ELK-STACK BEI FRONTLINE

ERFAHRUNGEN AUS ZWEI JAHREN PRODUKTIVEM EINSATZ

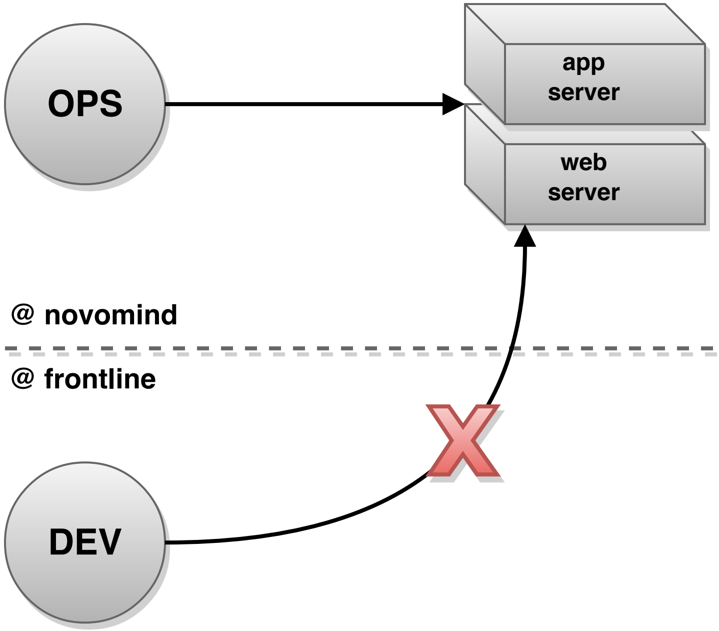
Timo Zingel / @pyspam

Jens Fischer / @jensfischerhh

frontlineshop

- Onlineshop für Streetfashion
  - 1986 als Musik-Mailorder gegründet
  - 1996 Streetfashion: *www.ziehdichan.de*
  - 2004 *www.frontlineshop.com*
- ca. 80 Mitarbeiter
- Scrum Team mit 6 Entwicklern

MOTIVATION



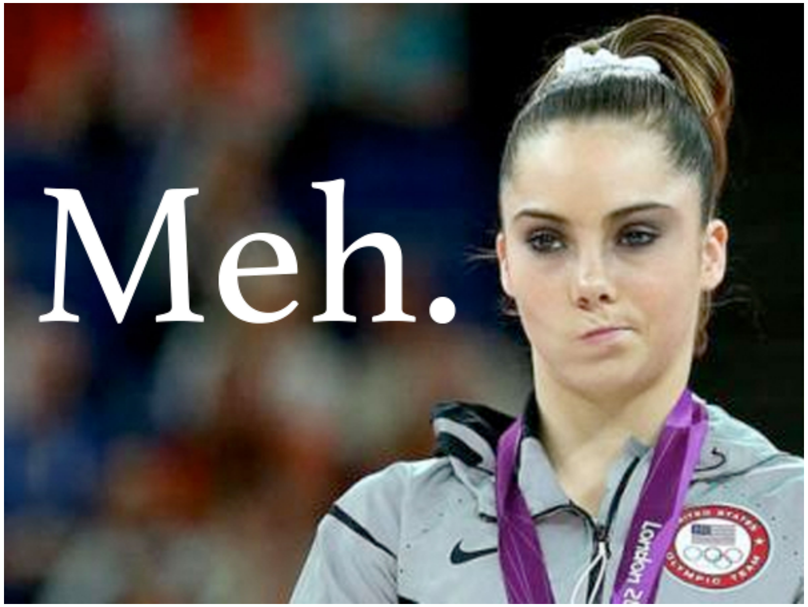
Ops @Novomind: Shell Zugriff

Devs @Frontline: **keinen** Shell Zugriff

BUG FIXING

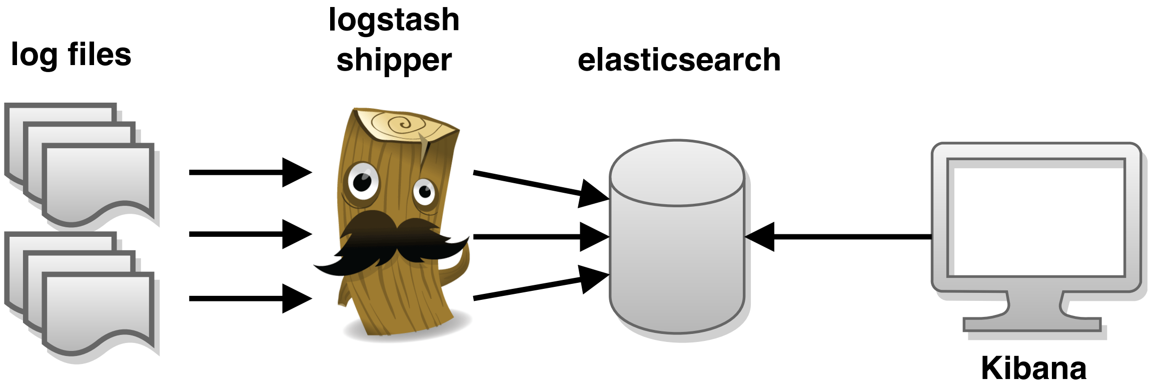
ABER BITTE MIT LOGS!

