



SPARK  
SUMMIT



## PRODUCTIONIZING BEHAVIOURAL FEATURES FOR MACHINE LEARNING WITH SPARK STREAMING

Roman Studenikin & Ben Teeuwen  
Booking.com | Priceline Group

#EUstr4





# Timeline

•  
•  
•  
•  
•  
•  
•  
•  
•



**2014: Joined Booking.com**  
Data analysis through Perl  
Push for R & python  
My first big ML project



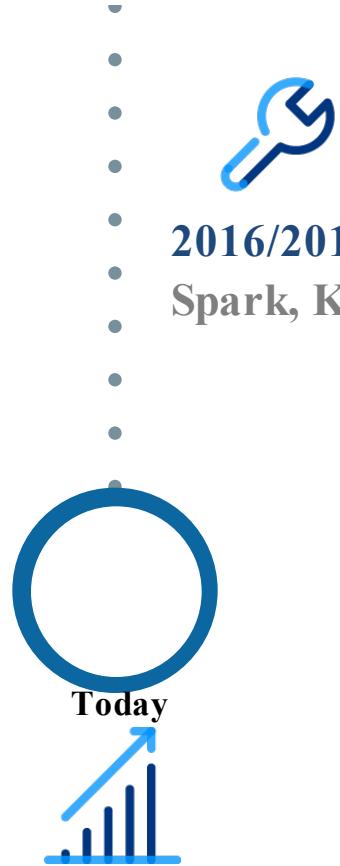
**Early 2015: Towards production**  
R randomForest ported  
Counter-based features implemented



**Mid 2015: Validating real-time service**  
Online predictions  
Feature inputs



**> Late 2015: Experimentation .. profit!**



**Today**

Integration of components into Feature Store

Trainings, trainings & trainings

## Objectives



Reduce skew between  
offline DS world & online  
DEV-guarded world



Autonomy for DS (fueled  
by scarcity of DEV  
capacity)



