Hops

@hopshadoop

http://github.com/hopshadoop

http://www.hops.io

SPARK SUMMIT 2017

Structured Spark Streaming-as-a-Service with Hopsworks

Jim Dowling

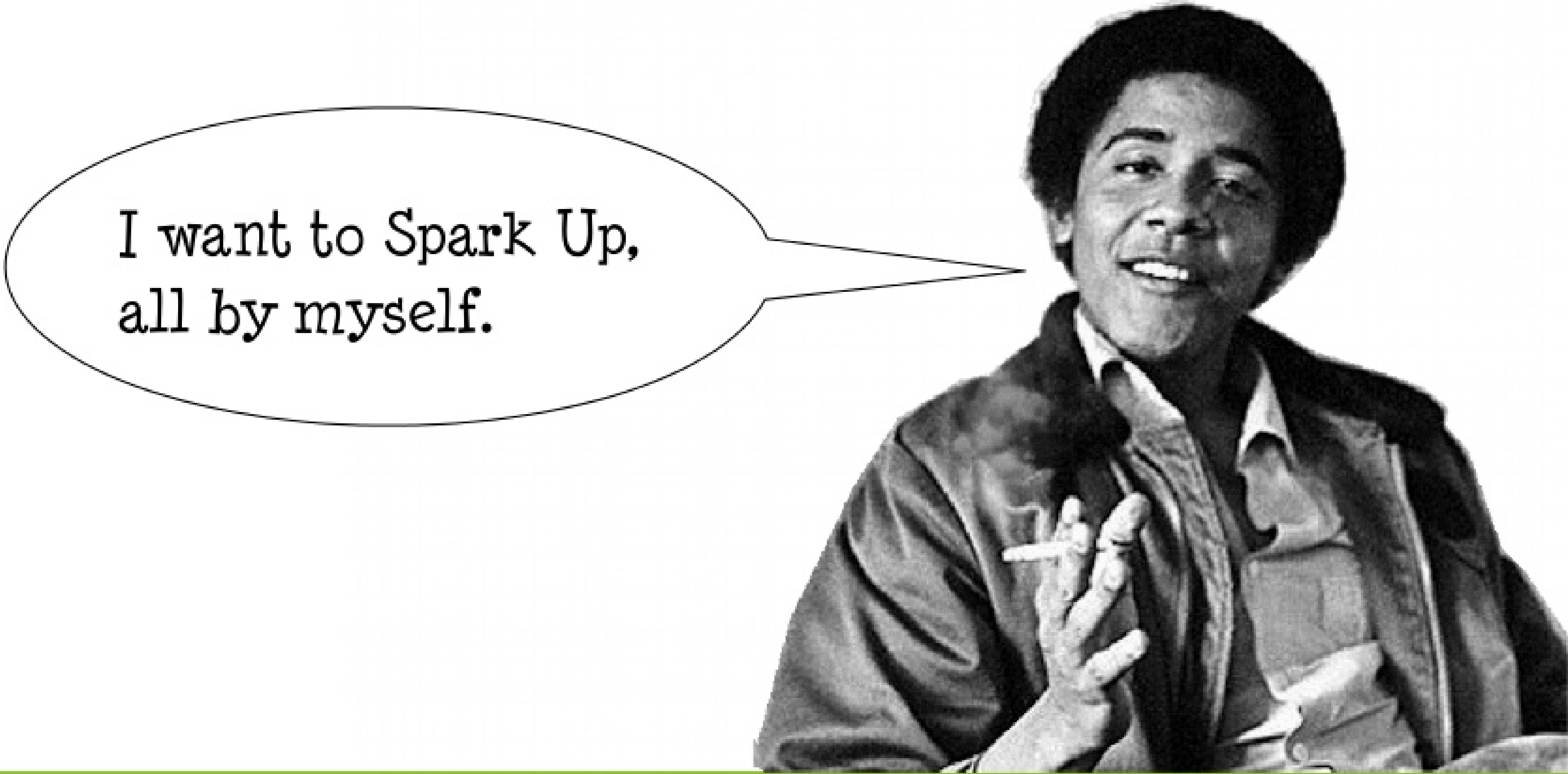
Assoc. Prof, Royal Institute of Technology, Stockholm

Spark Streaming-as-a-Service in Sweden

Swedish National Day Today!

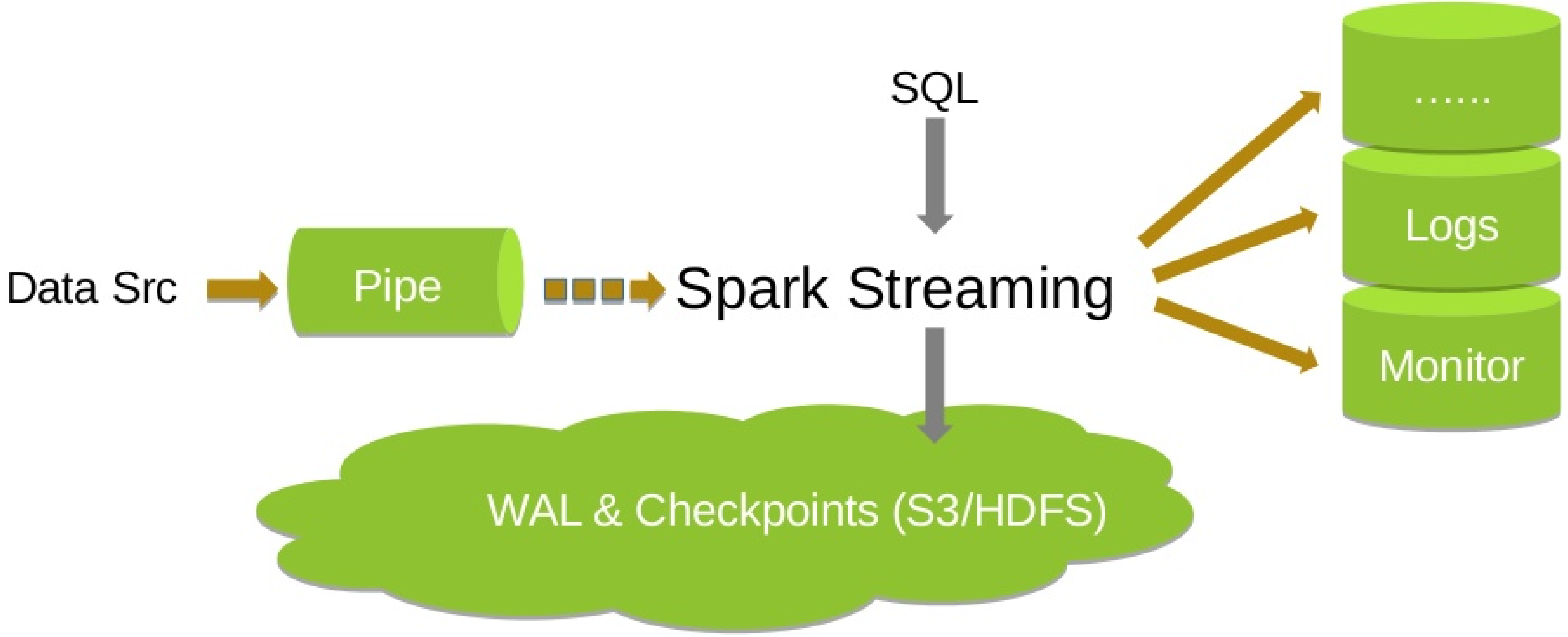
- **SICS ICE**
  Datacenter research environment
- **Hopsworks**
  Spark/Flink/Kafka/Tensorflow-as-a-service
  - Built on Hops Hadoop (www.hops.io)
    >150 active users

