Practical Machine Learning

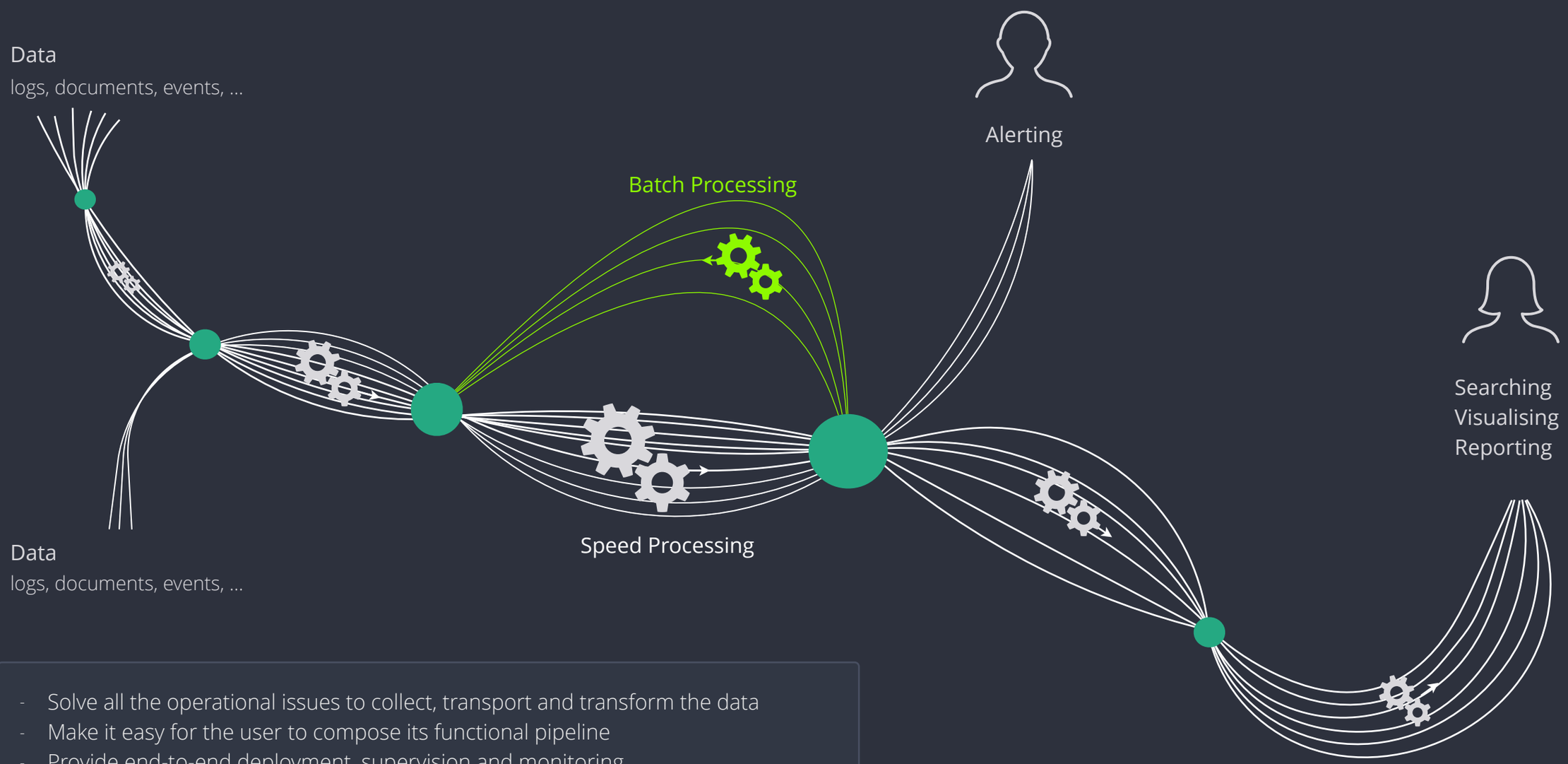
PunchPlatform team
Thales

Agenda

- Starting From Log Management
- Moving To Machine Learning
- Challenges
- Thanks