Re-usage of DS products



Speed up experimentation  
cycle with product owners

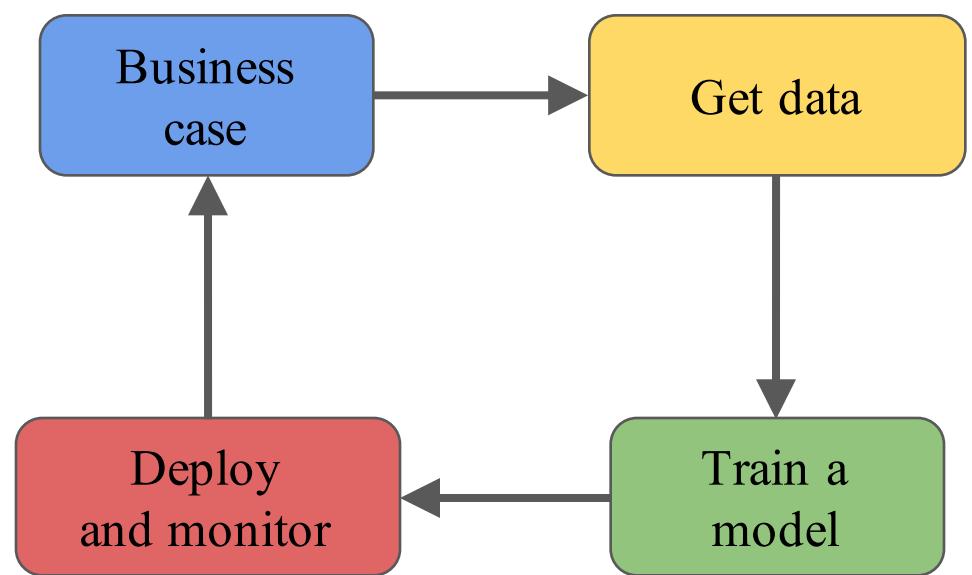
# Our tooling



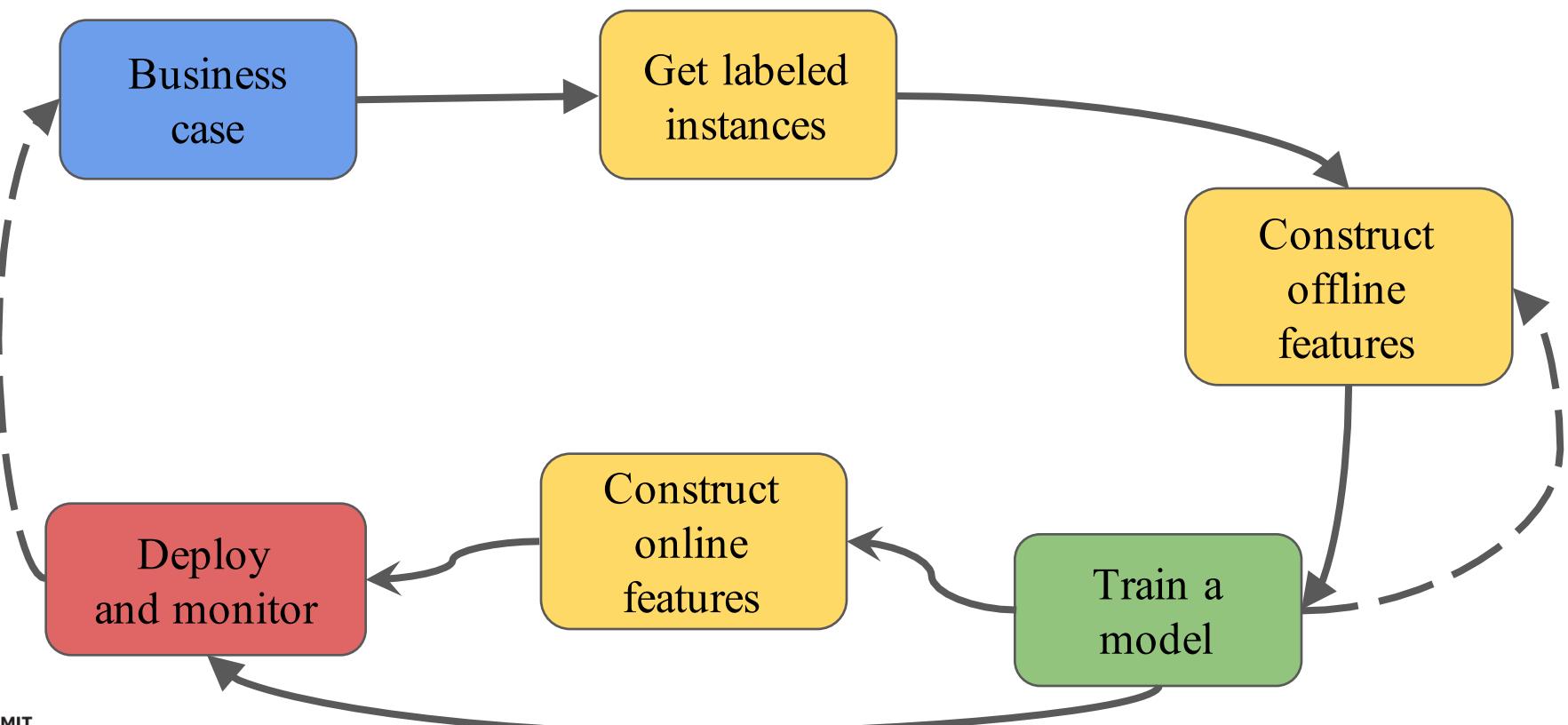
## Software development



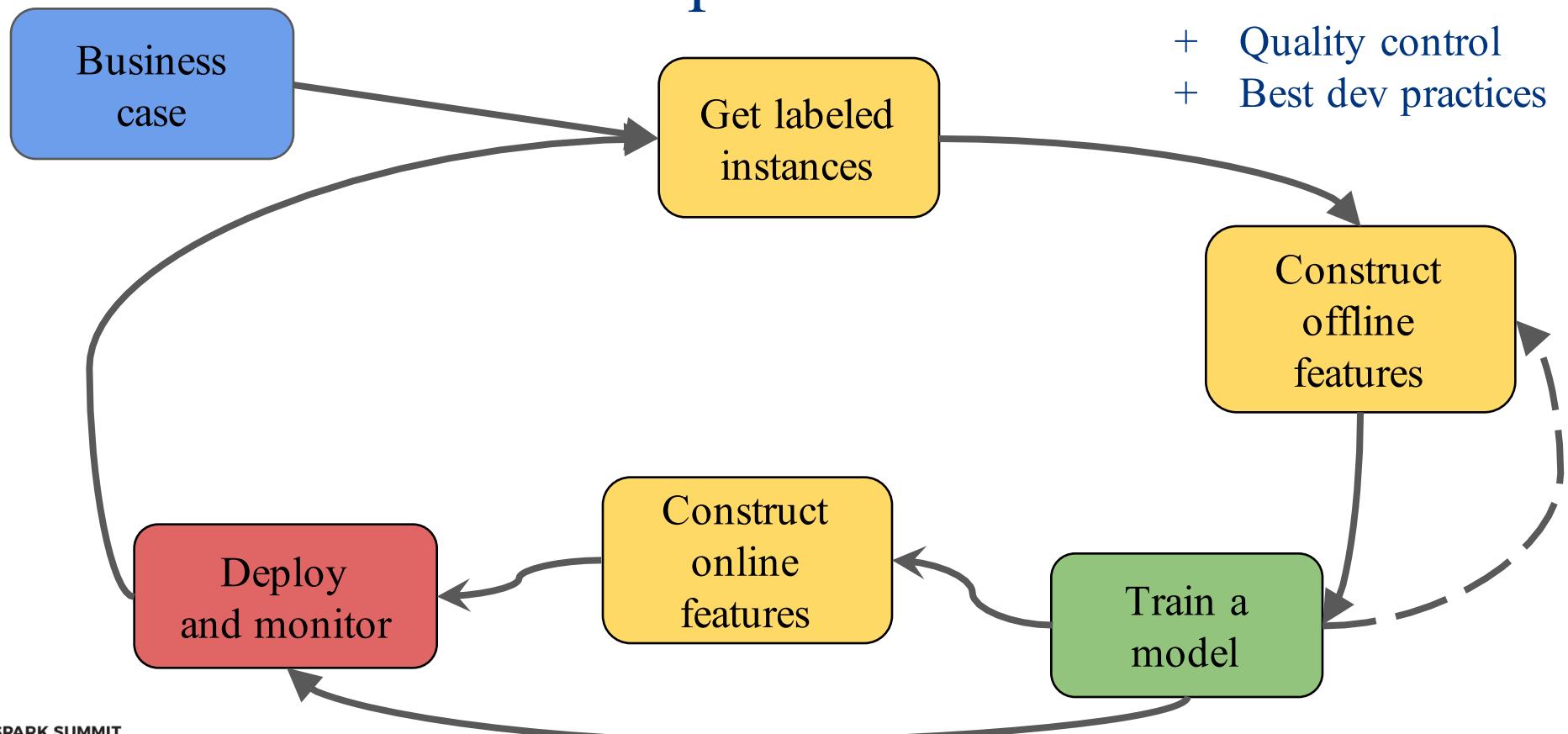
## Data science



# What can be improved?



# Reproducible



# Construct offline features

- FeatureVader - registry of all features
- Spark ML Transformer

```
data = spark.sql("....")
```

```
fv = FeatureVader()
```

```
fv.setNeededColumns(["feature1", "feature2"])
```

```
withFeatures = fv.transform(data)
```

userId	unixTime	label
1001	1508083925	1
1002	1508189936	0



userId	unixTime	label	feature1	feature2
1001	1508083925	1	123.2	2 kids
1002	1508189936	0	213.5	0 kids

