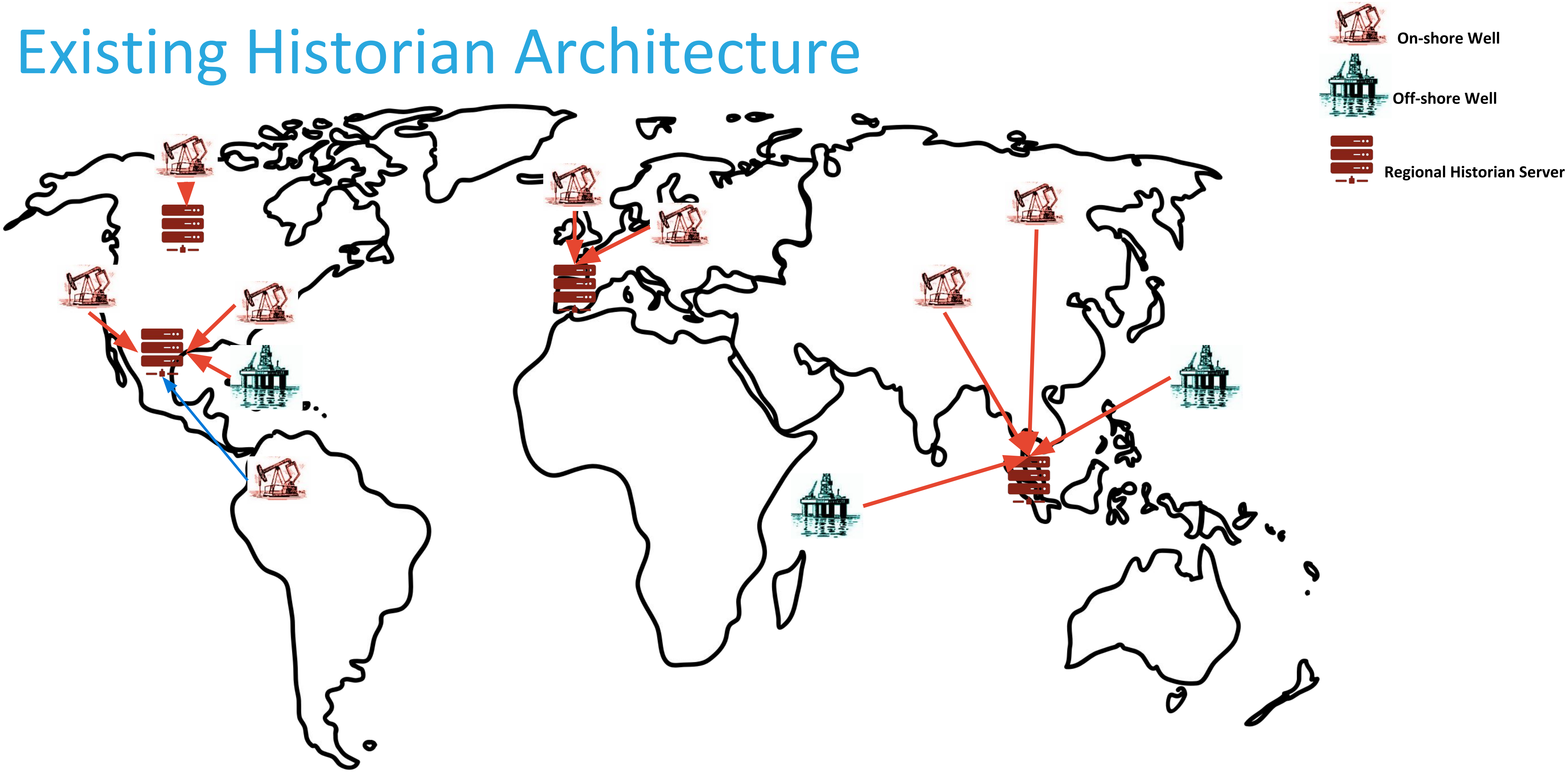
cloudera

Modern Historian Data Analytics

Samir Gupta, Sales Engineer
Sheridan McDonald, Account Executive
January, 2017

Existing Historian Architecture

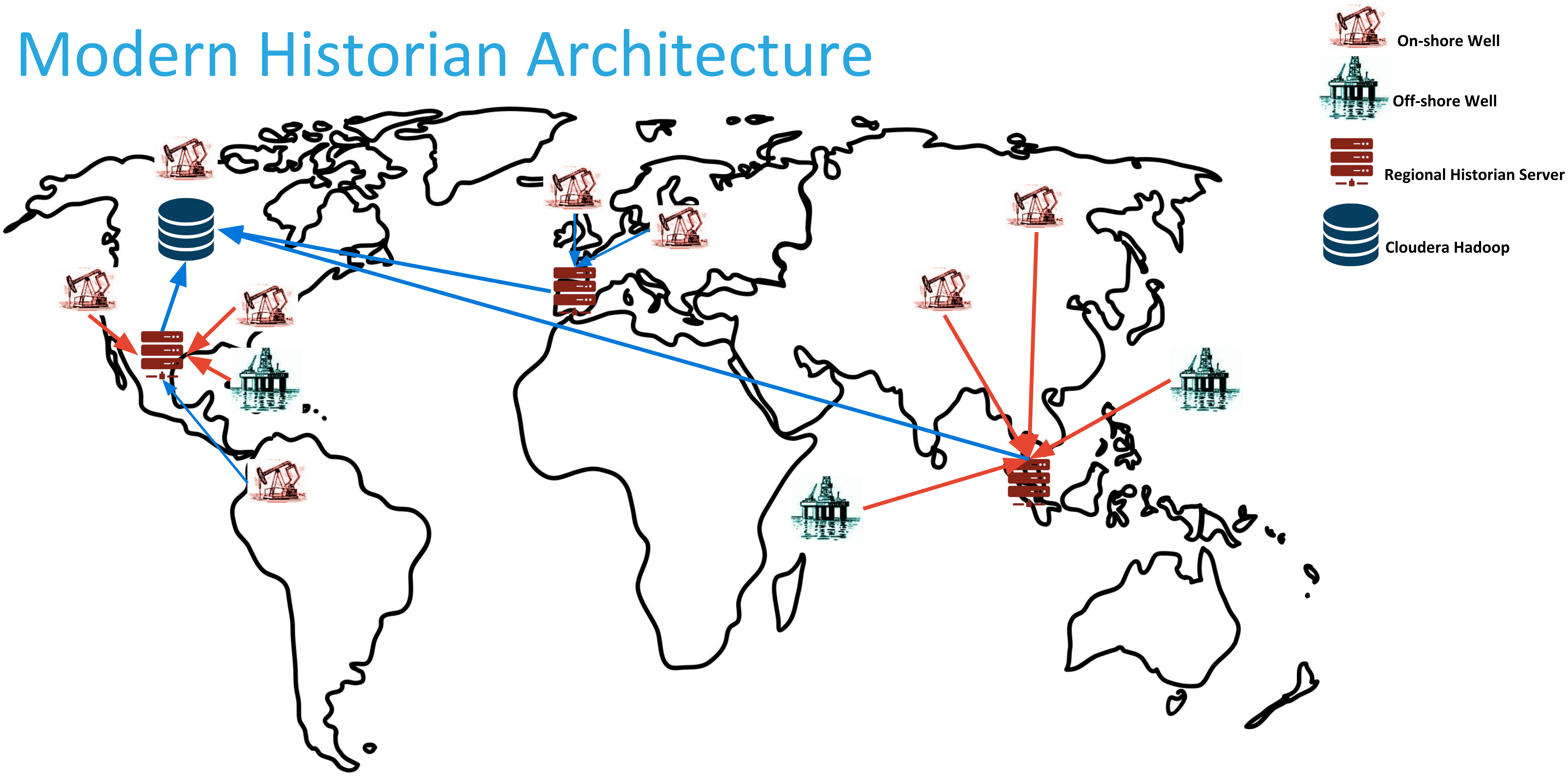


cloudera

Challenges with Existing Architecture

- Expensive
- Proprietary technology
  - Limited analytics capabilities
  - No inherent relationships between data and tags
  - Difficult to get data out in a useful format - ie. by individual tag ID
  - Ancient historian tooling to analyze the data
- No real-time access to the data
  - Raw data needs processing to be easily visualized
- Not well integrated with the data management ecosystem
  - Need to analyze in Excel, which as a 1M record limit (<.001% of data)

