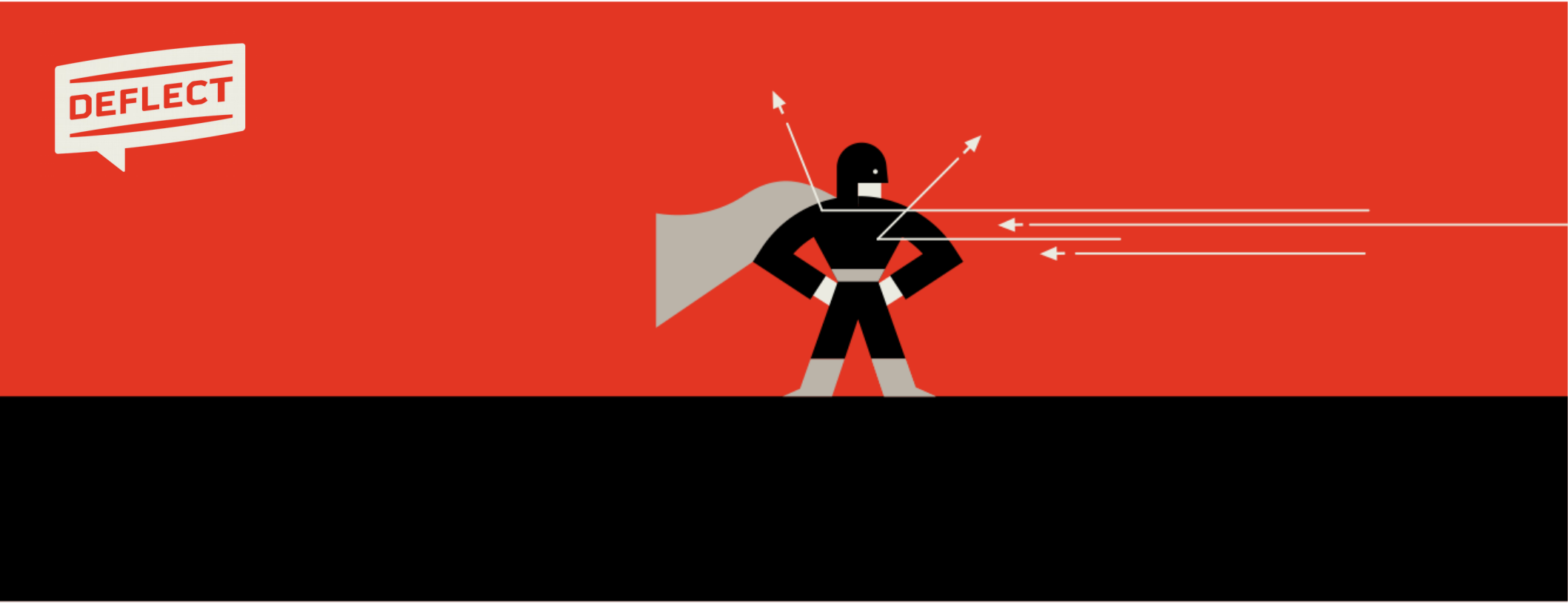


eQualitie

Deflect Labs and Baskerville – Identifying attacks with Machine Learning

Introduction – Why?

The need

- Manual identification and mitigation of (DDoS) attacks on websites is a difficult and time-consuming task with many challenges

The goal

- Create a system to identify the attacks directed to Deflect protected websites as they happen and give the infrastructure the time to respond properly.