# H2O.ai

0.0.1

## TRAINING DATA

### DATASET

default\_of\_credit\_card\_clients.csv

### ROWS

24k

25

0

1

### TARGET COLUMN

default payment next month

### TYPE

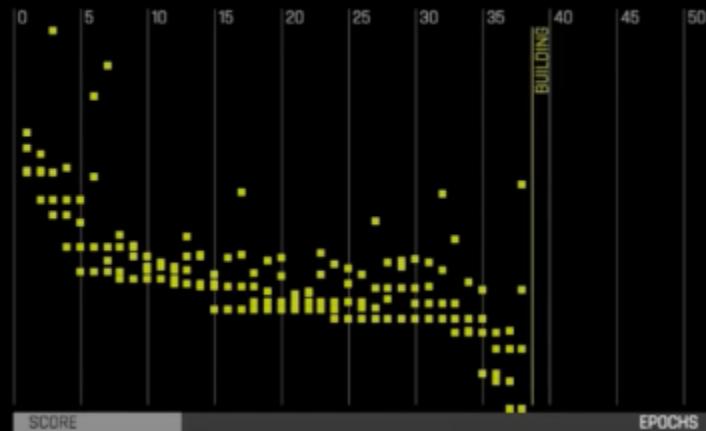
int64

0

MEAN  
0.22

STD DEV  
0.41

## ITERATION SCORES



## STATUS MONITOR



## EXPERIMENT SETTINGS

3 WORKERS

50 ITERATIONS

3 CV FOLDS

4 POPULATION

Detect IDs

Drop Dups.

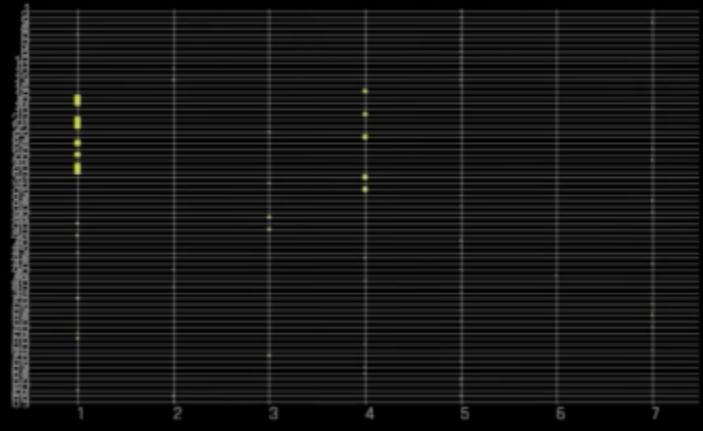
## GPU STATS



## VARIABLE IMPORTANCE

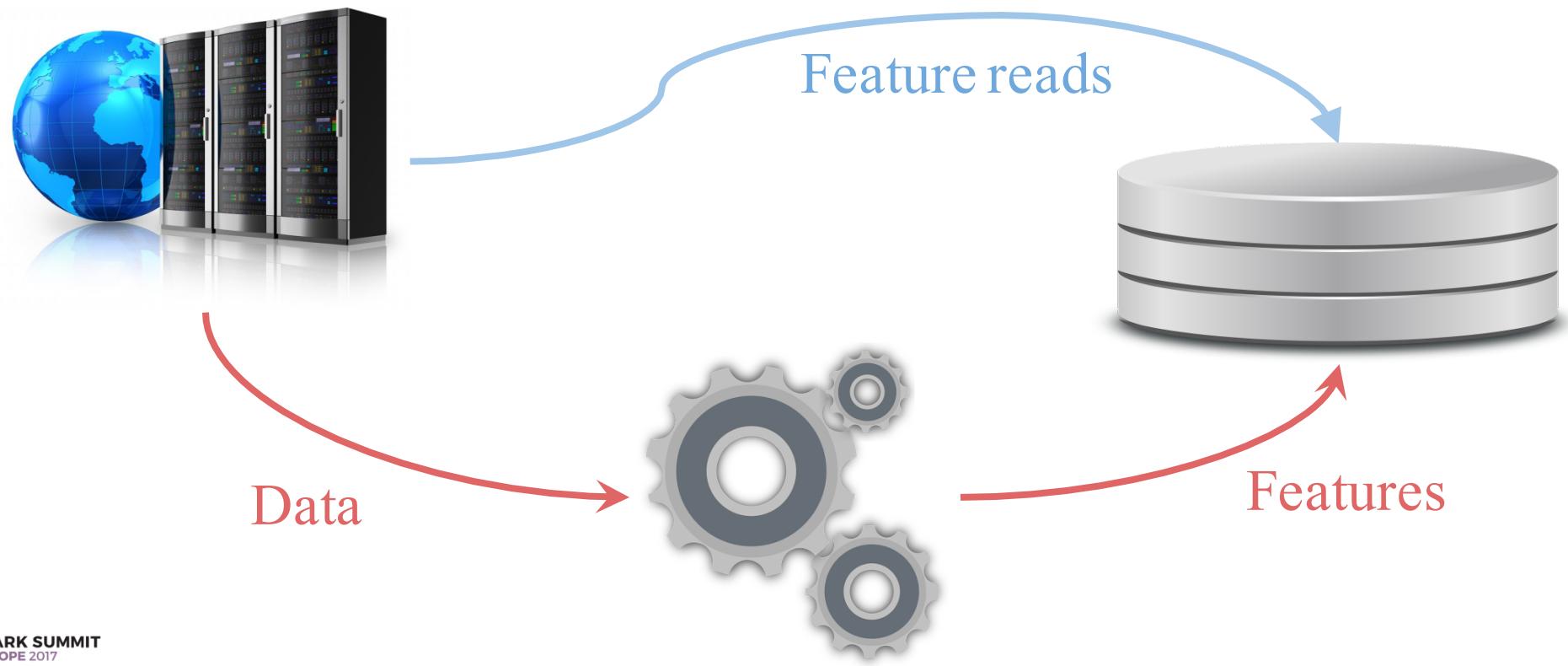
58_Interaction_PAY_0#multiply#BILL_AMT1	14194.18
10_PAY_0	8718.03
41_Interaction_PAY_0#safe_divide#PAY_AMT5	4911.97
70_PAY_0	3178.09
30_Interaction_BILL_AMT2#safe_divide#LIMIT_BAL	2883.98
66_Interaction_PAY_AMT3#multiply#LIMIT_BAL	2367.03
11_PAY_2	1728.54
56_Interaction_BILL_AMT2#multiply#LIMIT_BAL	1364.25
28_TruncSVD_BILL_AMT1_LIMIT_BAL_0	1199.49
71_Interaction_PAY_AMT5#subtract#PAY_AMT4	1147.27
52_CV_CatNumEnc_PAY_5_PAY_AMT5_std	1093.34
55_Interaction_PAY_0#multiply#BILL_AMT2	1046.90
12_PAY_3	1040.80
44_CV_CatNumEnc_PAY_5_PAY_2_std	1022.61

