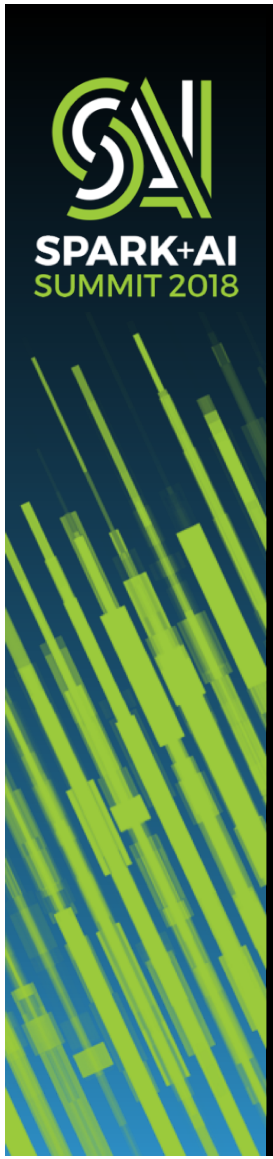


Scaling Machine Learning at Booking.com

Brammert Ottens, Booking.com
Luca Falsina, Booking.com

#ML8SAIS

Booking.com

Who are we?

Brammert Ottens

Senior Data Scientist

Luca Falsina

Senior Developer

Booking.com

- One of the **largest** travel e-commerce companies in the world
- **15.000** employees in 198 offices in 70 countries worldwide
- **133,577** destinations in 228 countries and territories worldwide
- Each day, more than **1,550,000** room nights are reserved on our platform

Outline

- Datascience @Booking.com
- The Machine Learning (ML) pipeline
- Offline ML
- Online model serving
- Model & features monitor and discoverability
- Future directions

Outline

- **Datascience @Booking.com**
- The Machine Learning (ML) pipeline
- Offline ML
- Online model serving
- Model & features monitor and discoverability
- Future directions

Datascience @ Booking.com

- We have a community of over 200 data scientists (DS)
- Each with different needs, expertise and preferred toolkits and methodologies

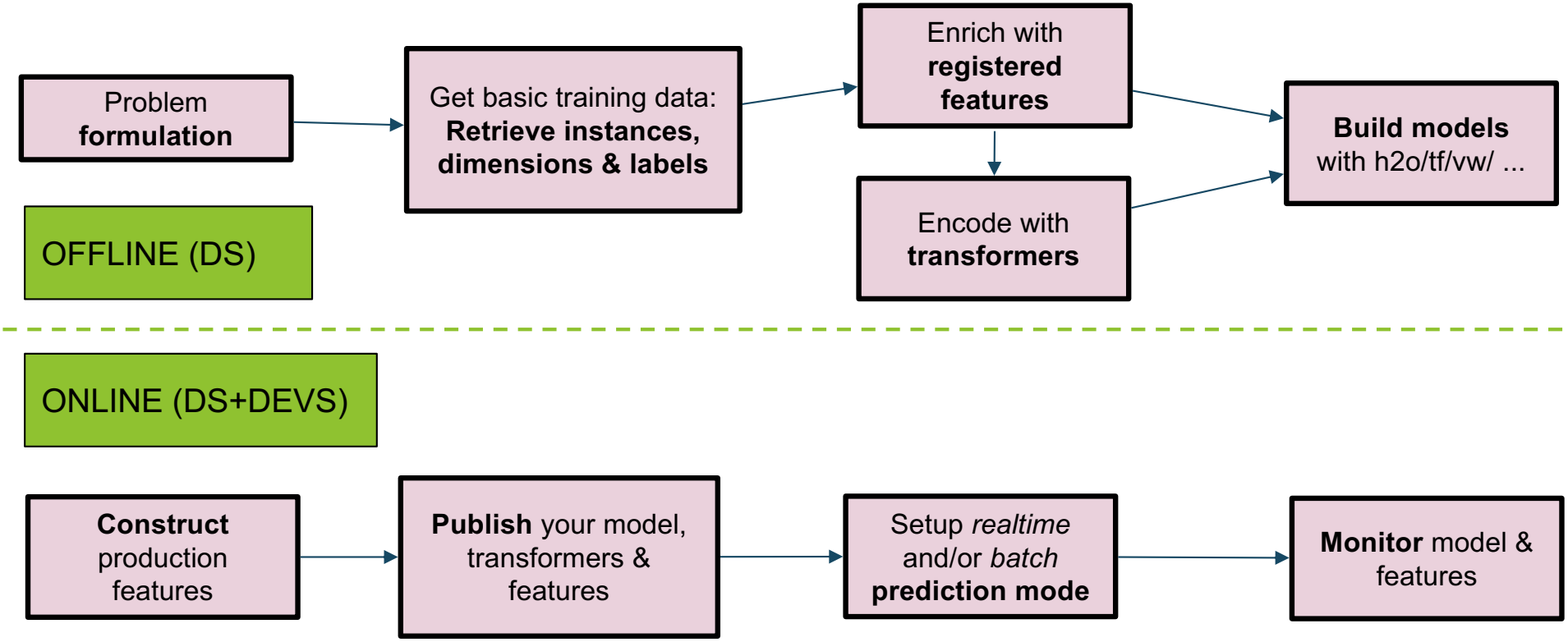
The mission of our track is to help scaling ML across our organization. How?

