Zentralisiertes Log Management

Ein Erfahrungsbericht

Michael Mimo Moratti
mimo@mimo.ch

Wer bin ich…

Michael Mimo Moratti, mimo@mimo.ch, jmimo on Github

Java, C, C++, Python, Lua, Bash, Linux, Web zeugs, …
Architektur, WAF / Entry Server, Code review, Kernel hacking.

3 RLK Installationen aufgebaut

Die Größte Installation: CSS Versicherung (+740 mio. Events)

Und wenn ich gerade mal nicht am Schreibtisch stehe dann:

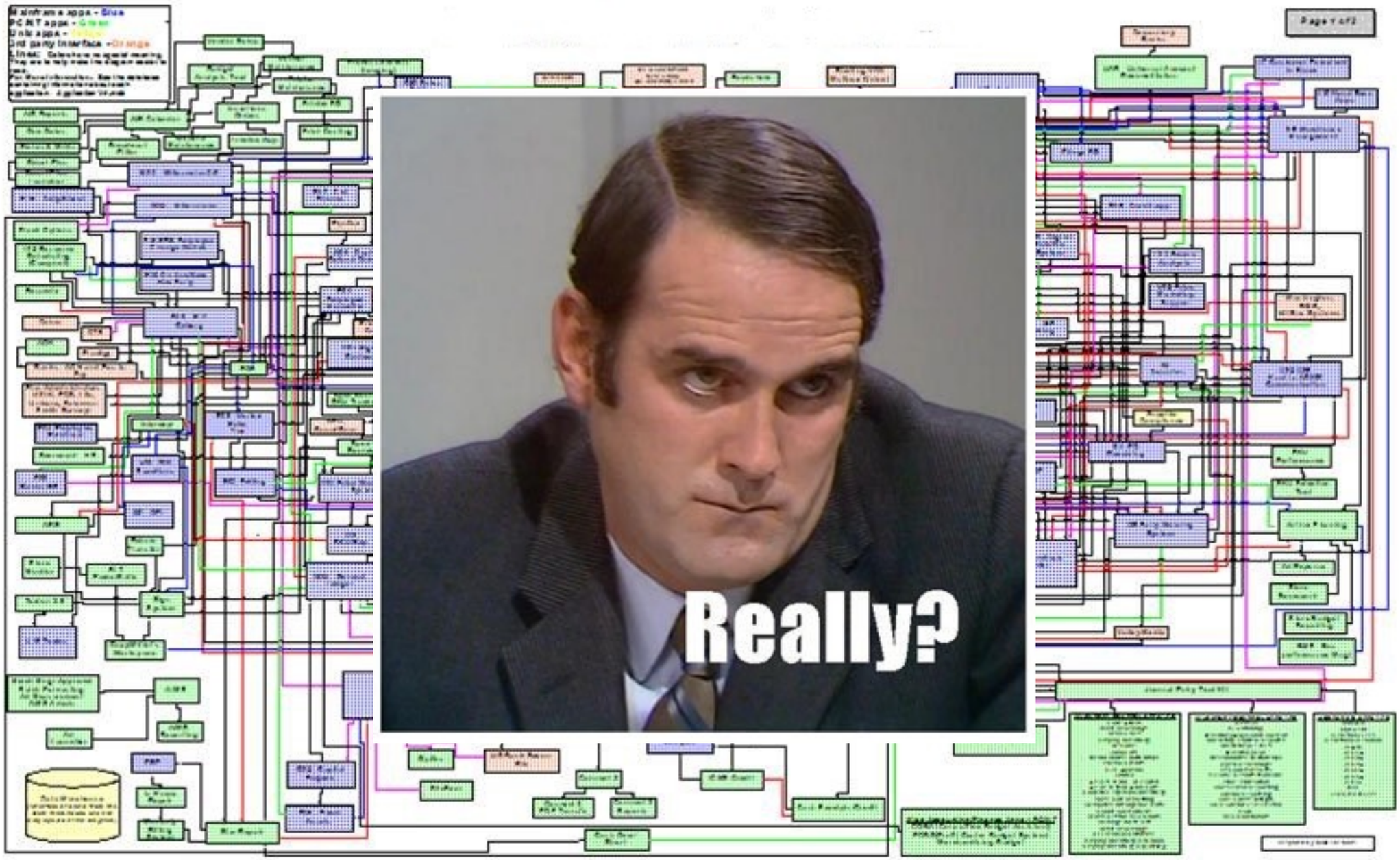
Warum sollten wir das tun?