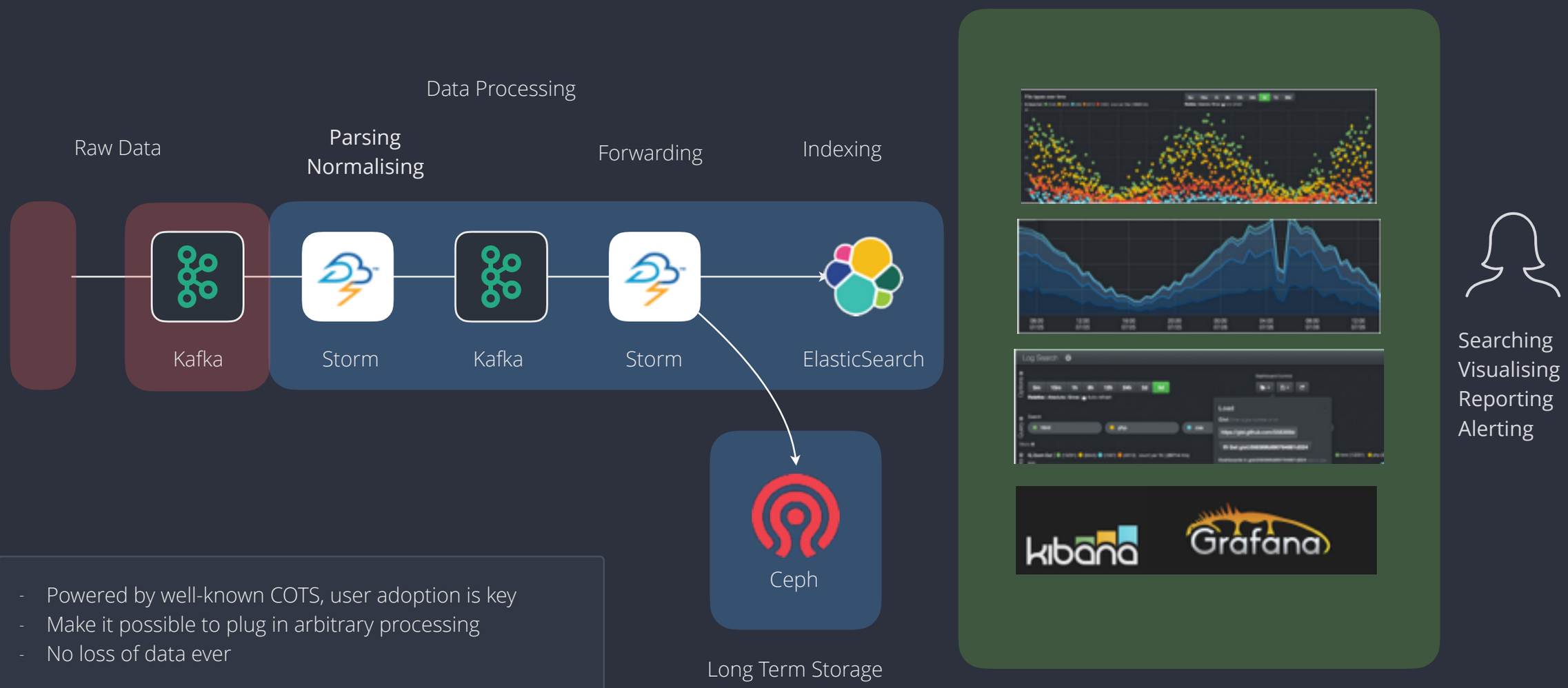
Starting From Log Management

Starting From Log Management



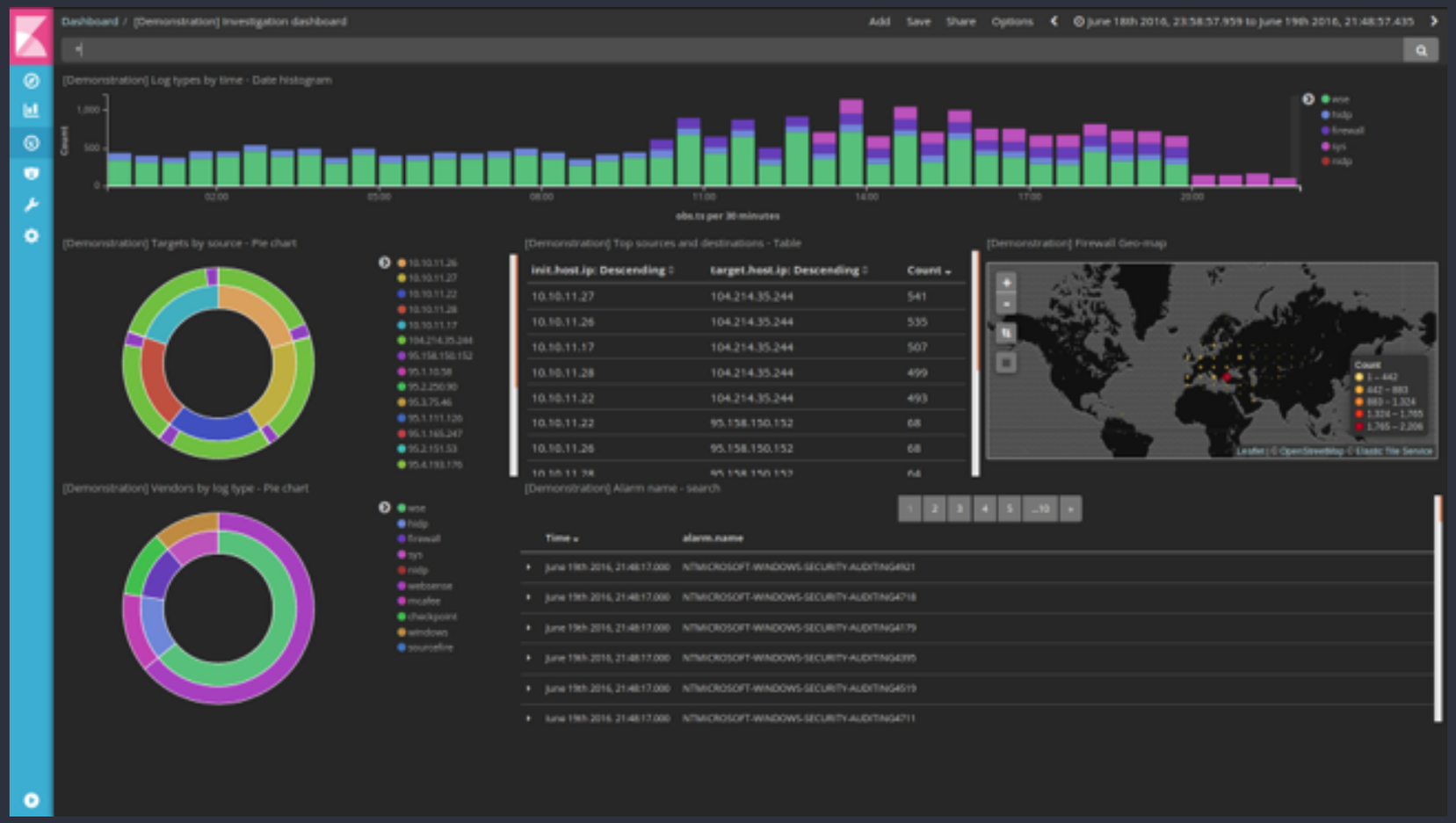
- Solve all the operational issues to collect, transport and transform the data
- Make it easy for the user to compose its functional pipeline