- Shell Zugriff?
- ad hoc anfordern?
- cron rsync?



ELK

- Elasticsearch
- Logstash
- Kibana



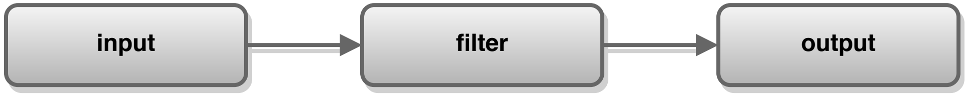
ERSTE SCHRITTE

- Q1 2013
  - Elasticsearch 0.90.x
  - Logstash 1.1.x
  - Kibana 2.x
- Analyse ERP Latenz
  - Logfile Historie
  - CSV Export

SCHNELLE ERGEBNISSE

- Wenig Aufwand
- tolle Visualisierung
- Naive Herangehensweise

LOGSTASH



LOGSTASH PLUGINS

INPUT

- file
- redis
- stdin
- syslog
- heroku
- ...

FILTER

- grok
- multiline
- geoip
- useragent
- csv
- ...

OUTPUT

- elasticsearch
- redis
- statsd
- graphite
- irc
- ...

logstash plugin documentation (logstash.net/docs/1.4.2)

community plugins (github.com/logstash-plugins)

LOGSTASH CONFIG

```
# simple.conf
input {
    stdin {
    }
}
output {
    stdout {
        codec => rubydebug
    }
}
```

```
echo "Hello Logstash" | ./bin/logstash -f simple.conf
```

```
{
       "message" => "Hello Logstash",
      "@version" => "1",
    "@timestamp" => "2015-03-15T10:00:42.149Z",
          "host" => "jfischer-mac"
}
```

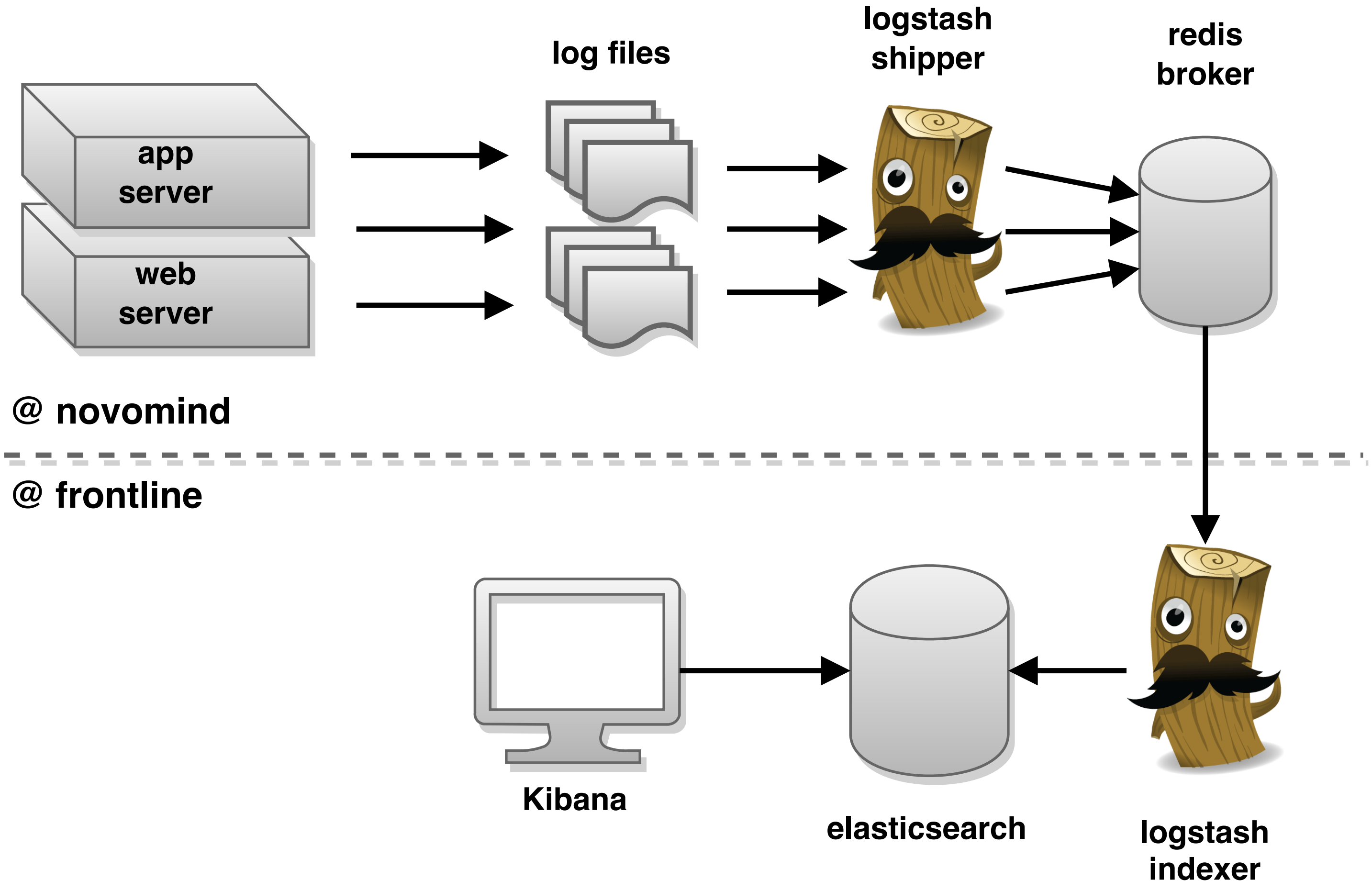
LIVE DEMO:

HELLO LOGSTASH

LIVE LOGS

- viele Log-Dateien
- unstrukturiert

ELK MIT BROKER



FILTER

- file-input liefert ein Event pro Zeile
- Log-Event mit Regex parsen
    - grok
    - multiline
- strukturiert in ES speichern

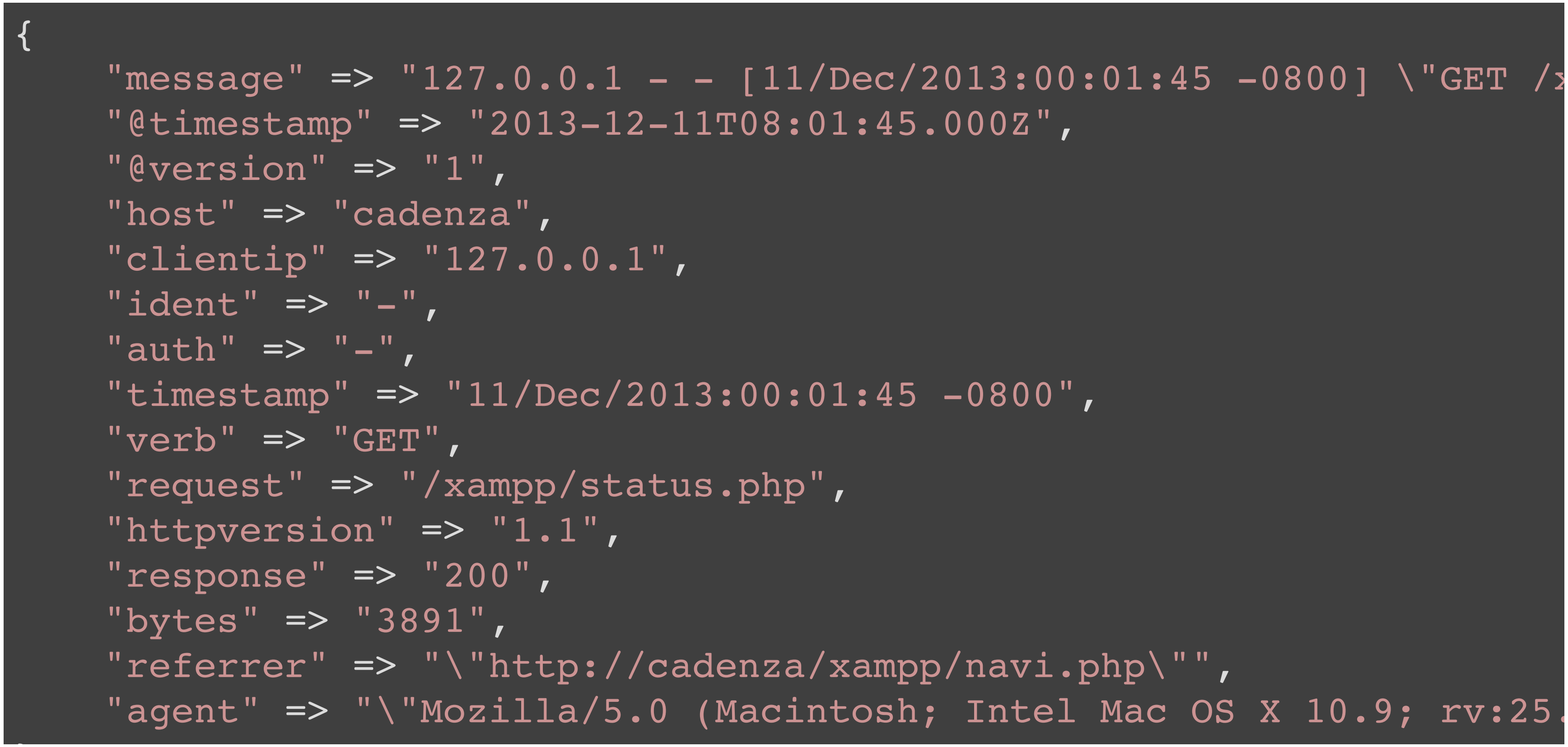
APACHE

```
127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] "GET /xampp/status.php HTT
```

```
input { stdin { } }

filter {
    grok {
        match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    date {
        match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
}

output {
    elasticsearch { host => localhost }
    stdout { codec => rubydebug }
}
```

