB Block Size, 100 GB JVM Heap for Apache Hadoop

HopsFS Architecture



Hops YARN Architecture



Extending Metadata in Hops

- JSON API (with and without schema)

  `public void attachMetadata(Json obj, String pathToFileorDir)`

  - Row added with jsonObj & foreign key to the inode (on delete cascade)
  - Use to annotate files/directories for search in Elasticsearch

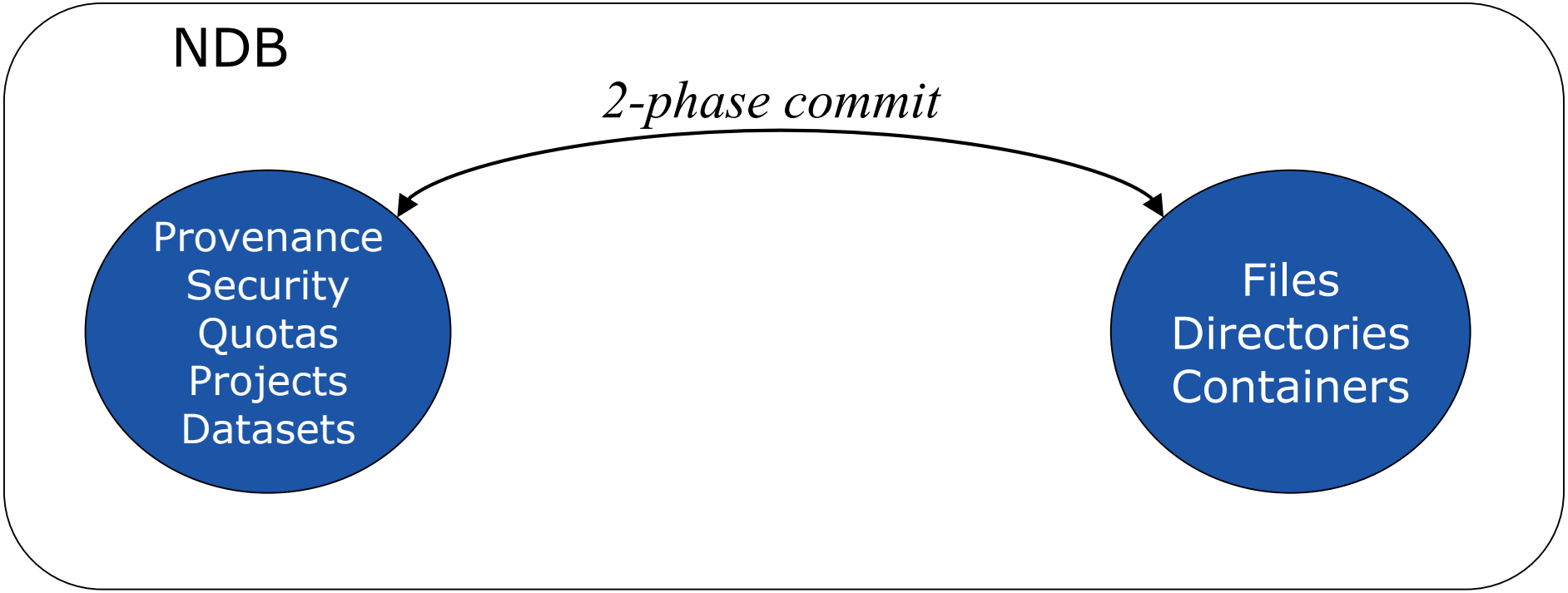
id	json_obj	inode (fk)
1	{ trump: non}	10101

inode_id	attrs
10101	/tmp…

- Add Tables in the database
  - Foreign keys to inode/applications ensure metadata integrity
  - Transactions involving both the inode(s) and metadata ensure metadata consinstency
  - Enables modular extensions to Namenodes and ResourceManagers

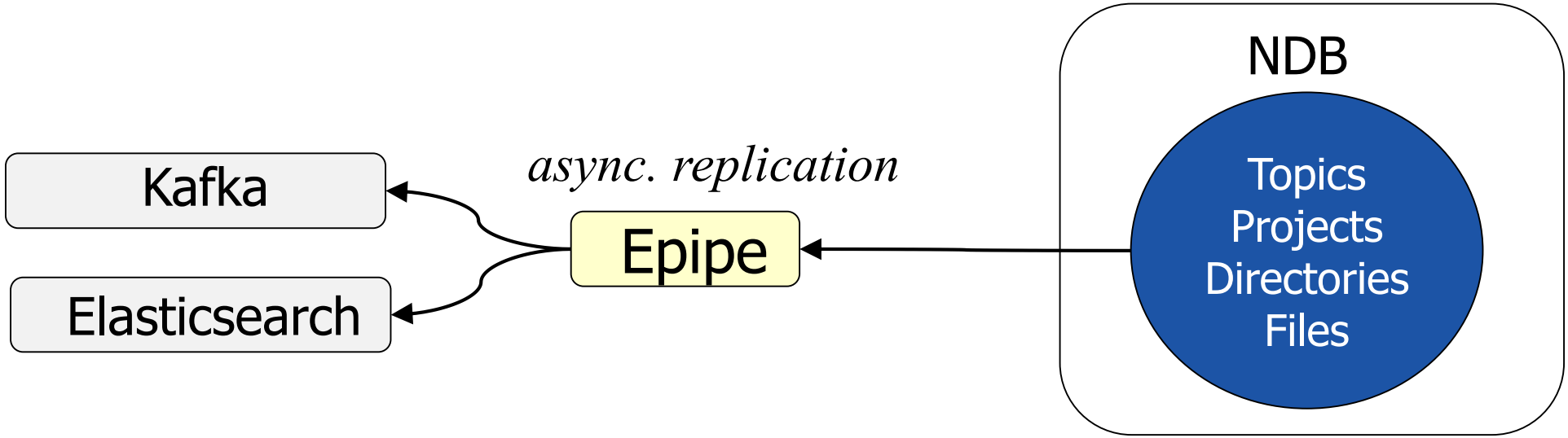
Strongly Consistent Metadata

Metadata Integrity maintained using
2PC and Foreign Keys.



