Sparkling Water 2.0

Michal Malohlava

michal@h2o.ai

H2O Open Tour 2016,
Dallas, TX
October 26, 2016

SPARKLING
WATER

H2O + Spark

=

Sparkling Water

Sparkling Water provides

Transparent integration of H2O machine learning platform with Spark ecosystem

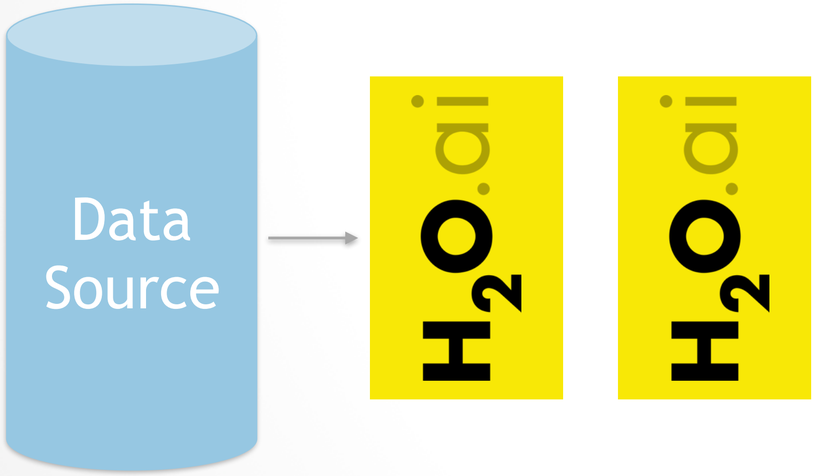
Transparent use of **H2O data structures** (H2OFrame) and **algorithms** with **Spark API**

Excels in existing Spark workflows requiring advanced Machine Learning algorithms

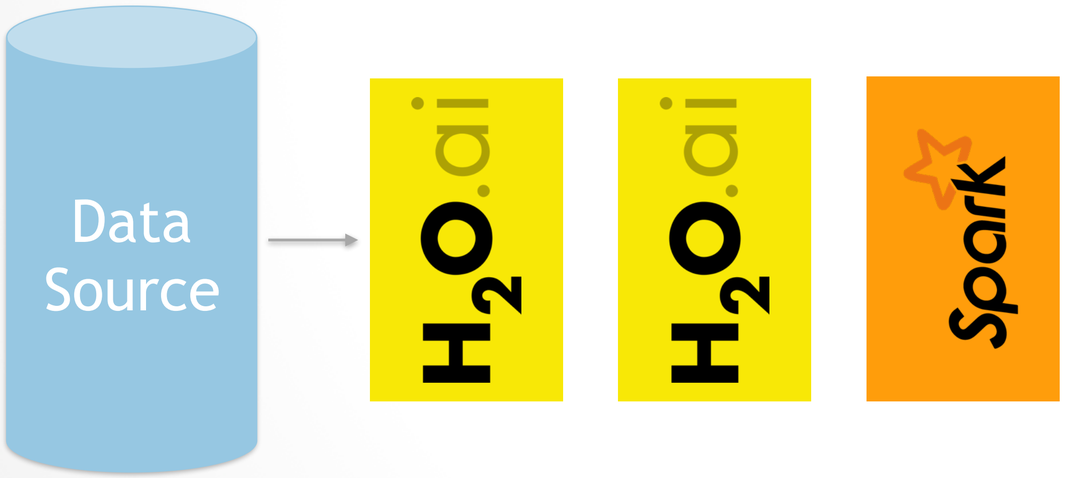
Functionality missing in H2O can be replaced by Spark and vice versa