- Identify **common problems** (i.e. historical feature reconstruction)
- **Implement infra and tooling** to solve them
- **Train, educate and consult** internal users on these solutions

Outline

- Datascience @Booking.com
- **The Machine Learning (ML) pipeline**
- Offline ML
- Online model serving
- Model & features monitor and discoverability
- Future directions

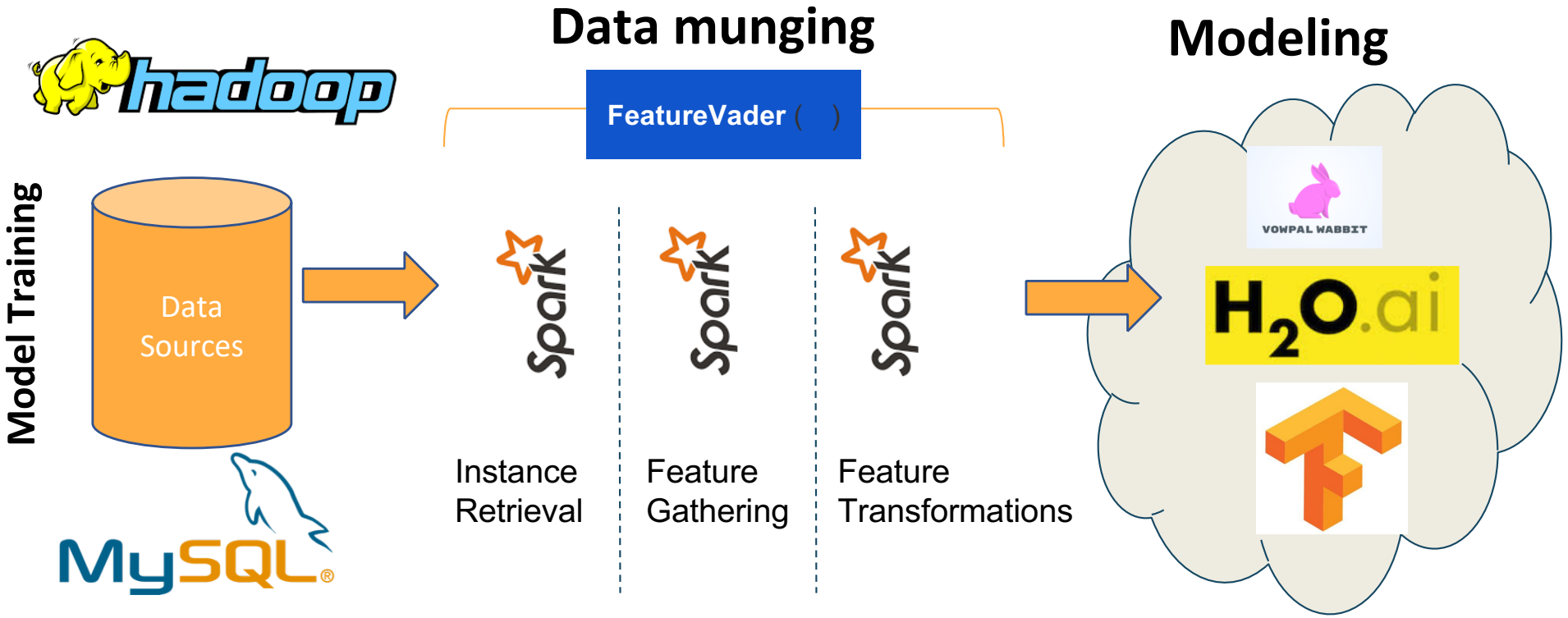
The “general” ML pipeline



Outline

- Datascience @Booking.com
- The Machine Learning (ML) pipeline
- **Offline ML**
- Online model serving
- Model & features monitor and discoverability
- Future directions

The “offline” world



OFFLINE DATA MUNGING

1. Extract relevant data from events
2. Feature engineering
3. Time coordinate matching
4. Scalability

Offline feature engineering

- **FeatureVader** (containing a feature registry):
  - Custom Spark ML Transformer:

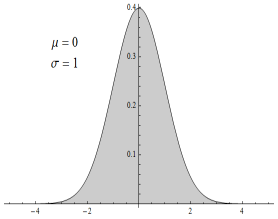
```
labeledInstances = spark.sql("""
            select
                userId
                , time
                , IF(x > 4, 1,0) label
            FROM data""")
fv = FeatureVader()
fv.setTimeStamp("time").setDesiredFeatures(["feature1", "feature2"])
withFeatures = fv.transform(labeledInstances)
```

userId	time	label
1001	25	1
1002	36	0

userId	time	label	feature1	feature2
1001	25	1	213.5	2 kids
1002	36	0	123.7	0 kids

Feature transformations

- Raw features are not enough.
- Transformations greatly increase model performance:



Feature transformations (WIP)

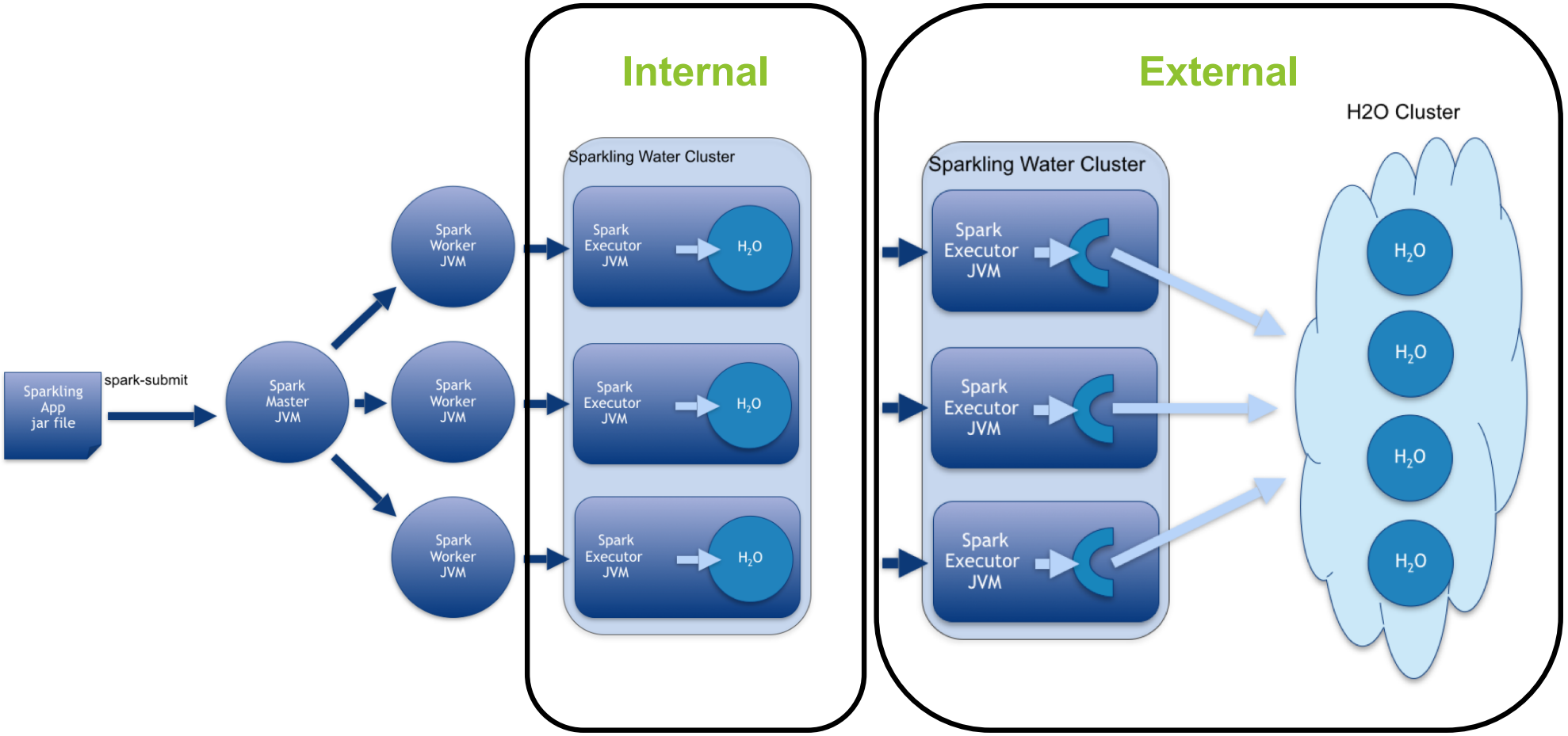
```
testimator = TargetEstimator()
                .setFeature("feature1")
                .setEncodedFeature("feature1_TE")
                .setResponse("label")

pipeline = new PipeLine(stages=[testimator])

pipeline_model = pipeline.fit(train)

pipelineWriter = PipelineWriter(pipelinemodel)

pipelineWriter.writeTo("pipeline_model.bojo")
```

Modeling

Several libraries in use at Booking.com

Modeling (H2O)