Modern Historian Architecture

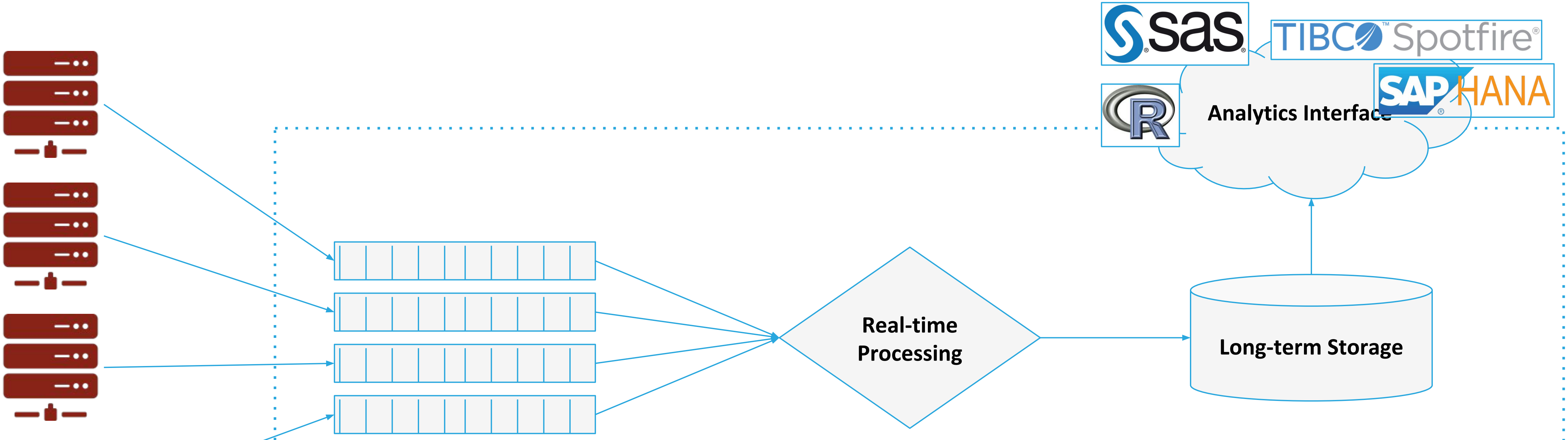


cloudera

Benefits with Modern Architecture

- Cost Effective
- Open technology
  - Advanced analytics capabilities
  - Easily create relationships between raw data and tag information
  - Integrate with any analytic tooling
- Real-time access to the data
  - Real-time processing of data
- Analyze all sensor data, instead of a small subset
  - 100B+ rows vs. 1M

Ingesting PHD Data into Cloudera



Regional Historian Servers
Flume agent sends sensor data to CDH

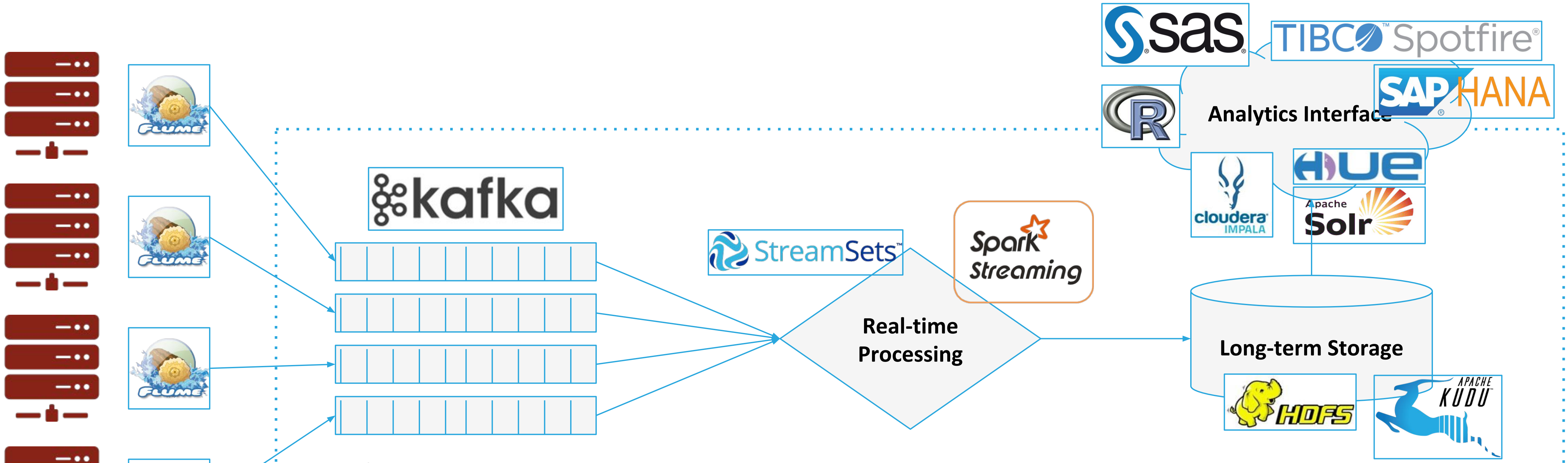
Pub-Sub Messaging
Reads data from Flume and makes available for stream processing with CDH

