Realtime Risk Management

USING KAFKA, PYTHON, AND SPARK STREAMING


UNDERWRITING CREDIT CARD
TRANSACTIONS IS RISKY
shopify


WE NEED TO BE QUICK AT
UNDERWRITING
shopify


WE ALSO NEED TO AVOID
LOSING MONEY
shopify

Some Numbers

$12Bn

CUMULATIVE PROCESSED

200k+

MERCHANTS

14k

EVENTS / SECOND

7

RISK ANALYSTS

Risk Analysts


WE NEEDED TO BUILD
SOMETHING THAT STOPS THE
HOPE FACTOR
shopify


Reebok
RBK


SPARK STREAMING
ALLOWS US TO DO REAL-TIME
DATA PROCESSING.
WE CAN DECIDE WHICH EVENTS
NEED A CLOSER LOOK

Intro to Kafka, Zookeeper, and Spark Streaming

Apache Kafka

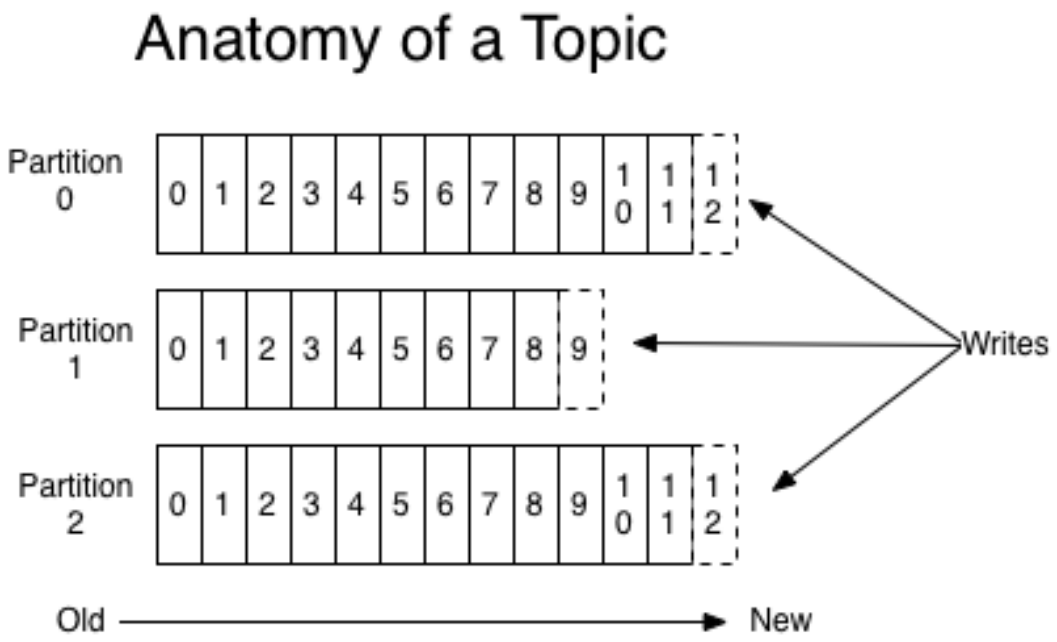


Image taken from Kafka docs

Apache Zookeeper

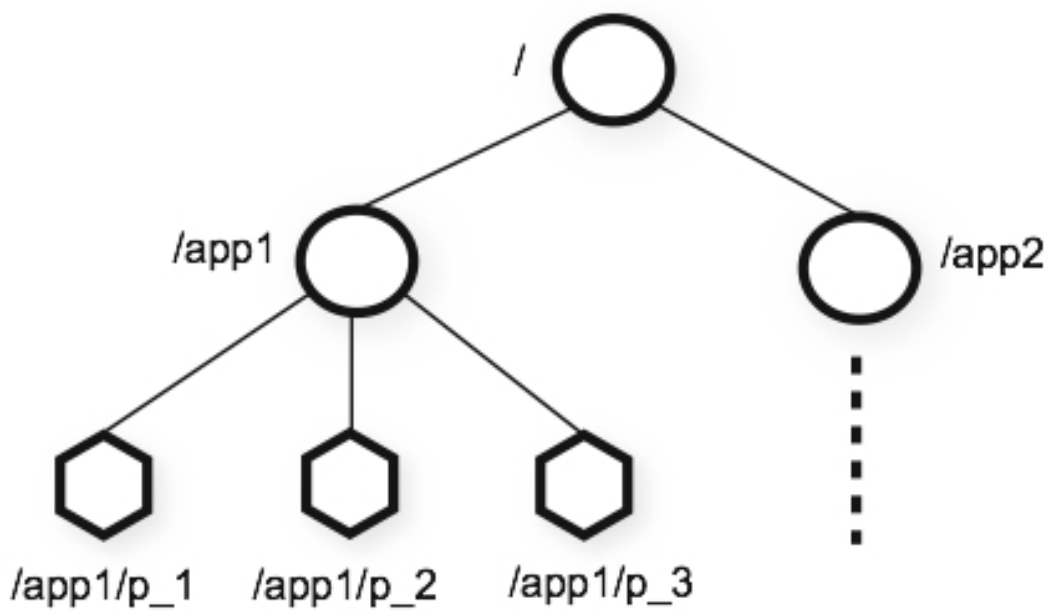
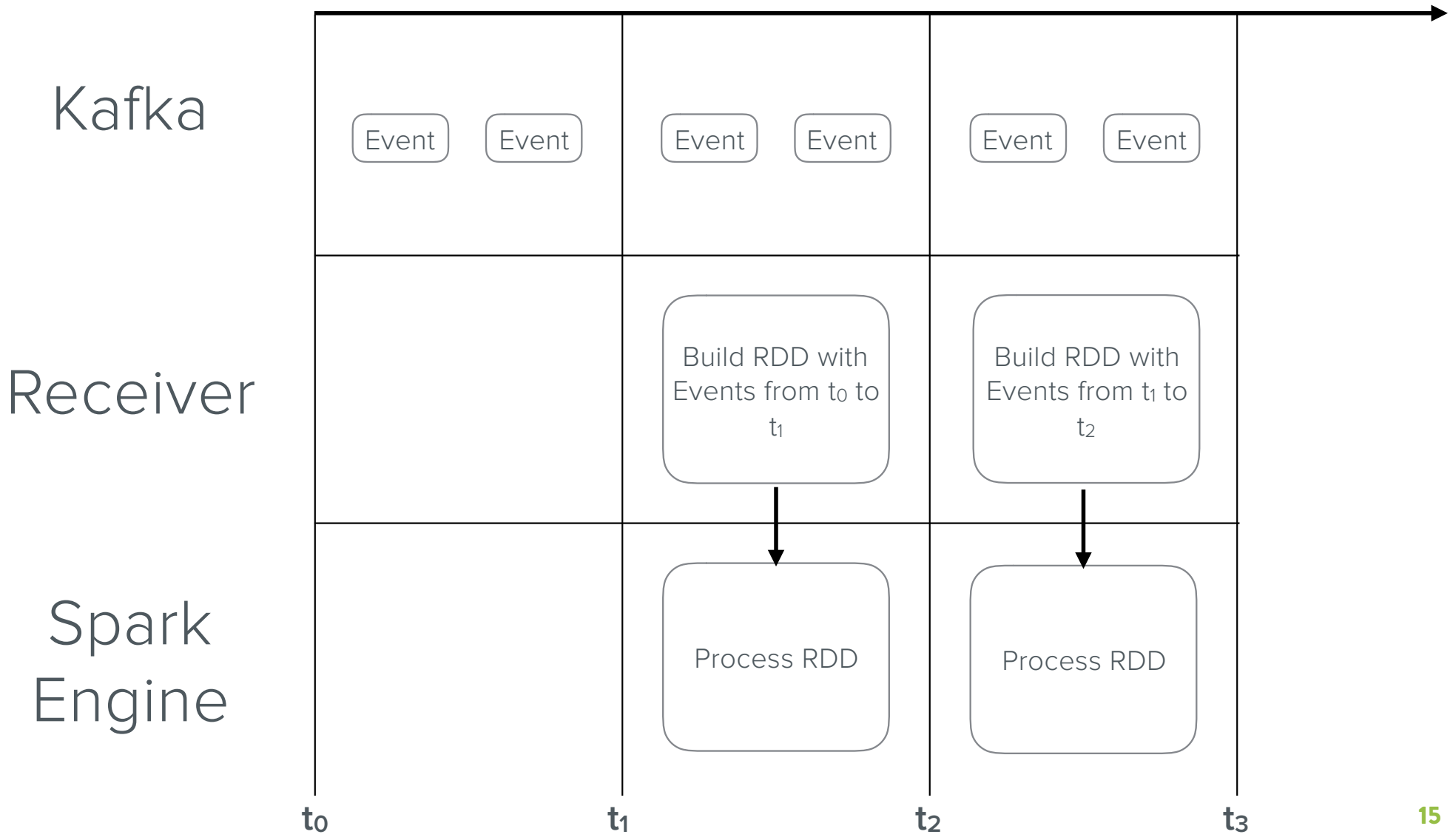