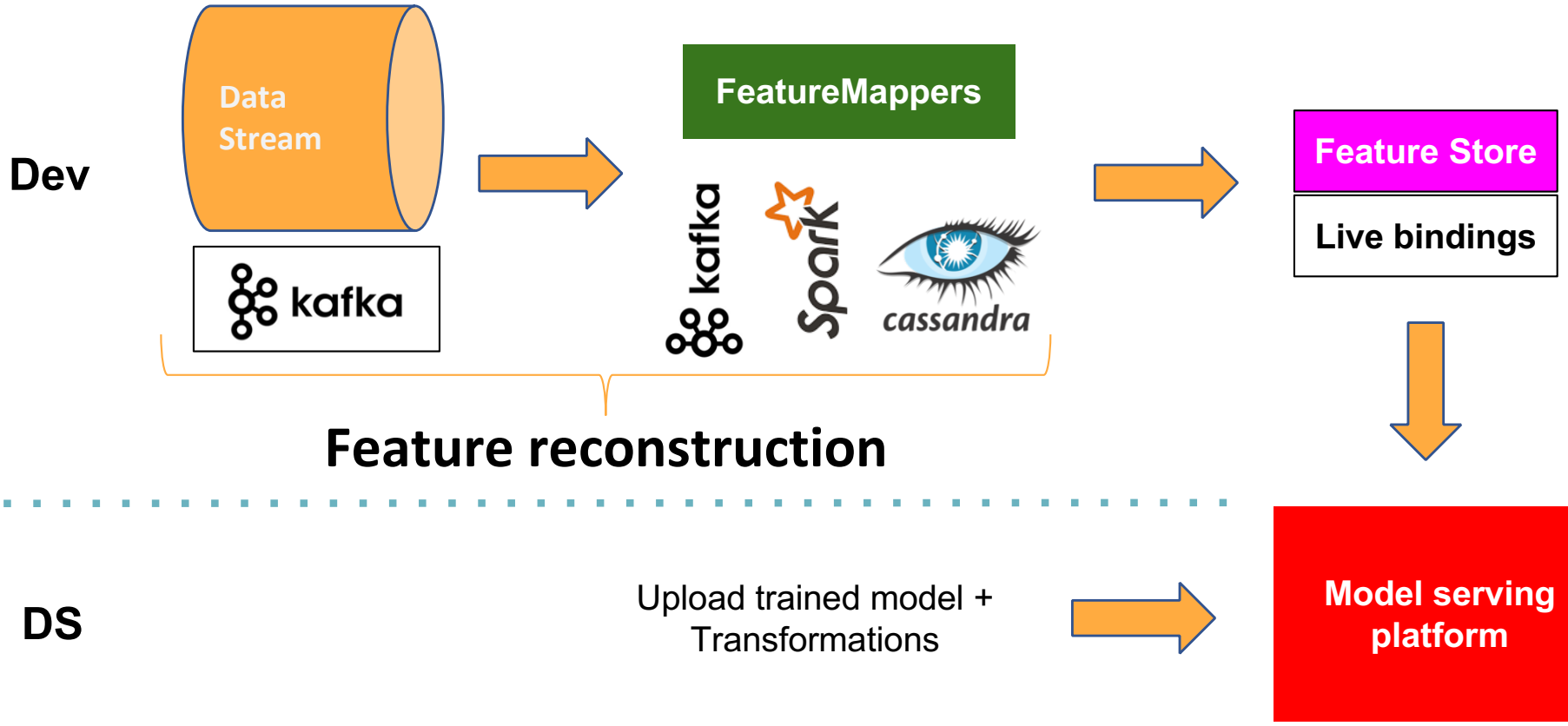
- Sparkling water
  - H2O integration with Spark
    - Best of both worlds: data munging in Spark + model training in H2O
  - Scalable to very large datasets (Billions of rows)
  - Distributed training
  - Internal vs external backend mode


Internal
External
Sparkling Water Cluster
Sparkling Water Cluster
H2O Cluster
Sparkling App jar file
spark-submit
Spark Master JVM
Spark Worker JVM
Spark Worker JVM
Spark Worker JVM
Spark Executor JVM
Spark Executor JVM
Spark Executor JVM
Spark Executor JVM
Spark Executor JVM
Spark Executor JVM
H2O
H2O
H2O
H2O
H2O

Outline

- Datascience @Booking.com
- The Machine Learning (ML) pipeline
- Offline ML
- **Online model serving**
- Model & features monitor and discoverability
- Future directions