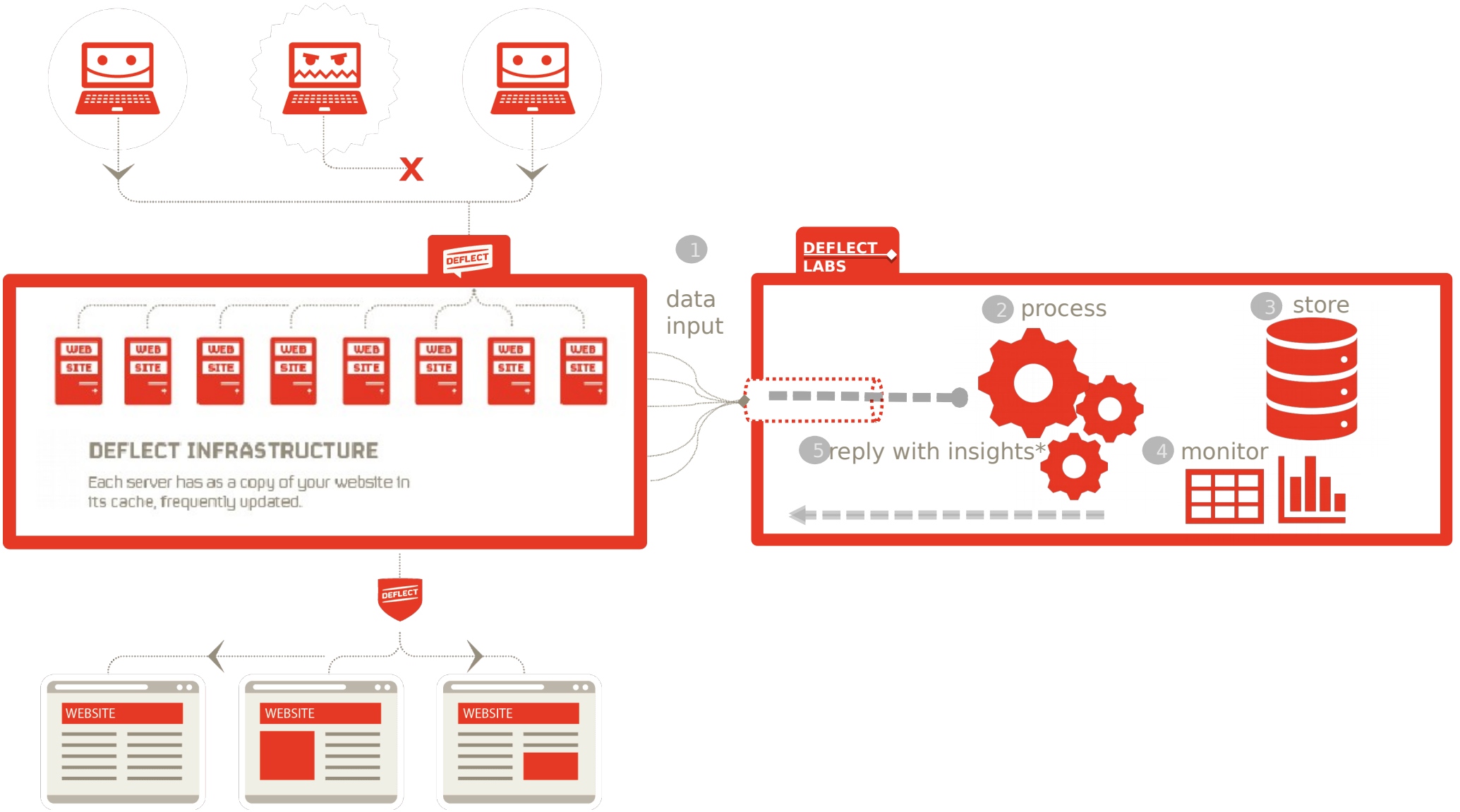
Introduction - The challenges

- Be fast enough to make it count
- Be able to adapt to traffic
- Provide actionable info
- Provide reliable predictions
- As with any ML project: not enough labelled data
- Make the system as generic as possible

DIGITAL SECURITY FOR CIVIL SOCIETY

Free. Open Source. Principled

Baskerville - Introduction



Baskerville - Architecture

Two main Components

- **Engine**
  - Main log processing Pipelines
- **Off-line analysis tools**
  - Model development
  - Visualization
  - Investigation

✓ Time-windowed – batch processing

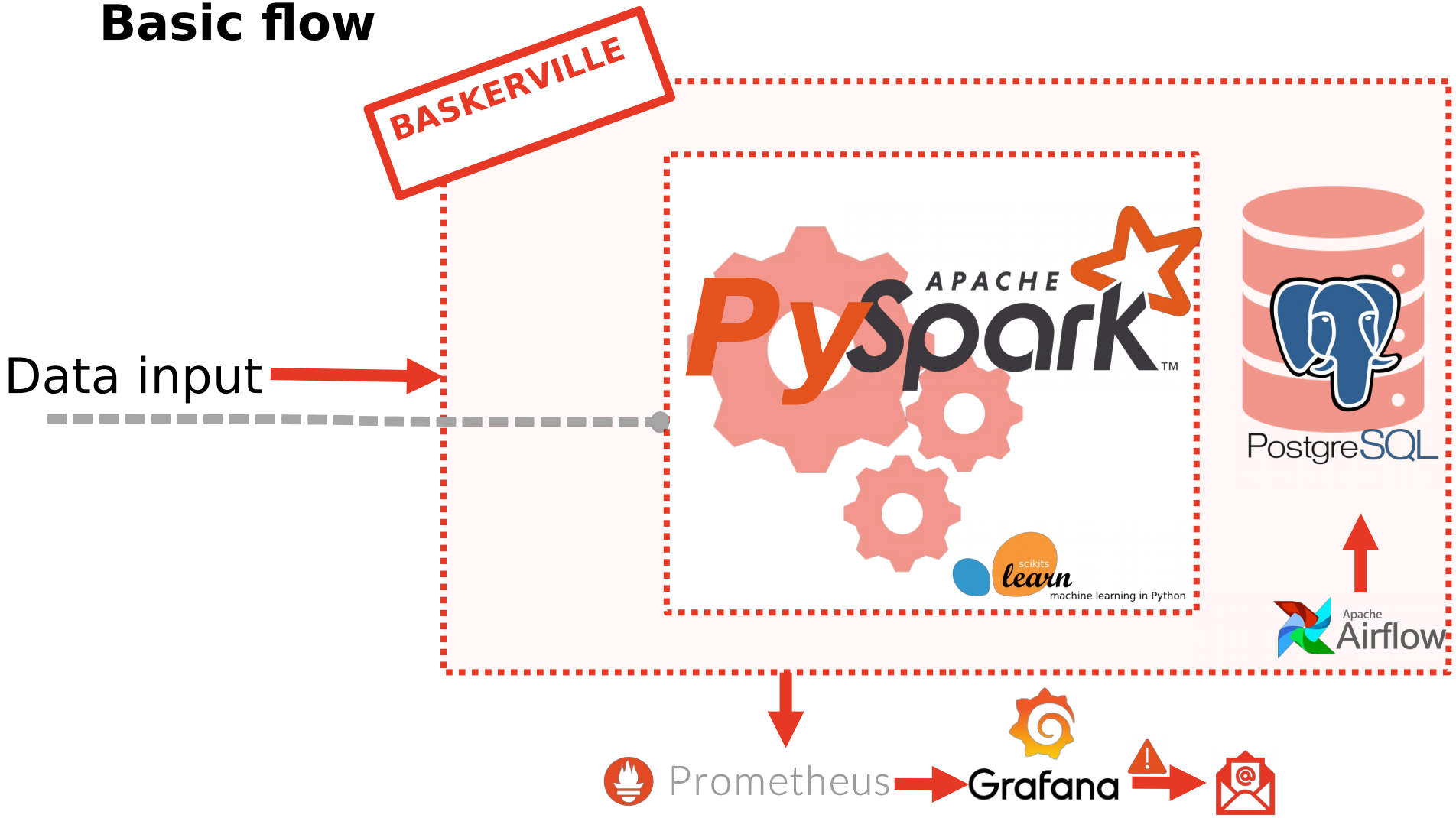
Baskerville – Engine: Pipelines

Pipelines

- Base Pipeline for the basic flow
- Elastic Search Pipeline - to process logs from ES
- Raw Logs Pipeline - to process log files
- Kafka Pipeline - to process logs from Kafka