Ich hatte gerade eine Fehlerantwort, was geschah da genau?

Irgendetwas geht nicht, schau doch bitte schnell nach!

Kannst du bitte die Logs von Gestern zwischen 14:36 und 15:12 auf Fehler prüfen!

Kannst du bitte alle Logs der letzen Woche auf Fehler überprüfen!


Really?

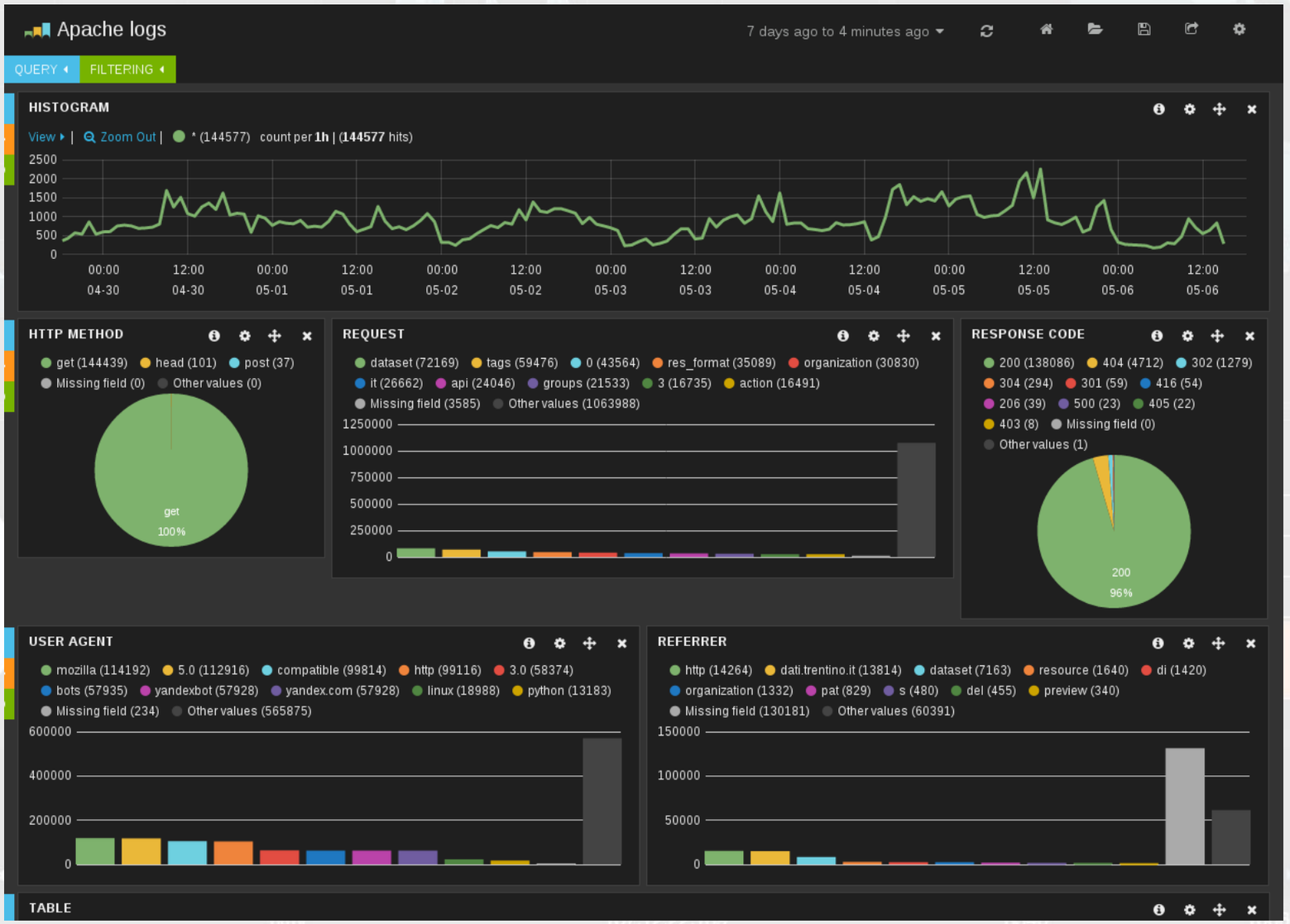
Und jetzt…

- Werden wir uns an einer Vielzahl von Systemen anmelden.
- Mit “find”, “grep” und anderen Tools die Logfiles nach Anzeichen von Fehler durchsuchen.
- Und versuchen die gefundenen Daten zu Korrelieren und Interpretieren.

```
Jan 10 00:51:37 BigThougt.local configd[25]: LINKLOCAL en0: parent has no IP
Jan 10 00:51:37 BigThougt.local networkd[198]: +[NETLedBelly stopFastFail] Clearing ledbelly failure cache
Jan 10 00:51:37 BigThougt.local configd[25]: network changed: v4(en0-:192.168.0.20) DNS- Proxy-
Jan 10 00:51:37 BigThougt.local discoveryd[50]: Basic Warn DD_Warn: Corrupt NSEC RDATA size
Jan 10 00:51:37 BigThougt.local discoveryd[50]: Basic WABServer NetResolverEvent no resolvers, resetting domains
Jan 10 00:51:37 BigThougt.local discoveryd[50]: Basic Bonjour Failed to delete registration recordId=518 error=4
Jan 10 00:51:38 BigThougt.local networkd[198]: +[NETLedBelly stopFastFail] Clearing ledbelly failure cache