Self-Service Spark-Streaming



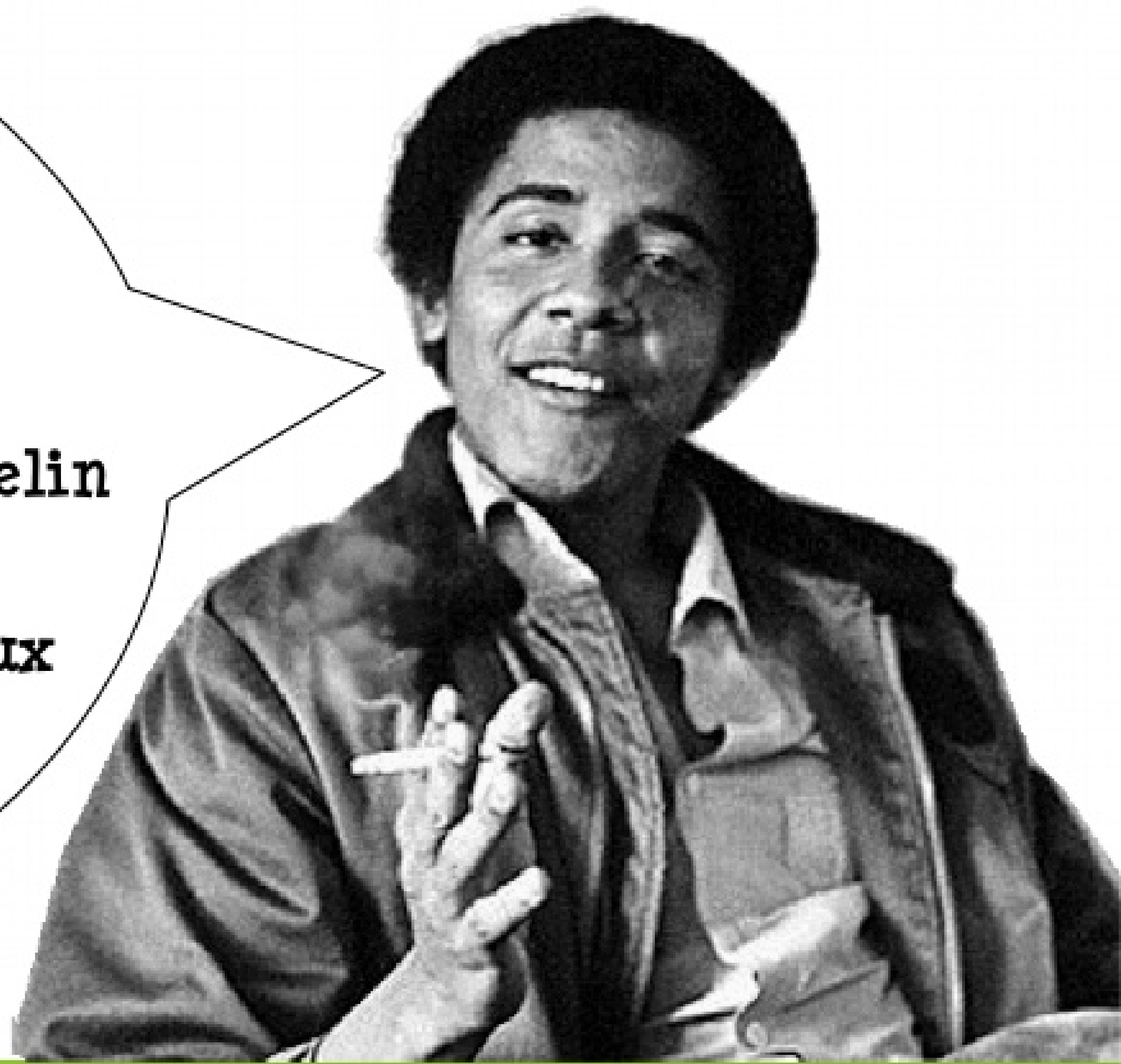
SPARK SUMMIT 2017

Structured Spark Streaming

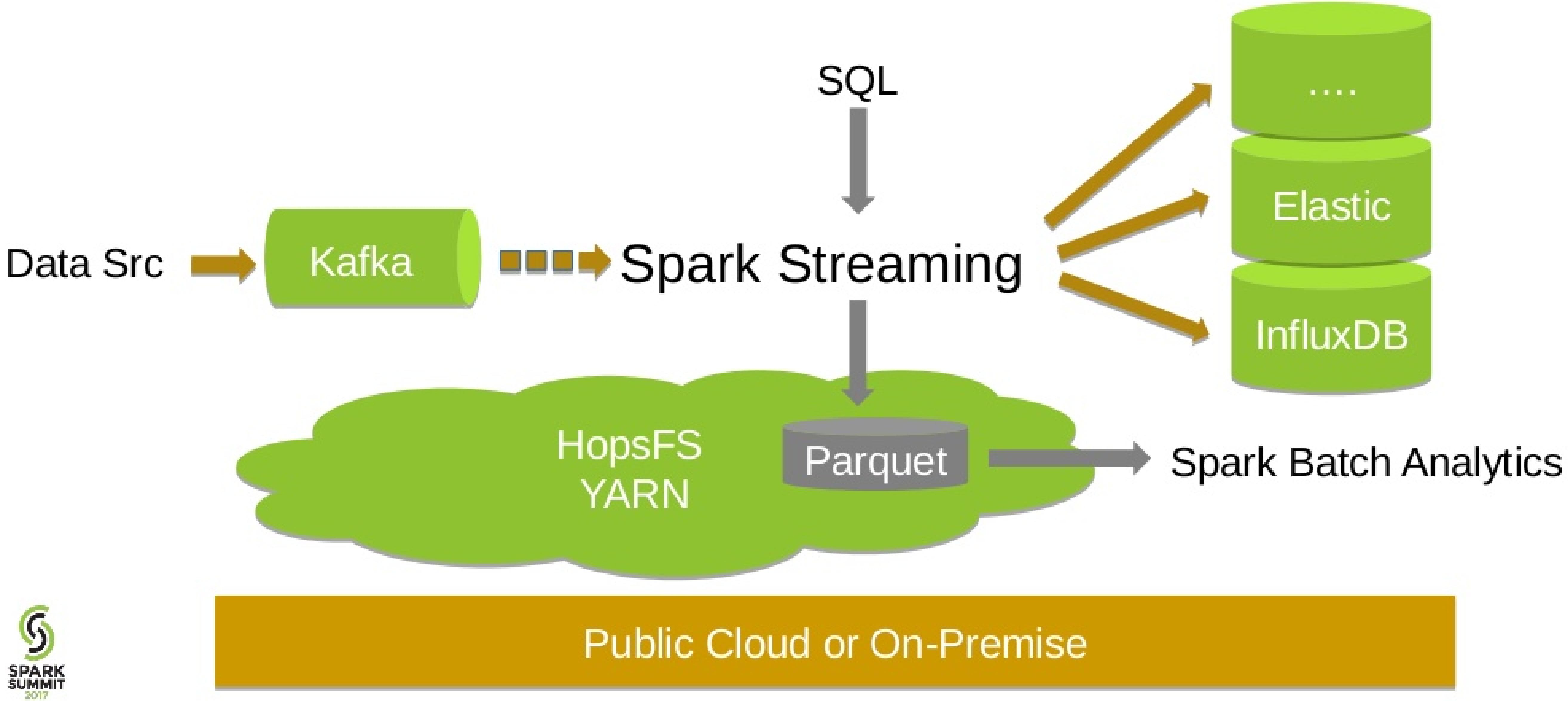


Smoking Spark Streaming

A **pipe** for my data:	Kafka
A **cloud** for my smoke:	Hops Hadoop
A way to **roll** A/B tests:	Jupyter/Zeppelin
A **lookout** for trouble:	Grafana/Influx
A **log** with evidence:	ELK Stack



Structured Spark Streaming (Hops)



General Data Protection Regulation (GDPR)

