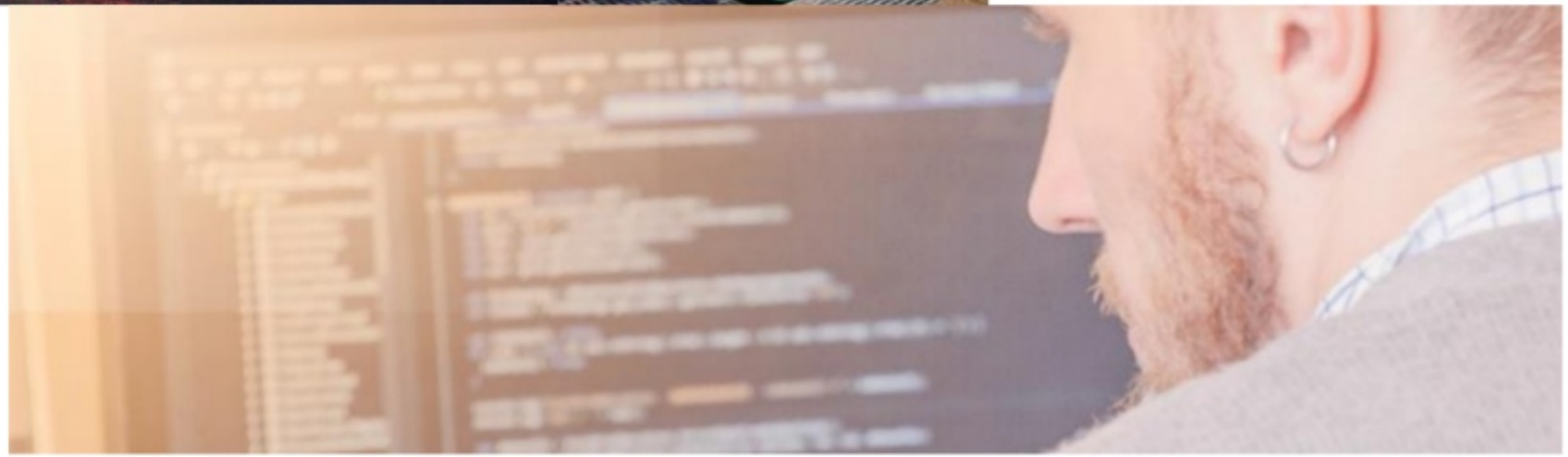


What is Intelligent Assistant?

1. A person who assists a specific person with their business tasks

2. A software agent that can perform tasks or services

3. Skynet

SPARK—UNIVERSAL COMPUTATION ENGINE FOR PROCESSING OIL INDUSTRY DATA

Andy Starzhinsky, Vice President
Yaroslav Nedashkovsky, System Architect
SoftEleganceData

Data analysis: from data collection to predictive analytics

1. Oil industry overview
2. Data Lake
3. Data Collection
4. Data analysis


Operations Excellence
Production Excellence
Remote Operations & Management
Collaborative working (CWEs)
Real-time Data And Visualization
Knowledge Management
Reliability and Maintenance
Integrated Planning
Major Projects
Production optimization
Drilling and Completions
Environmental Monitoring
Flow Assurance Analysis
Condition Based monitoring
Surveillance and Control of Subsea Production Systems
Asset Integrity
Intelligent Completions
IWIS
Well Optimisation
Umbilical and Flow line Monitoring
Reservoir Management