## FEATURE TRANSFORMATIONS

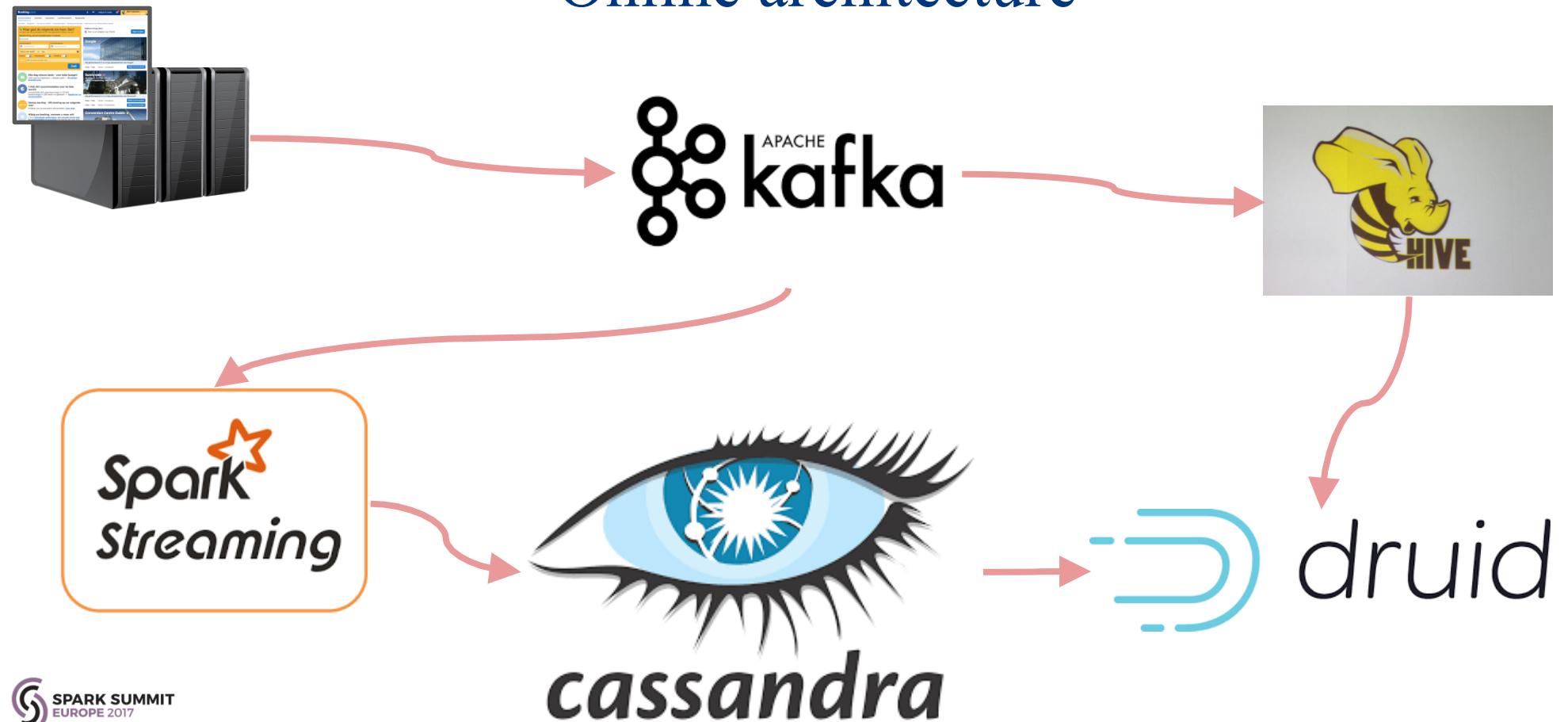


© 2017 H2O.ai. All rights reserved.