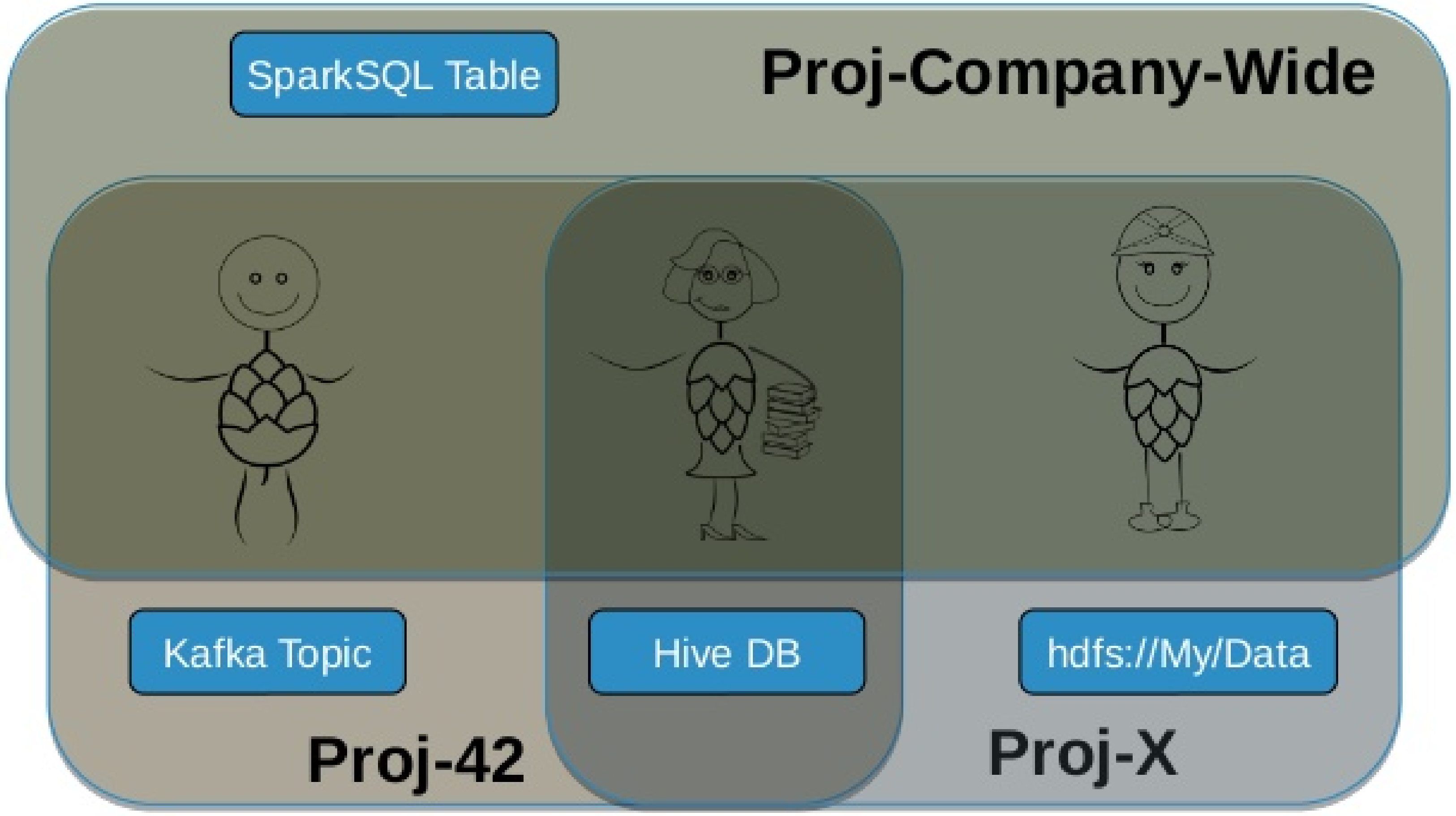
Ostrich Day: 2018-05-25

http://www.computerweekly.com/news/450295538/D-Day-for-GDPR-is-25-May-2018

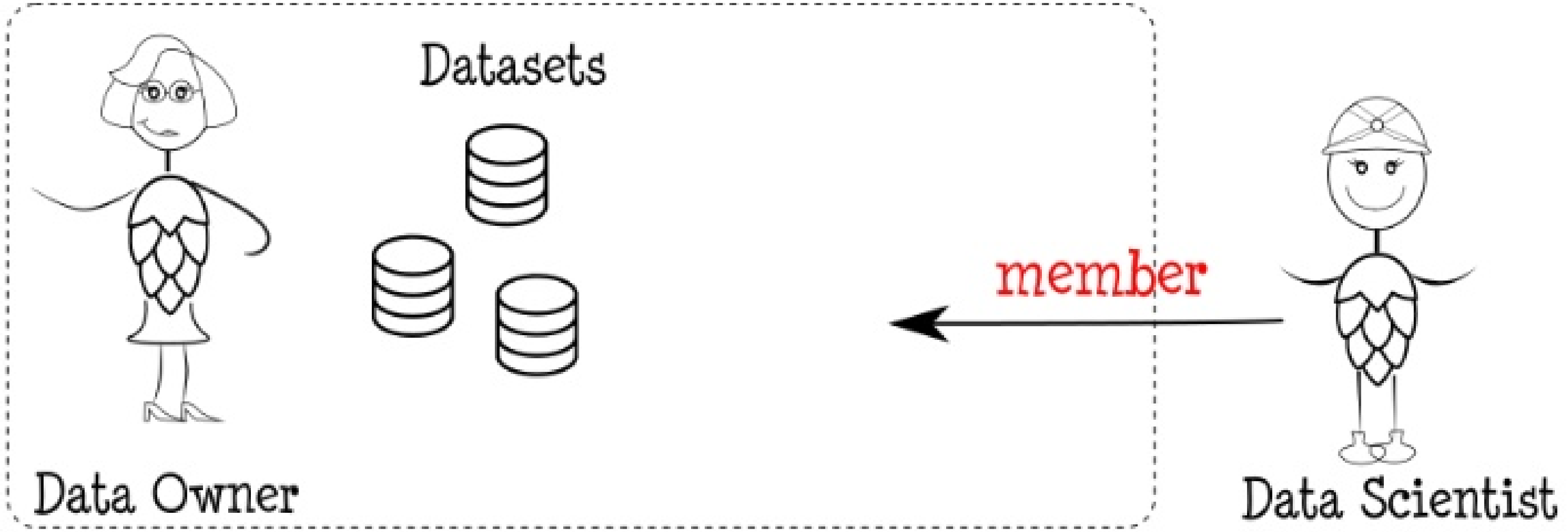
PRIVACY-BY-DESIGN WITH PROJECTS

Projects for Multi-tenancy

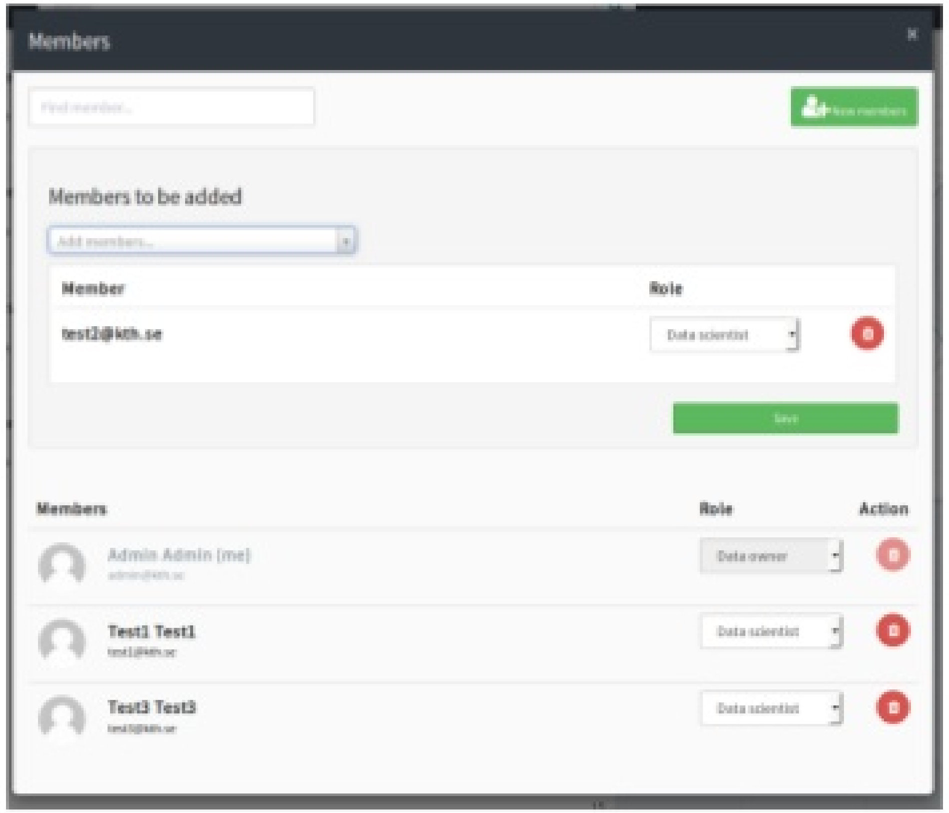
A **Project** is a Grouping of **Users** and **Data**



Manage Projects Like GitHub



Project Roles



Members to be added

Add members...

Member	Role
test2@kth.se	Data scientist

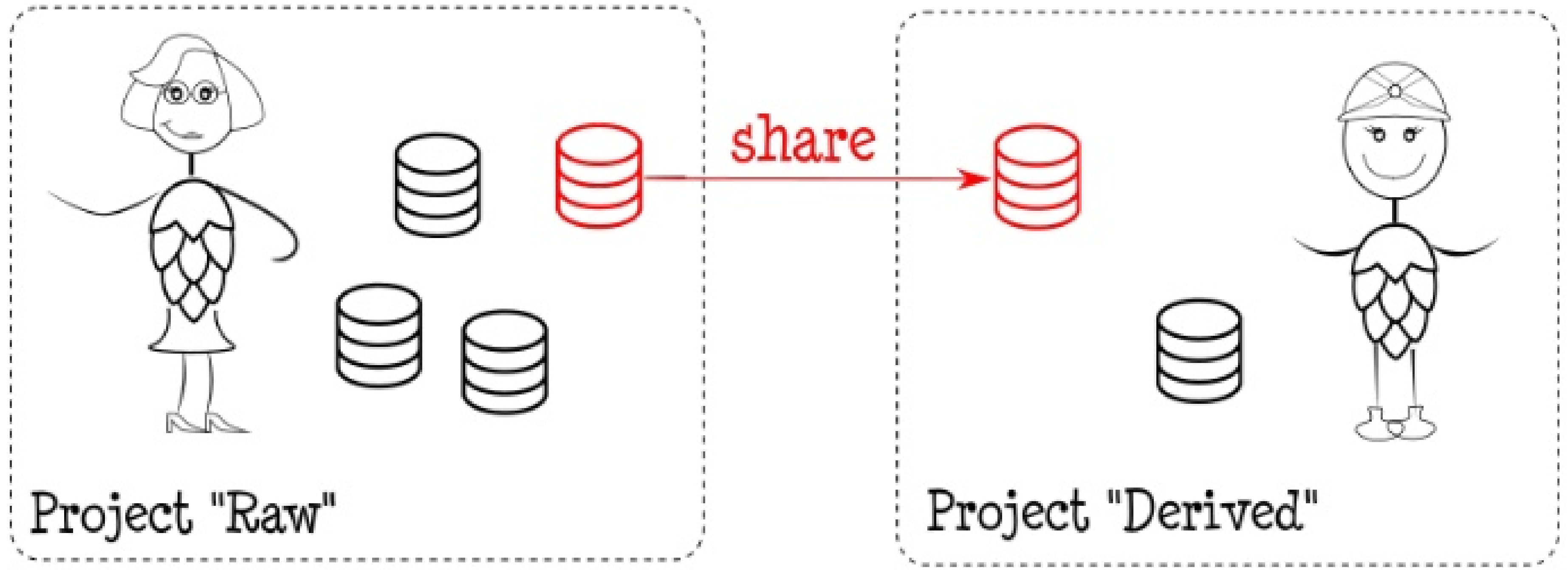
Members	Role	Action
Admin Admin (me) admin@kth.se	Data owner	
Test1 Test1 test1@kth.se	Data scientist	
Test3 Test3 test3@kth.se	Data scientist	