Data Flow

Challenges

- Subsurface monitoring

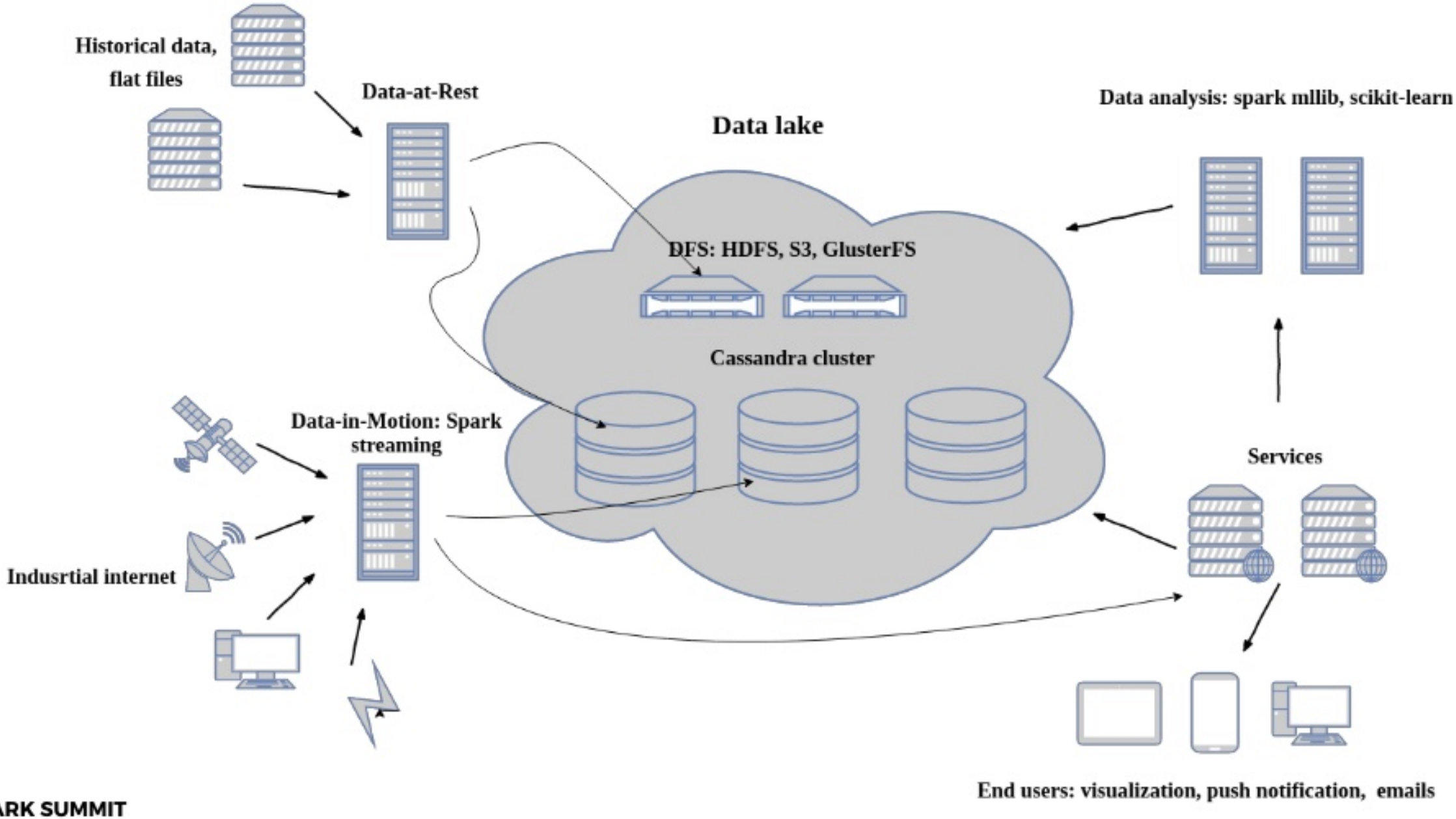
- Unification of data collection, monitoring and analysis

- Predictive analytics

Intelligent Assistance

- Failure Prediction
- Digital management
- Major overhaul

2. Data Lake


Historical data,
flat files
Data-at-Rest
Data lake
Data analysis: spark mllib, scikit-learn
DFS: HDFS, S3, GlusterFS
Cassandra cluster
Data-in-Motion: Spark
streaming
Services
Indusrtial internet
End users: visualization, push notification, emails

Data sources

- sensors readings
- flat files (for example las* file)
- legacy dataset

Data storage

- Cassandra
- distributed file system (GlusterFS)