Stream Processing
Data processing, enrichment and machine learning

Advanced Analytics
Columnar storage, open APIs, integration with 3rd party tools

cloudera

Ingesting Historian Data into Cloudera



Kafka Pub-Sub Messaging
Reads data from Flume and makes available for stream processing with CDH

Stream Processing
Data processing, enrichment and machine learning

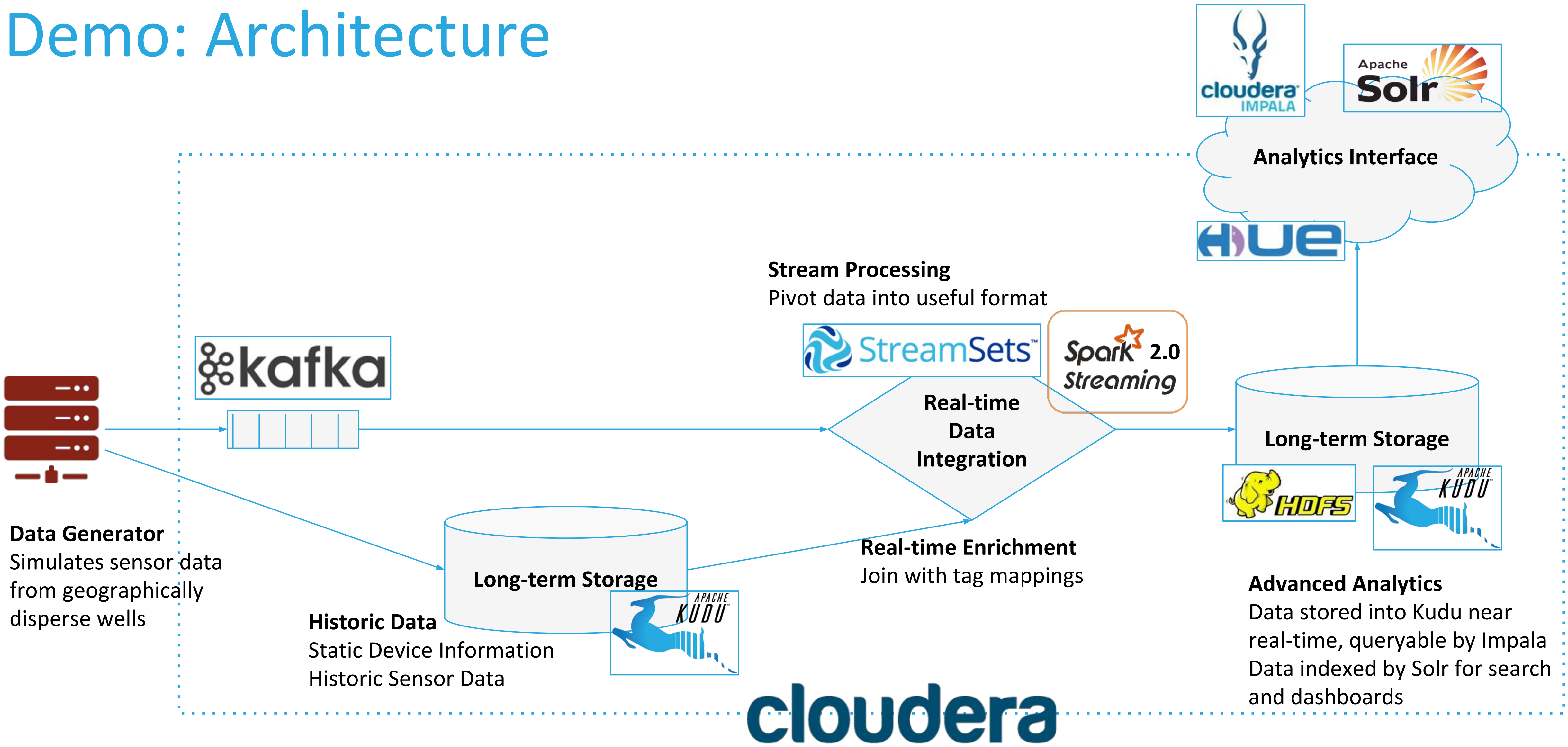
Advanced Analytics
Columnar storage, open APIs, integration with 3rd party tools