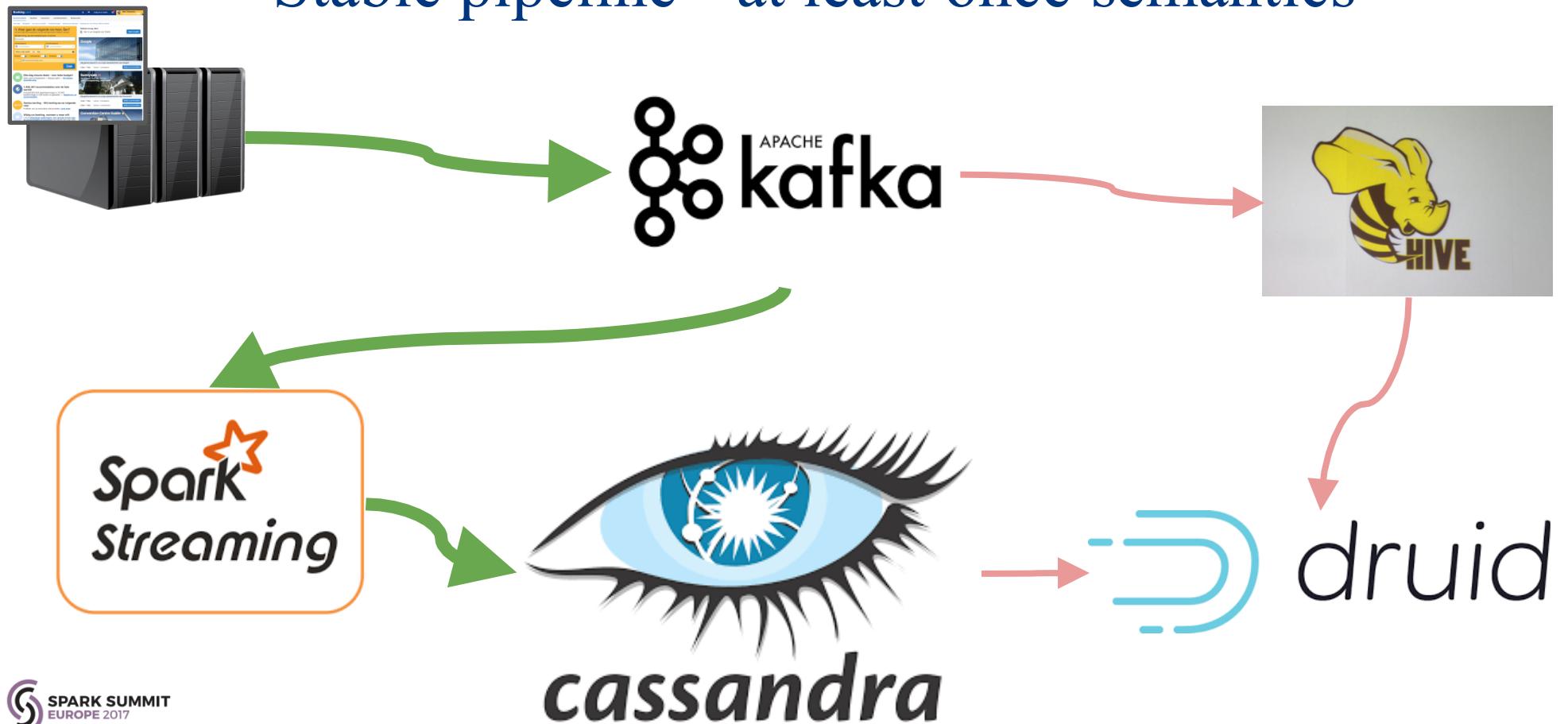
## Construct online features



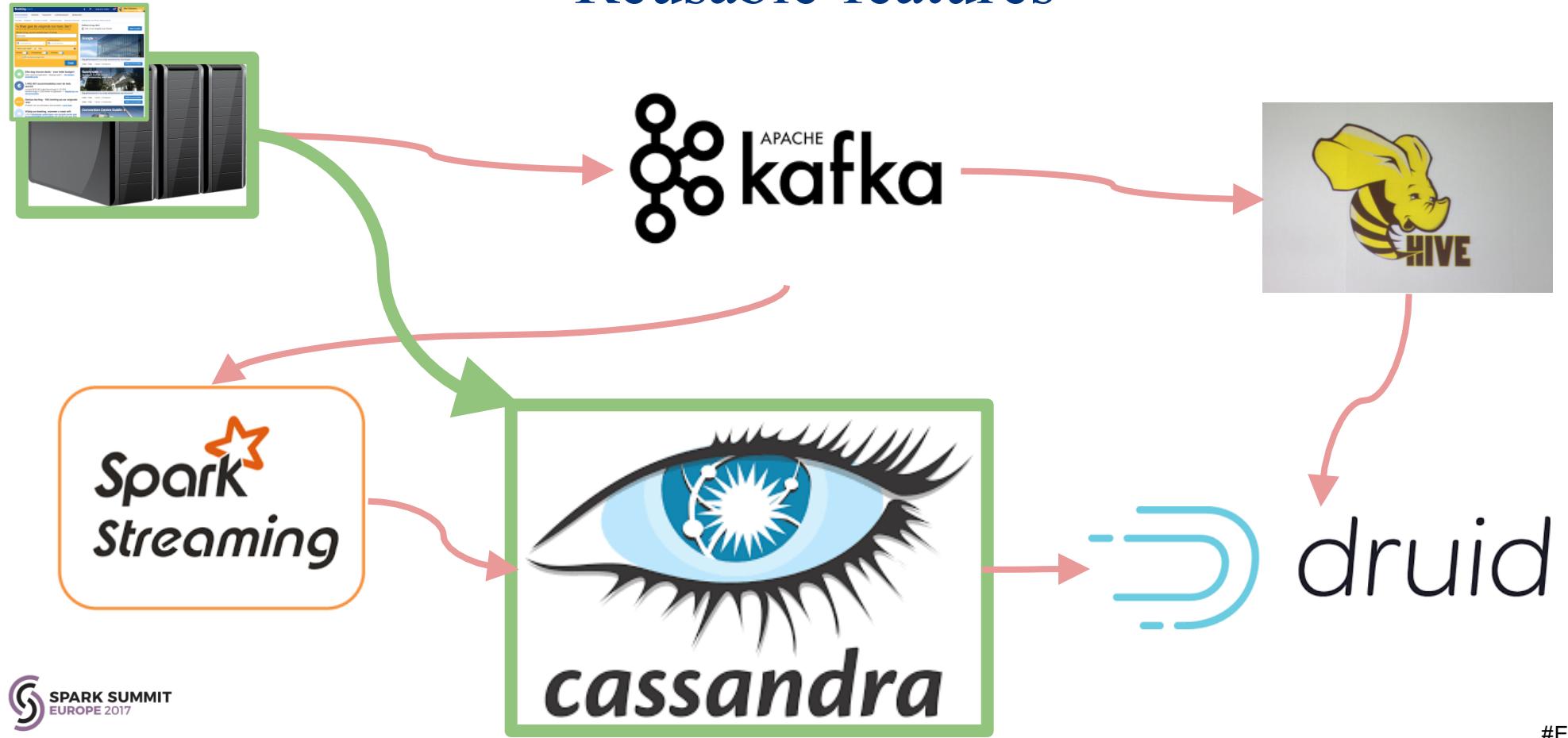