Infrastructure

MESOS

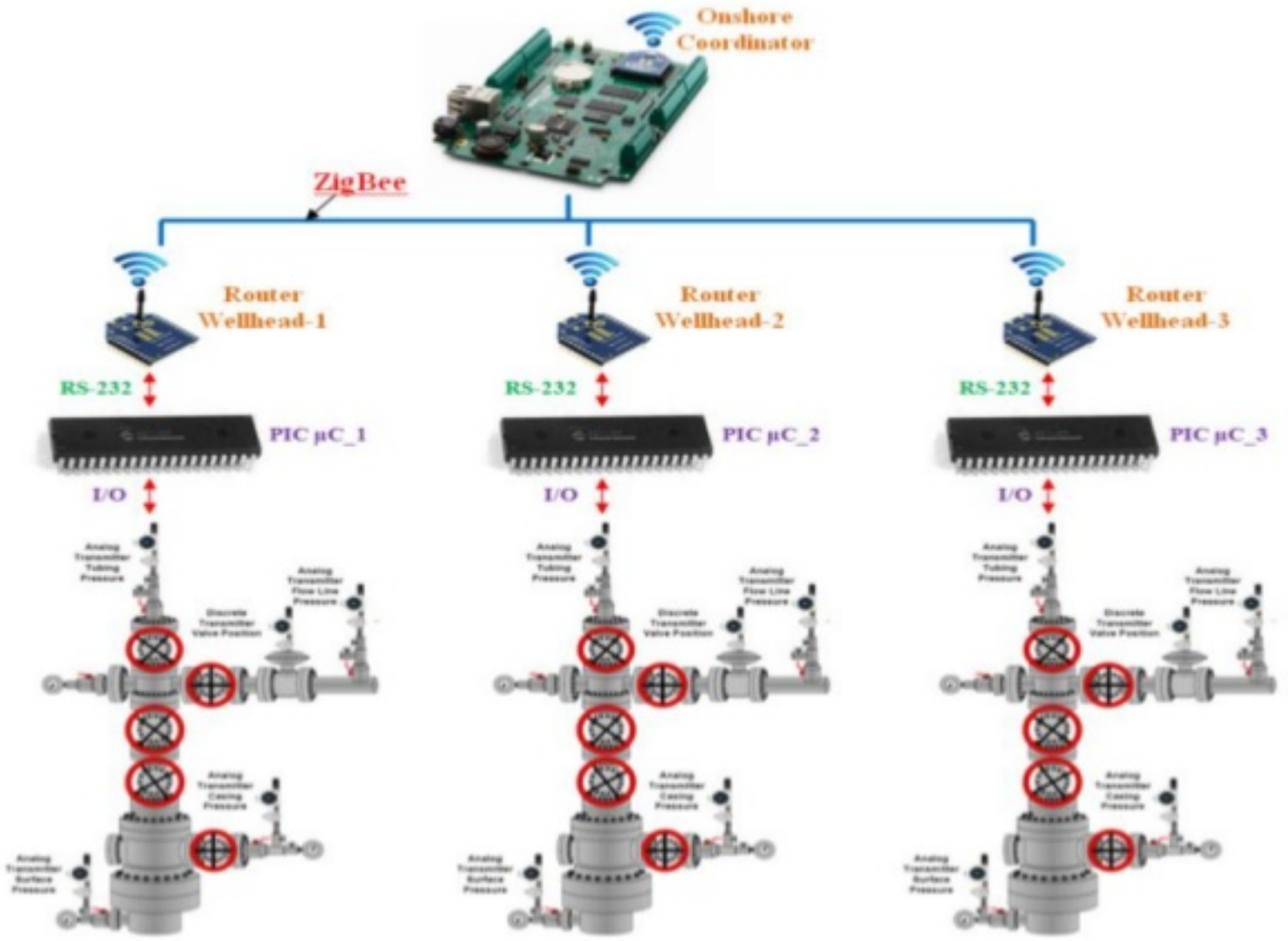
ANSIBLE

Jenkins

Cloud agnostic

3. Data Collection

Smart Well


Onshore
Coordinator
ZigBee
Router
Wellhead-1
Router
Wellhead-2
Router
Wellhead-3
RS-232
RS-232
RS-232
PIC µC_1
PIC µC_2
PIC µC_3
I/O
I/O
I/O