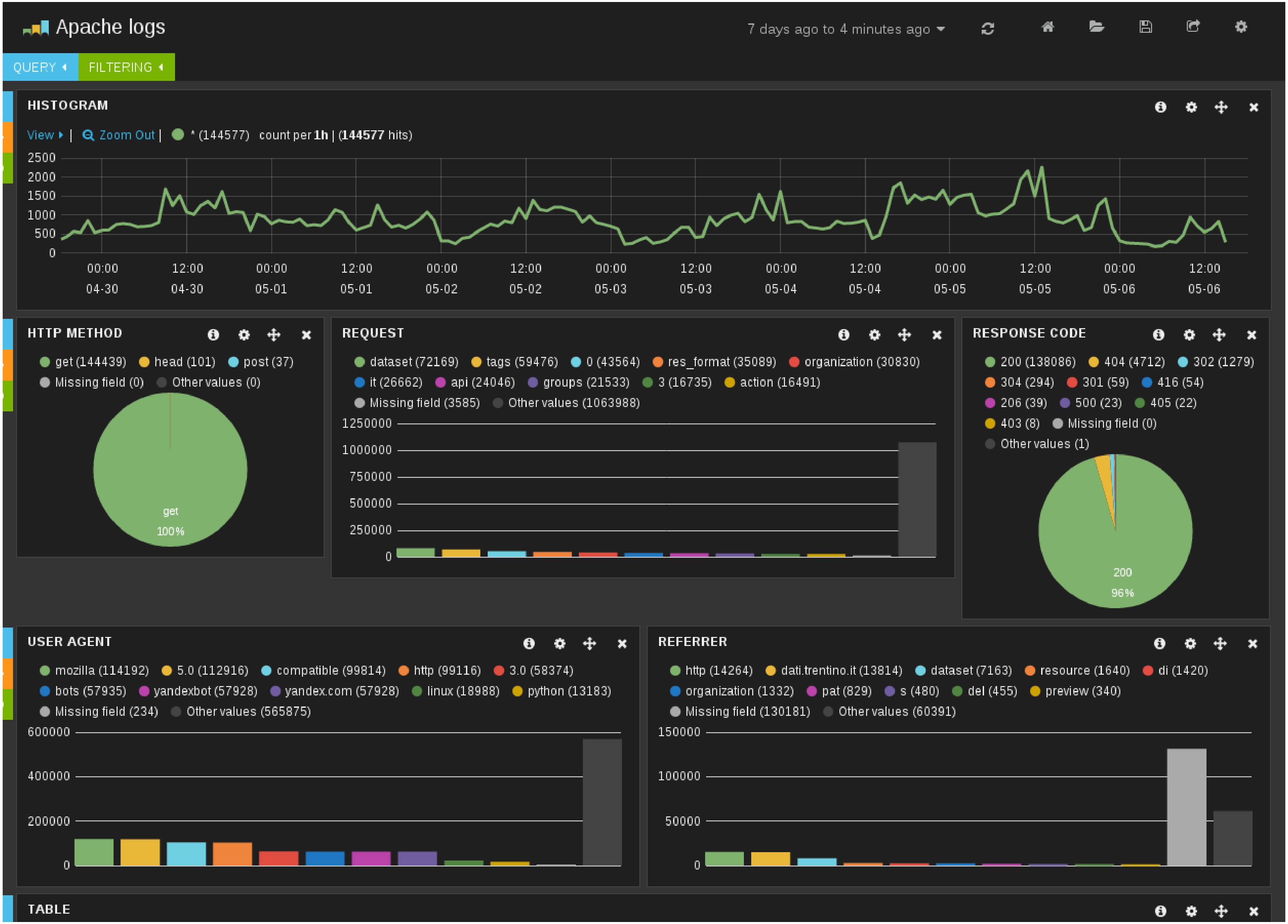


```
{
    "message" => "127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] \"GET /x
    "@timestamp" => "2013-12-11T08:01:45.000Z",
    "@version" => "1",
    "host" => "cadenza",
    "clientip" => "127.0.0.1",
    "ident" => "-",
    "auth" => "-",
    "timestamp" => "11/Dec/2013:00:01:45 -0800",
    "verb" => "GET",
    "request" => "/xampp/status.php",
    "httpversion" => "1.1",
    "response" => "200",
    "bytes" => "3891",
    "referrer" => "\"http://cadenza/xampp/navi.php\"",
    "agent" => "\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.
```

patterns (github.com/elastic/logstash/tree/v1.4.2/patterns)