The “online” world



ONLINE DATA MUNGING

1. Nearly real-time processing

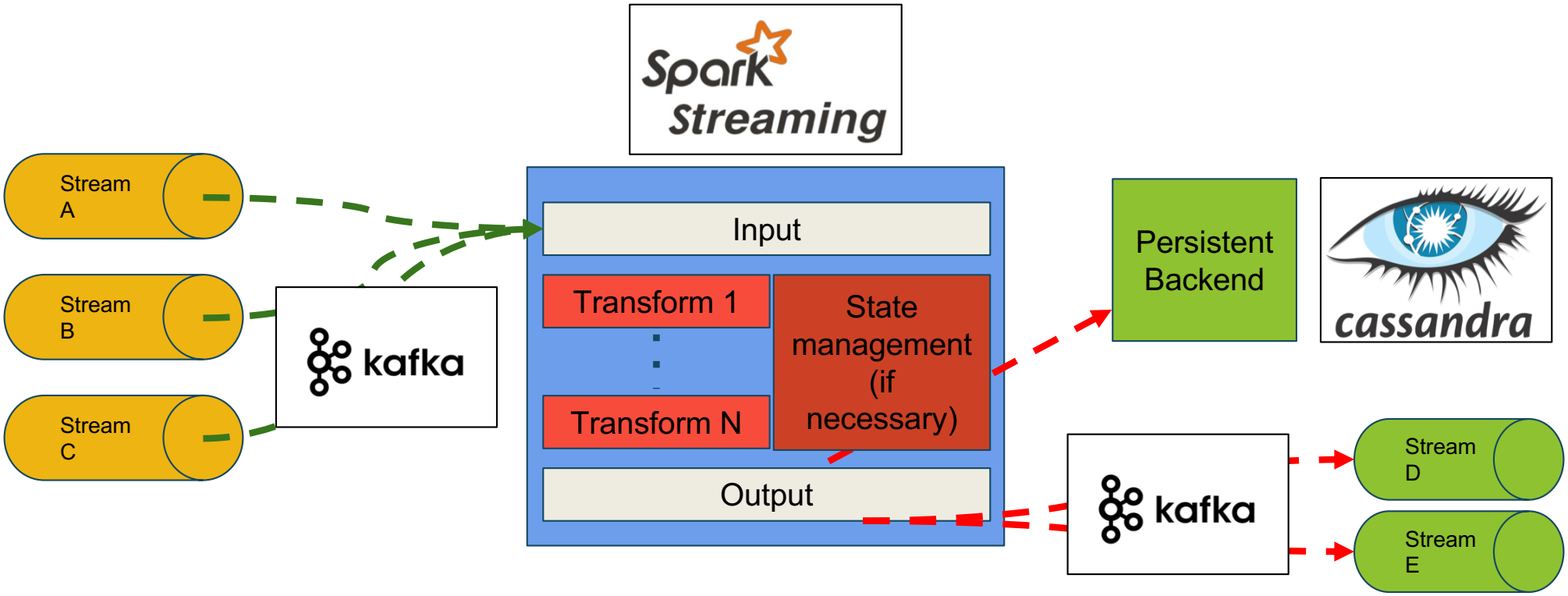
2. State(less/full) transformations

3. Containerized infrastructure

4. Nearly real-time serving

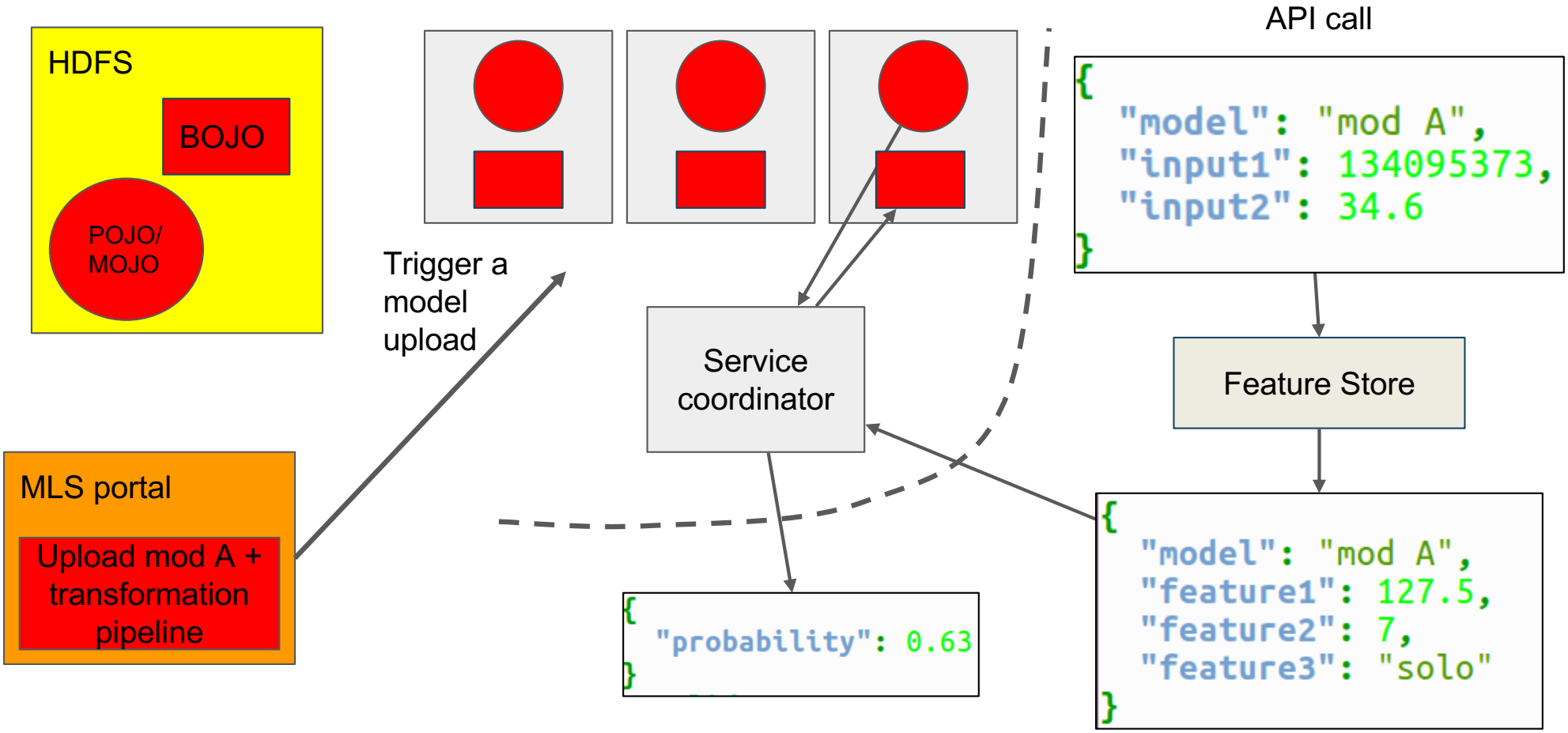
FeatureMappers

Online feature construction



Model serving platform

- Supports several model types (H2O, VW, TF, csv...)
- Allows both **single** and **batch** predictions
- Integration with *live feature bindings*