Data Munging & Modeling

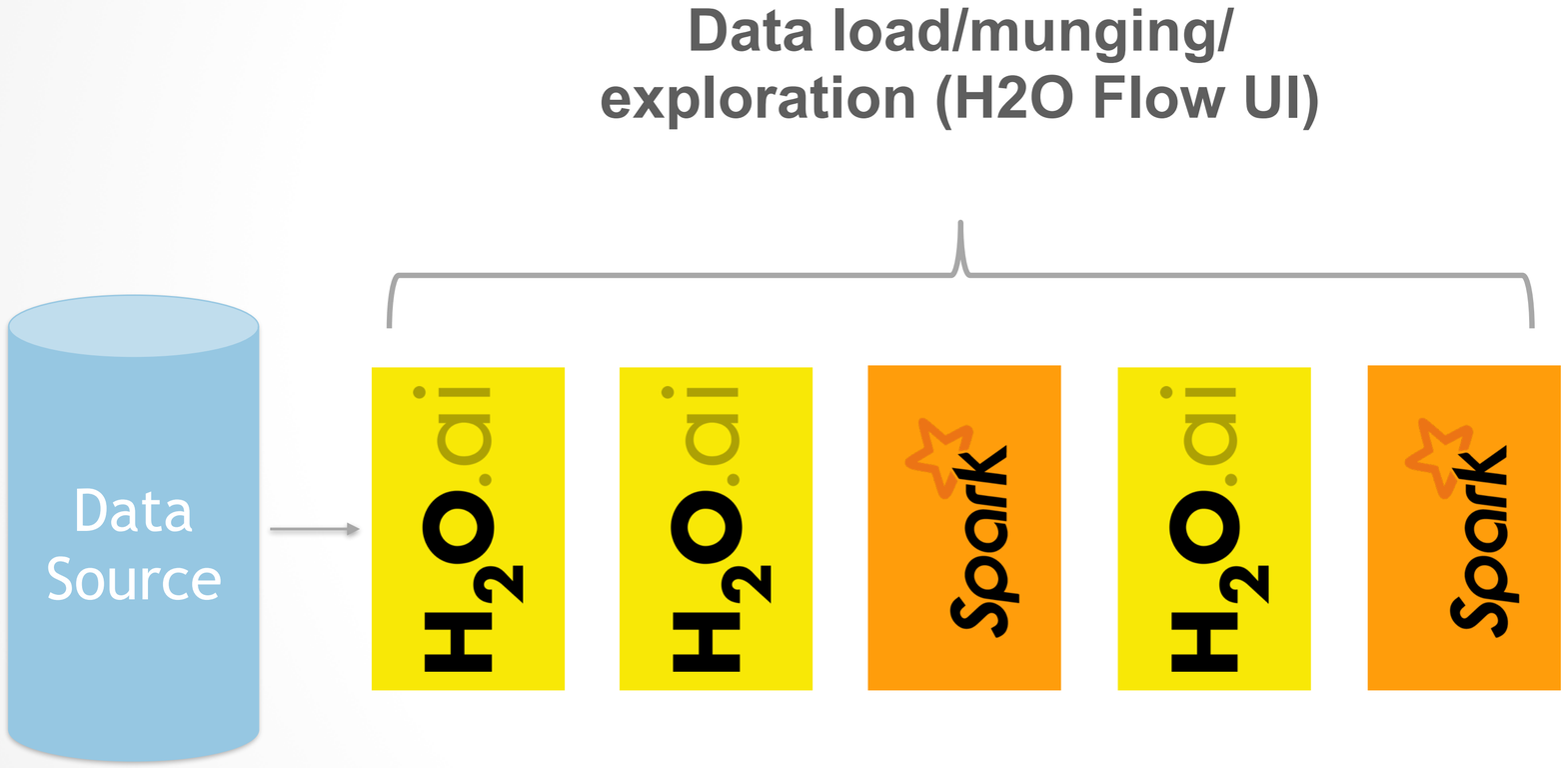
Data Munging & Modeling



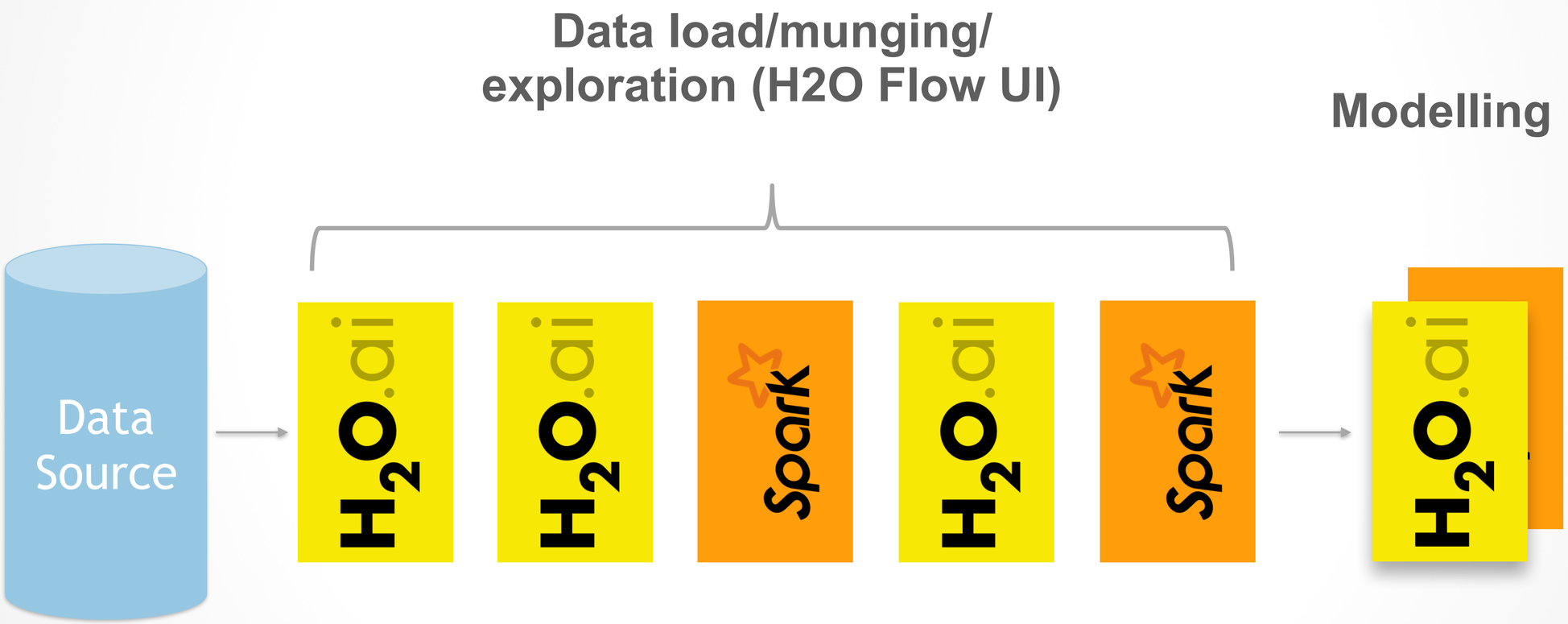
Data Munging & Modeling



Data Munging & Modeling



Data Munging & Modeling

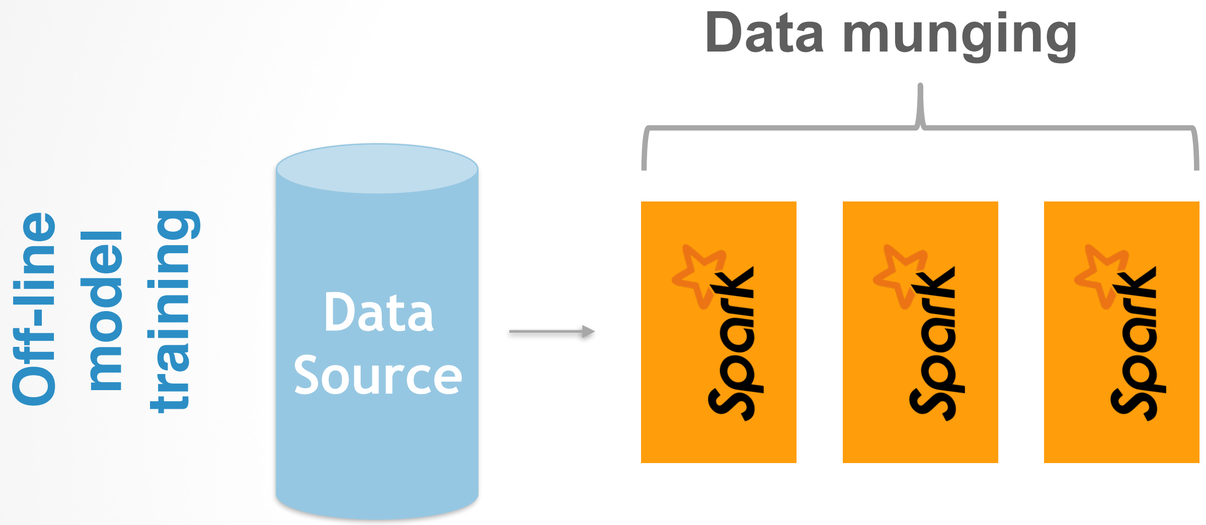


Stream processing