Jan 10 00:51:38 BigThougt.local configd[25]: network changed: v4(en0+:192.168.0.20) DNS+ Proxy+ SMB
Jan 10 00:51:38 BigThougt.local discoveryd[50]: Basic Sockets,Warn UDS FD=90 ERROR: Send failed errno=32
Jan 10 00:51:39 BigThougt.local discoveryd[50]: Basic Bonjour Failed to delete registration recordId=522 error=4
Jan 10 00:51:46 BigThougt.local sharingd[7192]: 00:51:46.555 : SDStatusMonitor::kStatusWirelessPowerChanged
Jan 10 00:51:46 BigThougt.local sharingd[302]: 00:51:46.555 : SDStatusMonitor::kStatusWirelessPowerChanged
Jan 10 00:51:46 BigThougt.local sharingd[7192]: 00:51:46.604 : SDStatusMonitor::kStatusWirelessPowerChanged
Jan 10 00:51:46 BigThougt.local sharingd[302]: 00:51:46.604 : SDStatusMonitor::kStatusWirelessPowerChanged
Jan 10 00:51:47 BigThougt.local ntpd[199]: wake time set +0.335678 s
Jan 10 00:51:55 BigThougt.local configd[25]: [0x7fe689e54af0] [m]DNS query timeout (query time = 13.752251), [46TE]
Jan 10 00:52:10 BigThougt.local configd[25]: [0x7fe689d43ee0] [m]DNS query timeout (query time = 29.040382), [46TE]
Jan 10 00:52:13 BigThougt.local configd[25]: [0x7fe689d42920] [m]DNS query timeout (query time = 32.261717), [46TE]
```

Aber wirklich wollen wir…

Eine Analyse über alle Logdaten aller Systeme mit Hilfe einer Query und Filter basierten Visualisierung im Web-browser machen.