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

We delegate administration of privileges to users

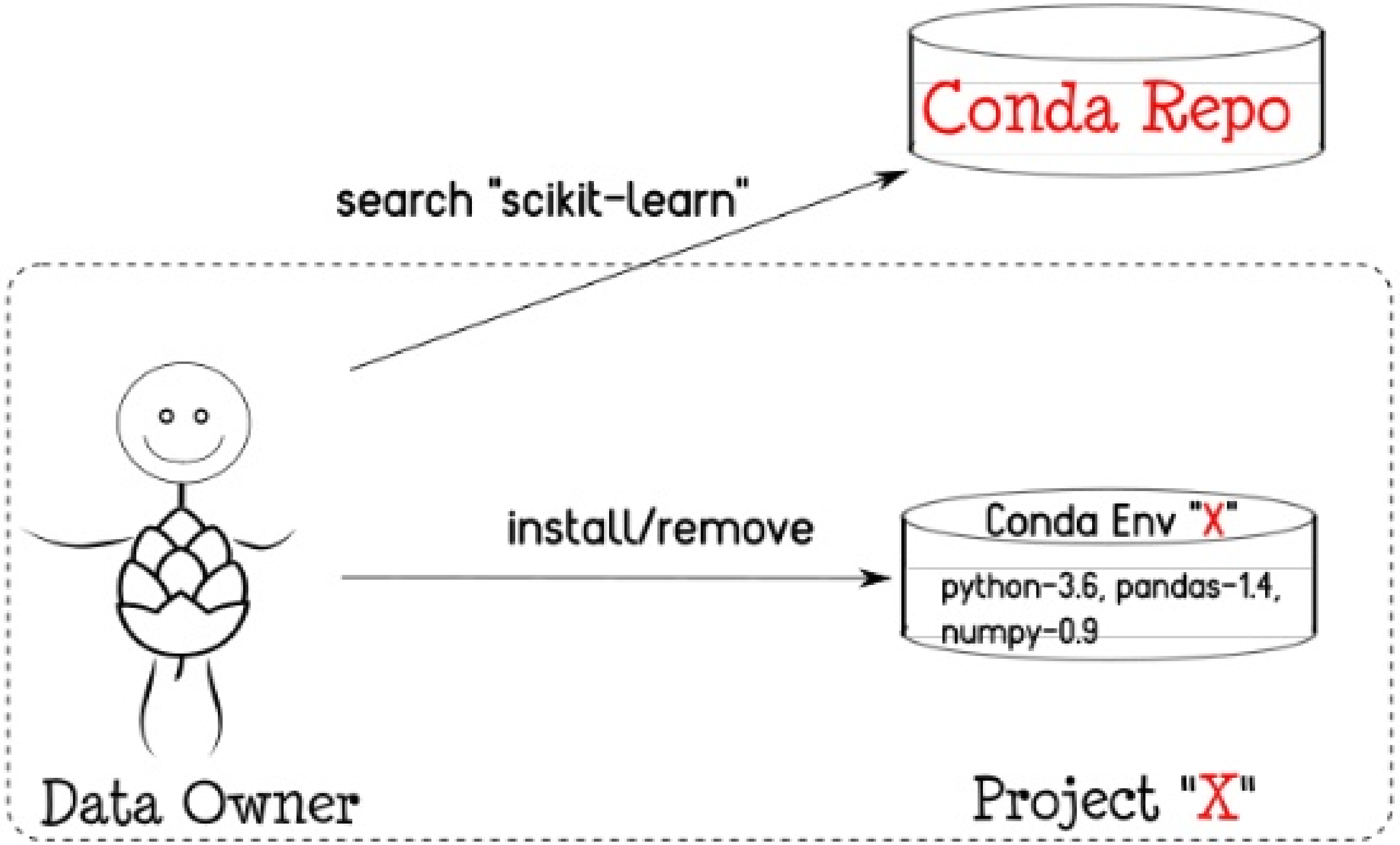
EASY SHARING IN A SECURE ENVIRONMENT

Share Datasets/Topics like Dropbox



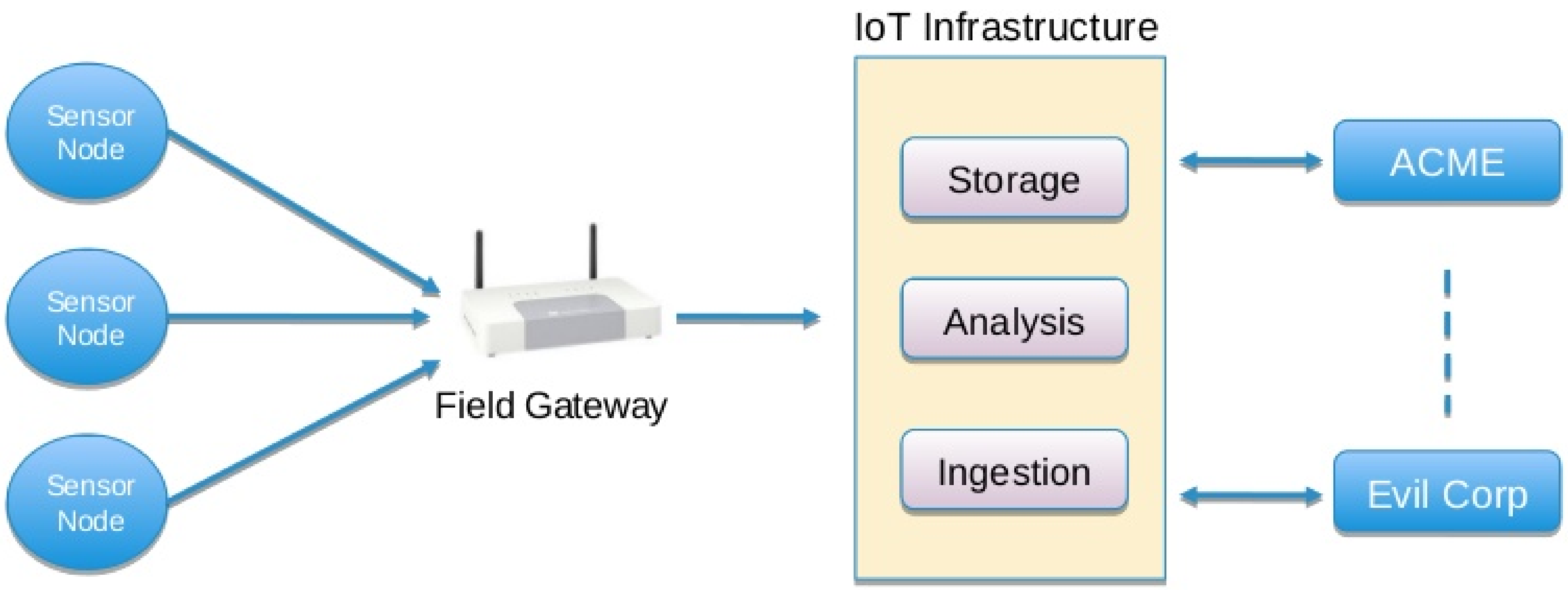
Share any Data Source/Sink: HDFS Datasets, Kafka Topics, etc

Custom Python Environments with Conda

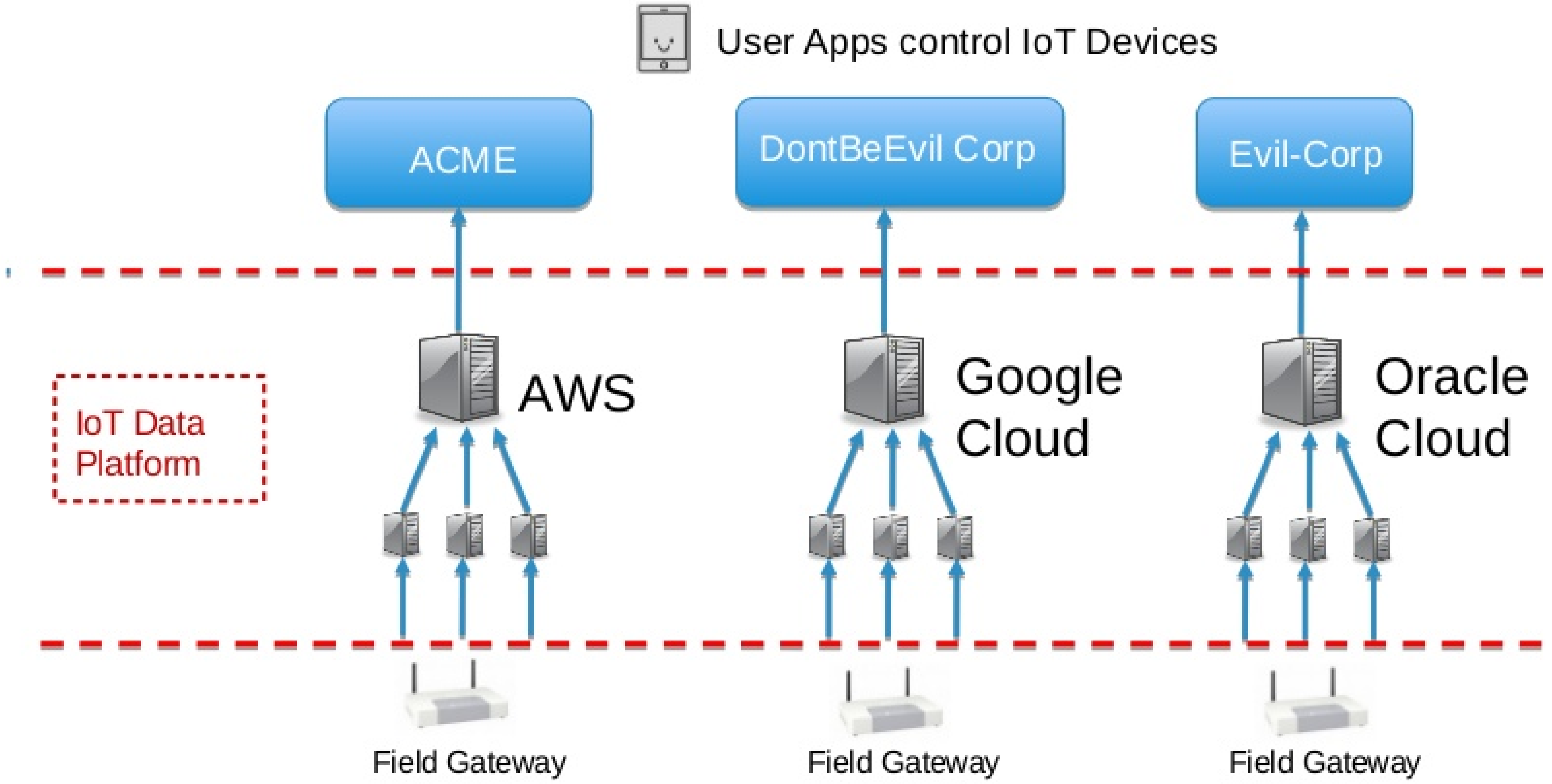


Python libraries are usable by any framework (Spark/Tensorflow)