cloudera

Regional Historian Servers
Flume agent sends sensor data to CDH

cloudera

Demo: Architecture



Demo: Data Model

WELL_INFO
well_id
latitude
longitude
depth
well_type
well_chemical
leakage_risk

well_type → **“Offshore” or “Onshore”**

well_chemical → **“Oil” or “LNG”**

TAG_MAPPINGS
well_id
tag_id
sensor_name

sensor_name →

temperature_sensor
flow_rate
choke_value_opening
casing_pressure
tubing_pressure

RAW_MEASUREMENTS
tag_id
record_time
value

Real-Time Data
Generated in real-time

MEASUREMENTS
well_id
record_time
temperature_sensor
flow_rate
choke_value_opening
casing_pressure
tubing_pressure

Static Data
Generated during initial setup, configurable
Stored in Impala using KUDU for fast lookup

Real-Time Data
Enriched in real-time with Spark
Evaluator in Streamsets
Stored in near-real-time into KUDU

cloudera

Demo: Pivot in Spark

tag_id	record_time	value
1	9:00:00	50
2	9:00:00	12
3	9:00:00	80
1	9:00:30	48
2	9:00:30	14

tag_id	well_id	sensor_name
1	1	temp
2	1	flow
3	1	pressure

RAW_MEASUREMENTS
OUTER JOIN
TAG_MAPPINGS
ON tag_id

tag_id	record_time	value	well	sensor_name
1	9:00:00	50	1	temp_sensor
2	9:00:00	12	1	flow_rate
3	9:00:00	80	1	casing_pressu re
1	9:00:30	48	1	temp_sensor
2	9:00:30	14	1	flow_rate
3	NULL	NULL	NULL	pressure