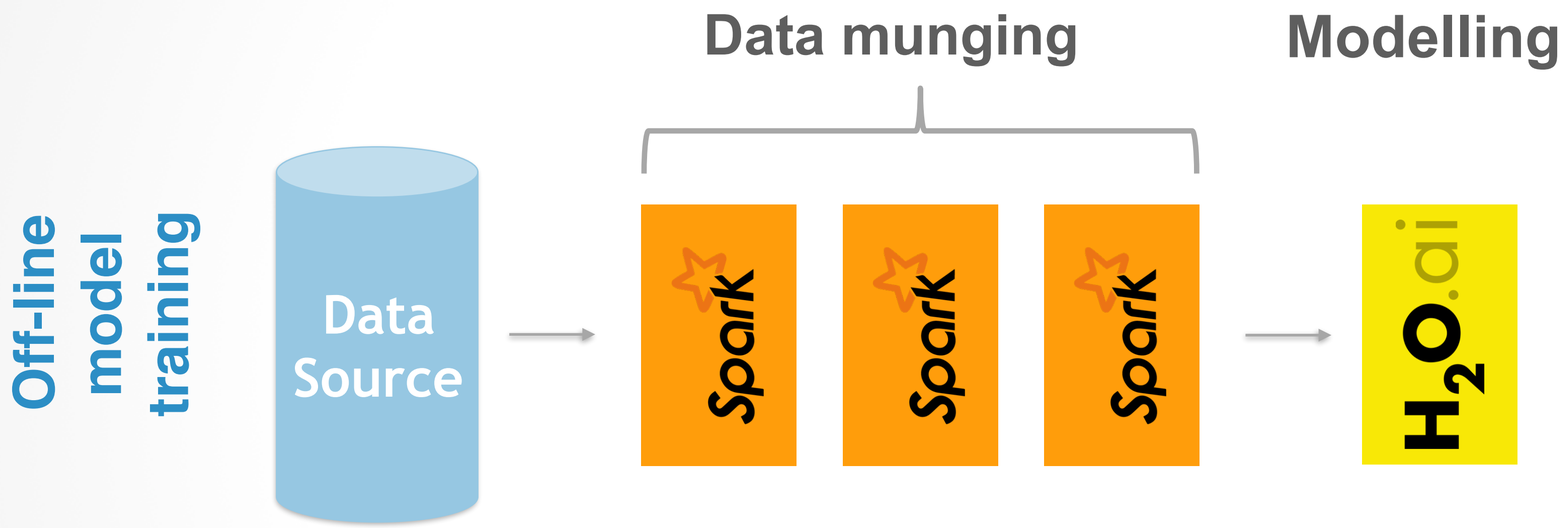
Stream processing

Off-line model training

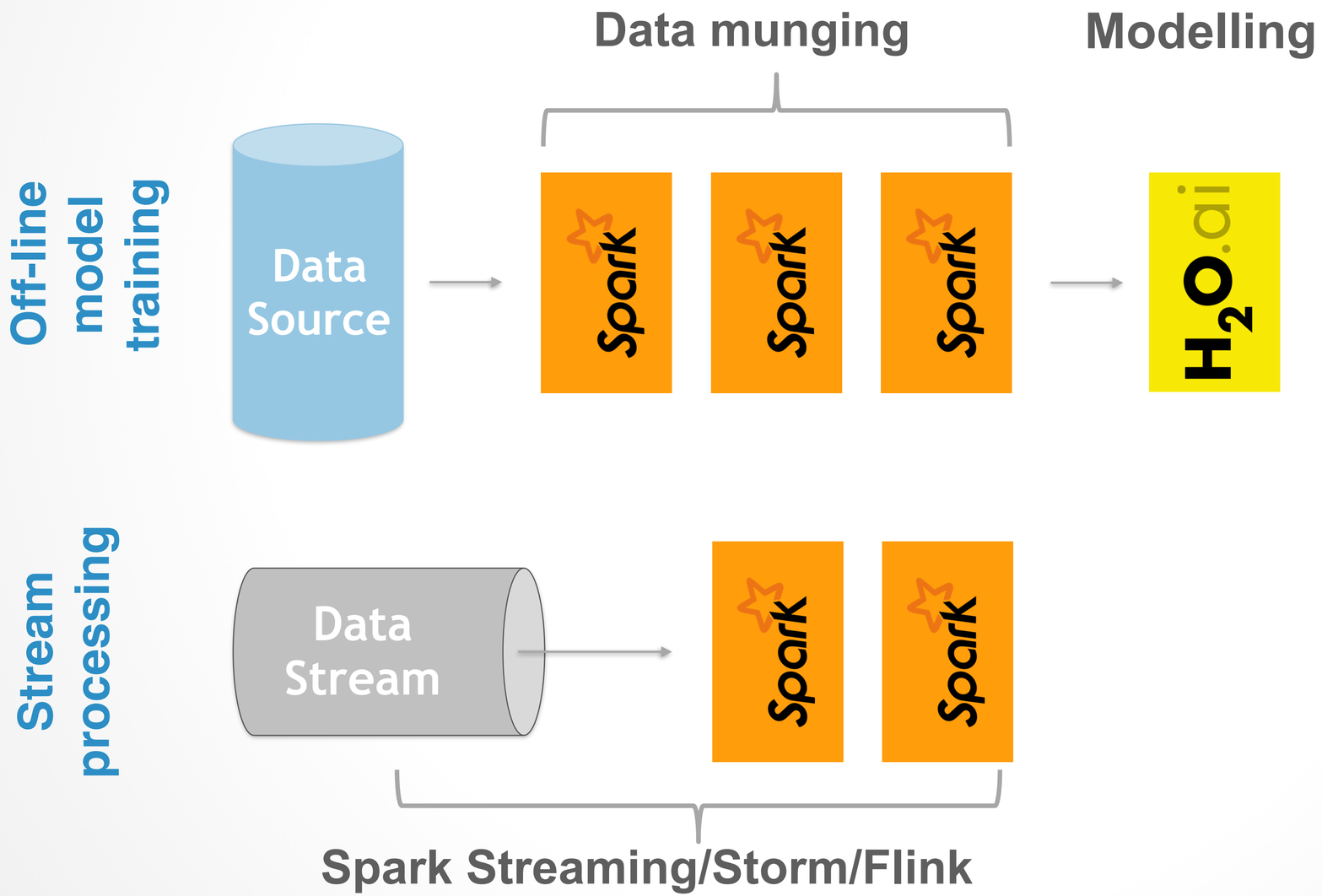
Stream processing



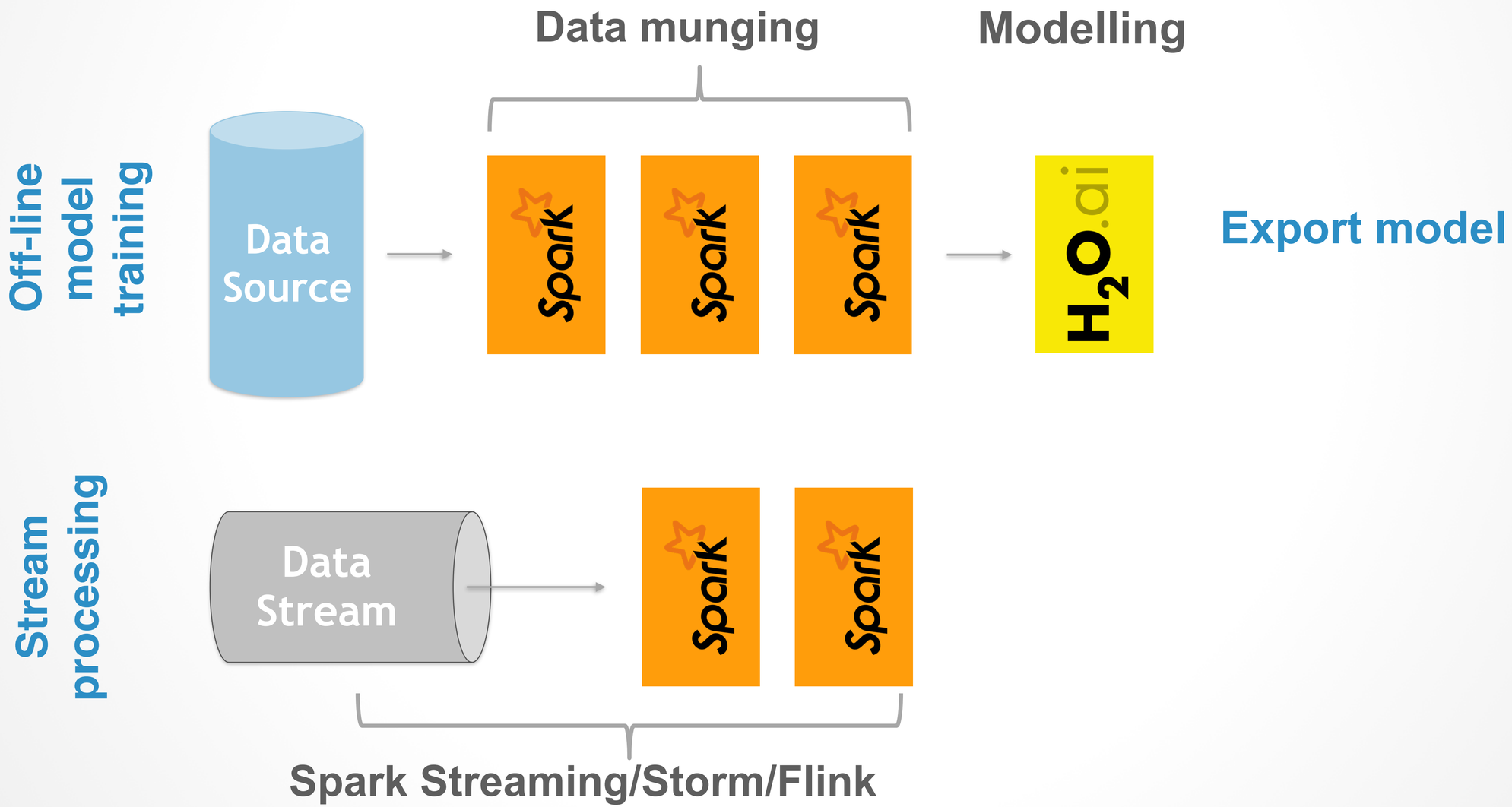
Stream processing



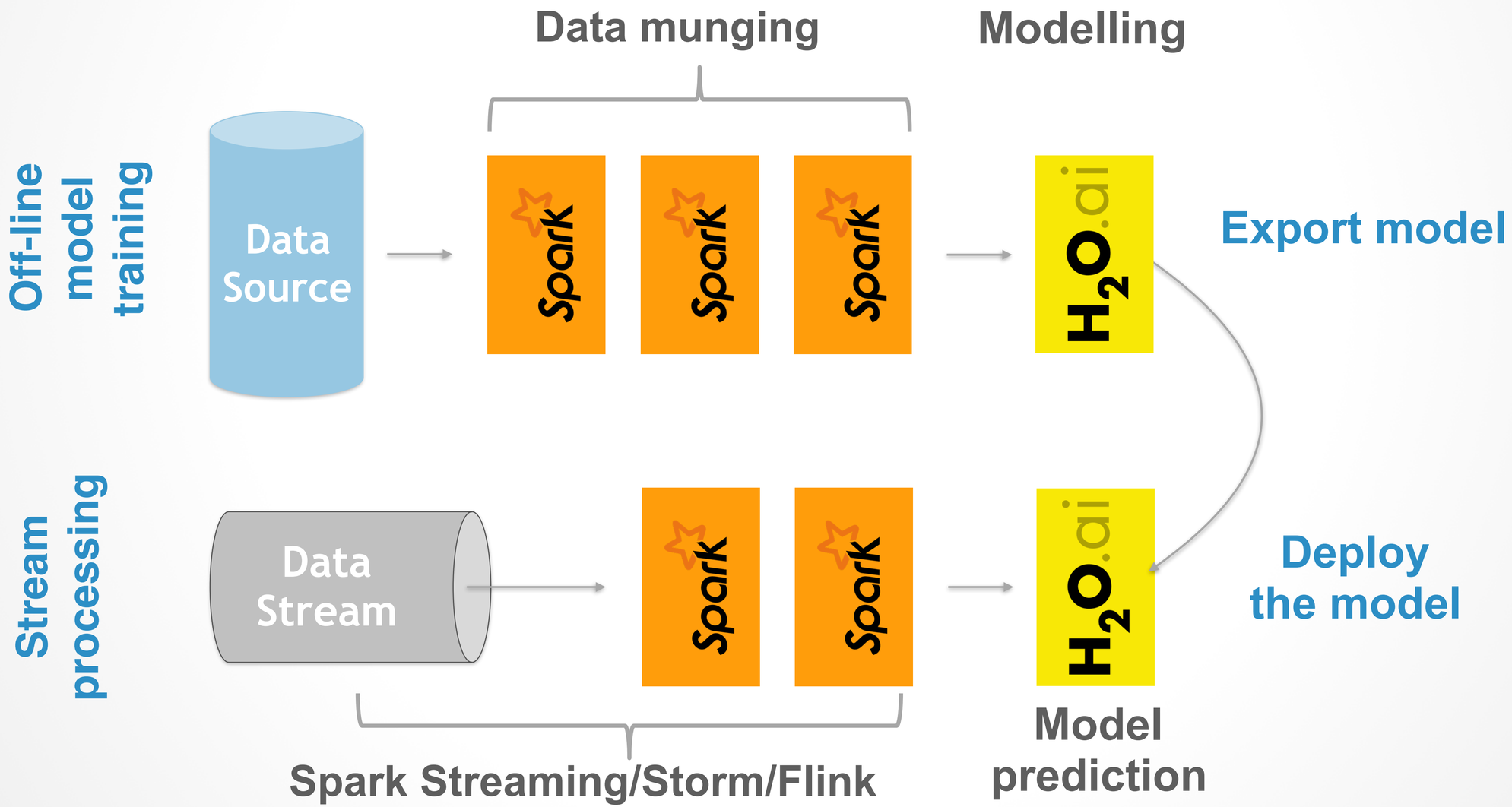
Stream processing