- Provide end-to-end deployment, supervision and monitoring

Log Management Technical Pipeline



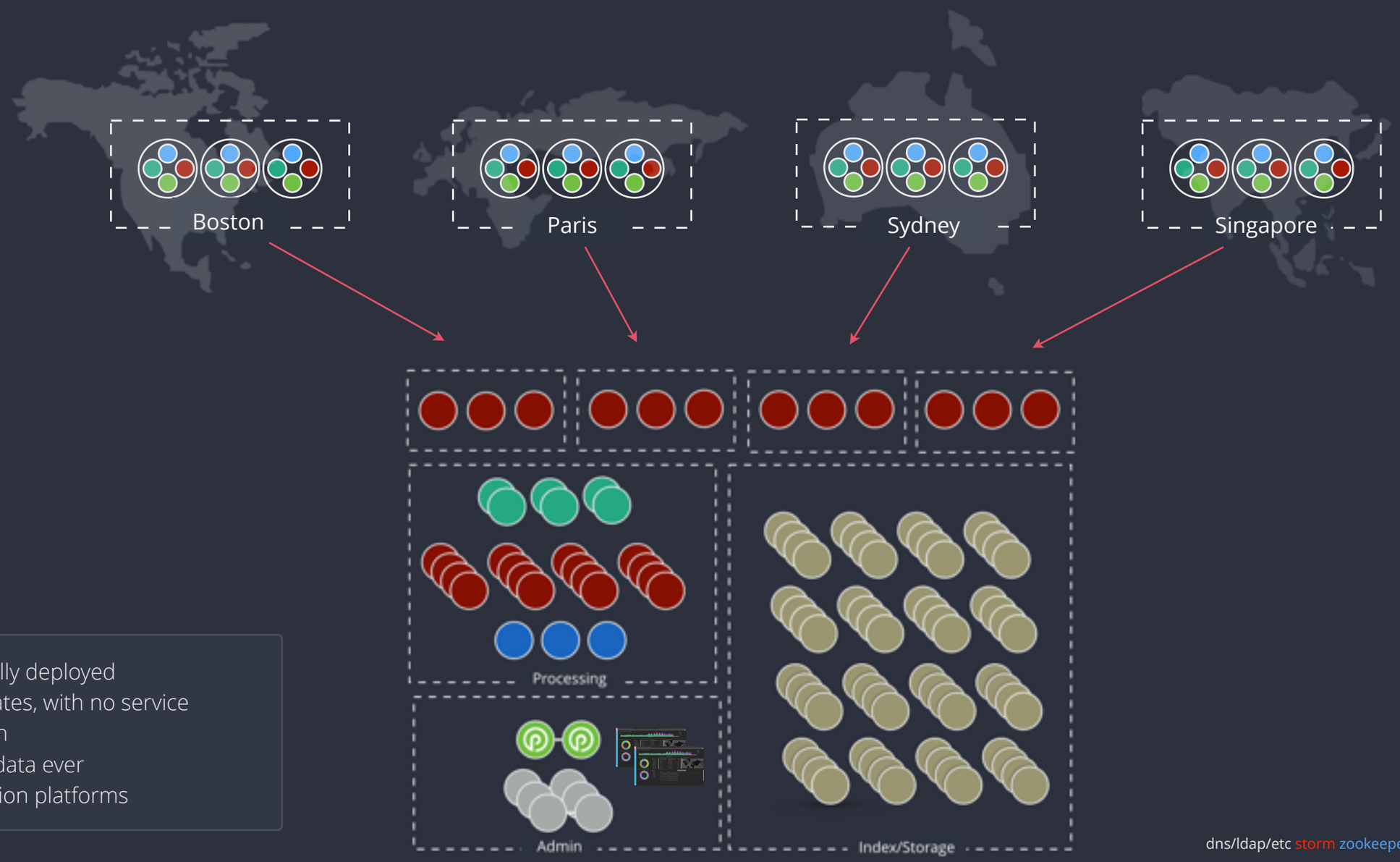
- Powered by well-known COTS, user adoption is key
- Make it possible to plug in arbitrary processing
- No loss of data ever