# Online architecture



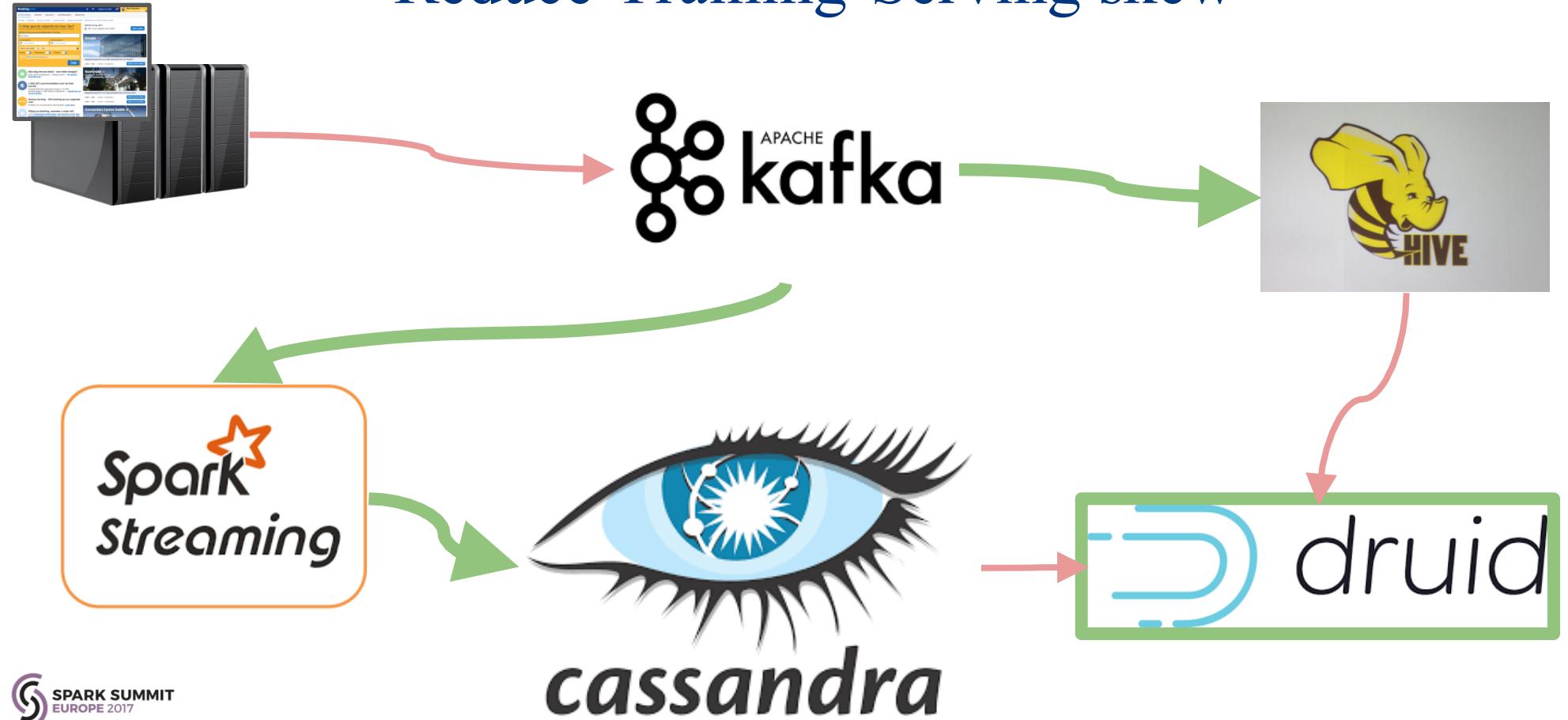
## Stable pipeline - at least once semantics