LIVE DEMO:

NASA ACCESS LOGS



Apache logs

7 days ago to 4 minutes ago

QUERY FILTERING

HISTOGRAM

View | Zoom Out | ● * (144577) count per **1h** | (**144577** hits)

HTTP METHOD

● get (144439) ● head (101) ● post (37)
● Missing field (0) ● Other values (0)

REQUEST

● dataset (72169) ● tags (59476) ● 0 (43564) ● res_format (35089) ● organization (30830)
● it (26662) ● api (24046) ● groups (21533) ● 3 (16735) ● action (16491)
● Missing field (3585) ● Other values (1063988)

RESPONSE CODE

● 200 (138086) ● 404 (4712) ● 302 (1279)
● 304 (294) ● 301 (59) ● 416 (54)
● 206 (39) ● 500 (23) ● 405 (22)
● 403 (8) ● Missing field (0)
● Other values (1)

USER AGENT

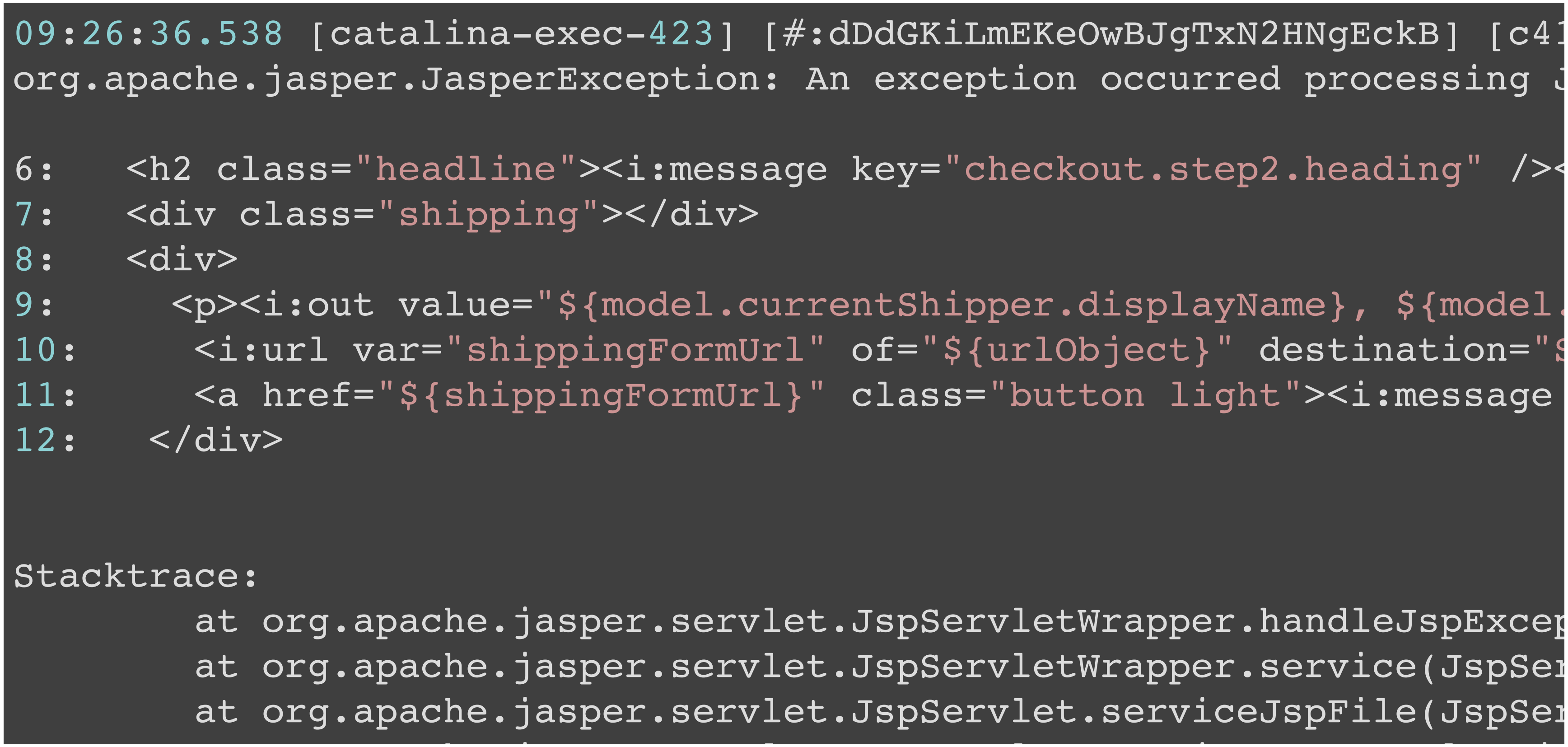
● mozilla (114192) ● 5.0 (112916) ● compatible (99814) ● http (99116) ● 3.0 (58374)
● bots (57935) ● yandexbot (57928) ● yandex.com (57928) ● linux (18988) ● python (13183)
● Missing field (234) ● Other values (565875)