End User Experience



- Forensics : searching, reporting, aggregating
- Real-time, dynamic
- Already a data scientist starter tool

CyberSecurity Platforms Architecture



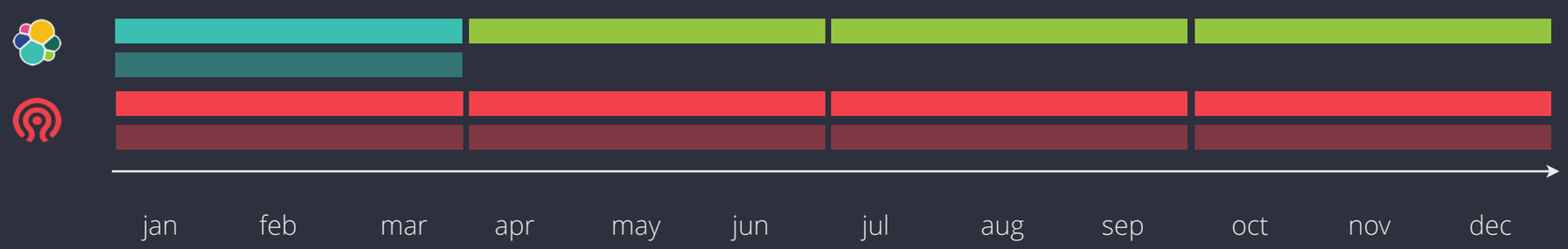
- automatically deployed
- yearly updates, with no service interruption
- No loss of data ever
- 16 production platforms