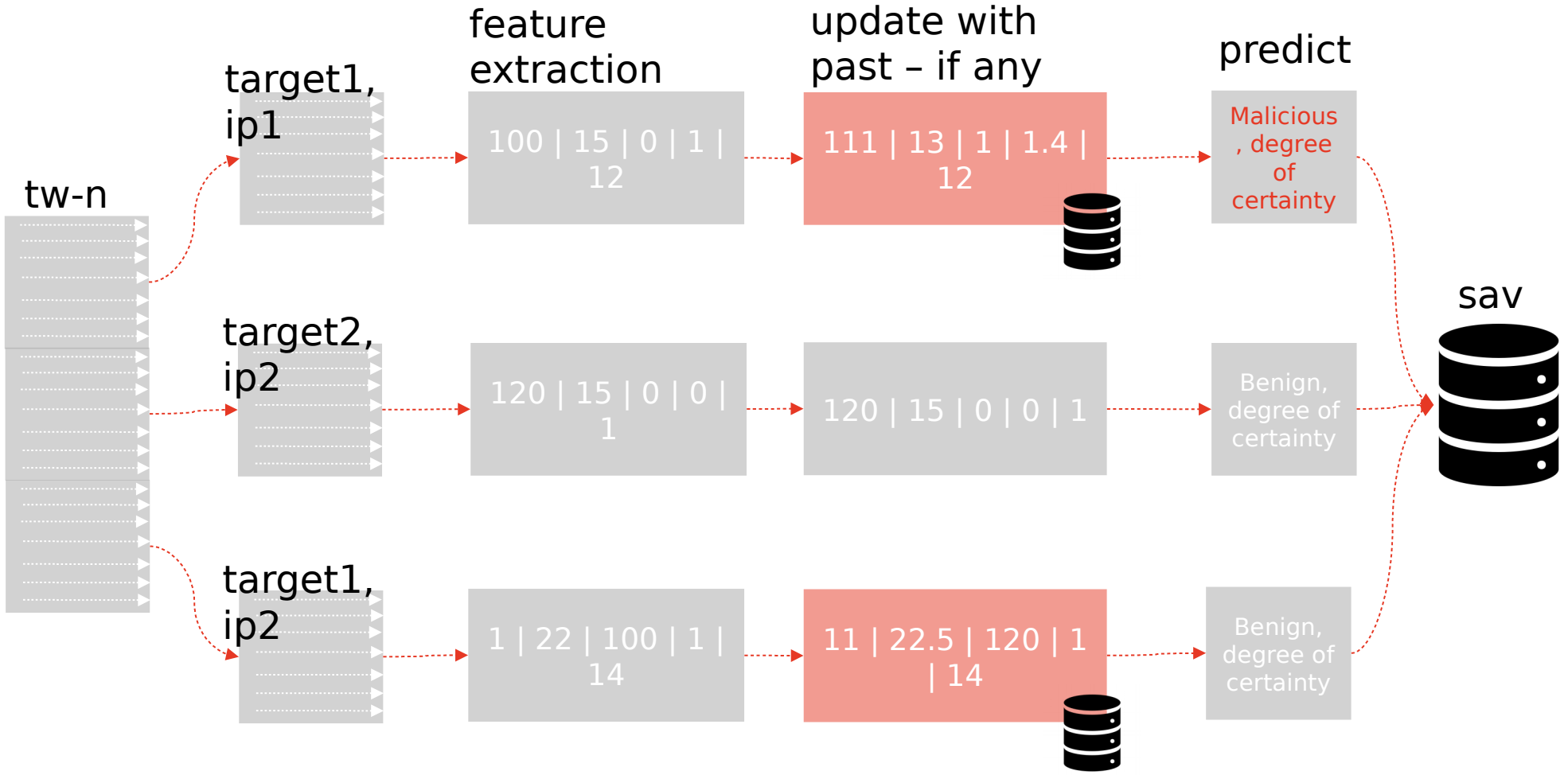
Baskerville – Engine

Basic flow



Baskerville – Engine: Pipelines

Basic flow



Baskerville – Features

Features and structure

- Css to html ratio
- Image to html ratio
- Js to html ratio
- Minutes total
- Path depth average
- Path depth variance
- Payload size average
- Payload size log average
- Request interval average
- Request interval variance
- Request total
- Response 4xx to request ratio
- Top page to request ratio
- Unique path rate
- Unique path to request ratio
- Unique query rate
- Unique query to unique path ratio
- Unique UA rate
- ...

Most of them are Updateable Features – they take past into account

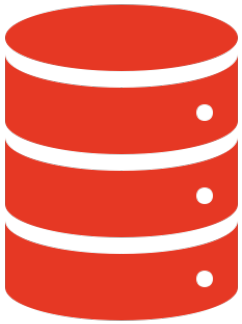
- ✓ compute
- ✓ update

Baskerville – Features

Cache and Feature update - Taking past into account

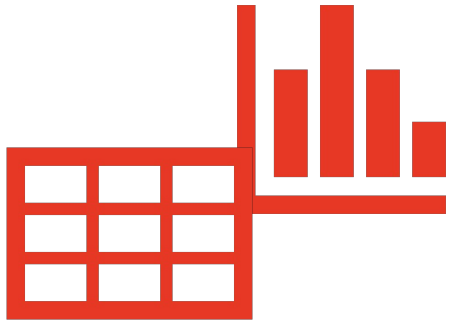
- Keep track of one week of traffic at any point
- Two level cache - short-term (in memory) and long-term (parquet)

Baskerville – Database



- ORM – SQLAlchemy
- Only inserts, no updates – except for the offline tools
- Data partitioning: per week of year
- Data archive / retention policy: keep data for one year
- Airflow to coordinate and schedule the database maintenance, create new partitions, detach old ones, archive.

Baskerville – Monitoring



- Prometheus with exporters for:
  - Baskerville itself
  - Spark
  - Postgres
  - Kafka
  - and Prometheus and Grafana of course
- Grafana for visualization