Image taken from Zookeeper docs

Apache Spark Streaming

Image taken from mapr.com

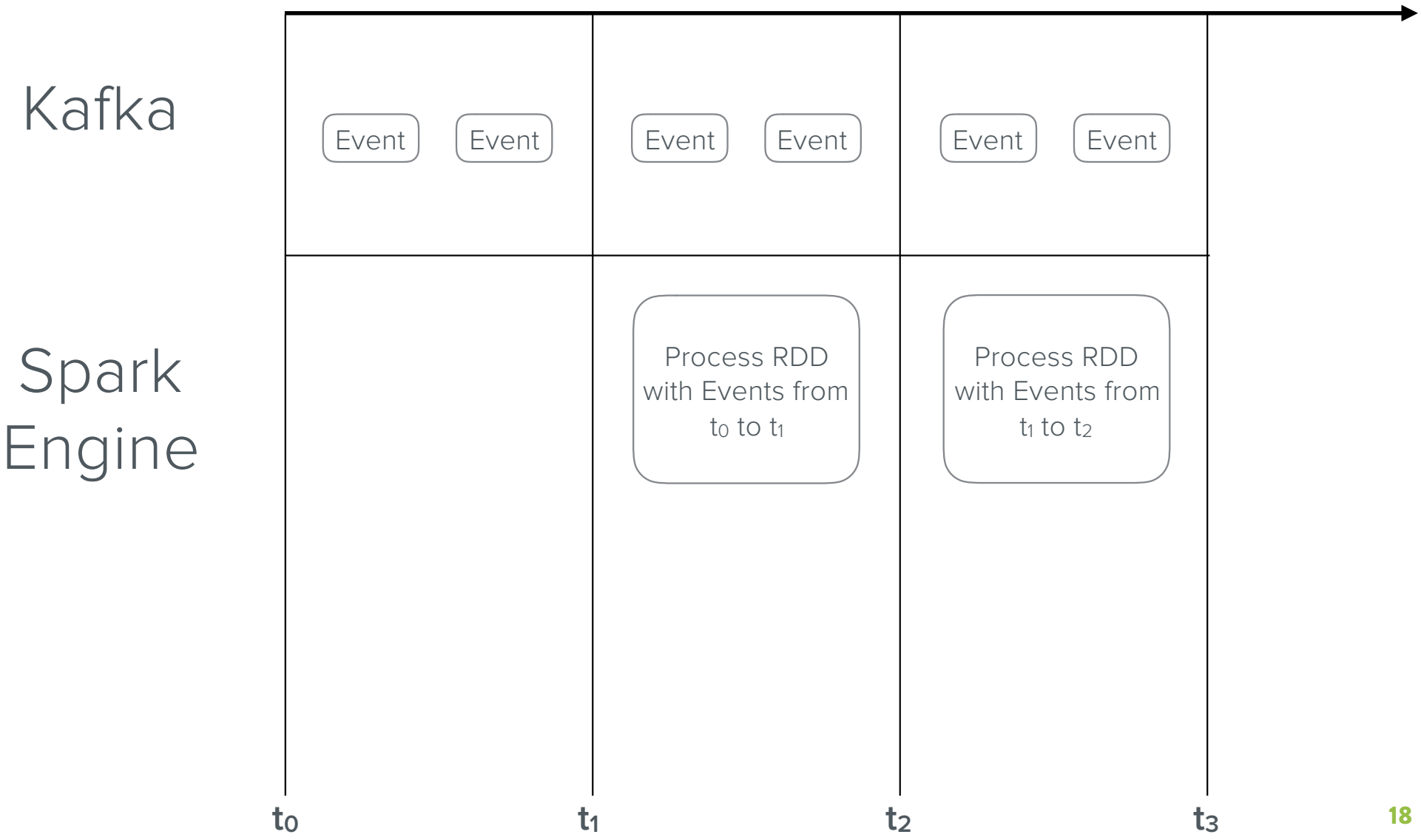

OLD WAY: RECEIVERS


Kafka
Receiver
Spark Engine
Event
Event
Event
Event
Event
Event
Build RDD with Events from t_0 to t_1
Build RDD with Events from t_1 to t_2
Process RDD
Process RDD
t_0
t_1
t_2
t_3