REFERRER

● http (14264) ● dati.trentino.it (13814) ● dataset (7163) ● resource (1640) ● di (1420)
● organization (1332) ● pat (829) ● s (480) ● del (455) ● preview (340)
● Missing field (130181) ● Other values (60391)

TABLE

STACKTRACE



```
09:26:36.538 [catalina-exec-423] [#:dDdGKiLmEKeOwBJgTxN2HNgEckB] [c41
org.apache.jasper.JasperException: An exception occurred processing J

6:   <h2 class="headline"><i:message key="checkout.step2.heading" /><
7:   <div class="shipping"></div>
8:   <div>
9:     <p><i:out value="${model.currentShipper.displayName}, ${model.
10:     <i:url var="shippingFormUrl" of="${urlObject}" destination="$
11:     <a href="${shippingFormUrl}" class="button light"><i:message
12:   </div>


Stacktrace:
        at org.apache.jasper.servlet.JspServletWrapper.handleJspExcep
        at org.apache.jasper.servlet.JspServletWrapper.service(JspSer
        at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspSer
```

multiline & grok filter!

```
filter {
    multiline {
        pattern => "^${TIME}"
        negate => true
        what => "previous"
    }
    grok {
        match => { "message" => "%{TIME:timestamp} \[%{DATA:thread}\]
    }
}
```

PROBLEME

GROK

- komplexe Regex
- performance

MULTILINE

- multithreading
- performance

- logstash-indexer stirbt
- Echtzeit geht verloren
- Logs gehen verloren

CODECS

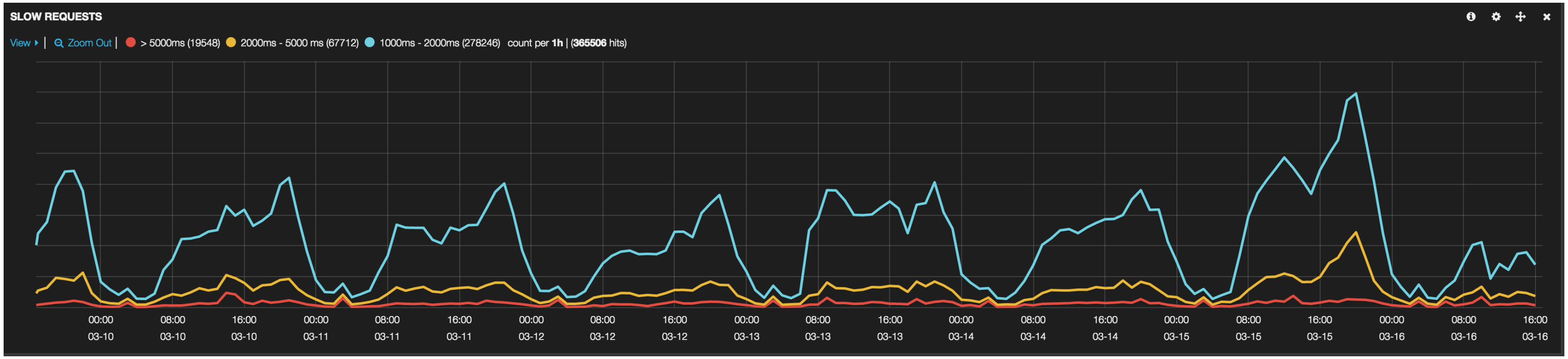
JSON?

JSON

- strukturiert
- typisiert

APACHE LOGFORMAT

```
LogFormat '{"@timestamp":"%{%Y-%m-%dT%H:%M:%S%z}t","@version":"1","me
```



LOGSTASH-LOGBACK-ENCODER

```
<configuration>
  <appender name="ISHOP" class="FileAppender">
    <file>${catalina.base}/logs/ishop.log</file>
    <layout class="PatternLayout">
      <pattern>
          %-28(%d{"yyyy-MM-dd'T'HH:mm:ss,SSS"} [%thread])
          [#:%exHash] [%X{sessionid}@%X{ipaddr}]
          [%X{rnd}/%X{username}] %-5level
          %marker %c - %m%n
      </pattern>
    </layout>
  </appender>

  <appender name="ISHOP_LOGSTASH" class="FileAppender">
    <file>${catalina.base}/logs/ishop.json</file>
    <encoder class="LogstashEncoder" />
```