Nur, was braucht es dazu…

Klassisches (manuelles) Vorgehen

Sammeln *(scp / ftp …)*

Analyse *(find / grep / sed / awk …)*

Visualisierung *(Excel / Birt / LogMX …)*

ELK Stack

Elasticsearch *(NoSQL / speichern / suchen)*

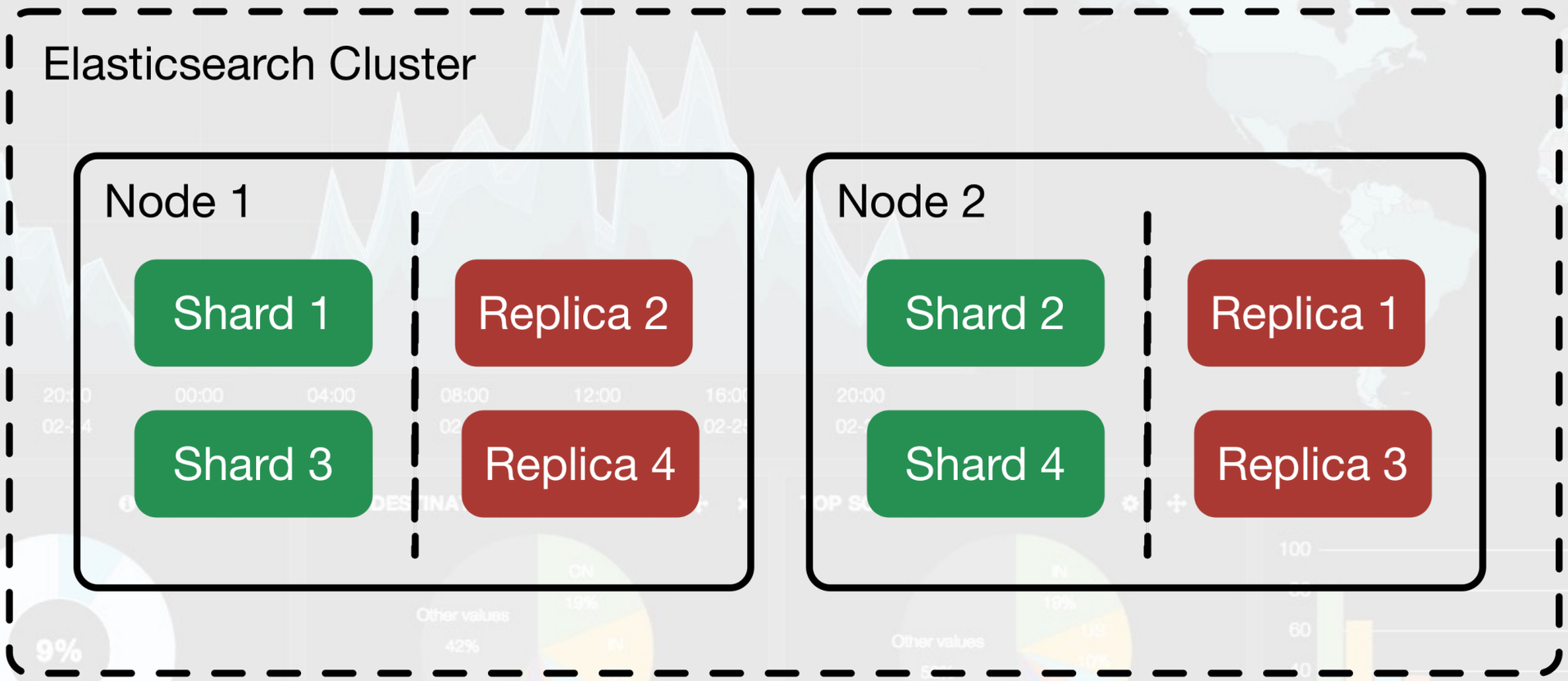
Logstash *(sammeln / transformieren)*

Kibana *(visualisieren)*

Elasticsearch

- NoSQL “Big data” Volltext Suchmaschine
- Java / basiert auf “Apache Lucene"
- Suchen und Indexieren
- Verteilt (shards & copies)
- Clustering
- API — JSON / RESTful

Elasticsearch



Logstash

- “Schweizer Sackmesser” zum Sammeln, Transformieren und Weiterleiten von Logdaten.

- JRuby

- MiMo (multiple in / multiple out)

- Sammeln

- Analysieren (Filter)

- Speichern / Weiterleiten

Logstash

Input

- File
- Syslog
- Redis
- Forwarder
- …

Filter

- JSON
- Grok
- Drop
- Mutate
- …

Output

- Elasticsearch
- Redis
- StatsD
- Graphite
- …

Kibana

- AngularJS
- Drill-down / Roll-up
- Visualisierung (Histogram / Pie / Term….)