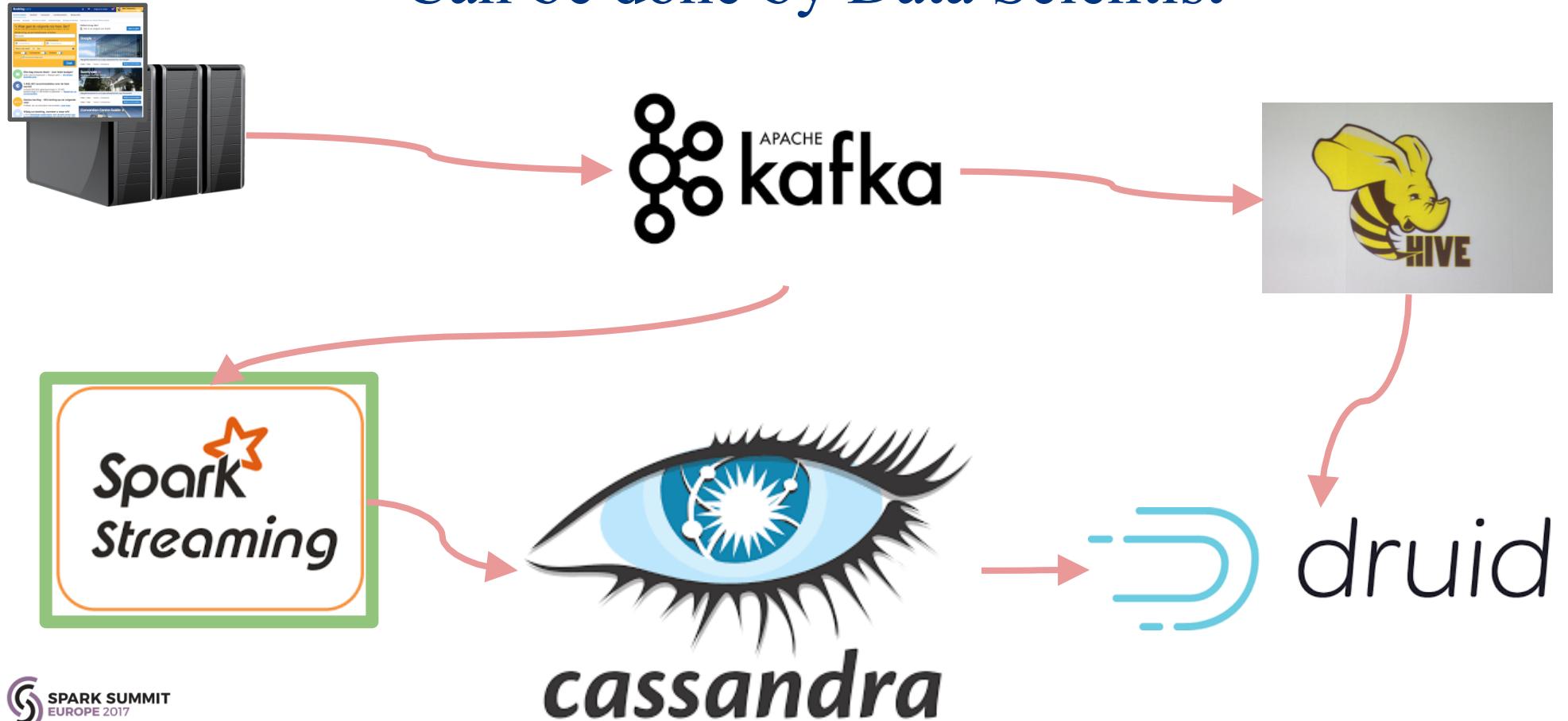
## Reusable features



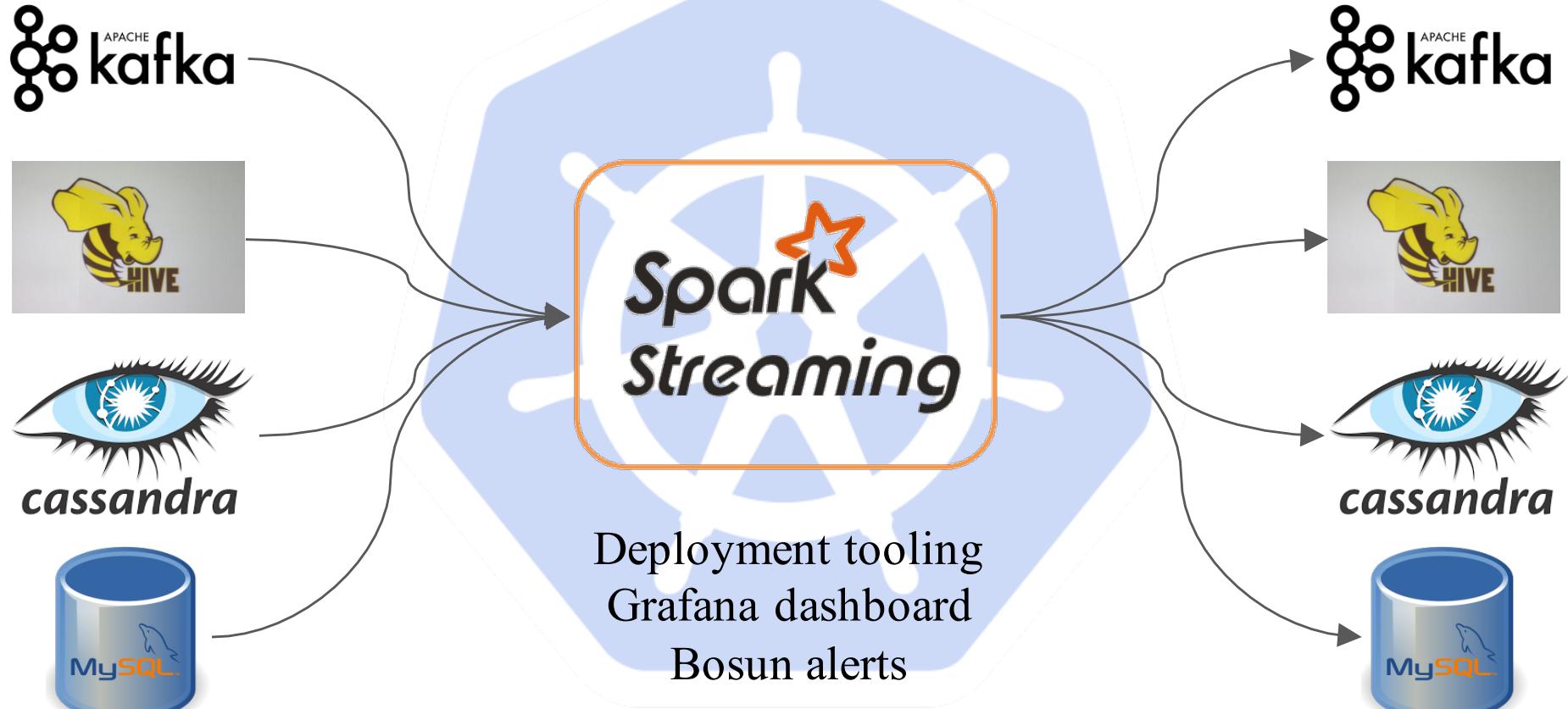
## Reduce Training-Serving skew



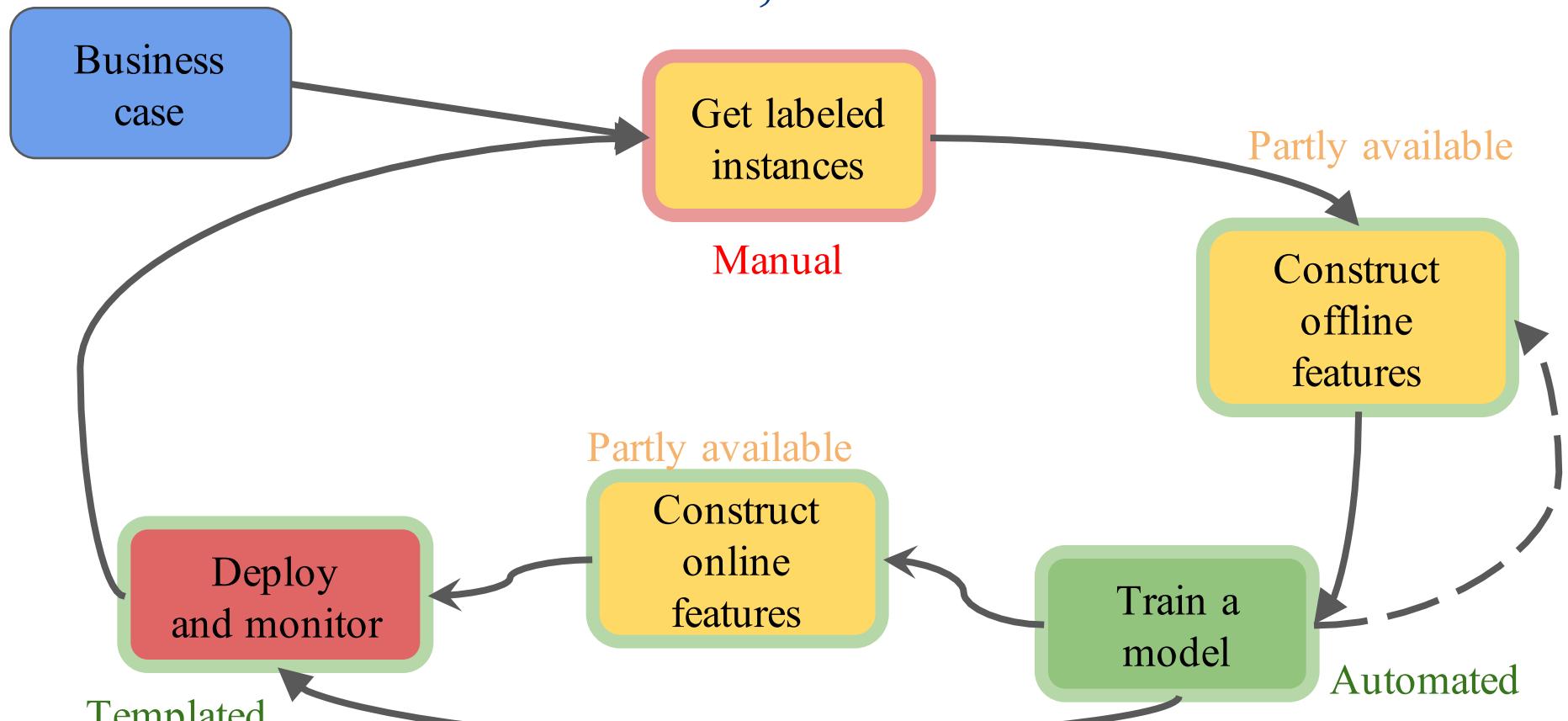
# Can be done by Data Scientist



# Online feature encoder



# Automated, standardised



# Learnings

Yes, it is possible =)