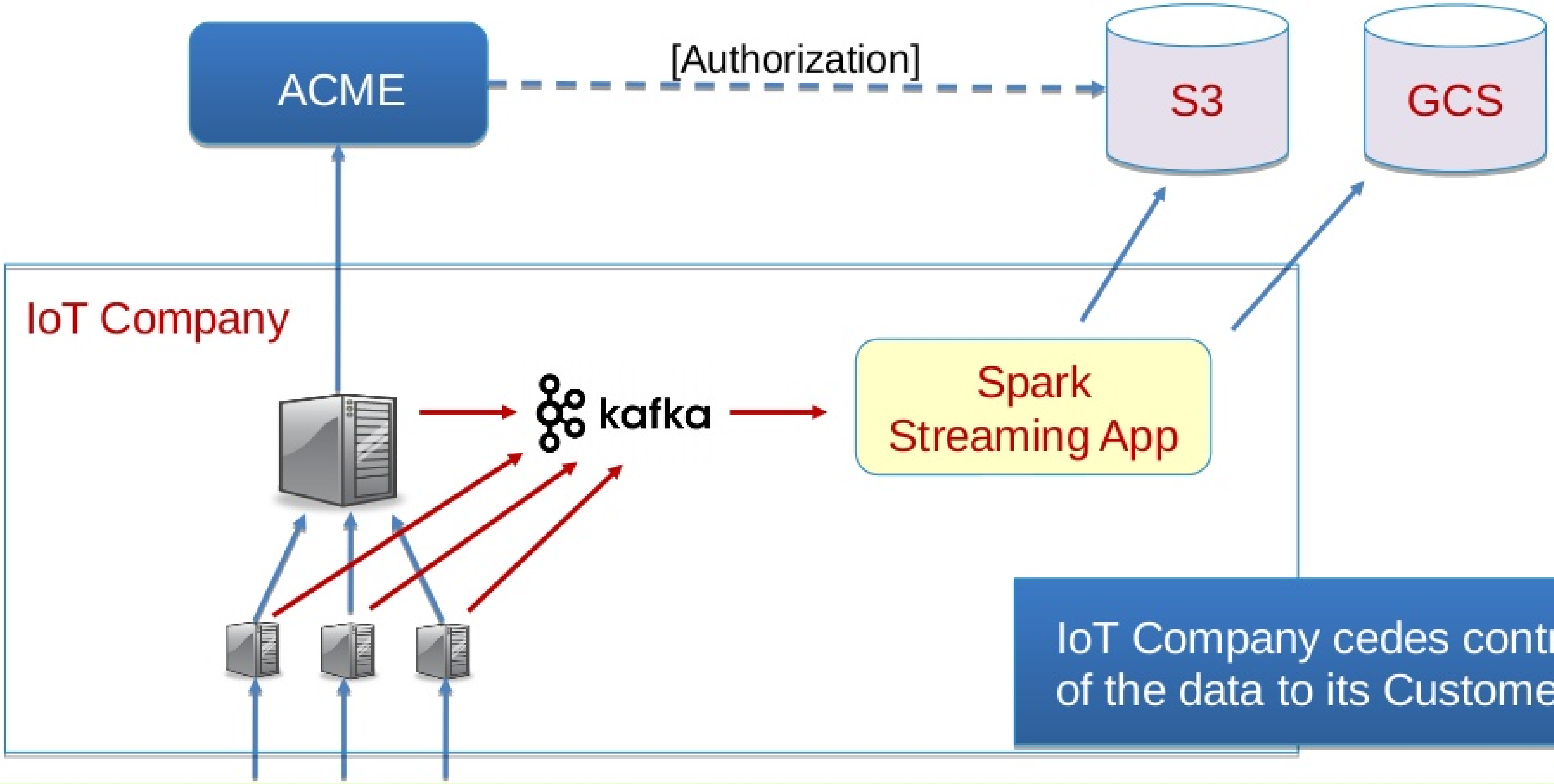
Case Study: IoT Data Platform



Multi-Cloud IoT Data Platform

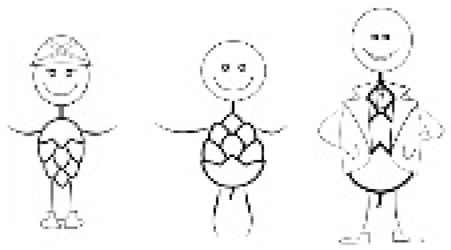


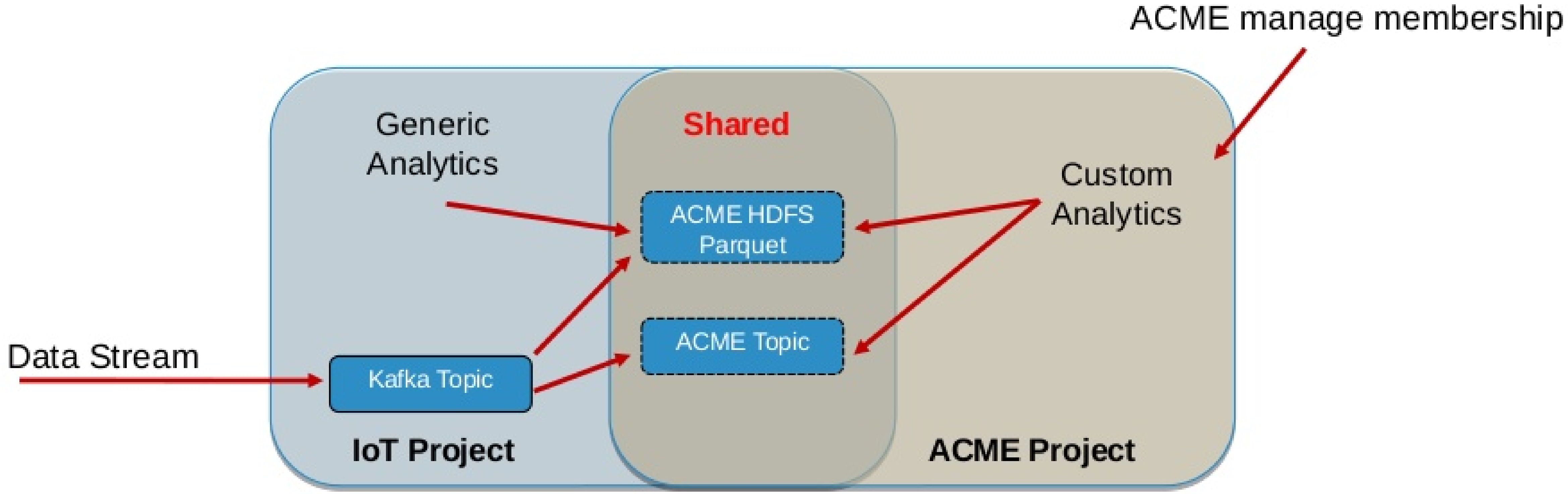
Cloud Native Solution



IoT Company cedes control of the data to its Customers

Hopsworks Solution: Project per Customer



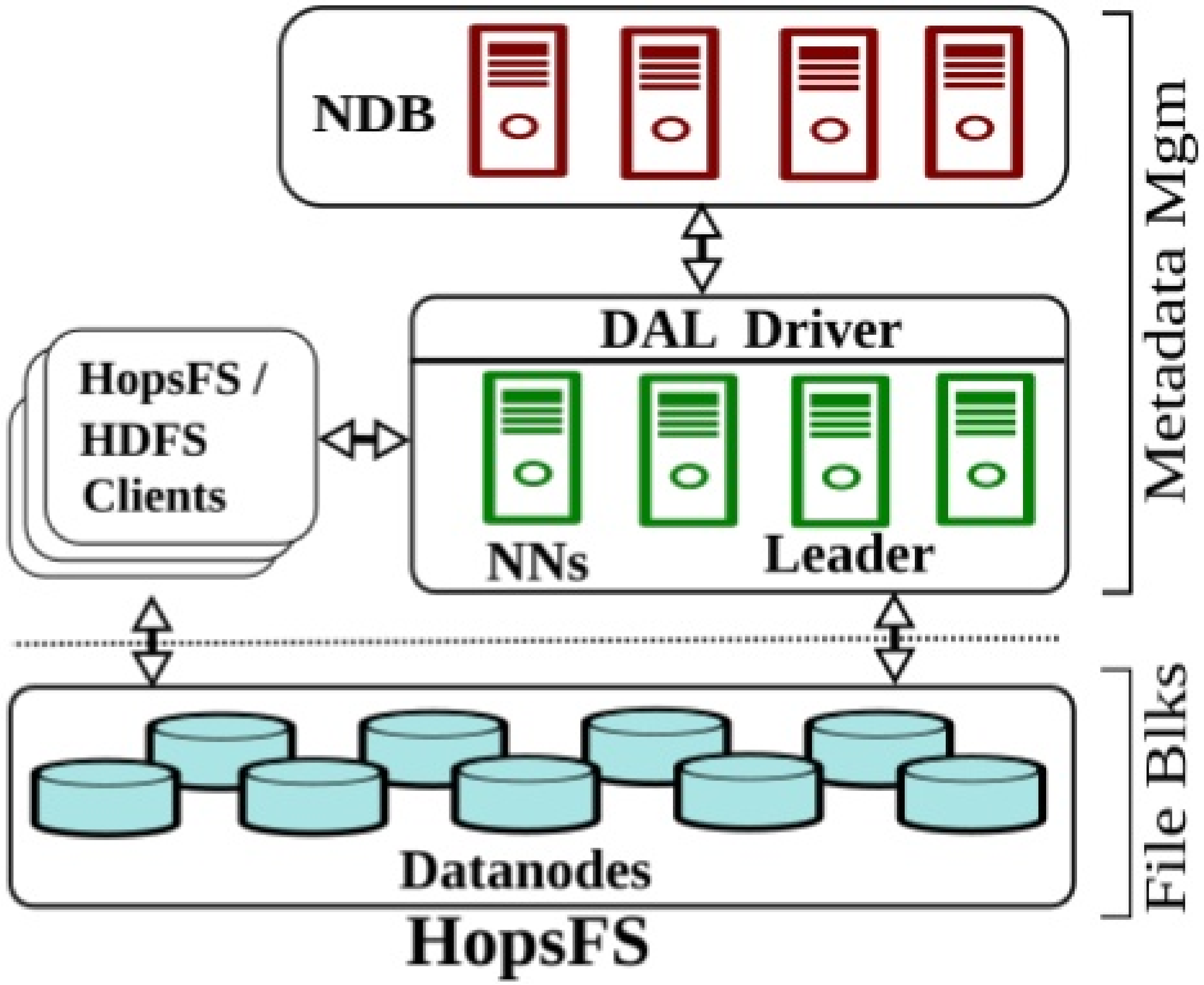


Spark Streaming Tooling

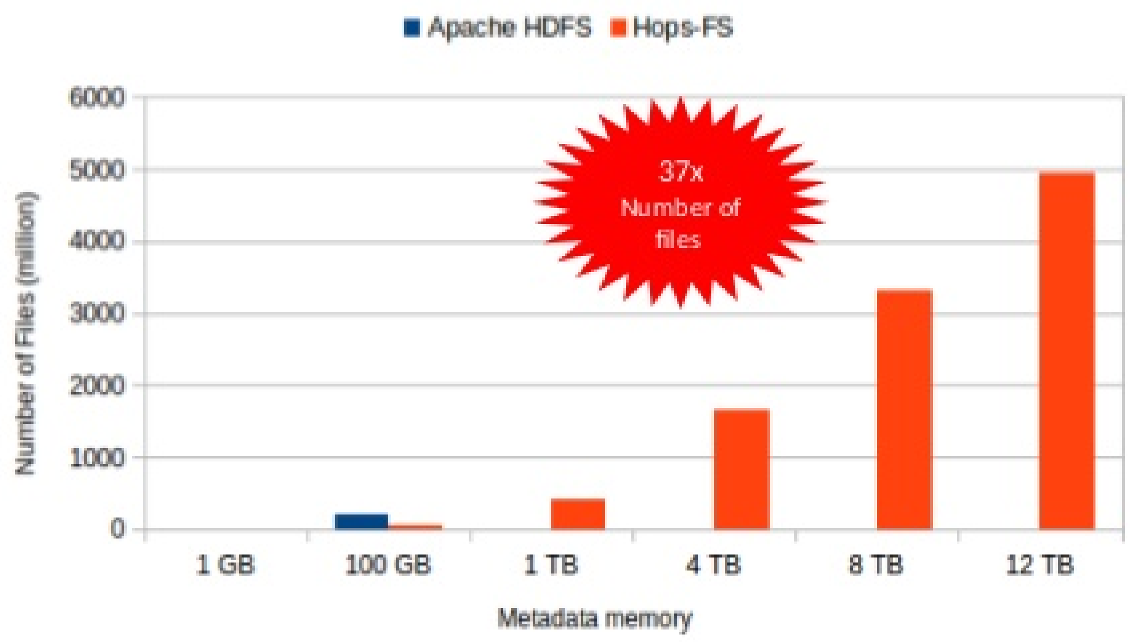
- Hops Hadoop
- Apache Kafka
- ELK Stack
- Grafana/InfluxDB
- Jupyter/Apache Zeppelin

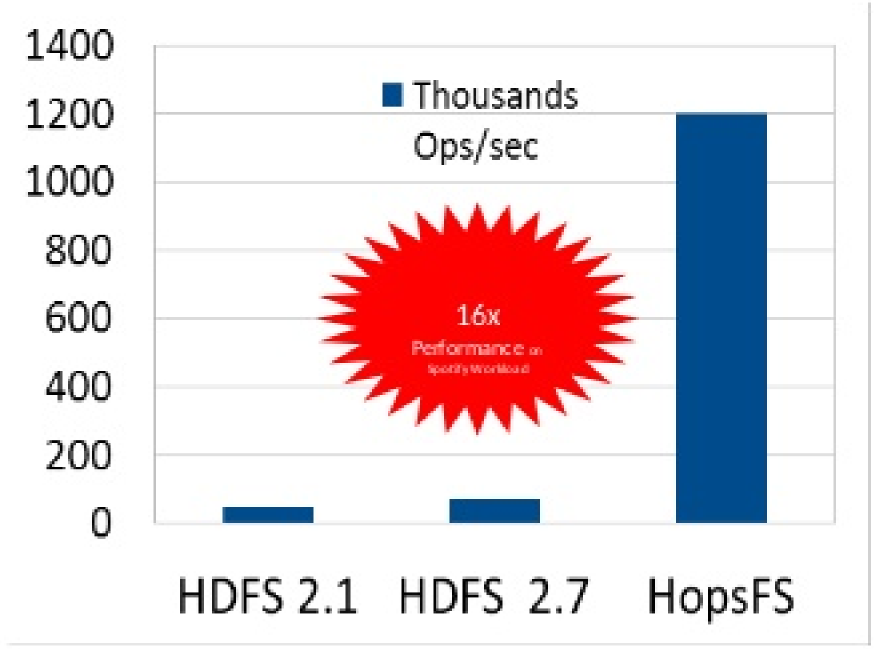
$\left.\right\}$ Hopsworks
Self-Service

HopsFS – Scale-out HDFS