dns/ldap/etc storm zookeeper kafka elasticsearch ceph

Data is at hand

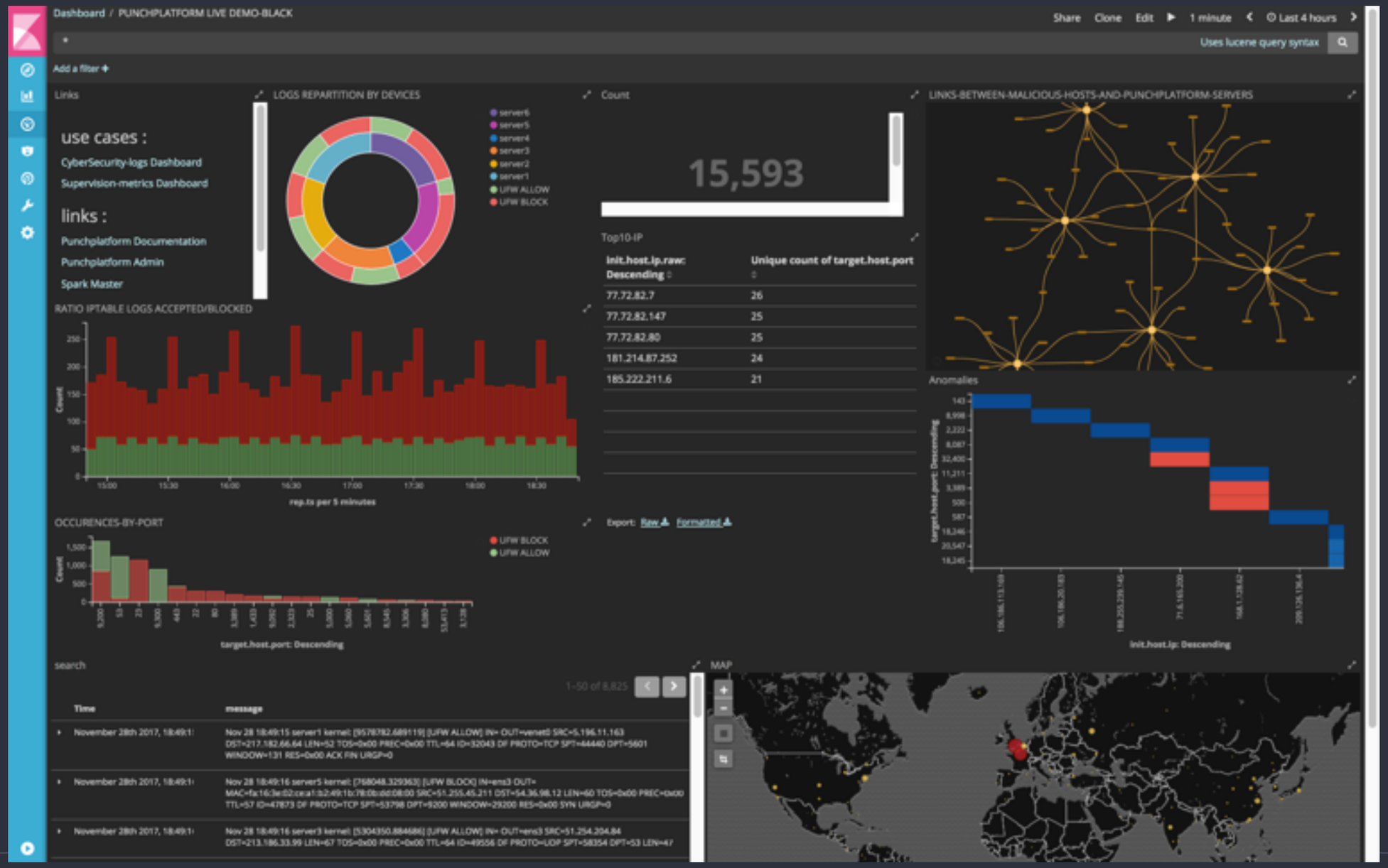
- 1 year of Indexed and normalised logs in Elasticsearch
- 1 year of compressed raw logs in distributed object storage
- days (up to a month) of normalised data in Kafka

The Punchplatform architecture and connectors make it simple and safe to deploy arbitrary processing on the data. Either batch or real-time (streaming).