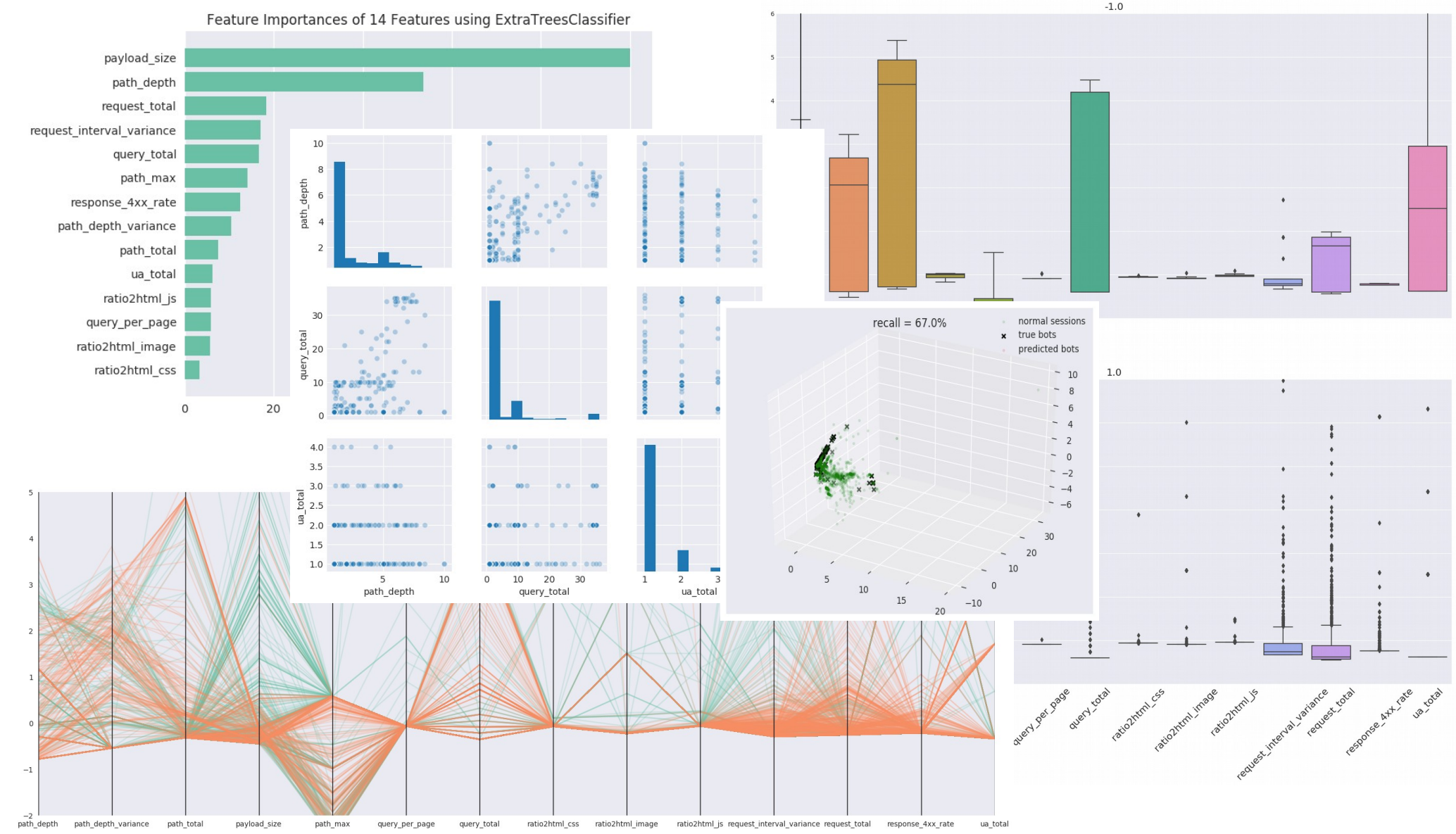
Baskerville – Offline tools

Offline Analysis Tools

Model development and investigations

- **Preprocess historic logs –** Elastic Search / log files
- **Preprocessing**: Label training / testing sessions based on MISP database of attacks
- **Training**: Train novelty detection classifier on labelled sessions from historic logs
- **Predicting**: Classify sessions as malicious / benign using newly trained models
- **Clustering**: Group sessions based on their features to investigate botnets
- **Visualization**: Produce figures to aid model development and investigations

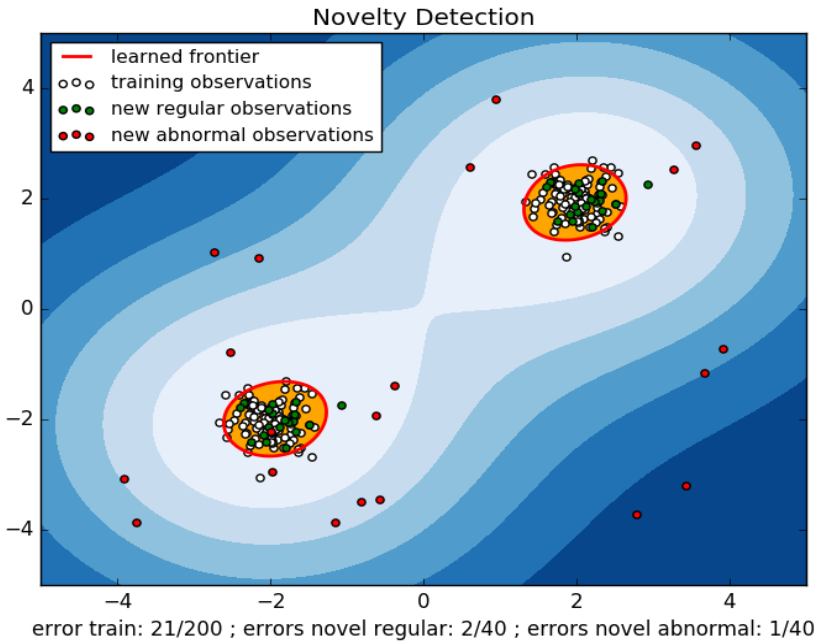
Baskerville – Offline tools



Baskerville – Machine Learning

How we use Machine Learning:

- **Novelty / Outlier detection** algorithms – One Class SVM, Isolation Forest
- **Lots of normal data**
- **Train** on **normal data** – allowed to include a small amount of abnormal data
- **Test** on known **past attacks**
- **Cross-reference** results with **Banjax bans**



Baskerville – Machine Learning

Processing the Attacks against Vietnamese Civil Society

- **8 attacks** in total considered
- Processed **attack periods** and **normal traffic** (separately) with **Baskerville**
- **Train dataset**: normal traffic
- **Test dataset**: a combination of normal and abnormal traffic