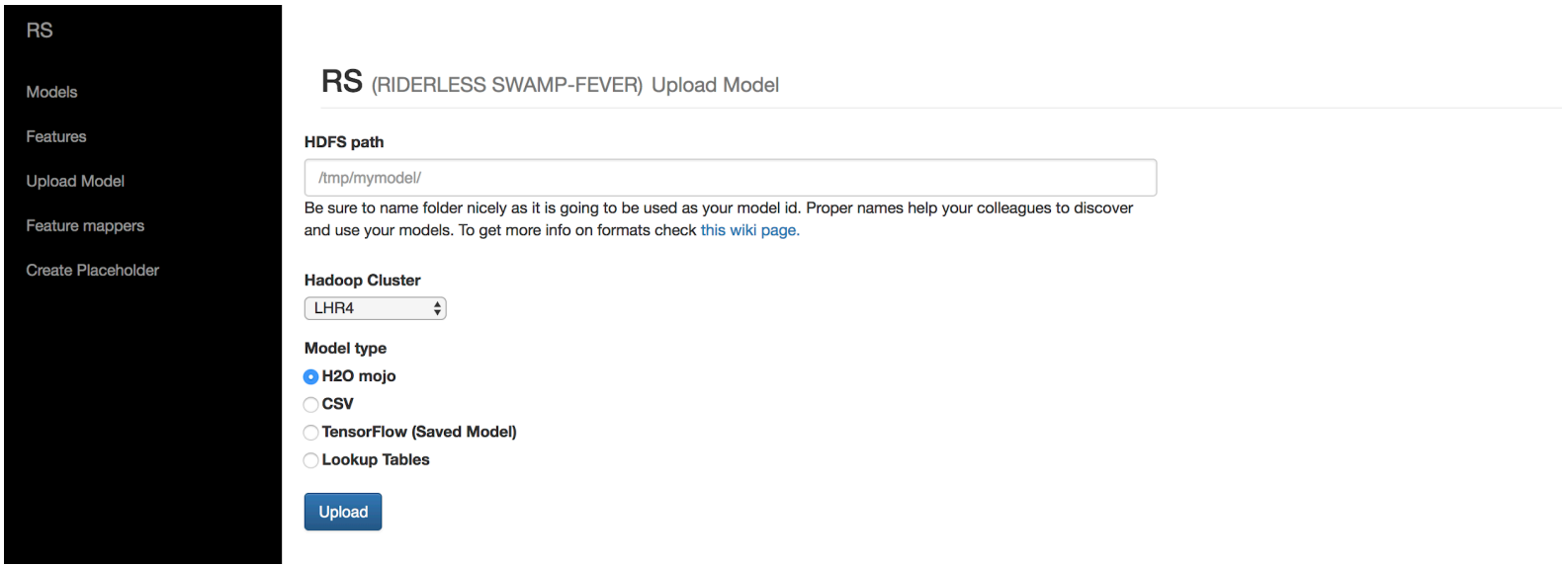


Example: Model upload & inference from frontend hosts

Outline

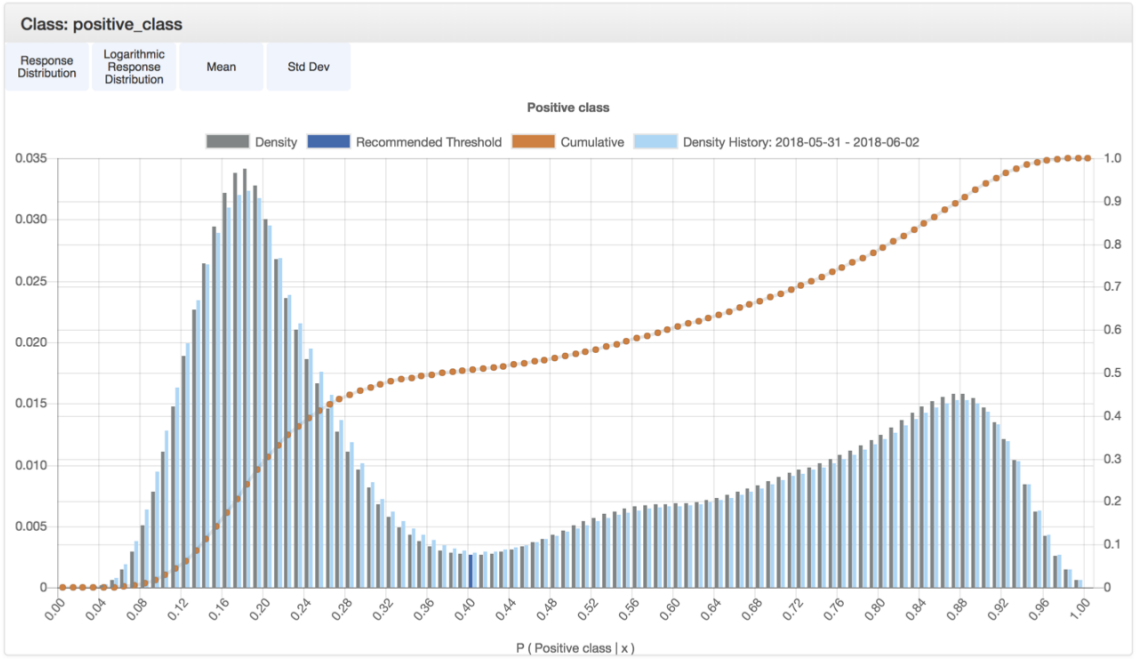
- Datascience @Booking.com
- The Machine Learning (ML) pipeline