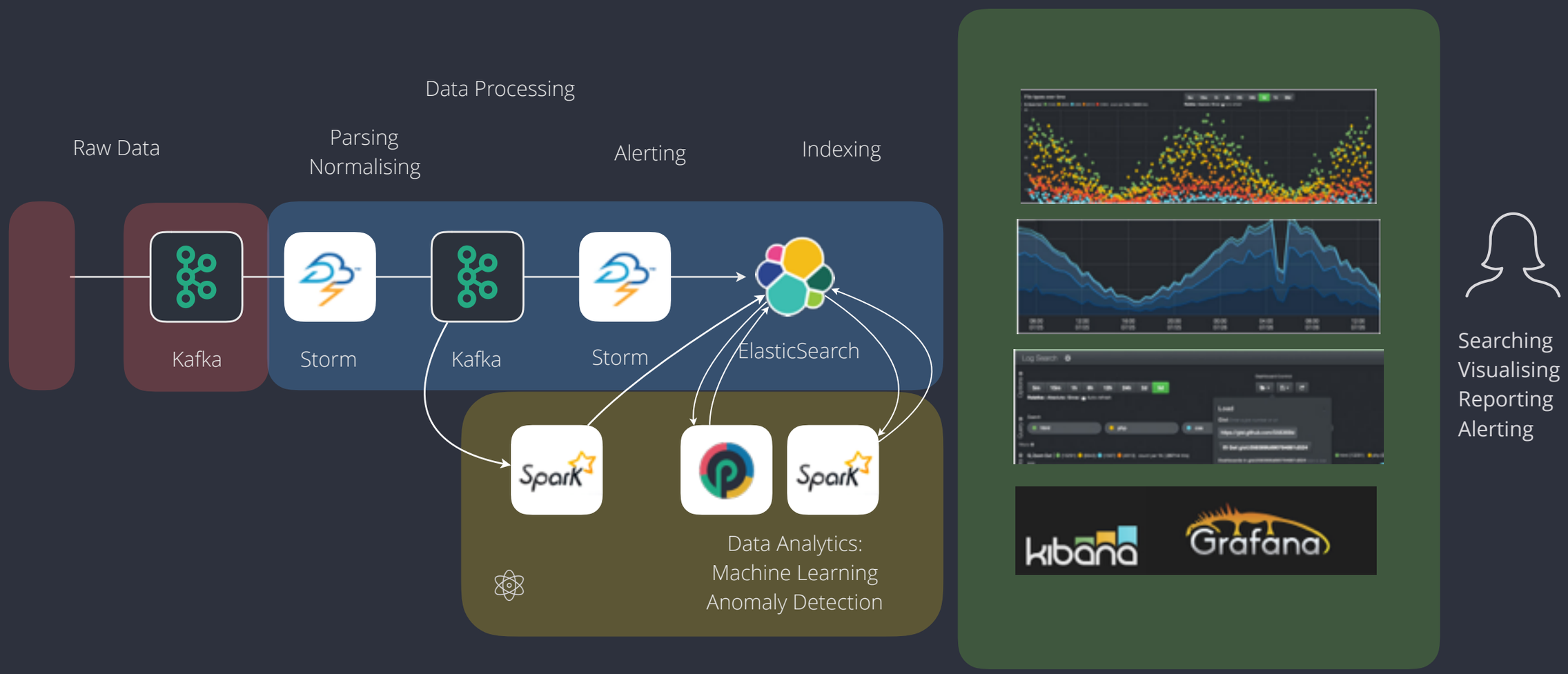


1-3 months are replicated : Each log is stored on 2 (or more) Elasticsearch servers.
4-12 months are not replicated : but online and indexed in ElasticSearch.
1-12 months are replicated. Each log is stored on 2 (or more) servers.

Live Demo


Dashboard / PUNCHPLATFORM LIVE DEMO-BLACK
Share
Clone
Edit
1 minute
Last 4 hours
Uses lucene query syntax
Add a filter
Links
use cases :
CyberSecurity-logs Dashboard
Supervision-metrics Dashboard
links :
Punchplatform Documentation
Punchplatform Admin
Spark Master
LOGS REPARTITION BY DEVICES
server6
server5
server4
server3
server2
server1
UFW ALLOW
UFW BLOCK
Count
15,593
Top10-IP
init.host.ip.raw: Descending
Unique count of target.host.port
77.72.82.7 26
77.72.82.147 25
77.72.82.80 25
181.214.87.252 24
185.222.211.6 21
LINKS-BETWEEN-MALICIOUS-HOSTS-AND-PUNCHPLATFORM-SERVERS
RATIO IPTABLE LOGS ACCEPTED/BLOCKED
Count
rep.ts per 5 minutes
Anomalies
target.host.port: Descending
init.host.ip: Descending
OCCURENCES-BY-PORT
UFW BLOCK
UFW ALLOW
Count
target.host.port: Descending
Export: Raw Formatted
search
1-50 of 8,825
Time
message
November 28th 2017, 18:49:1
Nov 28 18:49:15 server1 kernel: [9578782.689119] [UFW ALLOW] IN= OUT=venet0 SRC=5.196.11.163 DST=217.182.66.64 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=32043 DF PROTO=TCP SPT=44440 DPT=5601 WINDOW=131 RES=0x00 ACK FIN URGP=0
November 28th 2017, 18:49:1
Nov 28 18:49:16 server5 kernel: [768048.329363] [UFW BLOCK] IN=ens3 OUT= MAC=fa:16:3e:02:ce:a1:b2:49:1b:78:0b:dd:08:00 SRC=51.255.45.211 DST=54.36.98.12 LEN=60 TOS=0x00 PREC=0x00 TTL=57 ID=47873 DF PROTO=TCP SPT=53798 DPT=9200 WINDOW=29200 RES=0x00 SYN URGP=0
November 28th 2017, 18:49:1
Nov 28 18:49:16 server3 kernel: [5304350.884686] [UFW ALLOW] IN= OUT=ens3 SRC=51.254.204.84 DST=213.186.33.99 LEN=67 TOS=0x00 PREC=0x00 TTL=64 ID=49556 DF PROTO=UDP SPT=58354 DPT=53 LEN=47
MAP