LOGSTASH SHIPPER @ FRONTLINE

```
input {
  file {
    path => "/var/logs/ishop_logstash.log"
    type => "ishop_logstash"
    codec => "json"
  }
}
```

```
output {
  if [type] == "ishop_logstash" {
    redis {
      host => "redis"
      data_type => "list"
      key => "ftl-app-ishop-logstash"
    }
  }
}
```

keine Filter im Shipper

LOGSTASH INDEXER @ FRONTLINE

```
input {
  redis {
    host => "redis"
    type => "ishop_logstash"
    key => "ftl-app-ishop-logstash"
    data_type => "list"
    codec => json
    add_field => { "source_host" => "live" }
  }
}
```

```
filter {
  if [source_host] == "live" and [type] == "ishop_logstash" {
    mutate {
      rename => [ "ipaddr", "ip" ]
    }
  }
}
```

```
output {
  if [source_host] == "live" {
    elasticsearch_http {
      host => "elasticsearch"
    }
  }
}
```

AKZEPTANZ IM UNTERNEHMEN

- **Logs** interessiert nur IT
- **Bestellungen** interessieren ALLE
- Realtime Analytics fasziniert

DASHBOARDS AUF TFT

INHALTLICHES LOGGING

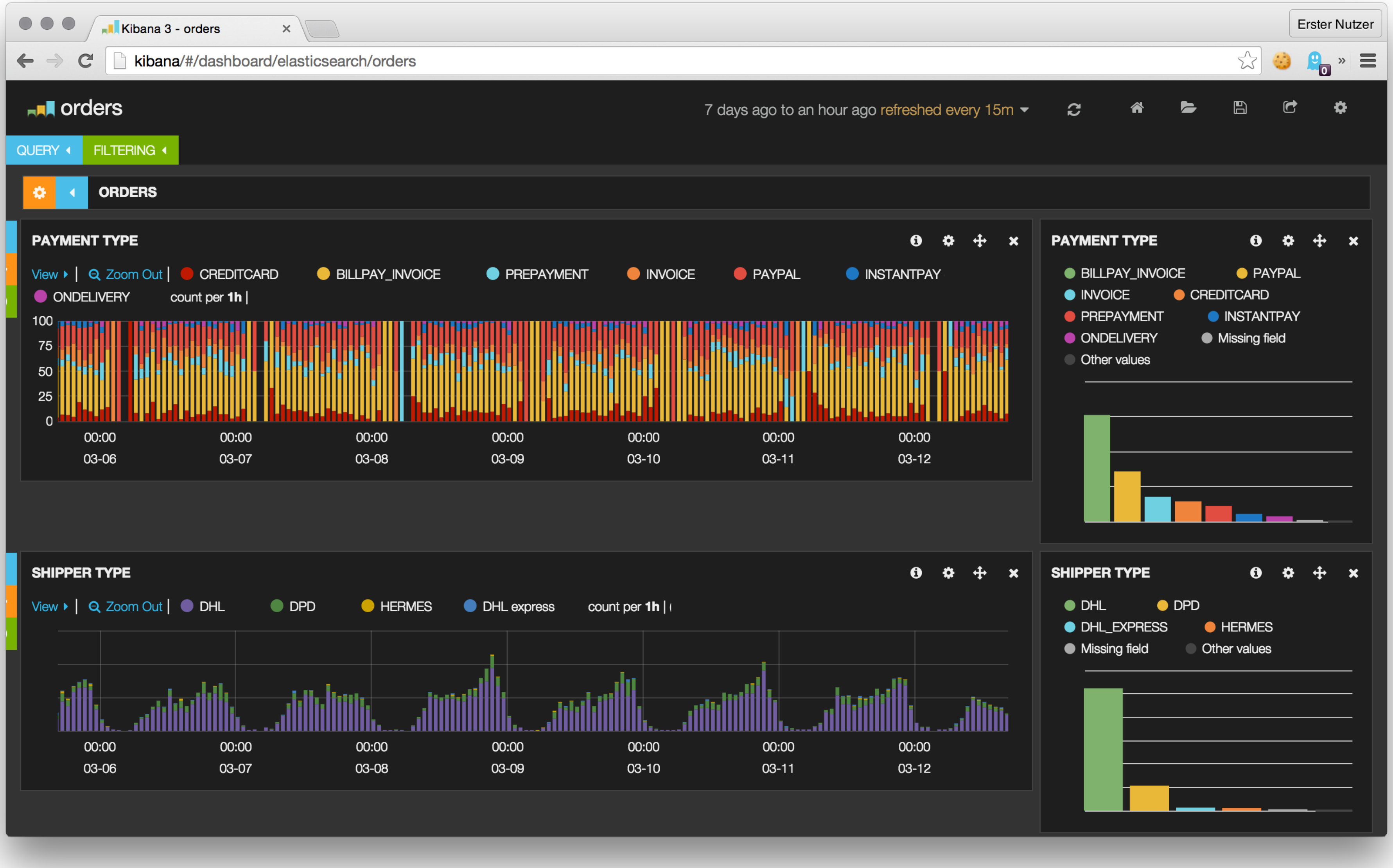
```
class OrderData {
    List<LineItem> lineItems;
    Money totalAmount;
    PaymentType paymentType;
    ShipperType shipperType;
}
```

```
Logger.info(Markers.append("order", orderData), "order success");
```