- Offline ML
- Online model serving
- **Model & features monitor and discoverability**
- Future directions

Model portal



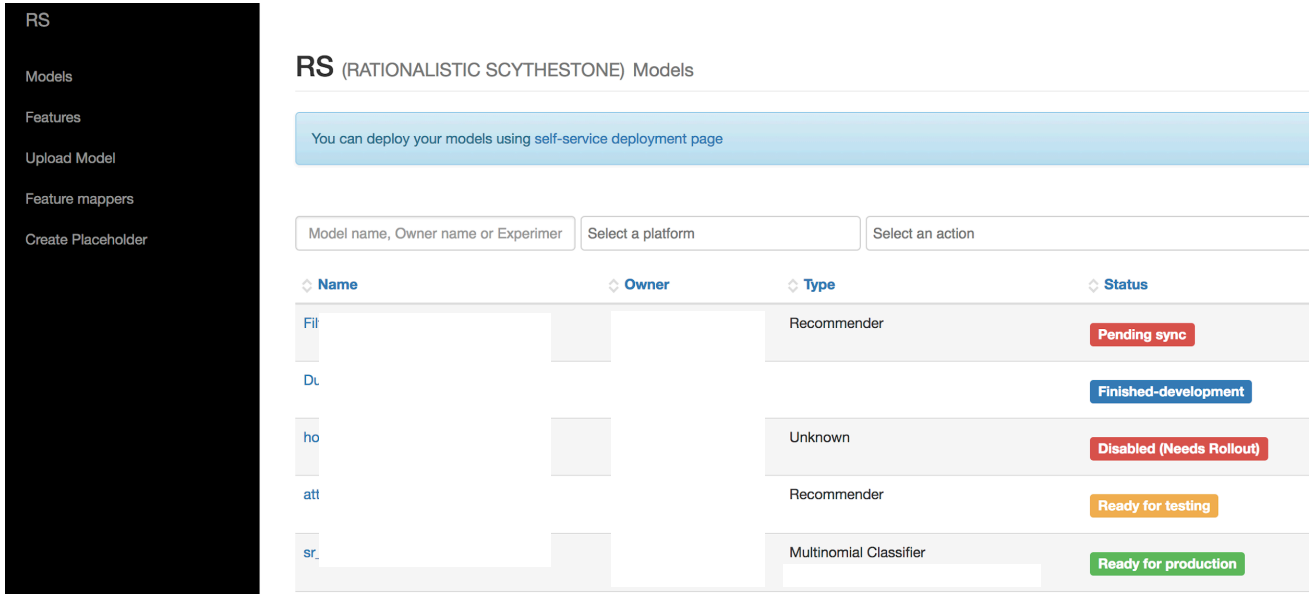
- Allow to define models and upload new versions