Eventually Consistent Metadata with Epipe

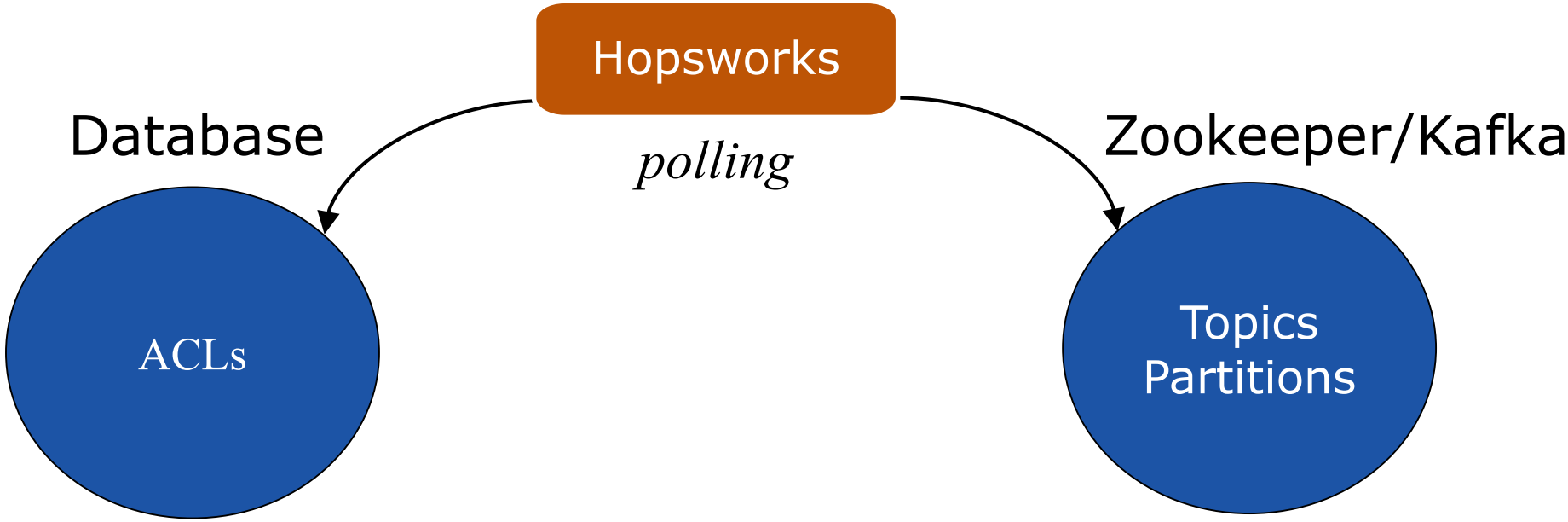
Metadata Integrity maintained by
***one-way* Asynchronous Replication**.



[ePipe Tutorial, BOSS Workshop, VLDB 2016]

Maintaining Eventually Consistent Metadata

Metadata integrity maintained by
custom recovery logic and polling.



Tinker-Friendly Metadata

- Erasure Coding in HopsFS
- Free-Text search of HopsFS namespace
  - Export changelog for NDB to Elasticsearch
- Dynamic User Roles in HopsFS
- **New abstractions**: **Projects** and **Datasets**

Extensible, tinker-friendly metadata is like 3-d printing for Hadoop

Leveraging Extensible Metadata in Hops

New Concepts in Hops Hadoop

Hops

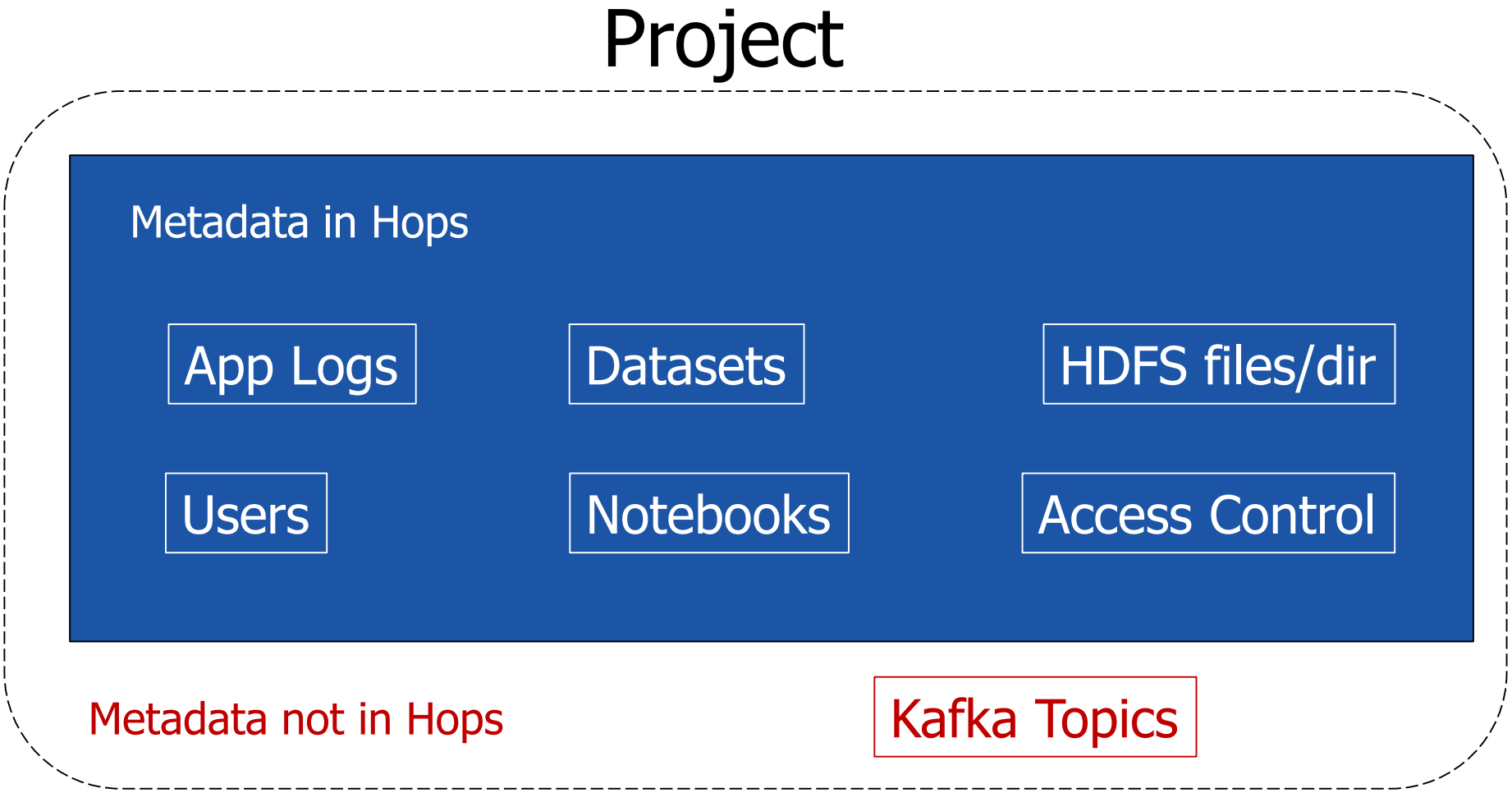
- Projects
  - Datasets
  - Topics
  - Users
  - Jobs
  - Notebooks