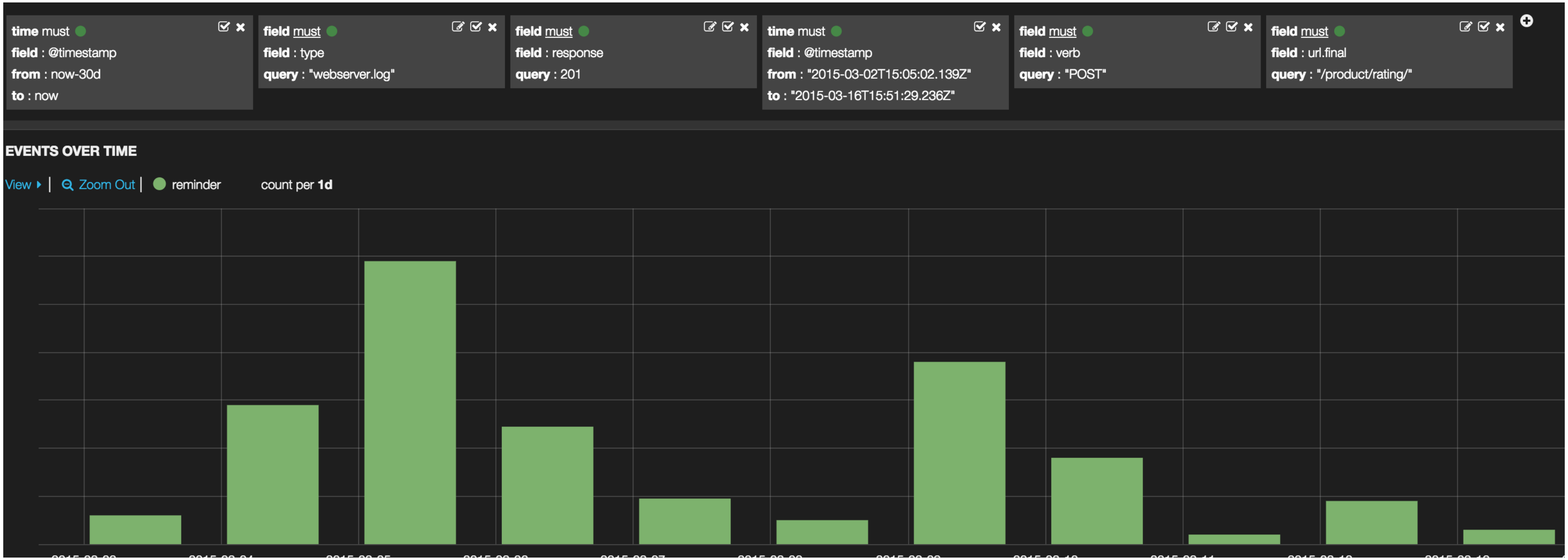
LOGSTASH OUTPUT

```
{
    "message": "order success",
    "@version": "1",
    "@timestamp": "2015-03-10T13:53:25.775Z",
    "host": "work",
    "order": {
        "lineItems": [{}, {}, ...],
        "totalAmount": 23.55,
        "paymentType": "PAYPAL",
        "shipperType": "DHL"
    }
}
```

KIBANA DASHBOARD



KIBANA DASHBOARD



time must ●
field : @timestamp
from : now-30d
to : now

field must ●
field : type
query : "webserver.log"

field must ●
field : response
query : 201

time must ●
field : @timestamp
from : "2015-03-02T15:05:02.139Z"
to : "2015-03-16T15:51:29.236Z"

field must ●
field : verb
query : "POST"

field must ●
field : url.final
query : "/product/rating/"

EVENTS OVER TIME

AUSBLICK: LOGFILES

- Nur noch JSON
- kürzere RollingPolicy
- Weniger Logfiles
  - `application.log` vs.
  - `[order|paypal|billpay|diva|epoq|js|...].l`
- SiftingAppender (http://logback.qos.ch/manual/appenders.html#SiftingAppender)
  - ein Logfile pro Thread
- LMAX Disruptor RingBuffer mit *AsyncDisruptorAppender (https://github.com/logstash/logstash-logback-encoder#usage)

AUSBLICK: ELASTICSEARCH

- Mehr Nodes
- Index Template (http://www.elastic.co/guide/en/elasticsearch/guide/current/index-templates.html)
    - _all Feld
    - Kompression
    - `not_analyzed` indizieren (oder `*.raw` nutzen)
- Doc Values (http://www.elastic.co/guide/en/elasticsearch/guide/current/doc-values.html)
    - `disk-based` statt `in-memory fielddata`

LOGVOLUMEN

- 45 Tage im ES
  - daily index
  - daily snapshots
  - daily cleanup
- Curator (github.com/elastic/curator)

STAGING ELK

- neue Konfigurationen testen
- Jede Komponente leicht skalierbar
  - docker?

VIELEN DANK

FRAGEN?