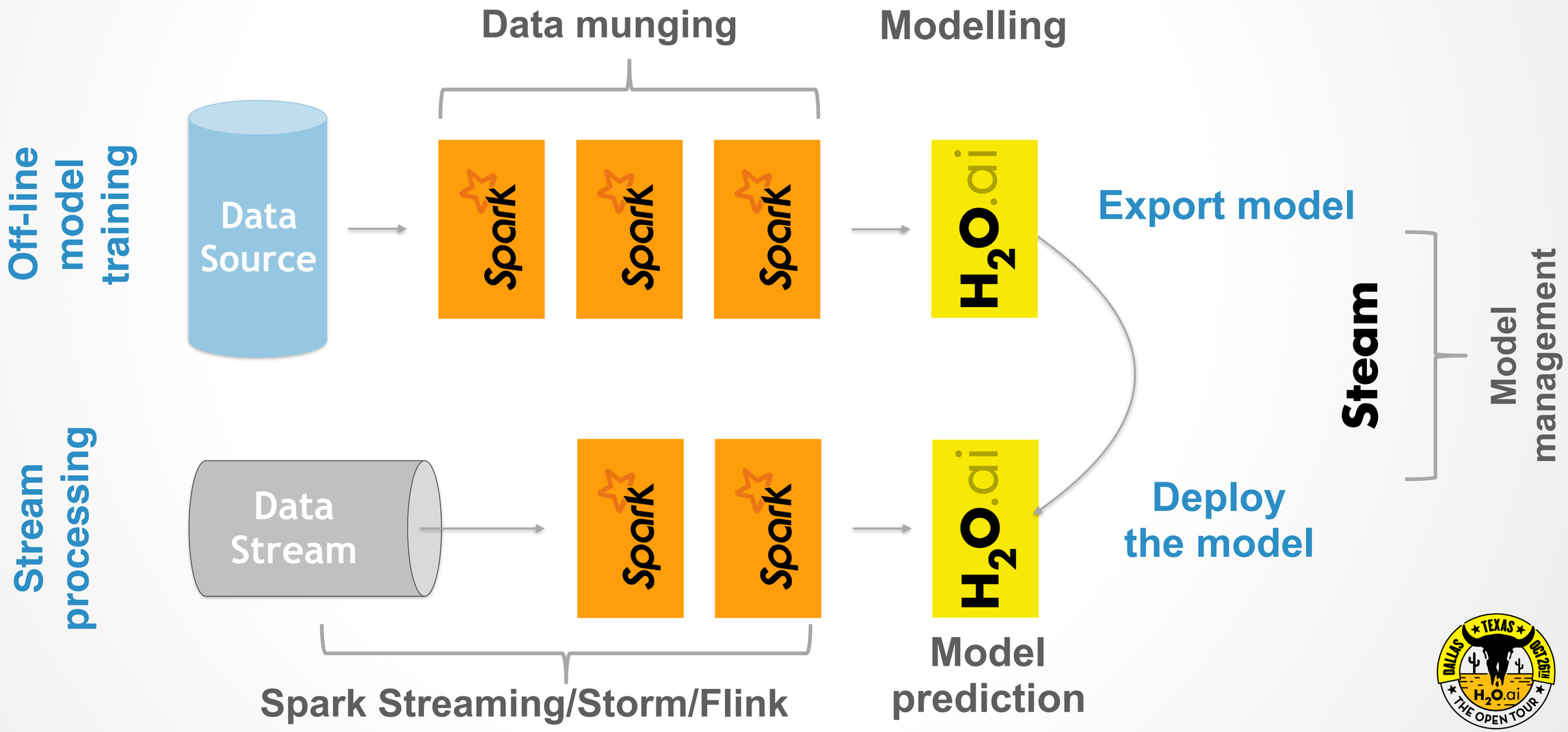
Stream processing



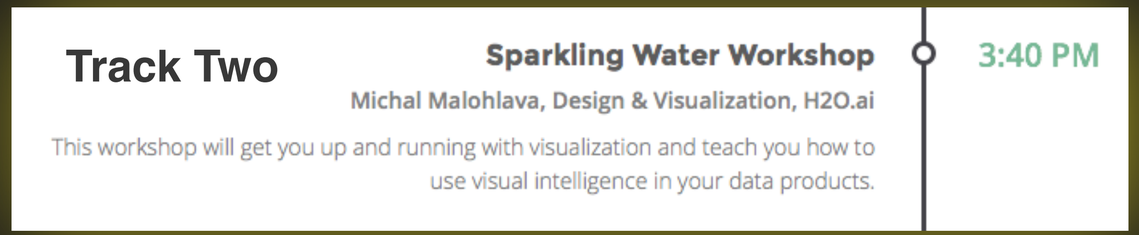
Stream processing



Stream processing



Interested in more details?



What's new in 2.0 ?

Releasing Schema

H2O.ai

Spark 1.5 + Latest ⇒ 1.5.x SPARKLING WATER

Back porting new features

Spark 1.6 + Latest ⇒ 1.6.x SPARKLING WATER

Back porting new features

Spark 2.0 + Latest ⇒ 2.0.x SPARKLING WATER

Spark 2.0 API Support

✓ Support of Spark **Session** based API

✓ More aggressive **optimization** during Spark DataFrame to H2OFrame transformation