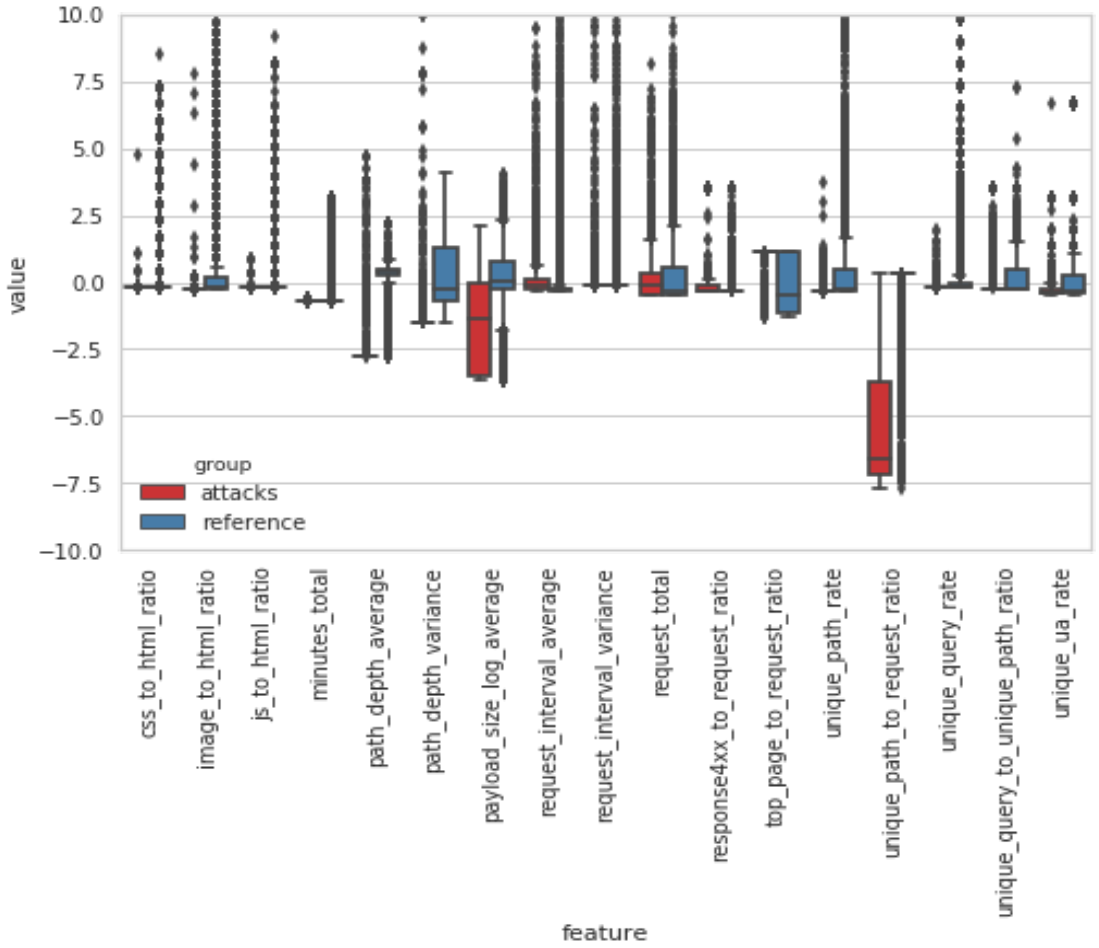
Isolation Forest

Precision	0.90
Recall	0.86
F1 score	0.88

- 90% of the IPs predicted as anomalous by Baskerville were also flagged by Banjax as malicious
- 88% of all the IPs flagged by Banjax as malicious were also identified as anomalous by Baskerville

Baskerville – Machine Learning

Processing the Attacks against Vietnamese Civil Society



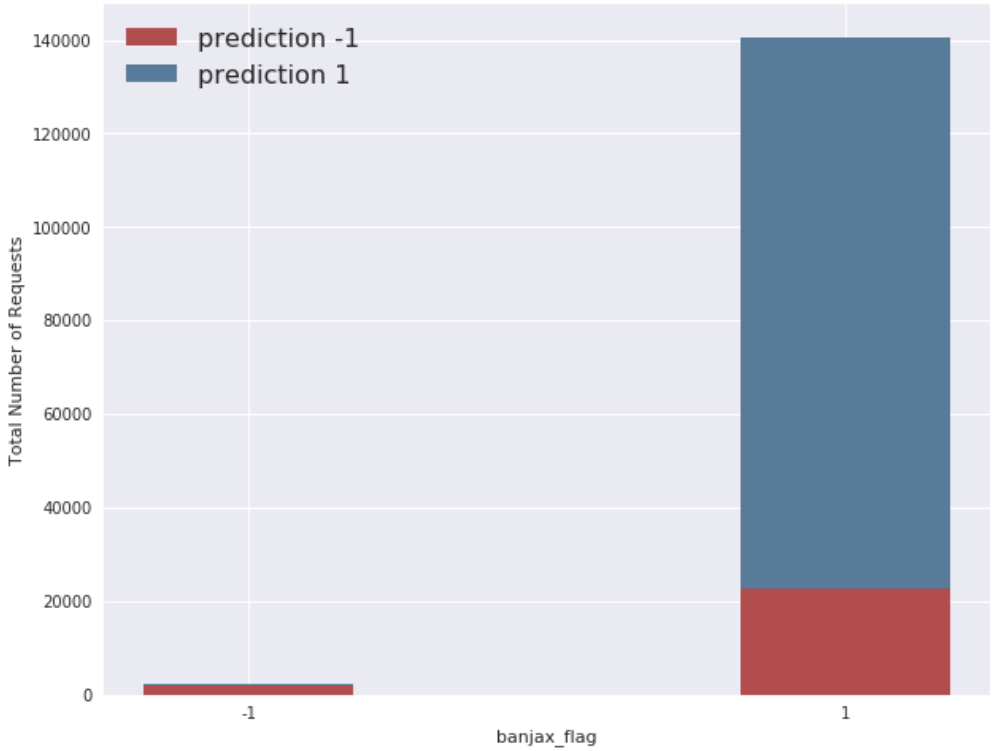
Attack characteristics

- far fewer **unique paths** requested
- a shorter **average path depth**
- a smaller variance in the **depth of paths requested**
- a lower **payload size**