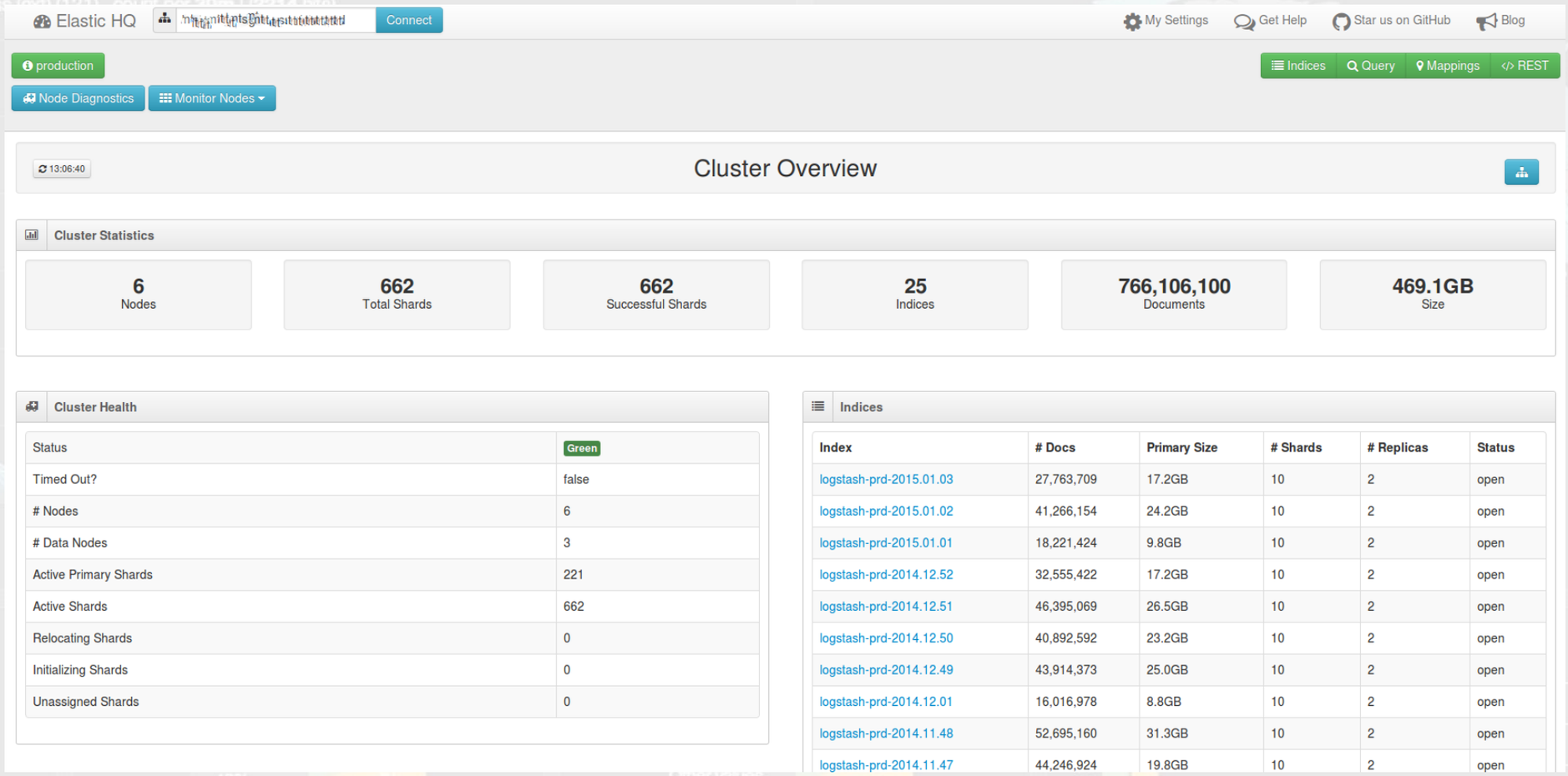
Version 4.0 ist in der pipeline, (beta 3) und wird aus dem leichtgewichtigen Javascript Frontend einen Server Prozess machen welcher dann aber auch (dank Elasticsearch 1.4) Daten aggregieren kann.

Kibana


Logstash Search
2 days ago to 3 minutes ago
QUERY
bytes:[0 TO 4000000] AND @tags:success
FILTERING
EVENTS OVER TIME
View | Zoom Out | html (ext) (888) php (ext) (659) png (ext) (317) gif (ext) (229) css (ext) (121) count per 30m | (2214 hits)
60
50
40
30
20
10
0
00:00 02-24
04:00 02-24
08:00 02-24
12:00 02-24
16:00 02-24
20:00 02-24
00:00 02-25
04:00 02-25
08:00 02-25
12:00 02-25
16:00 02-25
20:00 02-25
MAP
REVENUE
22%
VOLUME
9%
TOP DESTINATIONS
Other values 42%
CN 19%
IN 16%
TOP SOURCES
Other values 52%
IN 19%
US 10%
GEO PAIRS
100
80
60
40
20
0
ALL EVENTS
0 to 100 of 500 available for paging
Fields
All (31) / Current (28)
Type to filter...
@message
@tags
@tags
geo.srcdst
extension
clientip
bytes
id
phpmemory
response
success,security MY:VN html 167.12.22.189 8540 1066 200
success,info IT:MM png 164.87.170.73 2045 1903 200
success,info AR:ES html 222.23.102.238 1801 1133 200
success,info IN:DZ html 138.226.66.81 7029 1801 200