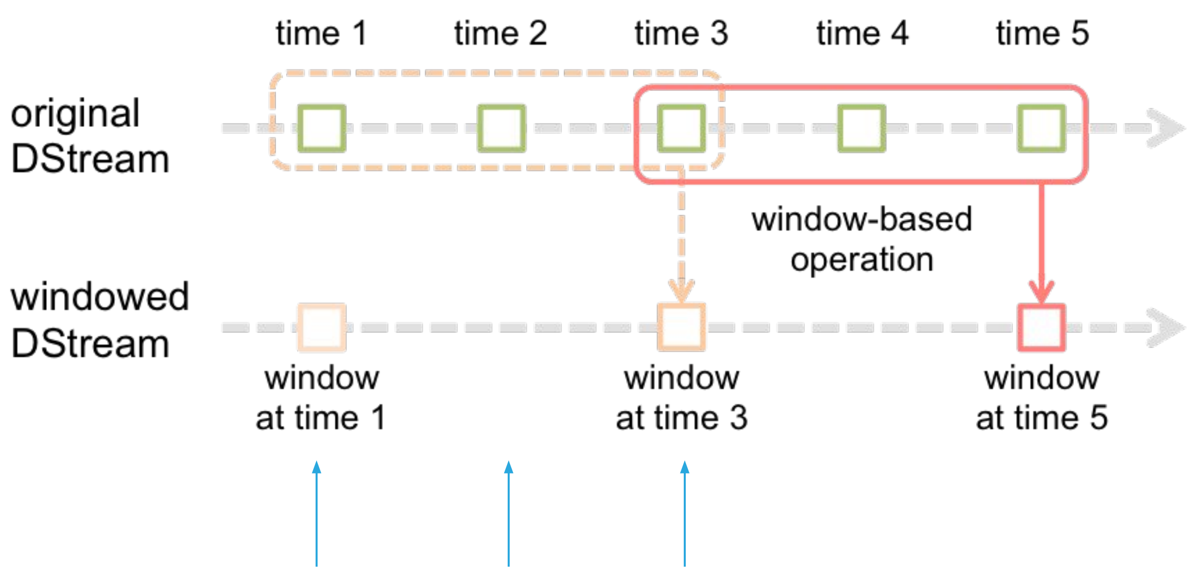
PIVOT ON sensor_name

well_id	record_time	temp	flow	pressure
1	9:00:00	50	12	80
1	9:00:30	48	15	NULL

UPSERT INTO KUDU

cloudera

Demo: Spark Streaming Windows



Streaming Interval = 30 seconds

Window Interval = 90 seconds

well_id	record_time	pressure
1	9:00:00	50
2	9:00:00	80
1	9:00:30	52
2	9:00:30	NULL at W1, 81 at W2
1	9:01:00	49
2	9:01:00	76
1	9:01:30	55
2	9:01:30	86
1	9:02:00	50
2	9:02:00	NULL

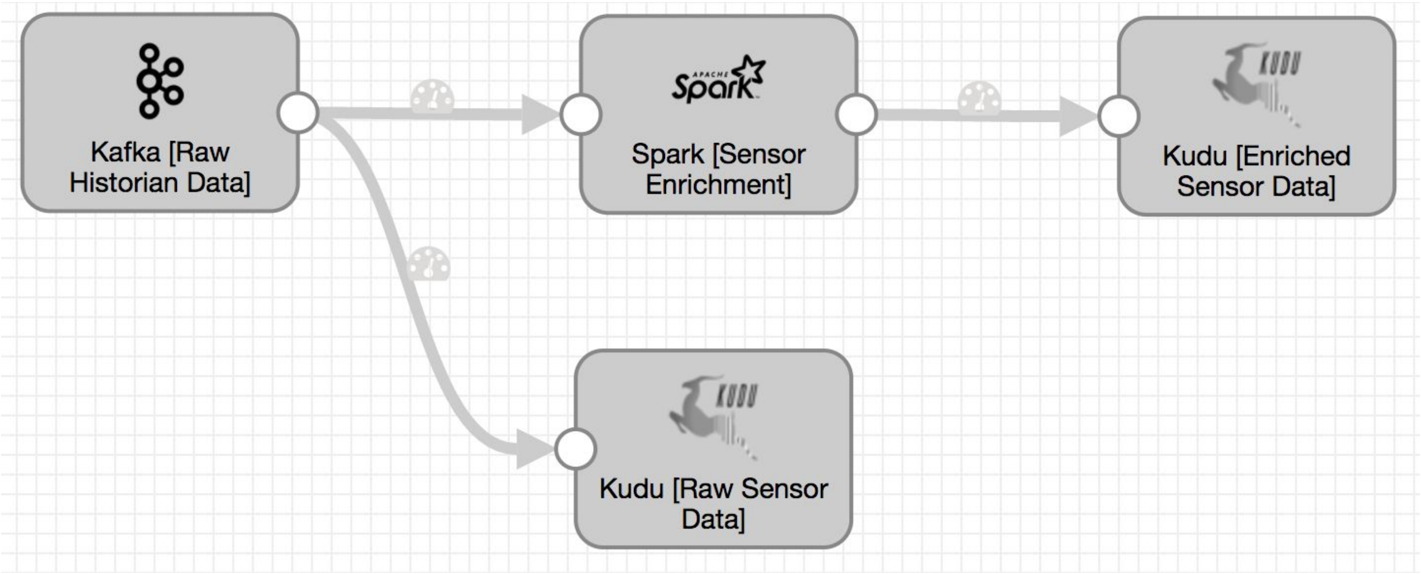
Window 1

Window 2

Window 3

cloudera

Demo: Streamsets Data Collector



Kudu Table
Sends enriched sensor data to Impala table to be queried and used in BI

Kafka Source
Reads raw sensor data from Kafka cluster

Custom Spark Evaluator
Real-time lookup to Kudu to enrich sensor data
Pivot and aggregate before sending to Kudu

Kudu Table
Raw sensor data also stored in Kudu for post-processing analysis/debug

Streamsets running in "Cluster Mode" to read directly from a Kafka cluster. The entire pipeline will run as a Spark streaming application inside CDH.

cloudera

cloudera

Thank you

Samir Gupta, Systems Engineer
sgupta@cloudera.com

Sheridan McDonald
smcdonald@cloudera.com