✓ **Scala 2.11** support

- useful if you are building Spark based applications

Datasets Support

✓ Support of new Spark data structure **DataSet**

- **DataSet: fusion of strongly typed RDD[X] and DataFrame query API**

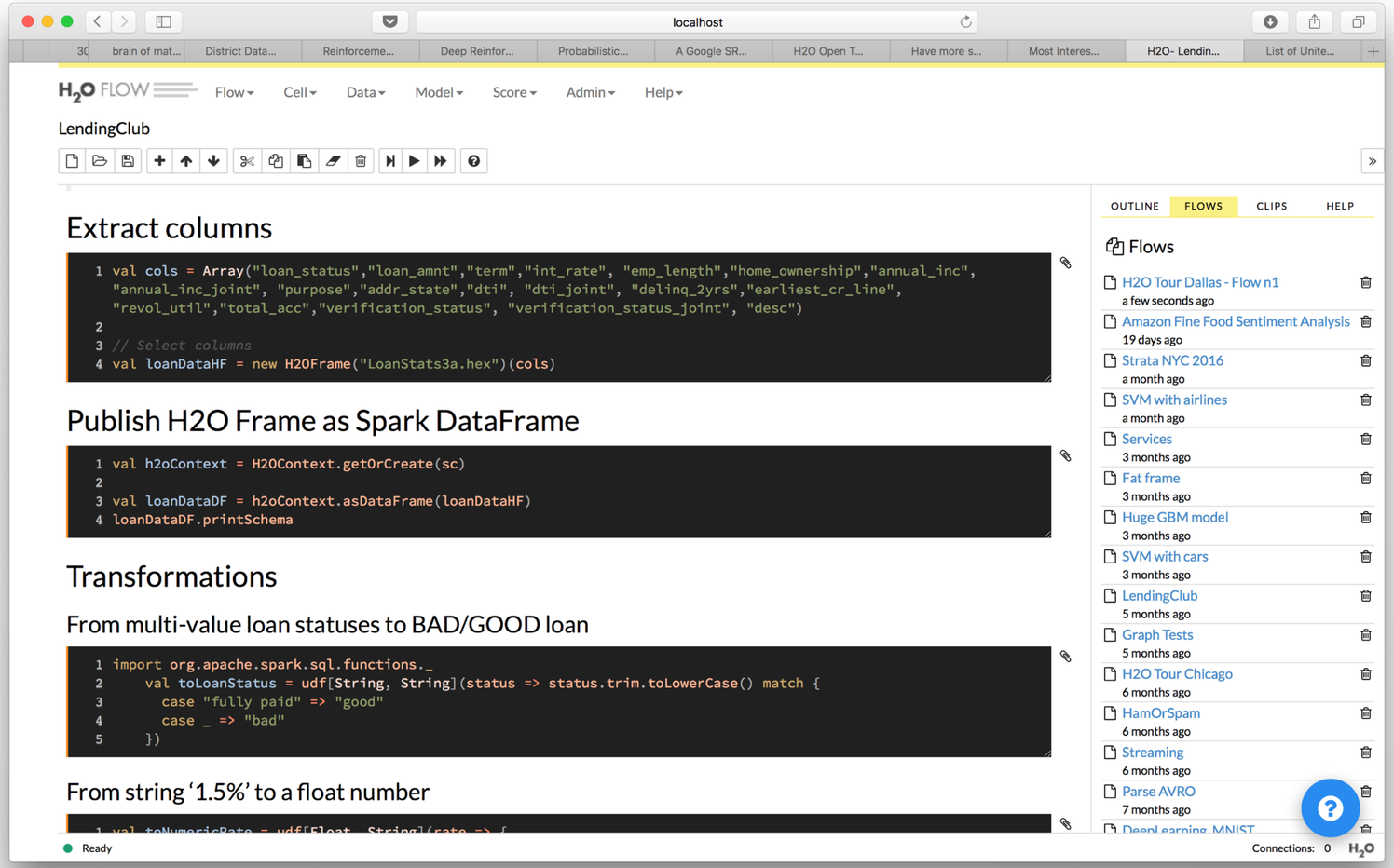
```
case class SamplePerson(name: String, age: Int, email: String)

val dataSource = ...

val samplePeople: List[SamplePerson] = dataSource map SamplePerson.tupled
// Create Dataset
val df: Dataset[SamplePerson] = sqlContext.createDataset(samplePeople)

// Publish Dataset as H2OFrame
val hf: H2OFrame = h2oContext.asH2OFrame(df)
```

Available in Sparkling Water 1.6!

Scala Code from Browser

✓We can now execute Scala code directly from H2O Flow UI!

Available in Sparkling Water 1.6!



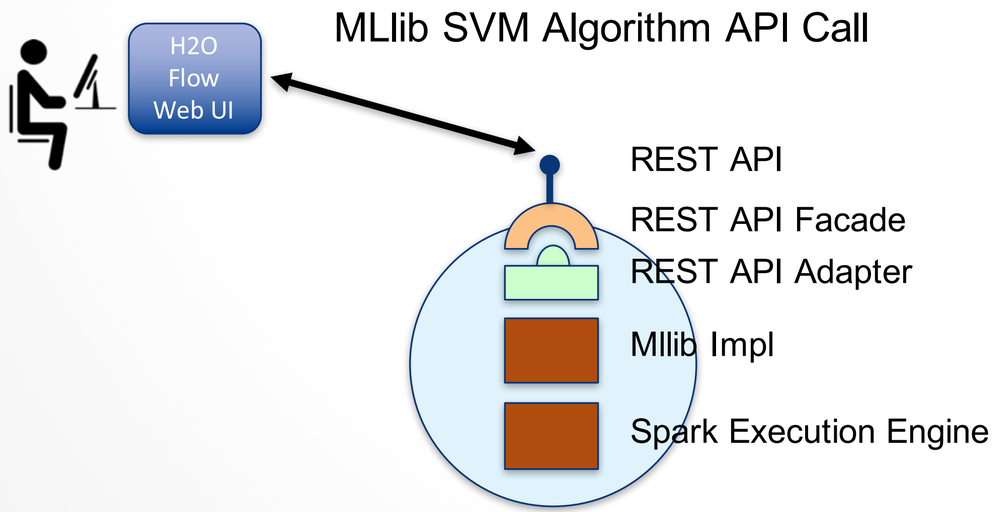
H2O.ai

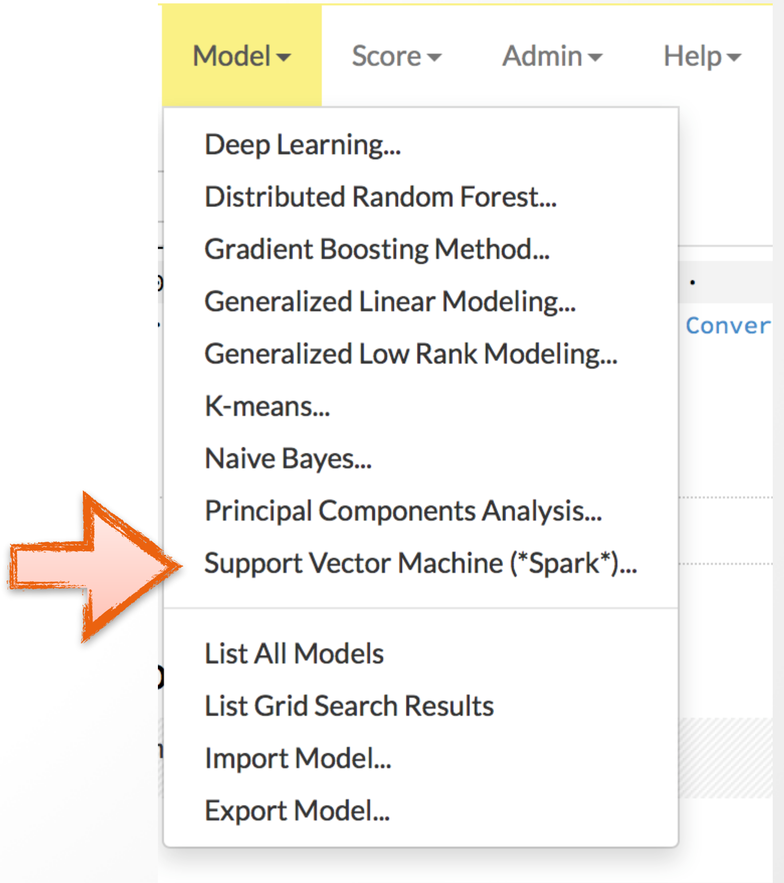
DALLAS TEXAS OCT 26TH H2O.ai THE OPEN TOUR