Enabled by Extensible Metadata

Hadoop

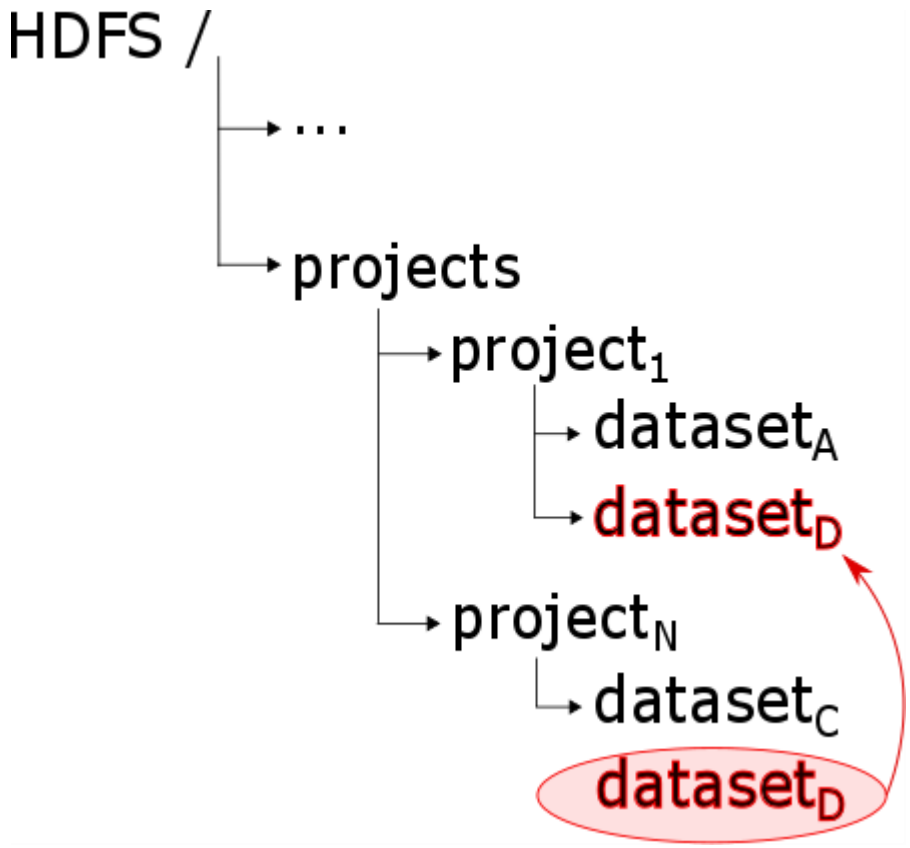
- Users
- Applications
- Jobs
- Files
- Clusters
- ACLs
- Sys Admins
- Kerberos

Simplifying Hadoop with Projects



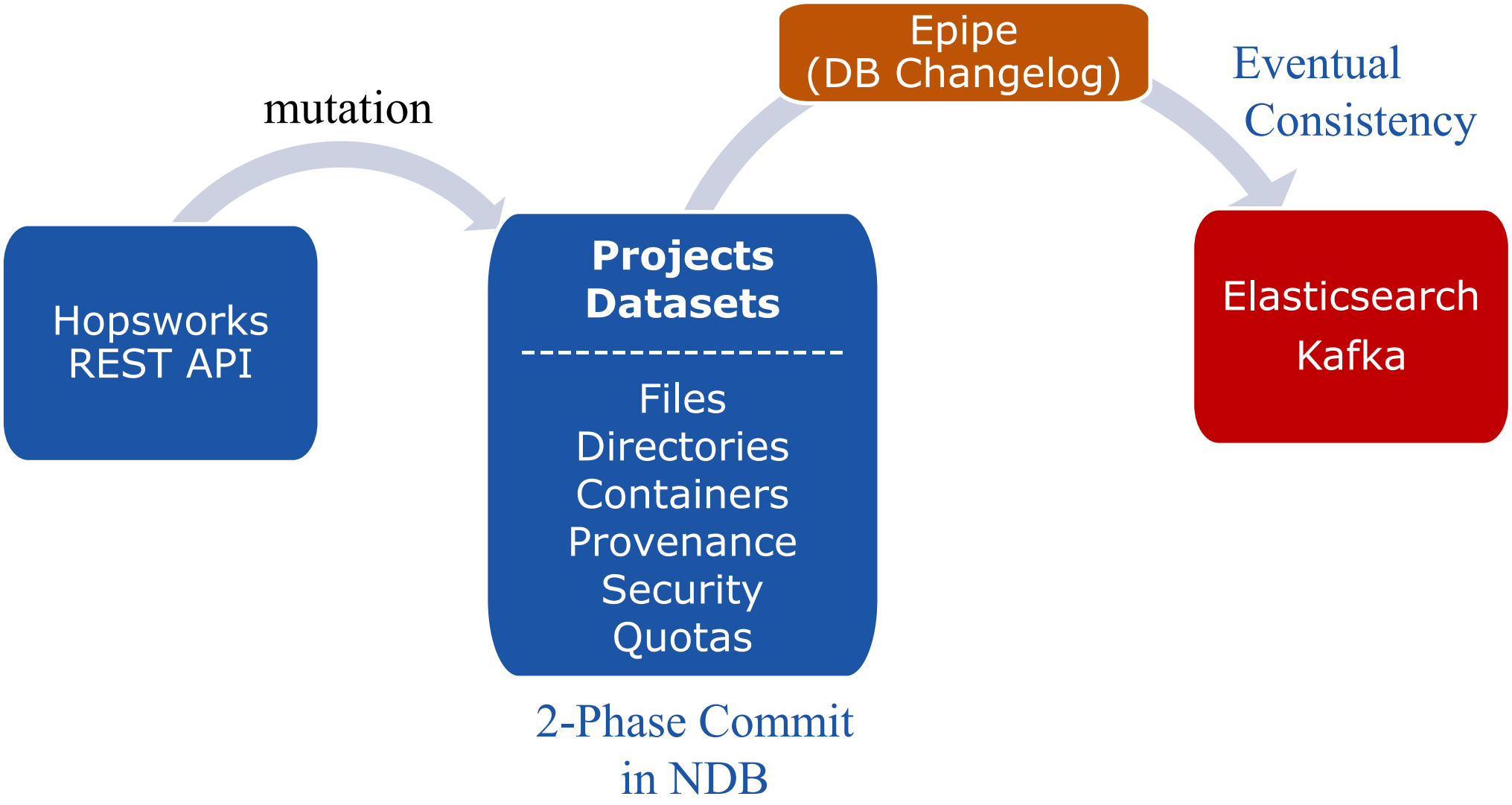
Simplifying Hadoop with Datasets

- Datasets are a directory subtree in HopsFS
  - Can be shared securely between projects
  - Indexed by Elasticsearch
- Datasets can be made public and downloaded from any Hopsworks cluster anywhere in the world