Tools

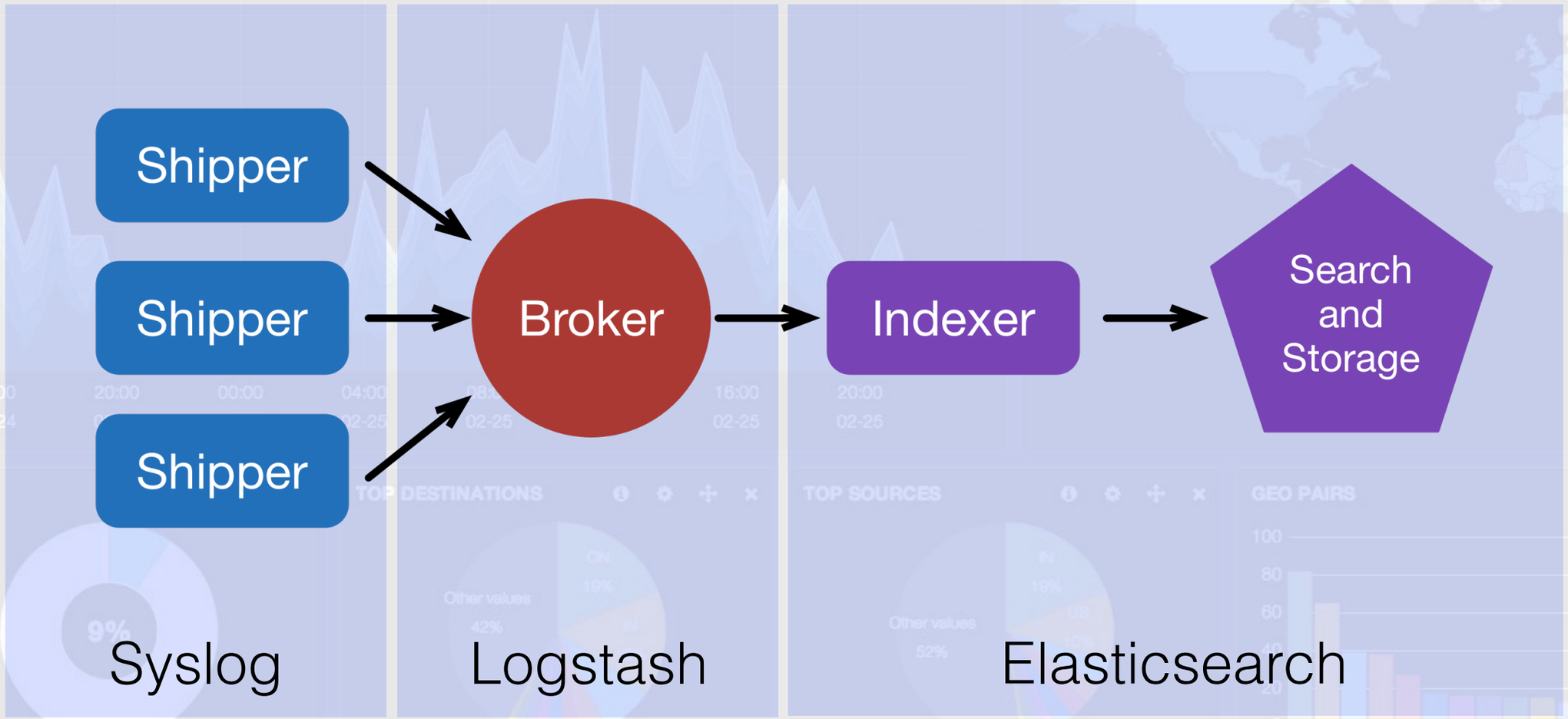
ElasticHQ



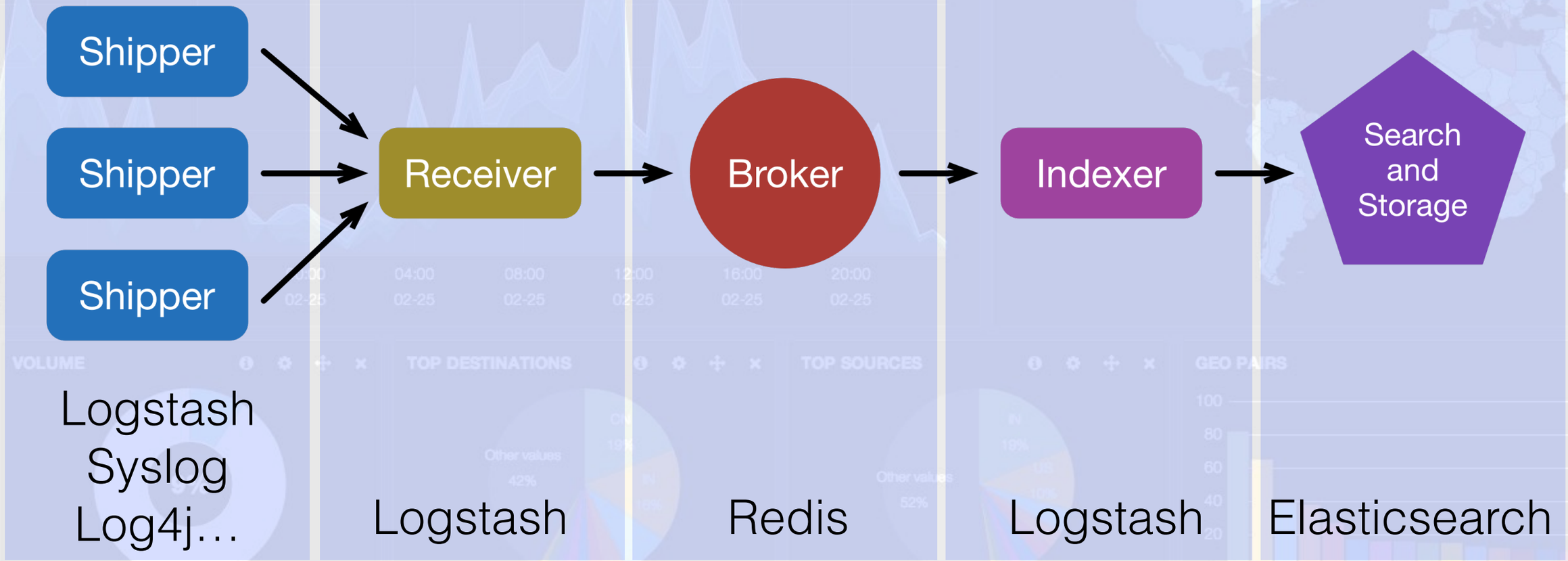
Curator

```
curator –host localhost –port 9200 delete –time-unit months –older-than 2
```