4. Data Analysis

Add-hoc query with Spark Notebook

SPARK NOTEBOOK visualization (unsaved changes)

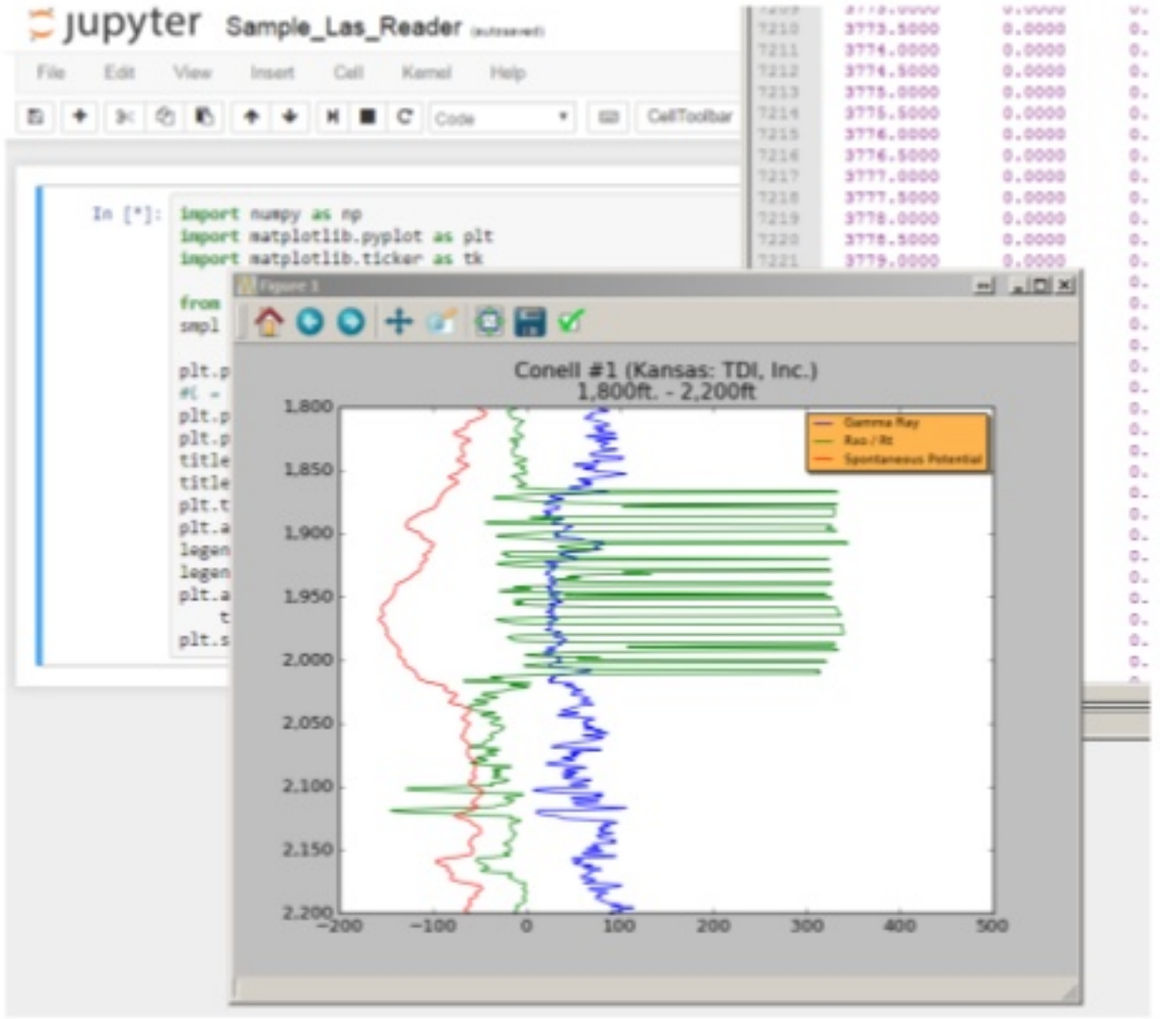
File Edit View Insert Cell Kernel Help | Scala [2.11.8] Spark [2.0.0] Hadoop [2.2.0] (Hive ✓)