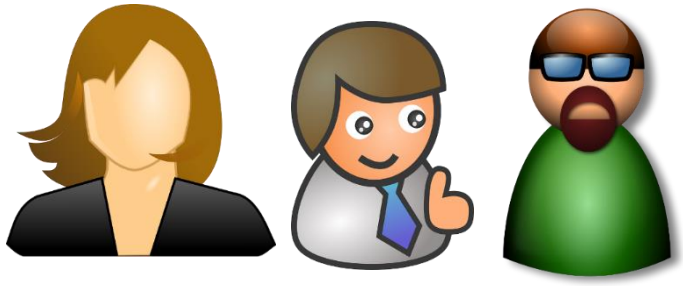
Hops Metadata Summarized

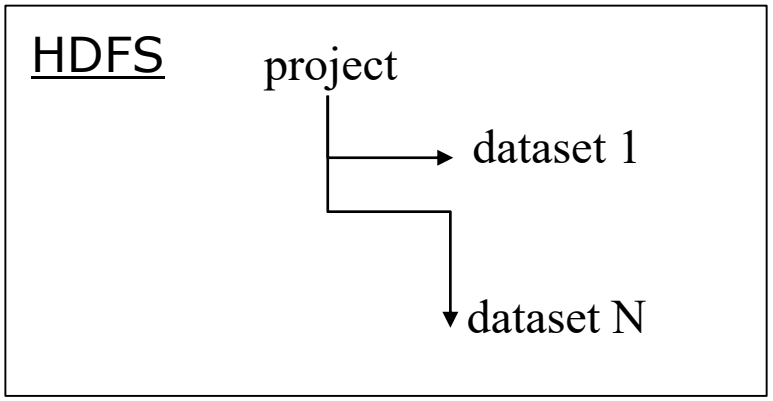
The Distributed Database is the Single Source-of-Truth for Metadata

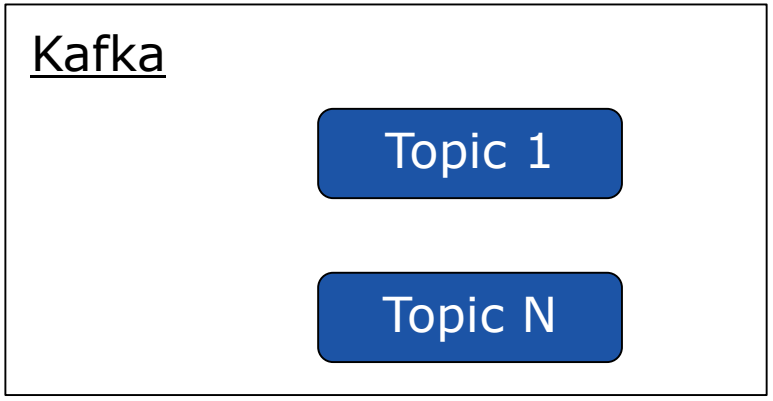


Hopsworks

Project-Based Multi-Tenancy

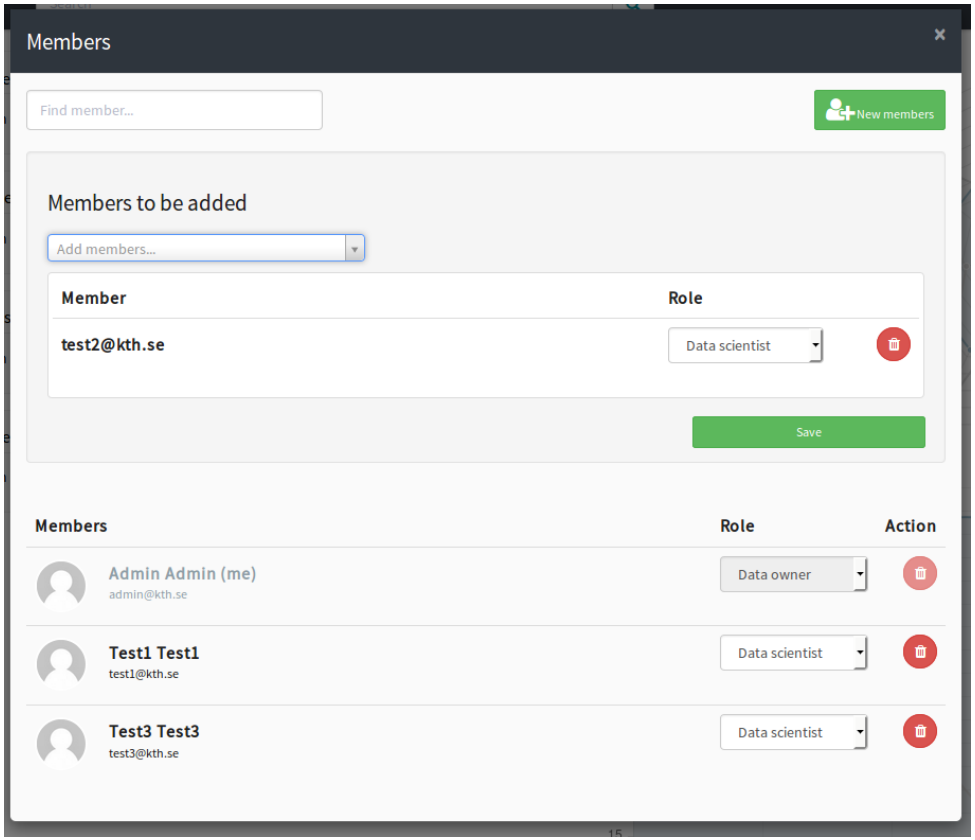






- A project is a collection of
  - Users with Roles
  - HDFS DataSets
  - Kafka Topics
  - Notebooks, Jobs
- Per-Project quotas
  - Storage in HDFS
  - CPU in YARN
    - Uber-style Pricing
- Sharing across Projects
  - Datasets/Topics