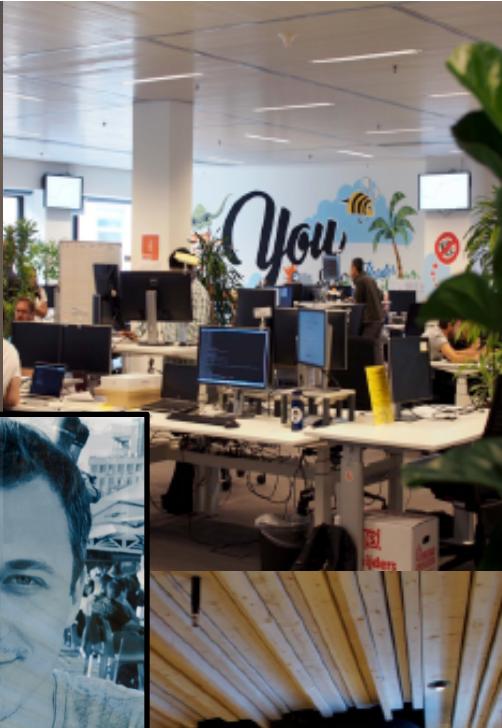
Offline (simple) feature construction is still easier.

Give data, not tooling.

Start building your vision from something



[roman.studenikin@booking.com](mailto:roman.studenikin@booking.com)



[ben.teeuwen@booking.com](mailto:ben.teeuwen@booking.com)