HopsFS: Next Generation HDFS*





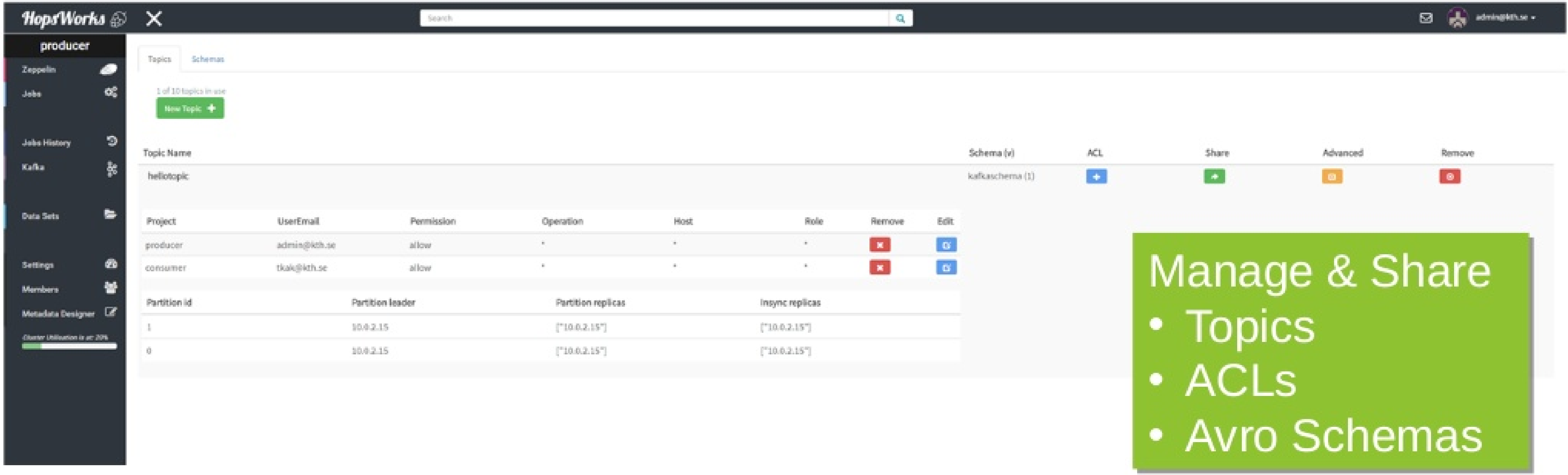
Bigger **Faster**

IEEE Scale Challenge Winner (2017)

SPARK SUMMIT 2017

*https://www.usenix.org/conference/fast17/technical-sessions/presentation/niazi

Kafka Self-Service UI



Topic Name	Schema (v)	ACL	Share	Advanced	Remove
hellotopic	kafkaschema (1)				

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

Tuning Kafka/Spark Streaming

- Key Kafka Tuning Parameters
  - Number of Topics
  - Number of Partitions per Topic
- Spark Streaming Tuning Considerations
  - Match # of Executors to the # of Partitions
  - Ensure balanced data across partitions