Model portal



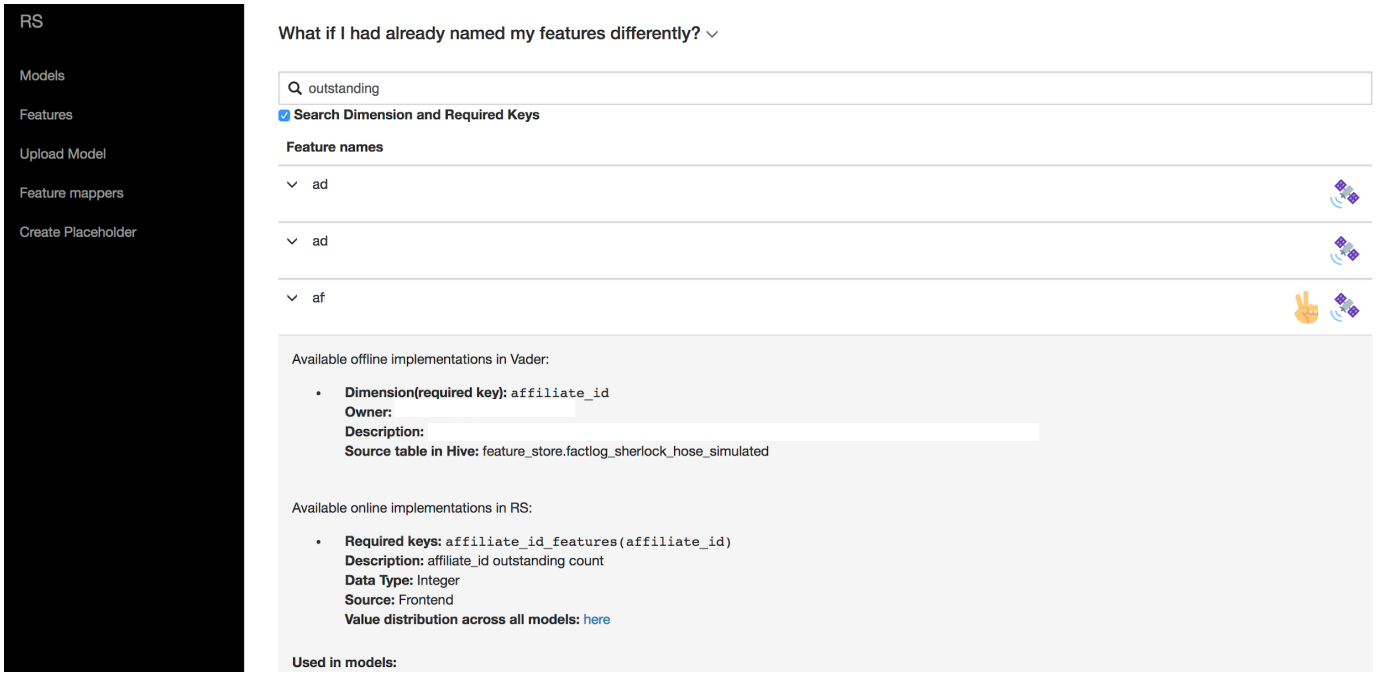
- Provide monitoring

Model portal



- Allow to discover and reuse existing models

Feature store portal

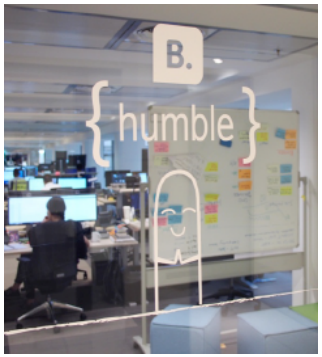


- Allow to discover and reuse features

Future directions

- Make the full pipeline smoother for faster iterations
- Limit involvement of devs (scarce resource) for ML work
  - Automatic reconstruction of features online from offline





Thank you for joining our talk!

We are hiring: Join the world's #1 website for booking hotels and other accommodations

https://workingatbooking.com

Luca Falsina **luca.falsina@**
Brammert Ottens **brammert.ottens@**

Booking.com