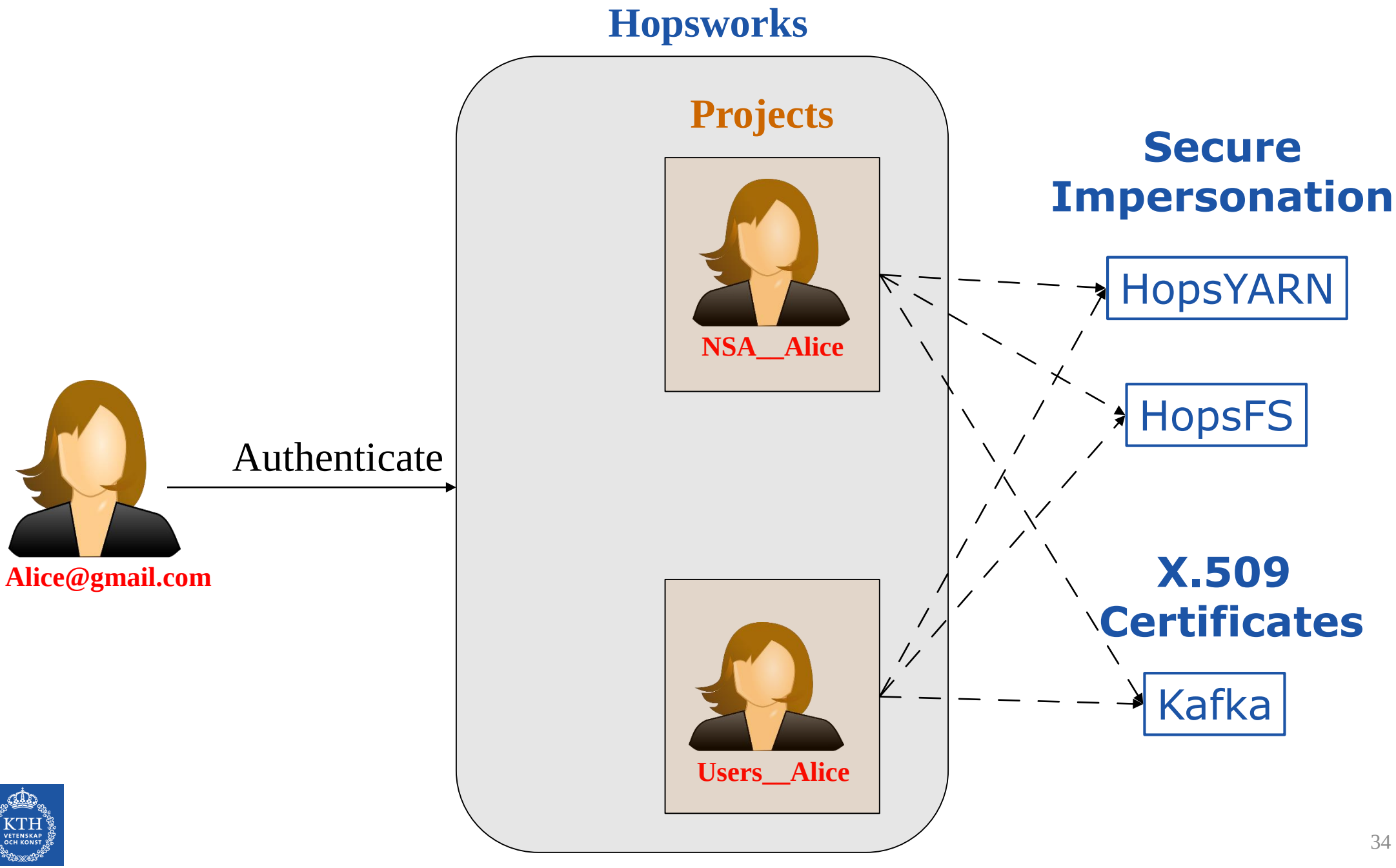
Project Roles



- 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 – Dynamic Roles



X.509 Certificates Everywhere

- User and service certs share same self-signed root CA

- Project-Specific User Certificates
  - Every user in every project is issued with a X.509 certificate, containing the project-specific userID.
    - Scales using intermediate CAs at each Hopsworks instance.
    - Inspired by Netflix' BLESS system.
- Service Certificates
  - Services use a host-specific certificate that is signed by the root CA. Process managed by an agent program (kagent).
  - Services identify SSL clients by extracting the CommonName from client certificate in RPC calls. Kerberos keytab gone.