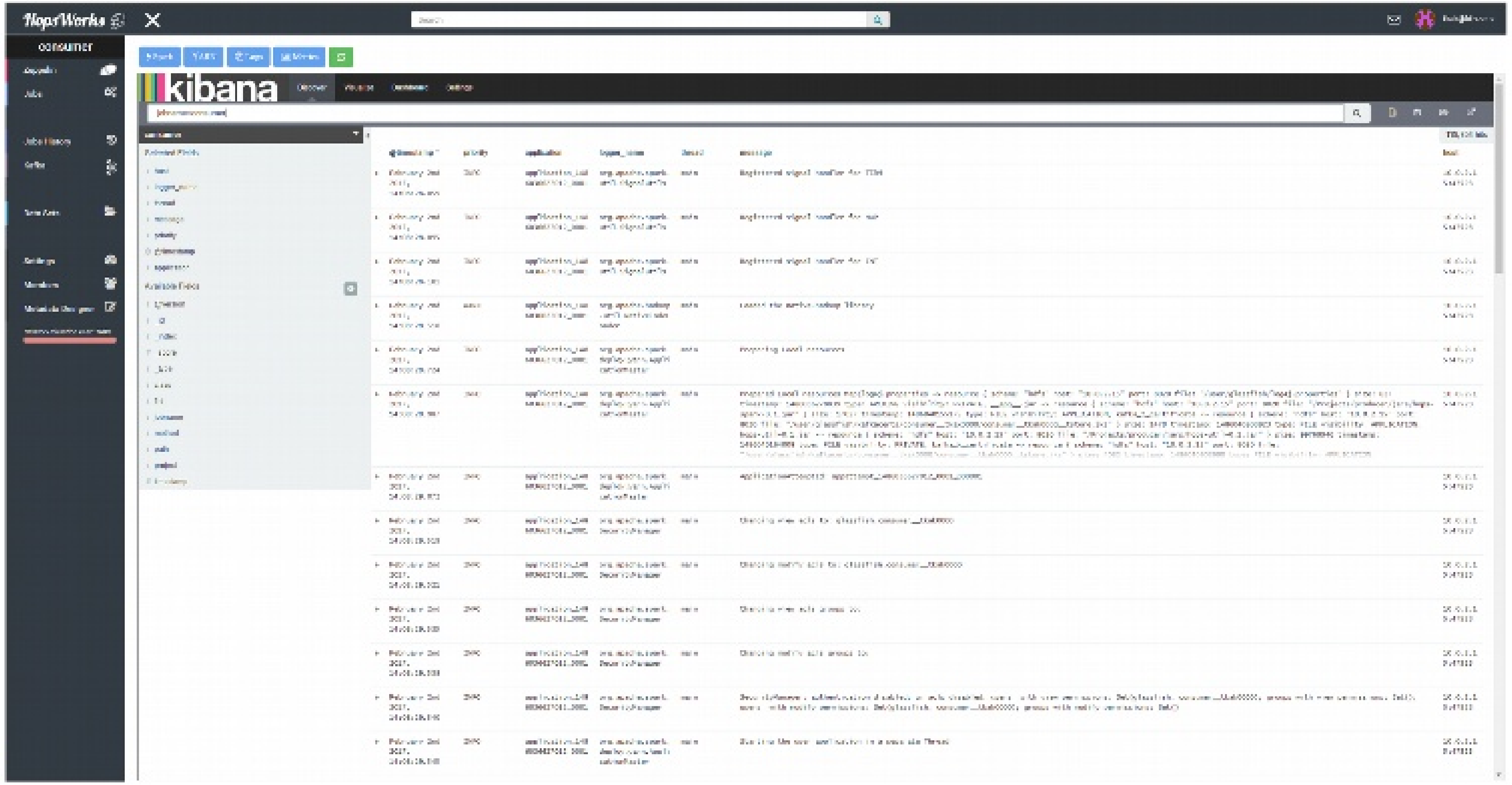
Realtime Logs

- 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

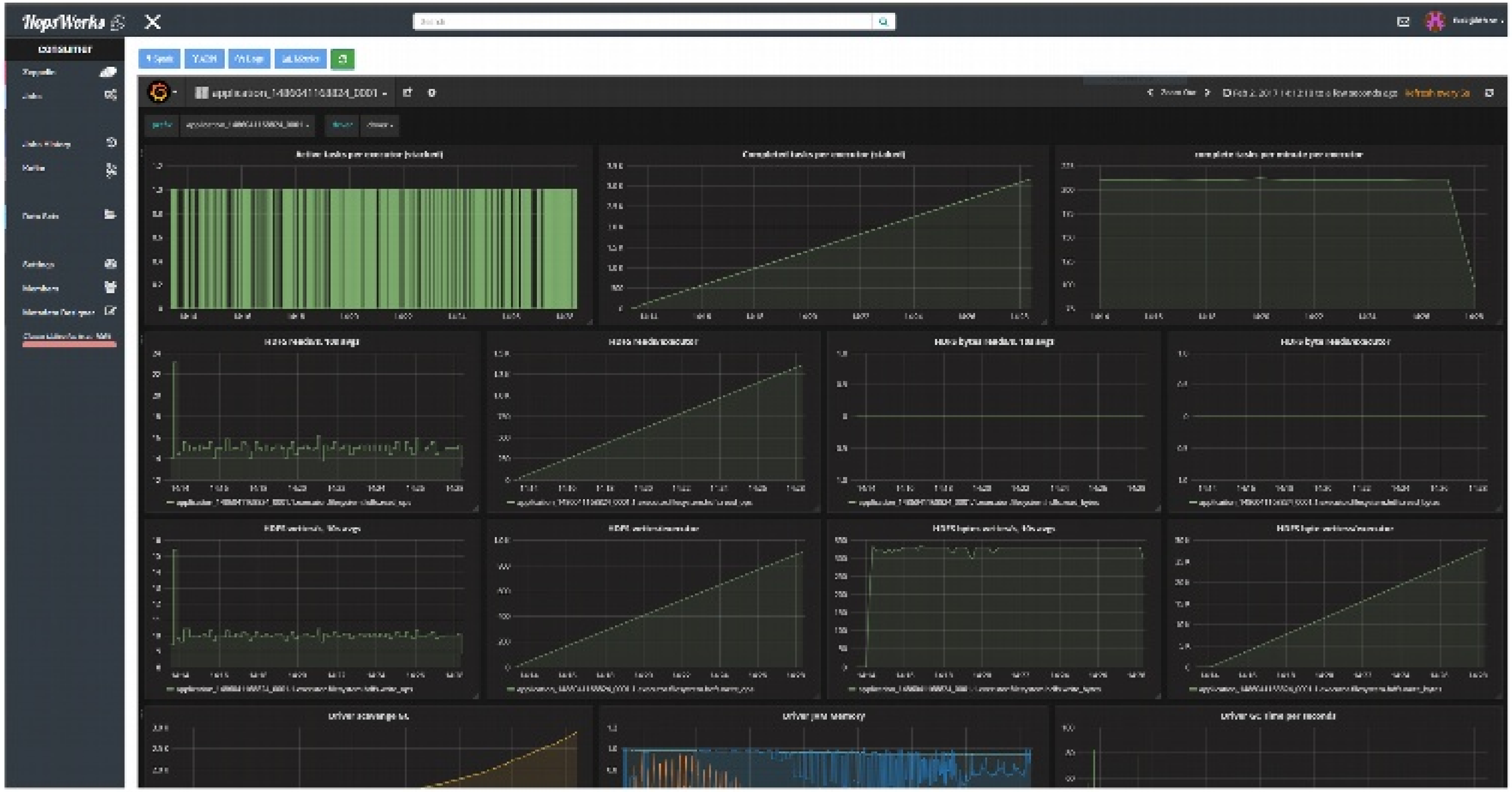


Elasticsearch,
Logstash,
Kibana
(ELK Stack)

Resource Monitoring/Alerting

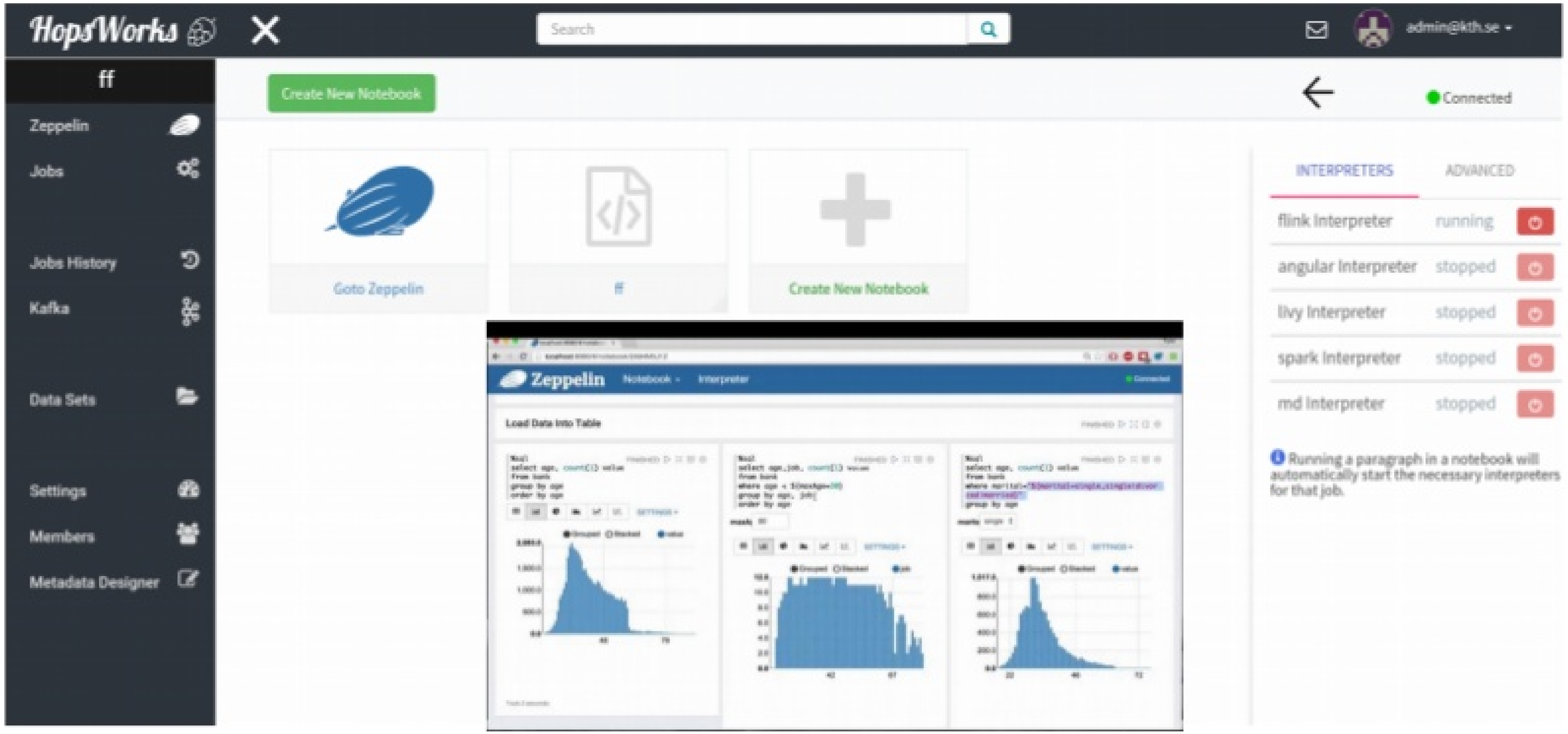
- $SPARK_HOME/conf/metrics.properties
  - Different sinks supported
    - JMX, **Graphite**, Servlet/JSON, CSV, Console, Slf4j
- StructuredQueryListener
  - Send query progress to a Kafka topic for inspection
- StreamingQueryListener
  - Asynchronous Monitoring of all queries for a Spark session

Resource Monitoring/Alerting


HopsWorks
Active tasks per executor (stacked)
Completed tasks per executor (stacked)
HDFS reads/executor
HDFS bytes reads/executor
HDFS writes/executor
HDFS bytes written/executor
Driver JVM Memory

Graphite/
InfluxDB
and
Grafana

Zeppelin for Prototyping Streaming Apps



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

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

[https://github.com/knockdata/spark-highcharts]

Project Quotas

- Per-Project quotas
  - Storage in HDFS
  - CPU in YARN (Uber-style Pricing)

- Sharing is not Copying
  - Datasets/Topics

SURGE PRICING

Demand is off the charts! Rates have increased to get more Ubers on the road.