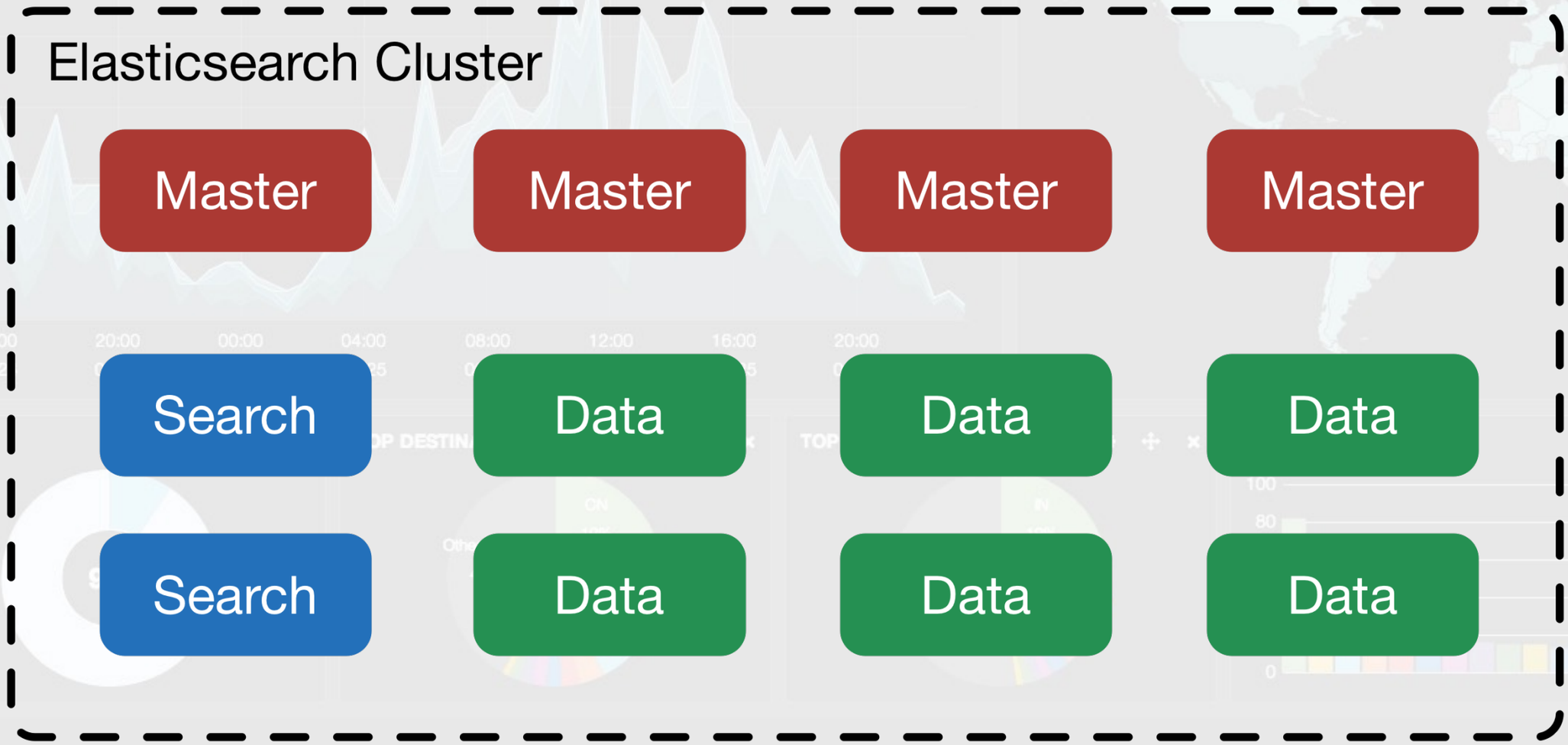
Architektur



Realistische Architektur :-)



Elasticsearch Node Architektur



Übliche Fallstricke

- Nicht genügend Abeitsspeicher (RAM, RAM, RAM…)
- Elasticsearch Server nutzt Swap Speicher!
- Langsamer Physischer Speicher.
- Java Memory Fragmentierung (Heap > 20GB)
- Elasticsearch mixed nodes (z.B.: master und data)
- Index Frequenz (Tag, Woche, Monat, Jahr)
- Shards und Copies (Split Brain)
- Logstash: CPU intensive Grok (regex) Ausdrücke.

Und nun…

- Die Daten sind zentral indexiert und können mit Volltextsuche analysiert werden.
- Wir haben die Möglichkeit uns eine Übersicht im Dschungel der Logdaten zu verschaffen.
- Aus den Daten können wir viel neues Lernen…

Anwendung

- Analyse von WAF Daten können Usability Probleme in Webapplikationen aufzeigen.
- Pattern welche durch das Korrelieren von Logdaten ersichtlich werden, können aufgedeckt und analysiert werden.