Code Cell Toolbar: None

```
import org.apache.spark._
import org.apache.spark.rdd._

import org.apache.spark.mllib.classification.{LogisticRegressionWithLBFGS, LogisticRegressionModel}
import org.apache.spark.mllib.evaluation.MulticlassMetrics
import org.apache.spark.mllib.regression.LabeledPoint
import org.apache.spark.mllib.linalg.Vectors
import org.apache.spark.mllib.util.MLUtils

val spark = SparkSession
  .builder()
  .appName("Wells visualization")
  .getOrCreate()

val rawDS = spark.read.
                  format("com.databricks.spark.csv").
                  option("header", "true").
                  load("/datalake/ks_wells.csv").cache()


val fDS = rawDS.filter("STATUS == 'OIL'").select("LATITUDE", "LONGITUDE")
fDS.count()
```

```
val rawData = fDS.rdd

val converter = (x:Any) => { if (x == null) 0.0 else x.toString.toDouble }
val points = rawData.map(row => (converter(row(0)), converter(row(1)))).collect()

GeoPointsChart(points)
```

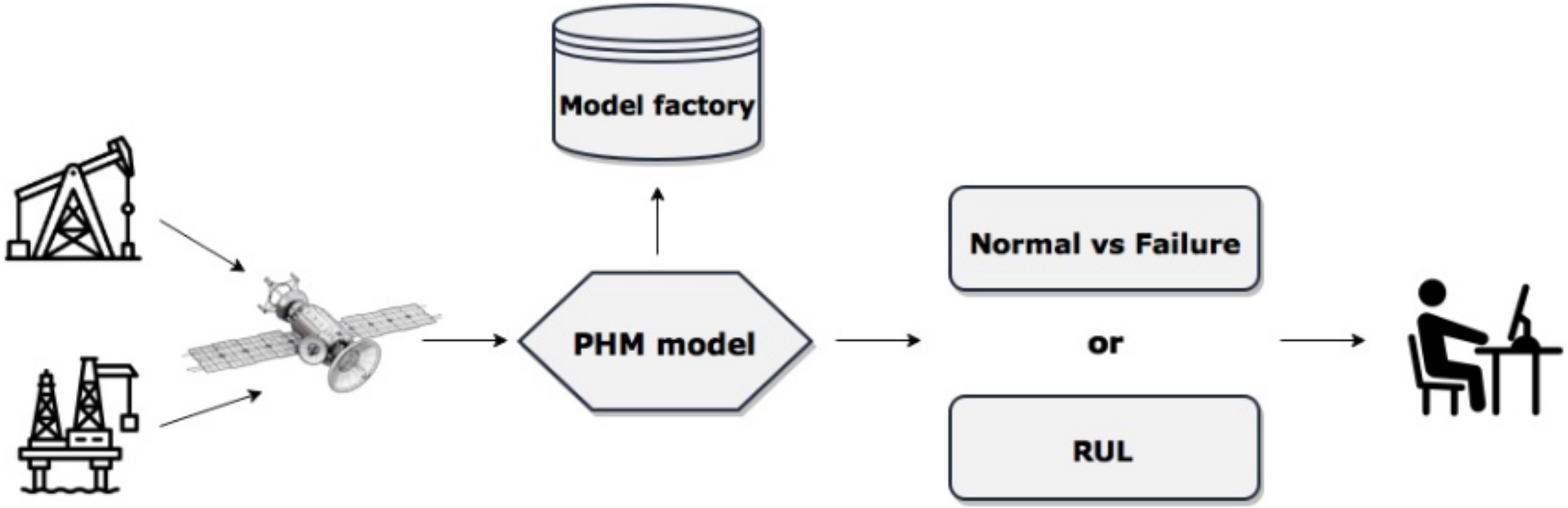
Jupyter + Python for LAS file analysis



Prognostic Health Monitoring

- determine remaining useful life
- predict failures before they occur

PHM workflow



Artificial lift

Estimation:
1% of improvement in ESP (Electric Submersible Pumps) performance world-wide →
provide over 0.5 M additional barrels of oil / day