Live Demo Explained

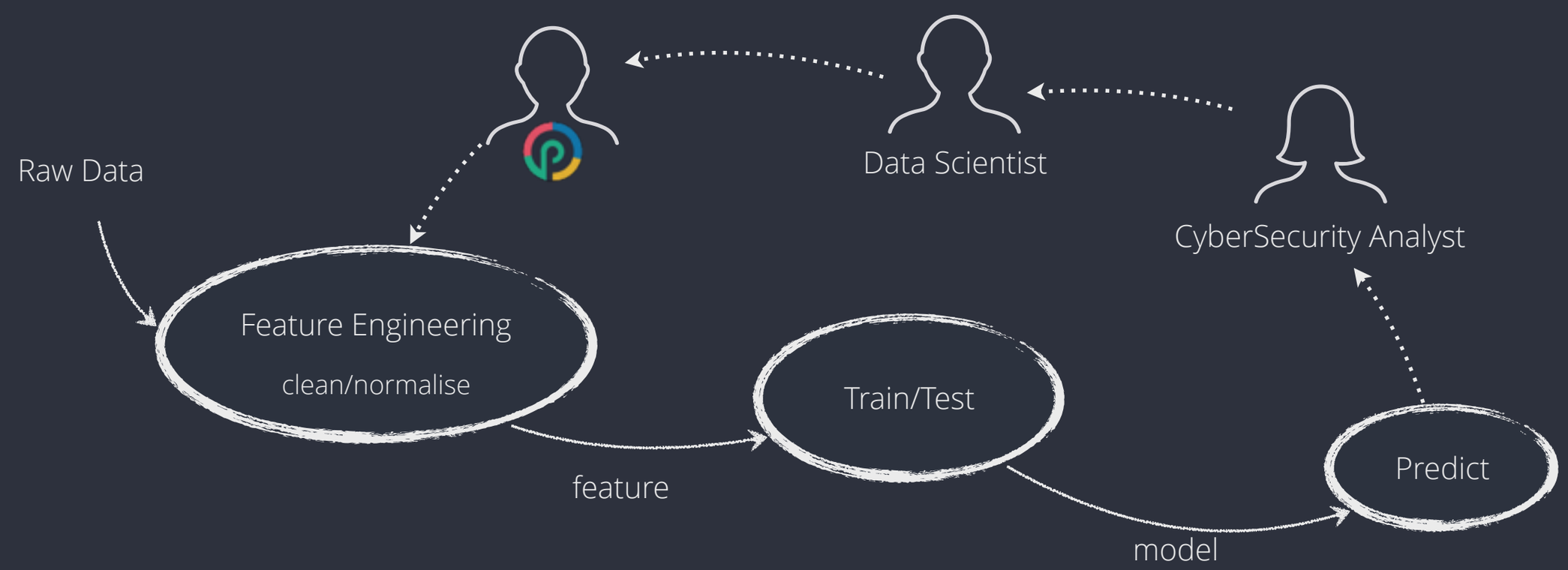


Challenges

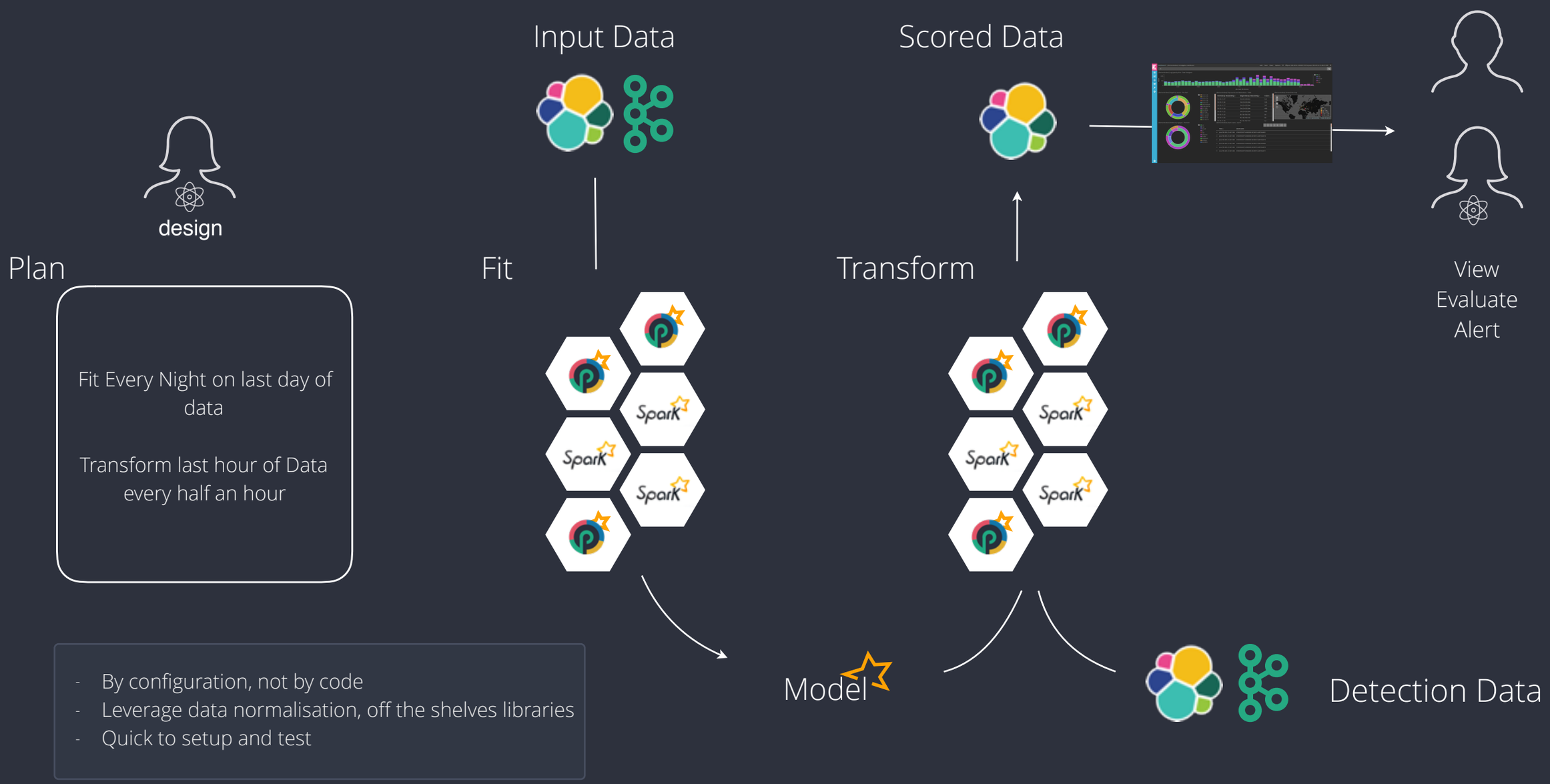
Our Process

- solve the data access and ownership issue
- small, agile, integrated team
- well defined process
- clear and shared vision of achievable steps : MVPs

..... and something like a



Making (Spark) ML simpler



- By configuration, not by code
- Leverage data normalisation, off the shelves libraries
- Quick to setup and test

Fit Job Example

Punch Stage
Field selection, enrichment, ..

Spark Stage
K-Means, Regression, …

Model Output
Save the computed Model

Data Input
ElasticSearch last day of firewall logs

Spark Stage
SparkQL

Spark Stage
Filter

- This is defined in a plain Json configuration file

Transform Job Example

Punch Stage
Field selection, filtering, …

Spark Stage
(say) K-Means/Regression/…

Data Output
Save the scored data to ElasticSearch

Model Input
from Fit Job

Live Input
Real Time Firewall Logs

Spark Stage
SparkQL

Spark Stage
select fields

- This is defined in a plain Json configuration file

Leveraging Spark MLib

https://spark.apache.org/mllib/

ML Algorithms

Classification:
logistic regression, naive Bayes,...
Regression:
generalized linear regression, survival regression,...
Decision trees, random forests,
gradient-boosted trees
Recommendation:
alternating least squares (ALS)
Clustering:
K-means, Gaussian mixtures (GMMs),...
Topic modeling:
latent Dirichlet allocation (LDA)
Frequent itemsets, association rules,
sequential pattern mining

ML Workflow

Feature transformations:
standardization, normalization, hashing,...
ML Pipeline construction
Model evaluation and hyper-parameter tuning
ML persistence:
saving and loading models and Pipelines

Utilities

Distributed linear algebra: SVD, PCA,...
Statistics: summary statistics, hypothesis testing,...

Conclusions

When embarking on AI projects you dramatically improve your chances of producing value by :

- *Operating in a build now, learn as you go fashion. Truly sophisticated products are arrived at via iteration and variation; not naive designs steeped in theory;*
- *Using nascent discoveries only in the context of a working product;*
- *Encouraging Agility from your Data Scientists as much as your developers and product managers;*
- *Closing the gap between lab and factory wherever possible, favoring quick and lean solutions that grow more valid with time;*
- ***Leveraging the machine learning already available in open source tools, only coding from the ground up when absolutely necessary;***
- *Passing user feedback into your data pipelines by exposing imperfect models to end users early.*

(Sean McClure)

Thanks !

http://doc.punchplatform.com

http://punchplatfom.io

http://kibana.punchplatform.com