- Performance kann ausgewertet werden.
- Business Intelligence…

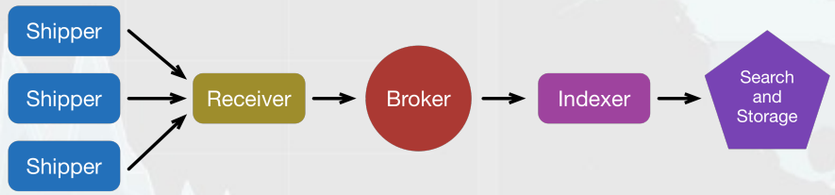
Demo

```
input {
    twitter {
        consumer_key => "vhFUbJ4vccROMirZesU9Br"
        consumer_secret => "v9xeZ0tCggwo0zqJWDuR02KhWnh1W2SLgPkQJmKGyF2GRzw"
        oauth_token => "51665235-TASGSSMzAFyjGlmeHd0mmkYrqJuUw50TZImZf"
        oauth_token_secret => "yf6PmWqOZJMwdvsUOxOd3zRNvIWpOqxWAmcaSe47R"
        keywords => [ "lauberhorn", "wengen", "abfahrt", "skirennen" ]
        full_tweet => true
    }
}
filter {
    prune {
        blacklist_names => [ "retweeted_.*" ]
    }
}
output {
    elasticsearch_http {
        host => "localhost"
        index => "lauberhorn"
    }
}
```

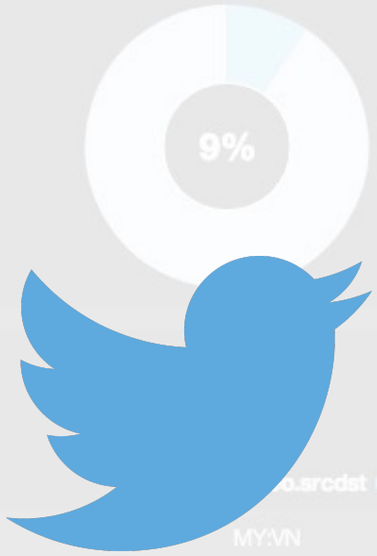
a fool with a tool is still a fool…


Any
Questions ?

elasticsearch.



DANKE



Elasticsearch Cluster

Master	Master	Master	Master
Search	Data	Data	Data
Search	Data	Data	Data