X.509 Certificate Generation

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

- LDAP,
- password,
- 2-Factor Authentication

Alice@gmail.com

Hopsworks

Add/Del Users

Project Mgr

Insert/Remove Certs

Intermediate Certificate Authority

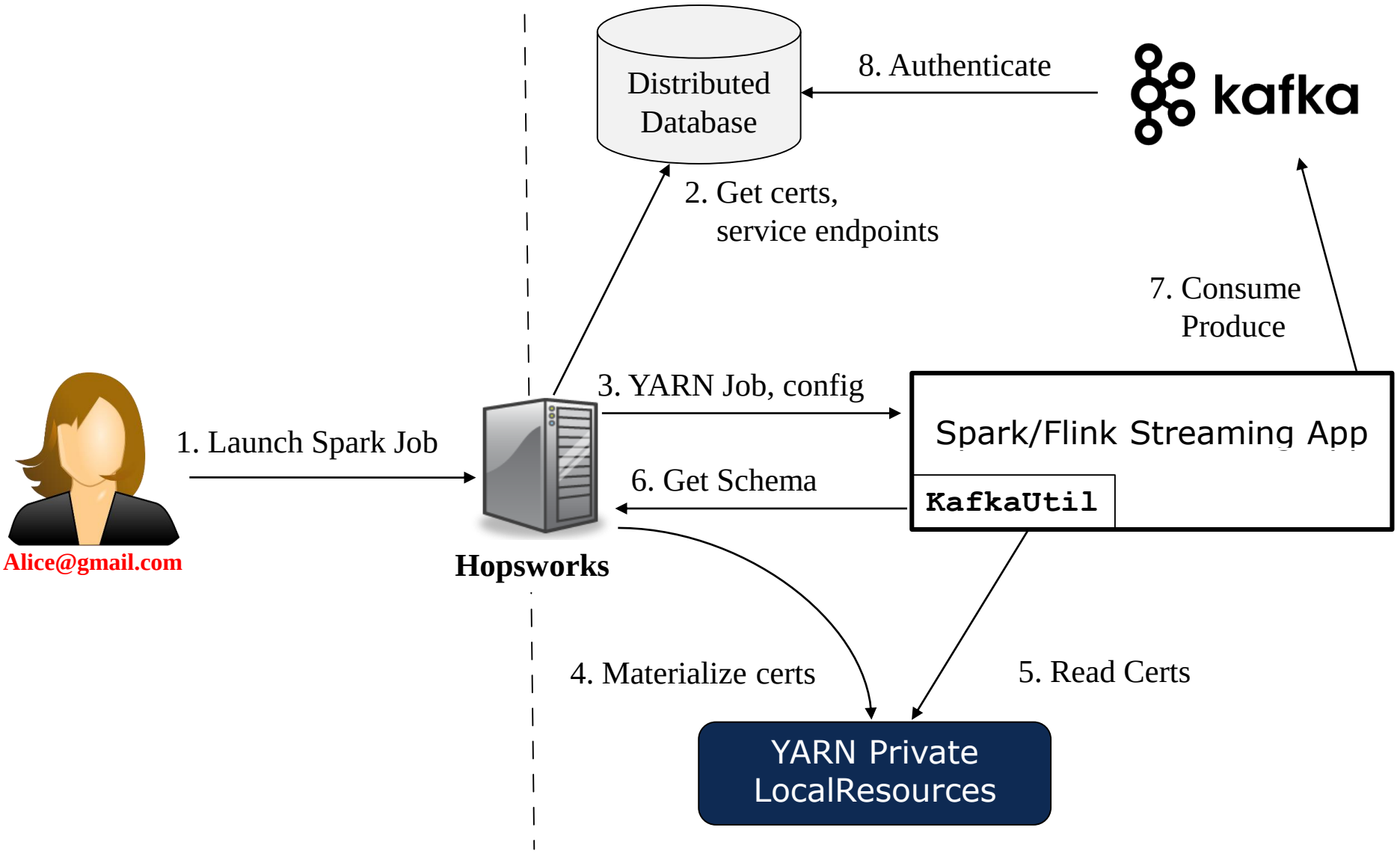
Distributed Database

Cert Signing Requests

Root CA

Kagent
HDFS
Spark
Kafka
YARN

Simplifying Flink/Spark Streaming



Secure Kafka Producer Application

1. **Discover:** Schema Registry and Kafka Broker Endpoints
2. **Create:** Kafka Properties file with certs and broker details
3. **Create:** producer 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

All of these steps are now down automatically by Hopsworks' KafkaUtil library

Hops simplifies Secure Flink/Kafka Producer

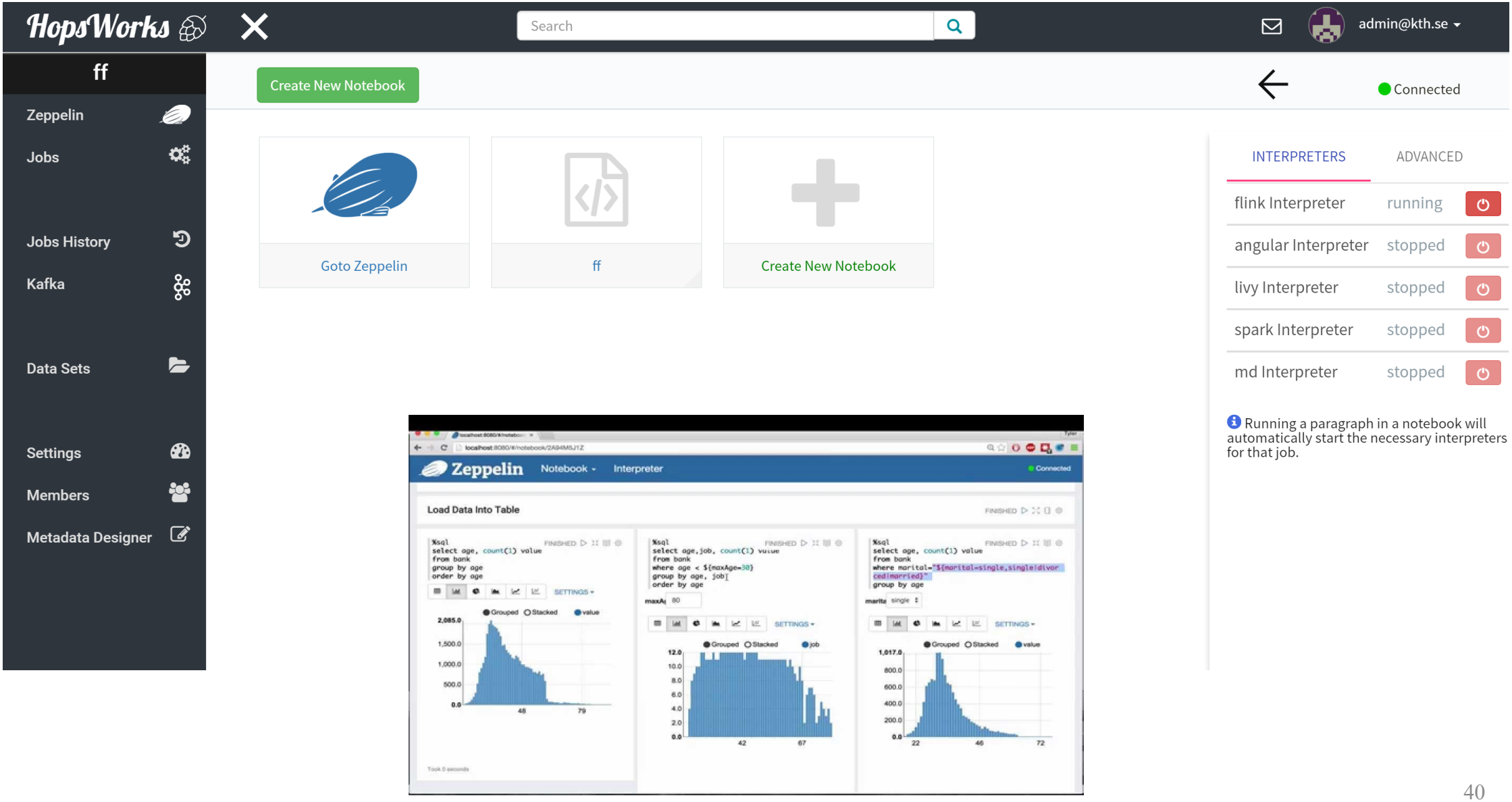
```
Properties props = new Properties();
props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, brokerList);
props.put(SCHEMA_REGISTRY_URL, restApp.restConnect);
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
org.apache.kafka.common.serialization.StringSerializer.class);
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
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.keysto
re.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);
```

Losts of Hard-Coded Endpoints Here!