Spark Models in Flow UI

Spark algorithms can be called from H2O Flow UI

- ✓ Quick visual feedback
- ✓ Model performance reports
- ✓ Model export (as code)





H2O.ai

RSparkling

- ✓ A New Member of Sparkling Water Ecosystem

sparklyr

RSparkling

✓ A New Member of Sparkling Water Ecosystem

Scala: Sparkling Water

Spark

```
val sc =
SparkContext.getOrCreate(...)

val df = sc.parallelize(1 to 10).toDF
```

sparklyr

```
val h2oContext =
H2OContext.getOrCreate(sc)

val hf = h2oContext.asH2OFrame(df)
```

H2O.ai

RSparkling

✓ A New Member of Sparkling Water Ecosystem

Scala: Sparkling Water	Python: PySparkling Water
Spark	PySpark
val **sc** = **SparkContext**.getOrCreate(…)	**sc** = **SparkContext**(…)
val df = sc.parallelize(1 to 10).toDF	df = sc.parallelize(range(1,11)) .toDF(“int”)
val h2oContext = **H2OContext**.getOrCreate(**sc**)	h2o_context = **H2OContext**.getOrCreate(**sc**)
val hf = h2oContext.**asH2OFrame**(df)	hf = h2o_context.**as_h2o_frame**(df)

sparklyr

H2O.ai

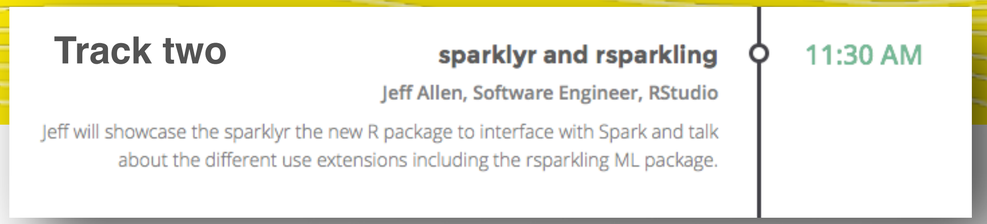
DALLAS TEXAS OCT 2017 H2O.ai THE OPEN TOUR

RSparkling

✓ A New Member of Sparkling Water Ecosystem

Scala: Sparkling Water	Python: PySparkling Water	R: RSparkling Water
Spark val **sc** = **SparkContext**.getOrCreate(…) val df = sc.parallelize(1 to 10).toDF	PySpark **sc** = **SparkContext**(…) df = sc.parallelize(range(1,11)) .toDF(“int”)	sparklyr **sc** <- **spark_connect**(…) tbl <- data_frame(c(1:10)) df <- copy_to(**sc**, tbl)
val h2oContext = **H2OContext**.getOrCreate(**sc**) val hf = h2oContext.**asH2OFrame**(df)	h2o_context = **H2OContext**.getOrCreate(**sc**) hf = h2o_context.**as_h2o_frame**(df)	hc <- **h2o_context**(**sc**) hf <- **as_h2o_frame**(**sc**, df)

RSparkling



✓ A New Member of Sparkling Water Ecosystem

Scala: Sparkling Water

Spark

```
val sc =
SparkContext.getOrCreate(...)

val df = sc.parallelize(1 to 10).toDF
```

```
val h2oContext =
H2OContext.getOrCreate(sc)

val hf = h2oContext.asH2OFrame(df)
```

Python: PySparkling Water

PySpark

```
sc = SparkContext(...)

df = sc.parallelize(range(1,11))
       .toDF("int")
```

```
h2o_context =
H2OContext.getOrCreate(sc)

hf = h2o_context.as_h2o_frame(df)
```

R: RSparkling Water

sparklyr

```
sc <- spark_connect(...)

tbl <- data_frame(c(1:10))
df <- copy_to(sc, tbl)
```

```
hc <- h2o_context(sc)

hf <- as_h2o_frame(sc, df)
```

Available in Sparkling Water 1.6!

H2O.ai

Coming soon...

Bringing both platforms together

API Convergence

- H2O model builders compatibility with Spark pipelines
  - Scala/Python level
- Reuse existing Spark functionality
  - e.g., applying Spark UDFs directly on H2OFrame
- Heterogenous (Spark/H2O) model ensembles/pipelines
- More Spark Algorithms in UI
  - e.g. ALS

Coming soon!