Baskerville – Machine Learning

Processing the Attacks against Vietnamese Civil Society

The need for a feedback mechanism



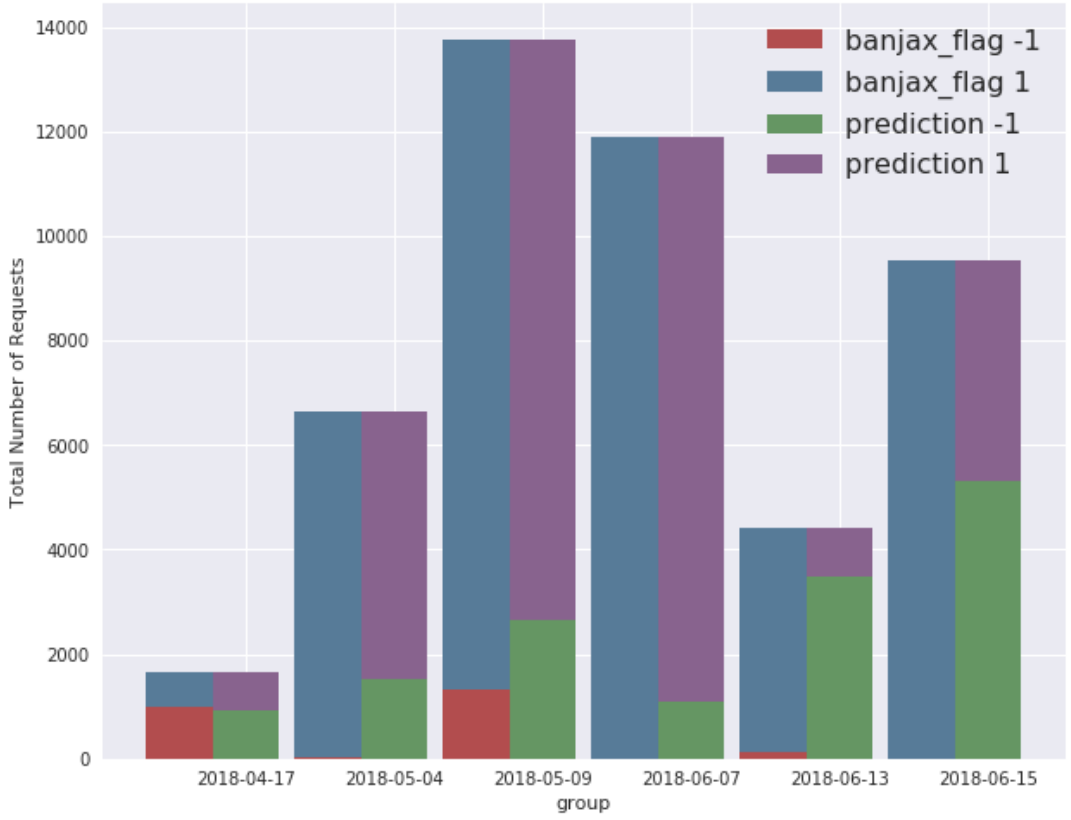
The overlap between the Banjax flag and the Baskerville prediction

-1 indicates malicious
+1 indicates benign

Baskerville – Machine Learning

Processing the Attacks against Vietnamese Civil Society

The need for a feedback mechanism



Baskerville is picking up more request sets as malicious than Banjax …

but does this indicate that Baskerville is too sensitive to anomalous behaviour, or that Baskerville is outperforming Banjax?

Read the report comparing human analysis vs baskerville

Baskerville – Machine Learning

What about Bias?

Perhaps there is a chance that people with slower internet connections display different than average browsing behaviour. So we need to be **careful feature selection** and **a good training dataset** with lots of examples of browsing.

To conclude

- So in general we are doing well, but there is a long way to go to properly evaluate the system and make sure we have low false positives and false negatives

Baskerville – How do I use it?

- **Just like any other Python project**, download, pip install it, configure it and run it
- **Docker compose** for Baskerville itself and the peripheral components like Postgres, Kafka, Prometheus, Grafana and various exporters
- Script for setting up a **stand-alone single node Spark cluster set up** (kubernetes - spark integration is still experimental in spark 2.4.0)

Baskerville – Extensibility & Use cases

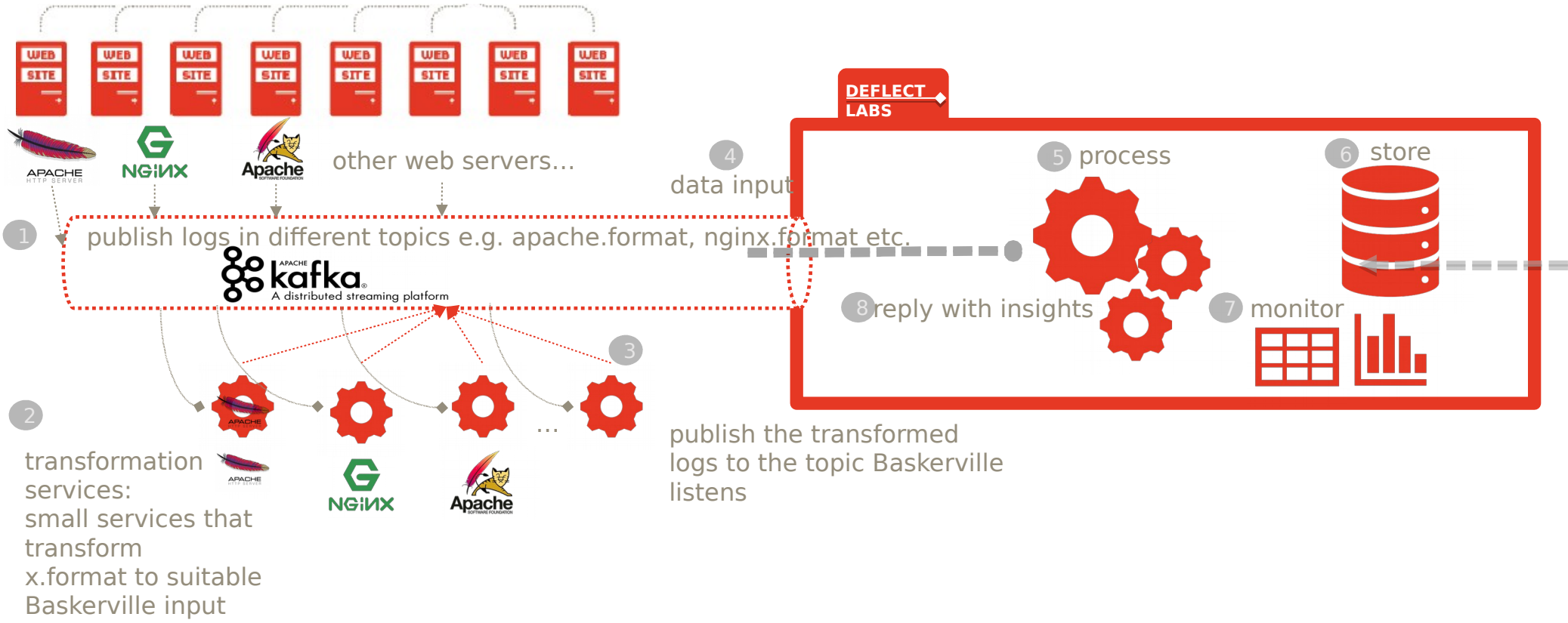
From a user's perspective:

- You can set it up as a stand-alone analysis engine to process and analyze old logs in various forms and formats:
  - text files
  - Elastic Search
  - through a queue / adapter
- Train models on own logs

From a developer's perspective:

- Easy to enable / disable features - or create new ones
- Easy to extend the pipelines or add new

What is Baskerville – Use cases



Baskerville - Current state

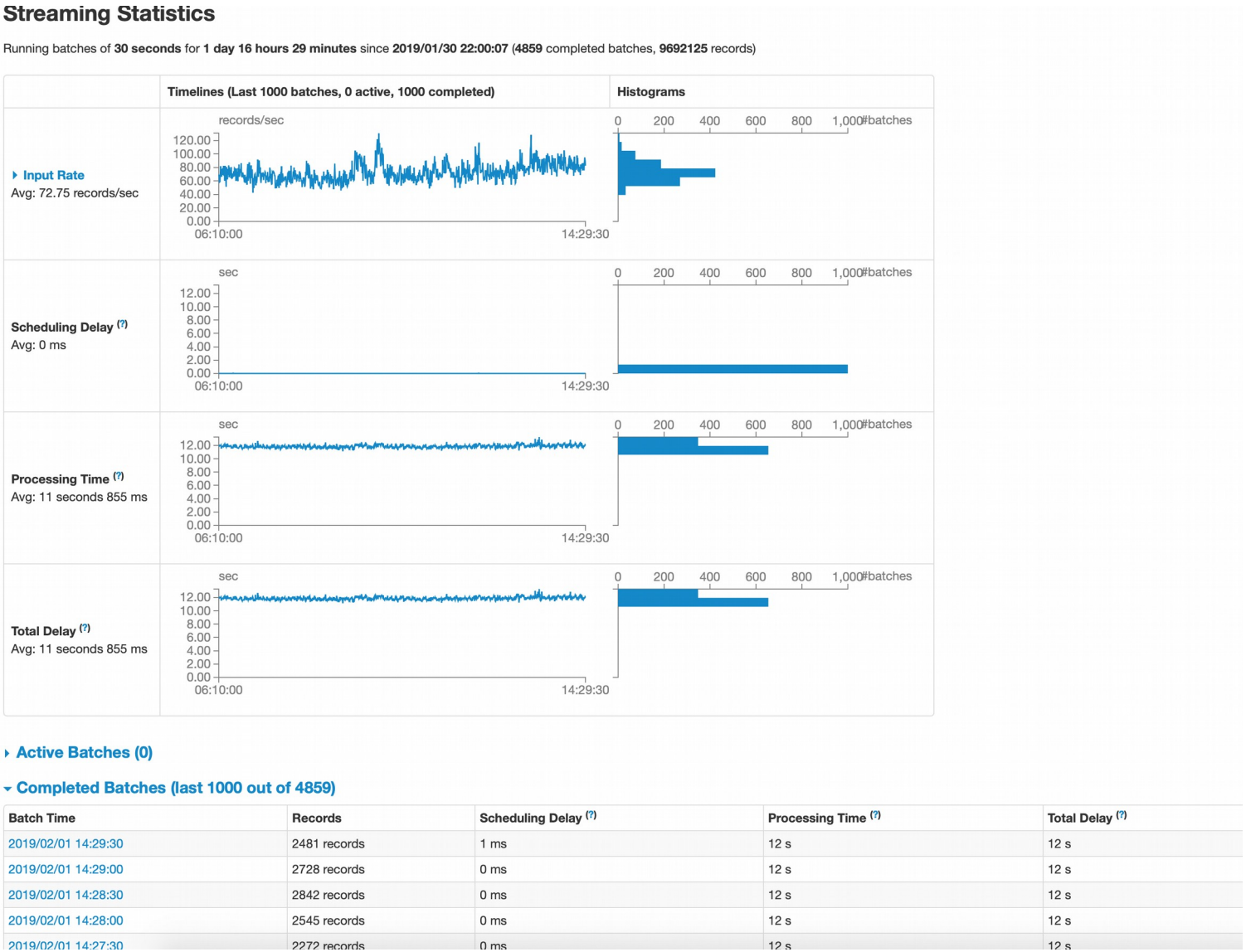
Dev deployment

- Consuming about **10%** of average daily traffic (~3M requests)
- **Multiplied by 10** on the Baskerville end to simulate the actual traffic
- Running on a **standalone single node spark cluster** with 1 worker, 4 executors with 2 cores and 6GB RAM each.
- **Performance tuning**: Java GC, Spark parameters, worker numbers, database tuning ...
- About **30M requests per day**
- Processing the data in a **2-minute time window** within ~30sec – about **¼ of the time window**

Model Development

- Feature selection, hyper-parameter optimization (Grid search)
- Training / Testing – gathering datasets and attacks