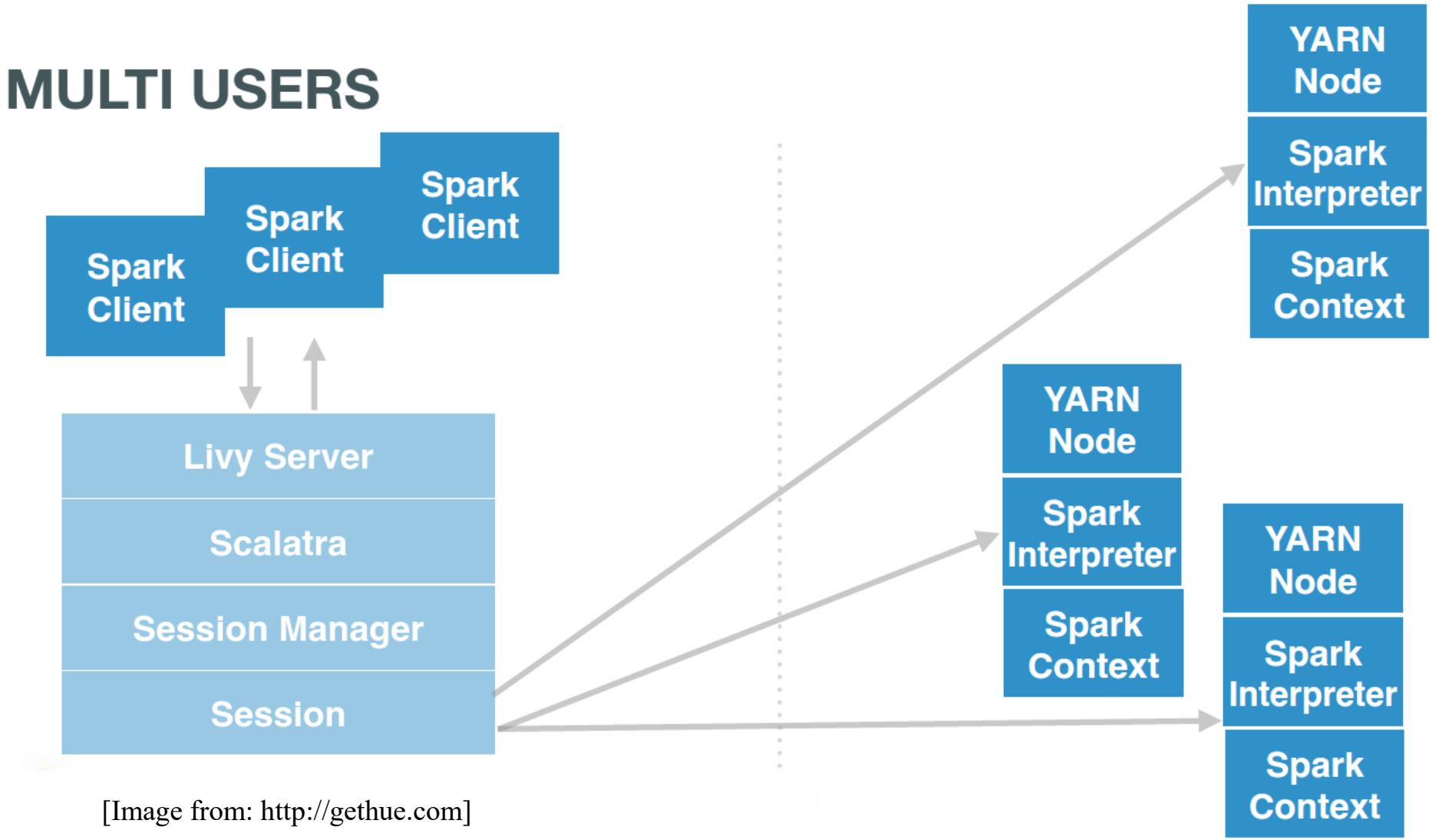
```
StreamExecutionEnvironment
env = …

FlinkProducer prod =
KafkaUtil.getFlinkProducer
(topicName);

DataStream<…> ms =
env.addSource(…);

ms.addSink(producer);
env.execute("Producing");
```

Massively Simplified Code for Hops/Flink/Kafka

Zeppelin Support for Spark/Livy



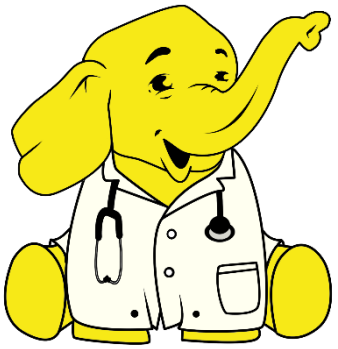
Spark Jobs in YARN with Livy

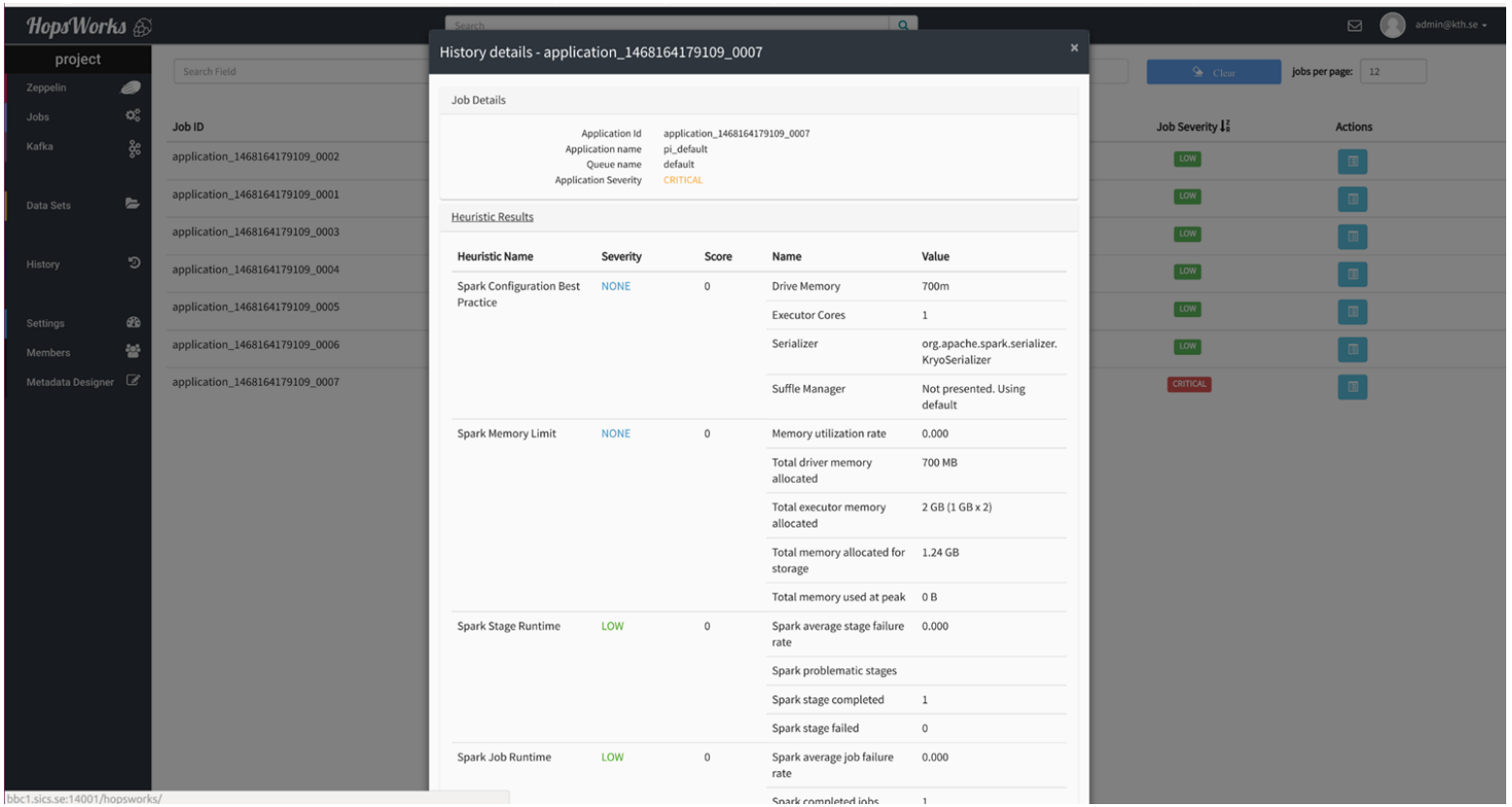


[Image from: http://gethue.com]