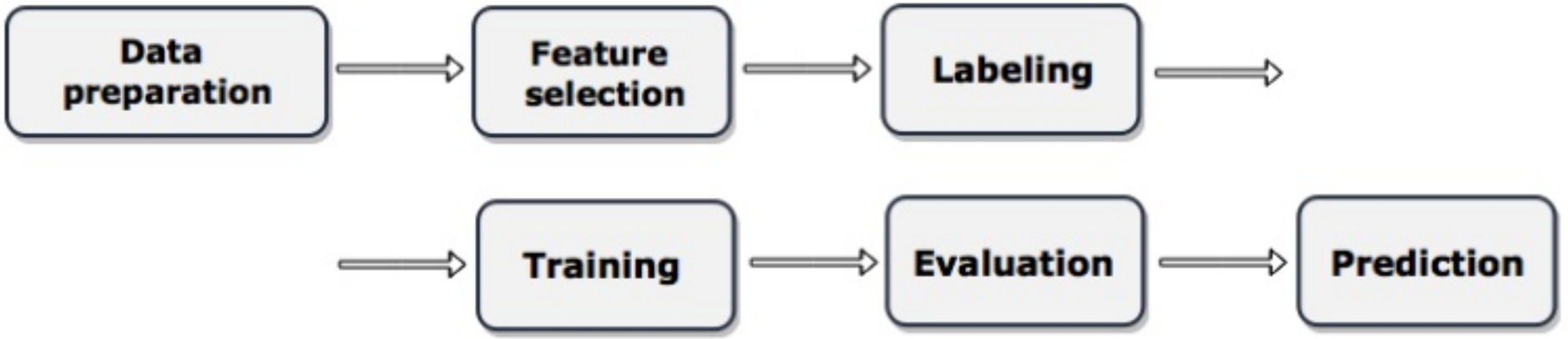
2 M oil wells in operation worldwide
1 M wells use some types of artificial lift
750,000 of the lifted wells use sucker-rod pumps

Problem: road pump failures (surface, tubing, down-hole) could lead ~ two weeks of down time for oil producing

Challenge: prevent/reduce downtime (especially handle pre-failure state to help support engineers service road pump), reduce cost

Dream: build global failure prediction model that could be scaled to all wells worldwide