Baskerville - Current state



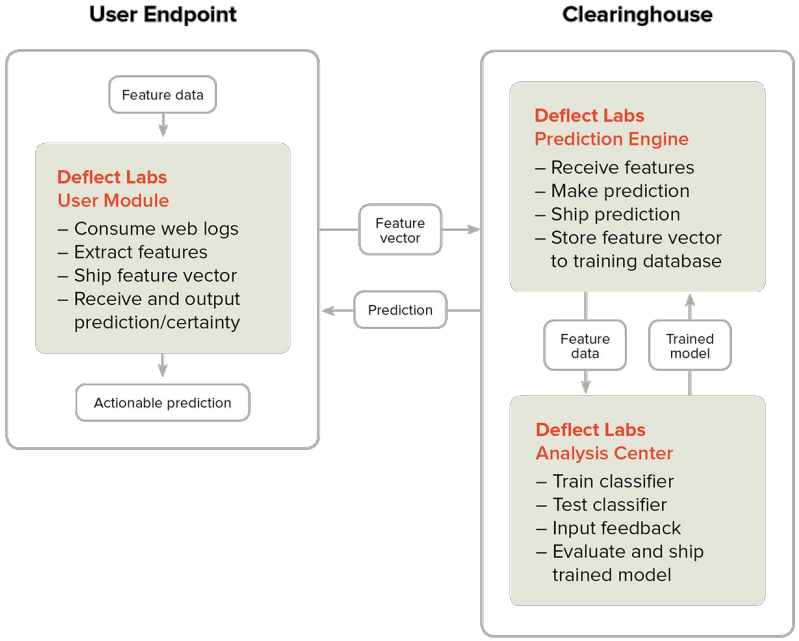
Baskerville: The next steps

- Deployment within April – Probation period
- Baskerville Dashboard
- On-going model development
- Feedback mechanism
- Release and Open Source by the end of Q2
- Create the challenge / ban communication component
- Work on k8s deployment for the spark cluster*

Baskerville - The next steps

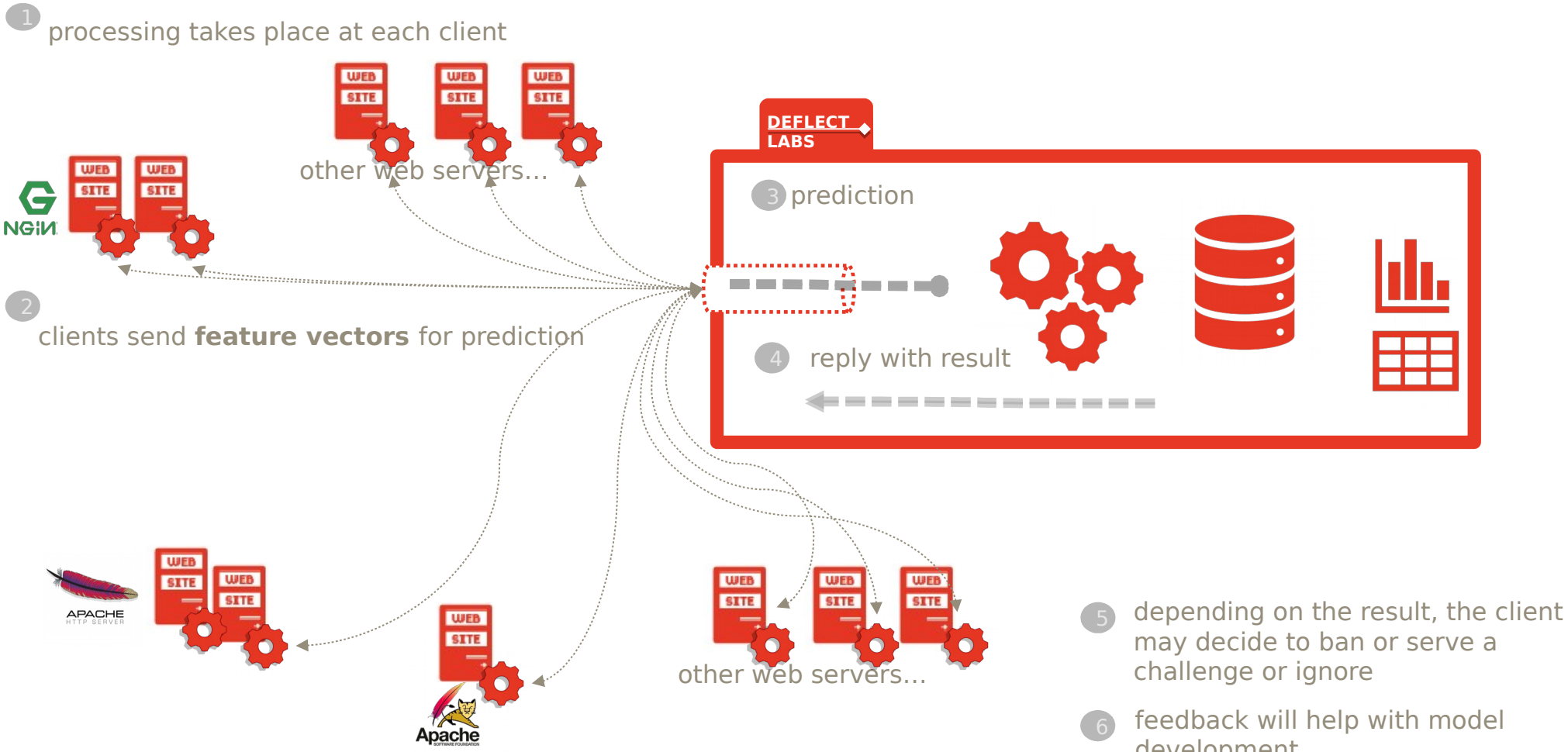
Further down the road: Information Sharing and Analysis Center (ISAC)

- **Clients run part of Baskerville**: the processing engine
- **Prediction** with the degree of confidence is **served by ISAC**
- Clients chose to use or not the prediction, e.g. ban or serve a challenge to the IP with the potentially malicious intent



Baskerville: The next steps

Further down the road: ISAC



Baskerville: The next steps

Further down the road: ISAC

Pros

- **No sensitive data sharing** - just the feature vectors are enough
- With a **feedback mechanism** we will be able to expand the training dataset and improve the model

Cons & Challenges

- The **size of the infrastructure** to be fast enough
- Convert part of Baskerville - the processing engine - to a “plugin” that can be used on the client side

Any Questions?

Website: https://docs.deflect.ca

GitHub: https://deflectca.github.io

- **Twitter:** @equalitie
- **Email:** info@equalit.ie
- **Public launch:** Fall 2019