1.75x

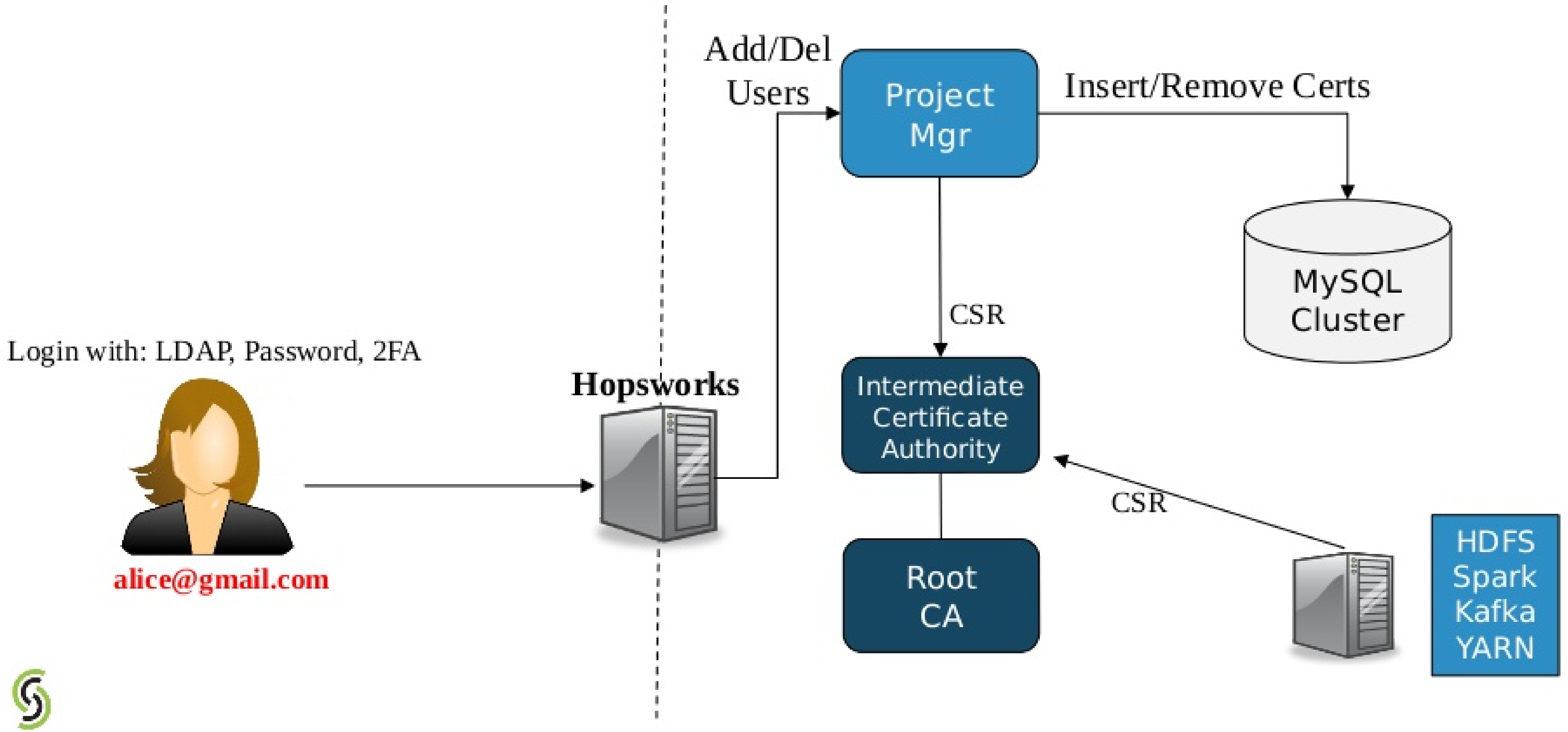
TIMES THE NORMAL RATE

Secure Spark Streaming

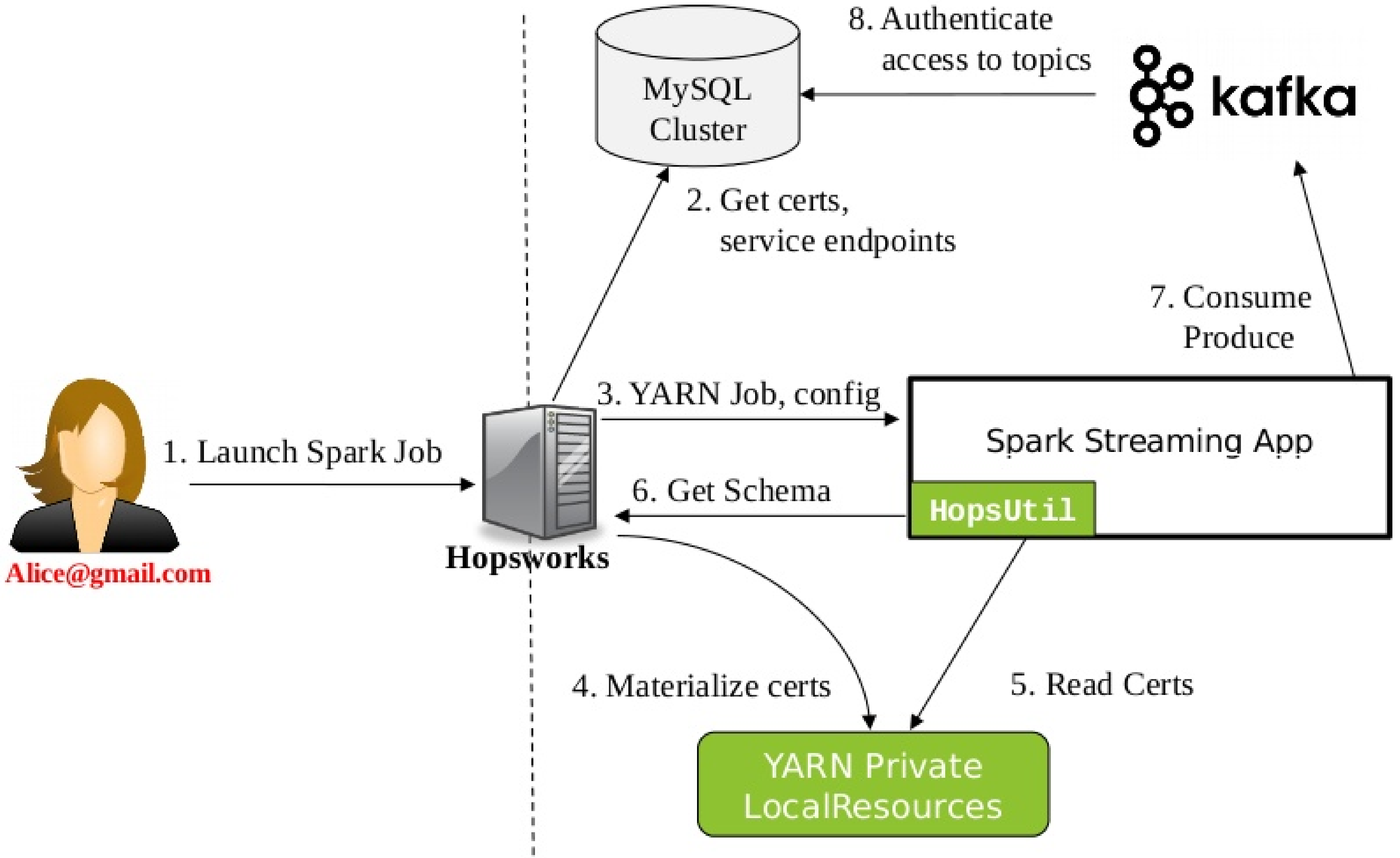
SSL/TLS Everywhere

- For each project, a user has a different SSL/TLS (X.509) certificate.
  - Client-side SSL certs for authentication
- Services are also issued with SSL/TLS certificates.
  - Kafka performs access control to topics using certs

SSL/TLS Certificate Generation



Distributing Certs for Spark Streaming



Simplified Structured Spark Streaming

Basic Structured Spark Streaming Program

- Query
- Input source
- Output sink
  - Mode: append/complete/update
- Trigger Interval
- Checkpoint Location

```
val cloudtrailEvents = …

val streamingETLQuery =
 cloudtrailEvents
  .withColumn("date",
    $"timestamp".cast("date")
  .writeStream
  .trigger(ProcessingTime
    ("10 seconds"))
  .format("parquet")
  .partitionBy("date")
  .option("path", "/cloudtrail")
  .option("checkpointLocation",
    "/cloudtrail.checkpoint/")
  .start()
```

https://databricks.com/blog/2017/01/19/real-time-streaming-etl-structured-streaming-apache-spark-2-1.html

Secure Structured Spark Streaming

- Streaming Apps also need to know:
  - Credentials, Endpoints
  - monitoring.properties
  - How to shutdown gracefully

- The **HopsUtil API** hides this complexity.

SPARK SUMMIT 2017

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.keystore.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\": \"U
ser\"," +
                        "\"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();
```

https://github.com/hopshadoop/hops-kafka-examples

Kafka Producer in HopsWorks

```
SparkConf sparkConf = new SparkConf().setAppName(HopsUtil.getJobName());
JavaSparkContext jsc = new JavaSparkContext(sparkConf);
...
SparkProducer producer = HopsUtil.getSparkProducer();
...
producer.produce(message);
...
HopsUtil.shutdownGracefully(jsc);
```

Streaming Consumer in HopsWorks

```
SparkConf sparkConf = new SparkConf().setAppName(HopsUtil.getJobName());
JavaSparkContext jsc = new JavaSparkContext(sparkConf);
...
DataStreamReader dsr = HopsUtil.getSparkConsumer().getKafkaDataStreamReader();
Dataset<Row> lines = dsr.load();
...
StreamingQuery queryFile = logEntries.writeStream()
...
HopsUtil.shutdownGracefully(queryFile);
```

Demo

Hops Roadmap

- HopsFS
  - Multi-Data-Center High Availability
  - Small files, 2-Level Erasure Coding
- HopsYARN
  - Tensorflow on Spark with GPUs-as-a-Resource
- Open Datasets
  - P2P Dataset Sharing

Spark Streaming on Hopsworks

- Hopsworks is a new Data Platform built on Hops
- Spark-Streaming is a First Class Citizen
- Built-in Tooling for Spark-Streaming

Hops Heads

Active:
Jim Dowling, Seif Haridi, Tor Björn Minde, Gautier Berthou, Salman Niazi, Mahmoud Ismail, Theofilos Kakantousis, Ermias Gebremeskel, Antonios Kouzoupis, Alex Ormenisan, Roberto Bampi, Fabio Buso, Fanti Machmount Al Samisti, Braulio Grana, Zahin Azher Rashid, Robin Andersson, ArunaKumari Yedurupaka, Tobias Johansson, August Bonds, Filotas Siskos.

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, Daniel Bali, Ioannis Kerkinos, Peter Buechler, Pushparaj Motamari, Hamid Afzali, Wasif Malik, Lalith Suresh, Mariano Valles, Ying Lieu.

SPARK SUMMIT 2017

Thank You.

Follow us: @hopshadoop

Star us: http://github.com/hopshadoop/hopsworks

Join us: http://www.hops.io