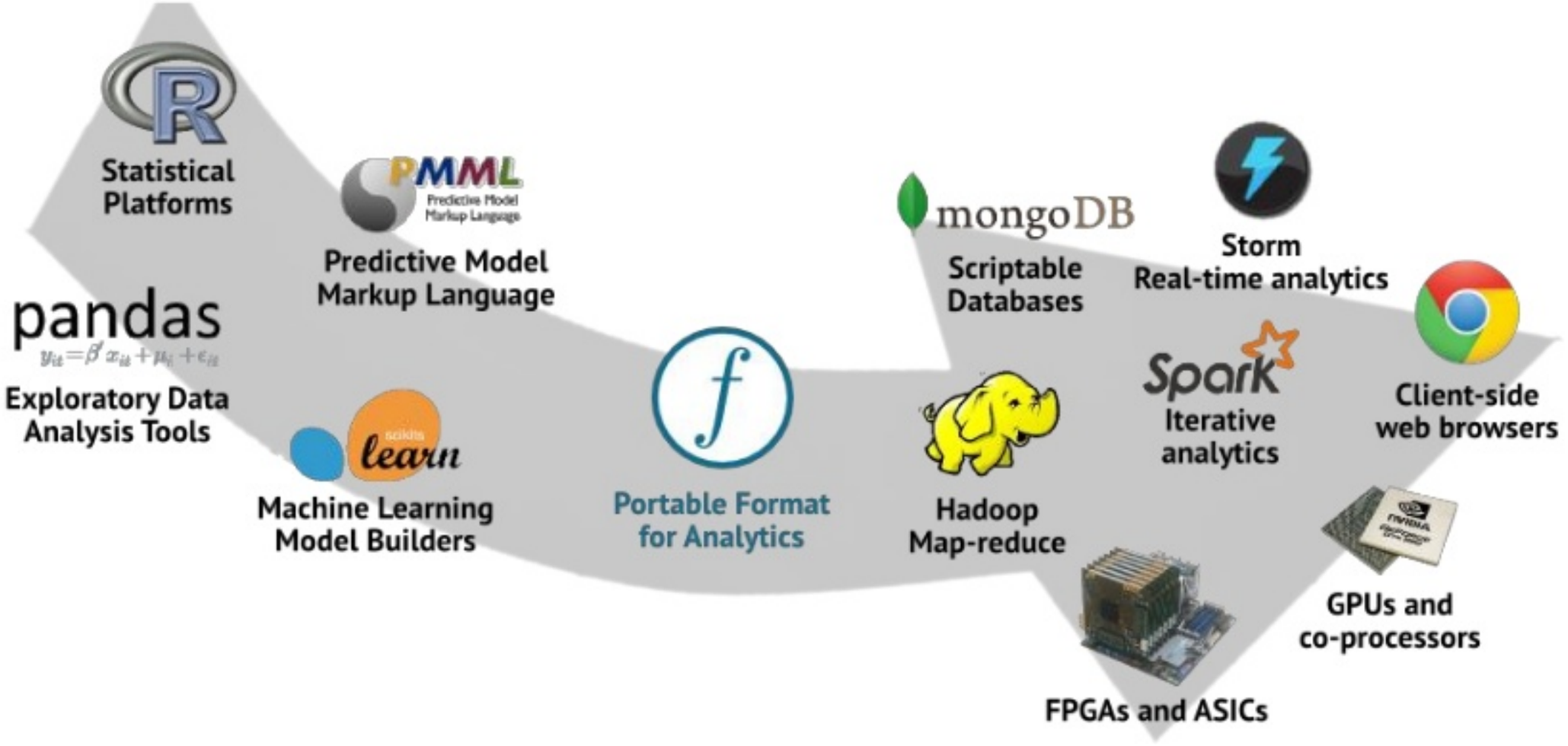
Workflow


Data preparation
Feature selection
Labeling
Training
Evaluation
Prediction

Training

- Logistic regression
- SVM
- Naive Bayes
- Decision Tree

Technical debt



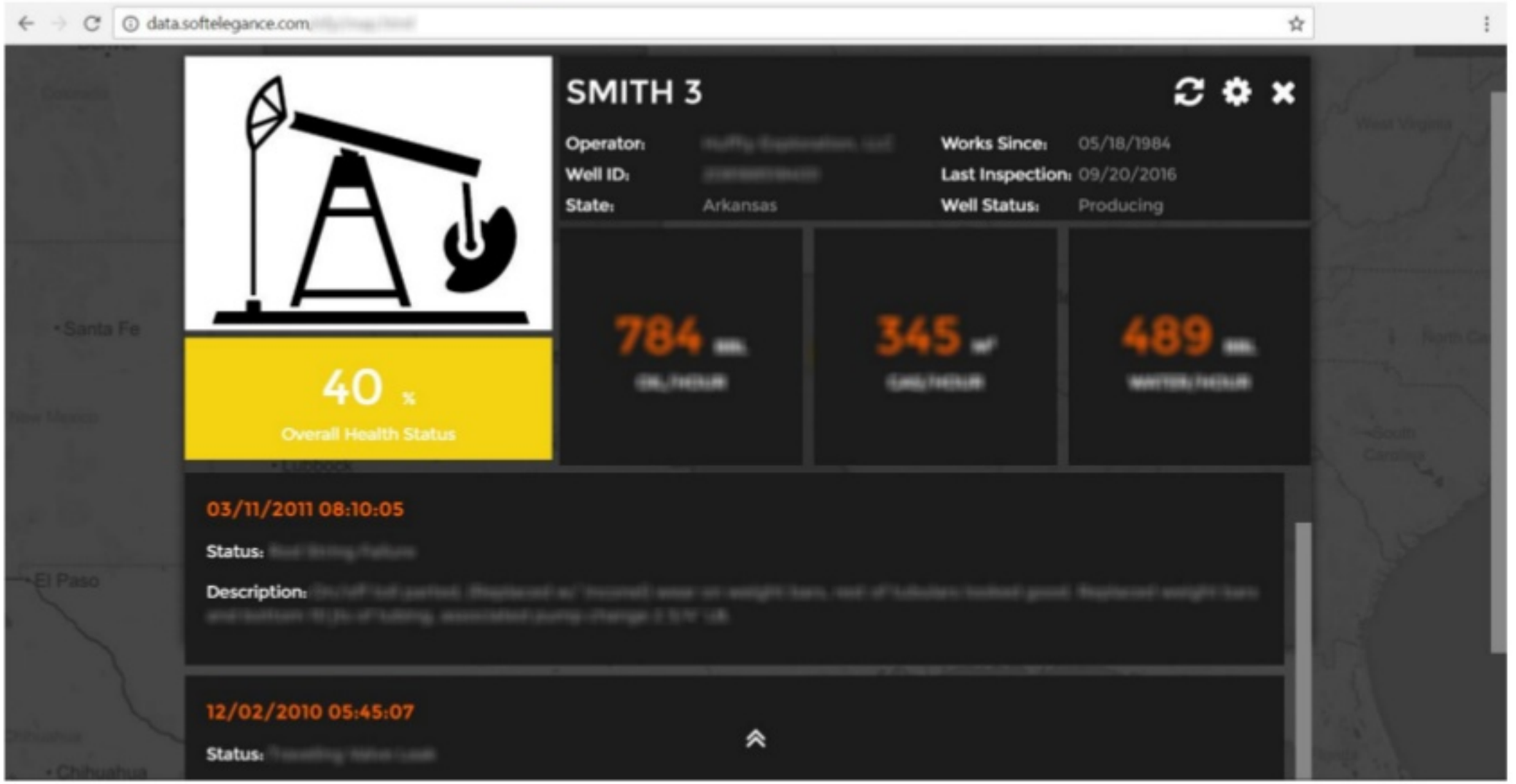
Spark streaming - data prediction at runtime

```
//at driver - load model to broadcast variable
val alSrpPredictionModel = {
  val model = *Model*.load("/datalake/models/al_srp/")
  ssc.sparkContext.broadcast(model)
}

//at client – make predictions
val predictions = alSrpPredictionModel.value.transform(inputDataset)
predictions.select("id", "time", "prediction")
  .collect()
  .foreach { case Row(meterId: String, time: Long, prediction: Double) =>

    //failure
    if (prediction != NORMAL) {
      session.execute(s"INSERT INTO meters.failures (time, meter_id, type)
                                                    VALUES ($time, '$meterId', $prediction)")
      //send notification about failure
    }
  }
```

! Scheduling delay

! Processing time

Intelligent Assistant



Conclusion

- digitalization collection intelligent analysis => better insight

- from artificial intelligent in Oil to Smart Cities, etc.

- is Spark a “Swiss knife” for Data Lakes? (SQL, batch, MLlib, streaming)

References

1. Predicting Failures from Oilfield Sensor Data using Time Series Shapelets - http://www-scf.usc.edu/~chelmis/pubs/spe14.pdf

2. Failure Prediction for Rod Pump Artificial Lift Systems - http://cdm15799.contentdm.oclc.org/cdm/ref/collection/p15799coll3/id/323256

3. Using ZigBee for Wireless Remote Monitoring and Control - http://www.ethanpublishing.com/uploadfile/2015/0608/20150608015338765.pdf

COLLECT. ANALYZE. INSIGHT.

data.softelegance.com
data@softelegance.com