Debugging Spark Jobs with Dr Elephant

- Project-specific view of performance and correctness issues for completed Spark Jobs
- Customizable heuristics
- Doesn't show killed jobs

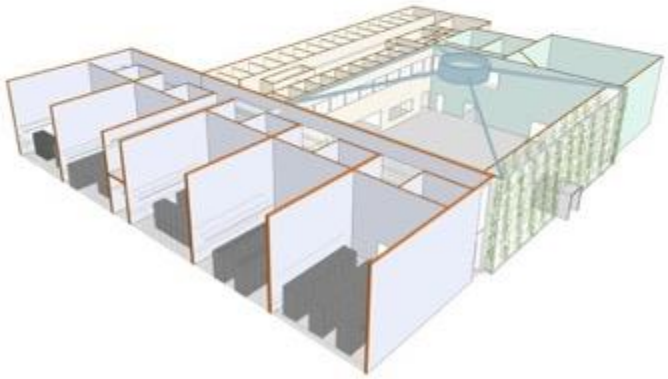




Hops in Production at www.hops.site

SICS ICE: A datacenter research and test environment

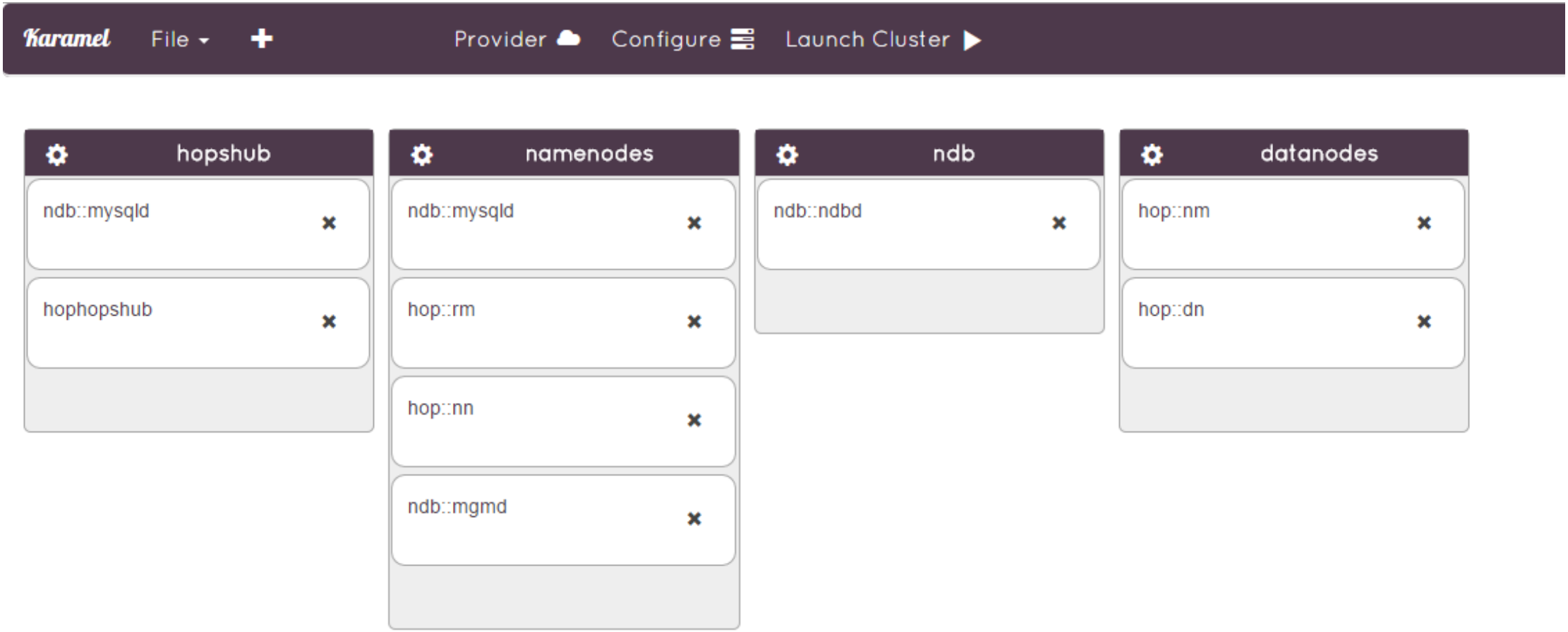
Purpose: Increase knowledge, strengthen universities, companies and researchers



SWEDISH ICT SICS

LULEÅ TEKNISKA UNIVERSITET

Karamel/Chef for Automated Installation



Demo

Summary

- Hops is the only European distribution of Hadoop
  - More scalable, tinker-friendly, and open-source.
- Hopsworks provides first-class support for Flink-/Spark-Kafka-as-a-Service
  - Streaming or Batch Jobs
  - Zeppelin Notebooks
- Hopworks provides best-in-class support for secure streaming applications with Kafka

Hops Team

Active: Jim Dowling, Seif Haridi, Tor Björn Minde, Gautier Berthou, Salman Niazi, Mahmoud Ismail, Theofilos Kakantousis, Antonios Kouzoupis, Ermias Gebremeskel.

Alumni: Vasileios Giannokostas, Johan Svedlund Nordström, Rizvi Hasan, Paul Mälzer, Bram Leenders, Juan Roca, Misganu Dessalegn, K "Sri" Srijeyanthan, 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, Ioannis Kerkinos, Peter Buechler, Pushparaj Motamari, Hamid Afzali, Wasif Malik, Lalith Suresh, Mariano Valles, Ying Lieu.

eit Digital

SEVENTH FRAMEWORK PROGRAMME

Stiftelsen för Strategisk Forskning

KTH Vetenskap och Konst

STREAMLINE.

SWEDISH ICT SICS

Hops

[Hadoop For Humans]

http://github.com/hopshadoop

http://www.hops.io