Problems w/ Receivers

- The only way to get at-least once delivery makes it hard to deploy new code
- Zookeeper is updated with which offsets to start from when data is received, not when it is processed
- We're actually duplicating Kafka


NEW WAY: RECEIVERLESS


Kafka
Event
Event
Event
Event
Event
Event
Spark Engine
Process RDD with Events from t_0 to t_1
Process RDD with Events from t_1 to t_2
t_0
t_1
t_2
t_3

General Structure

- Load Kafka offsets from Zookeeper
- Tell Spark Streaming to create a DStream that consumes from Kafka, starting at the specified offsets
- Define your processing step (ie. filter out non-risky events)
- Define your output step (ie. POST the data to the case management software)
- Save Kafka offset of most recently processed event to Zookeeper
- Start your streaming application, and grab some popcorn!

Example Filtering: Risky Products

hair extensions

gucci

vaporizer

wifi pineapple

pharmacy

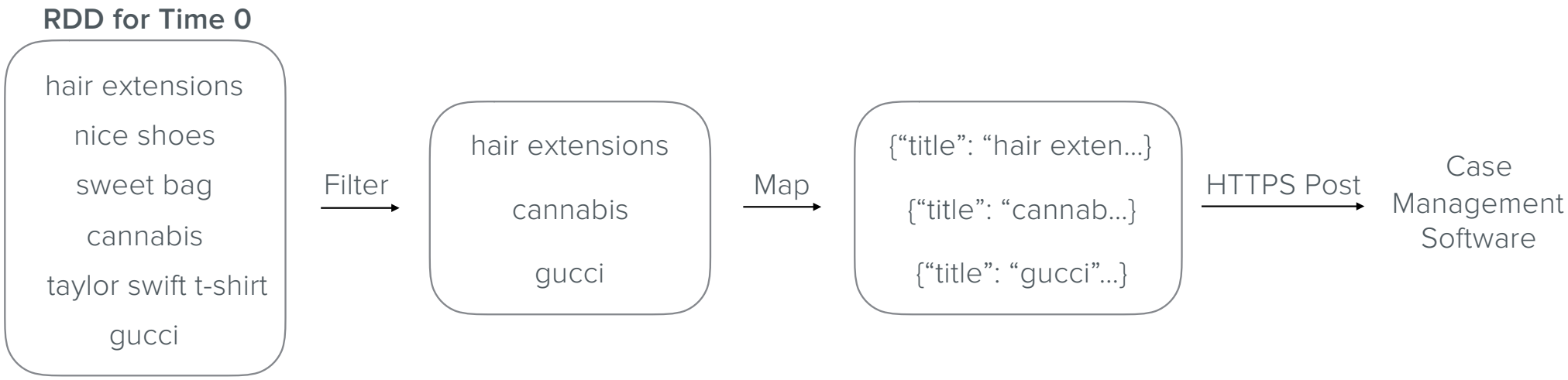
travel package

iPhone

gateway card

cannabis

Risky Products



```
return products\
    .map(self.coalesce_product_body_html)\
    .filter(self.shop_on_shopify_payments)\
    .filter(self.search_for_high_risk_words_in_products)\
    .filter(self.remove_blacklisted_shops)\
    .filter(self.is_not_verified_shop(verified_shops_bc))\
    .map(self.format_record)\
    .map(self.format_triggers)\
    .map(self.OUTPUT.project_row)
```


TIME-WINDOWED
AGGREGATIONS

The Future

- **Time-Windowed Functions** – A necessity for most of the non-trivial jobs
- **Performance Tweaks** – We haven't spent any time on this, so lots of potential for gains
- **Machine Learning** – We could use the Risk Analyst decisions to build a ML model
- **Improved Monitoring** – We are only monitoring the basics right now
- **Apache Cassandra** – Others use it as a fast key/value store for their jobs
- **Improved Receiverless API** – An API to access Kafka / Zookeeper without hard work

Icon Credits

Credit Card by Rediffusion from the Noun Project

Money Bag by icon 54 from the Noun Project


WE'RE HIRING
SHOPIFY.COM/CAREERS
shopify


@n_e_evans
THANK YOU!
shopify