More Tooling From H2O Ecosystem

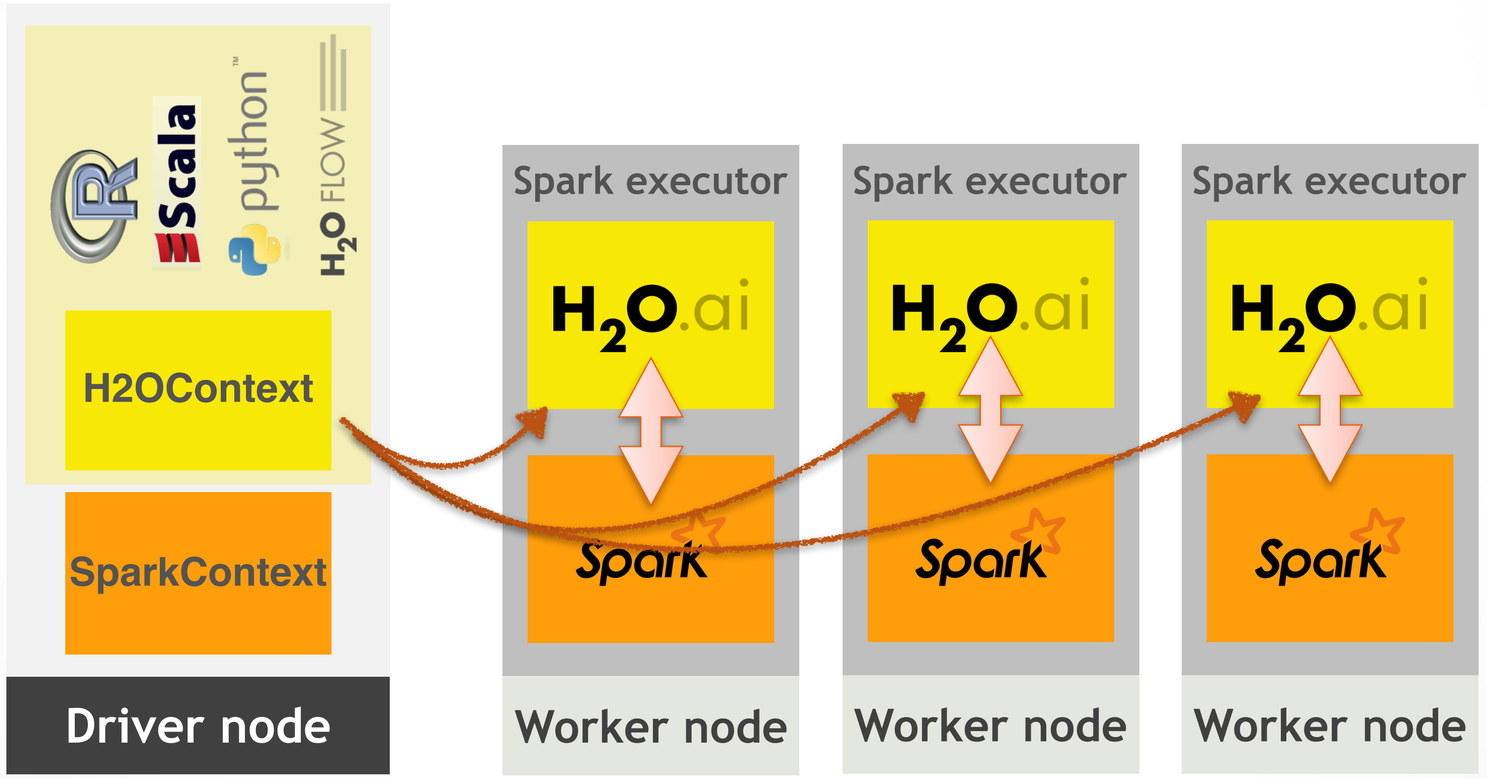
- Exposing **DeepWater** Algorithms

- Integration with **Steam**
  - Steam as Sparkling Water **launcher**
  - Steam as **model manager**

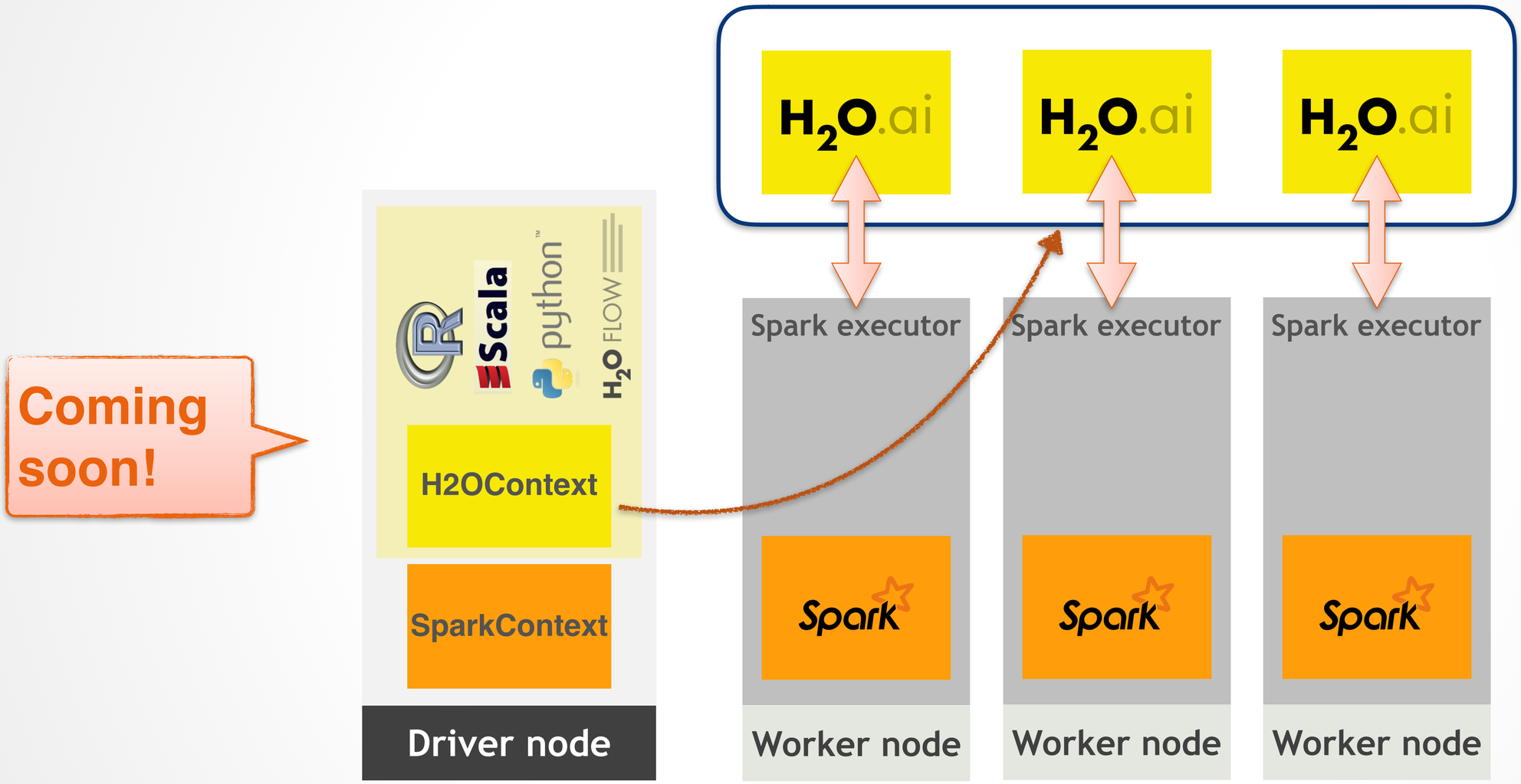
Coming soon!

High Availability

NOW



High Availability



Thank you!

Sparkling Water is
open-source
ML application platform
combining
power of Spark and H2O

Learn more at h2o.ai
Follow us at @h2oai