

Search-Driven Applications

Florian Hopf, freiberuflicher Softwareentwickler Tobias Kraft, exensio GmbH





Agenda

Motivation

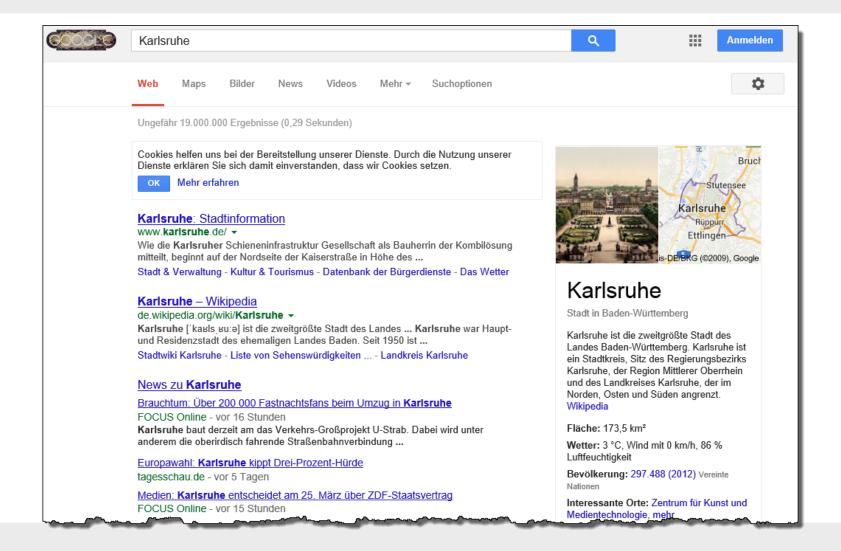
Aufbau der Such-Datenstruktur

Anwendungsfälle

Fallstricke



Was ist Suche?





Was wollen wir in diesem Vortrag zeigen

- Suche ist mehr als Volltextsuche
- Suche funktioniert im Kleinen wie im Großen
- Suche kann auch bei klassischen
 Anwendungsfällen unterstützen



http://www.morguefile.com/archive/display/861760



Strukturierte vs. unstrukturierte Daten

Strukturiert

- Datenbanken
- LDAP
- Spreadsheets
- (teilweise)Applikations- /API-Daten

Unstrukturiert

- Web Seiten
- Email
- Text- und Office-Dokumente
- PDF-Dokumente
- Log-Files



Probleme klassischer Applikationen

- Daten-Silos
 - → Konsolidierung Daten
- Daten-Menge steigt immer weiter
 - → Skalierbarkeit
- Häufige Änderungen
 - → Flexibles Schema und Abfragen



Suchserver vs. Datenbank

Eigenschaft	Suchserver	Datenbank
Semantisches Modell	Dokumenten-Modell	Relationales Modell
Speicherstruktur	Index	Tabelle
Ablage der Daten	Denormalisiert	Normalisiert



Search-Driven Applications

Suche ist Kernbestandteil

Suchserver als zentraler Lesespeicher



Agenda

Motivation

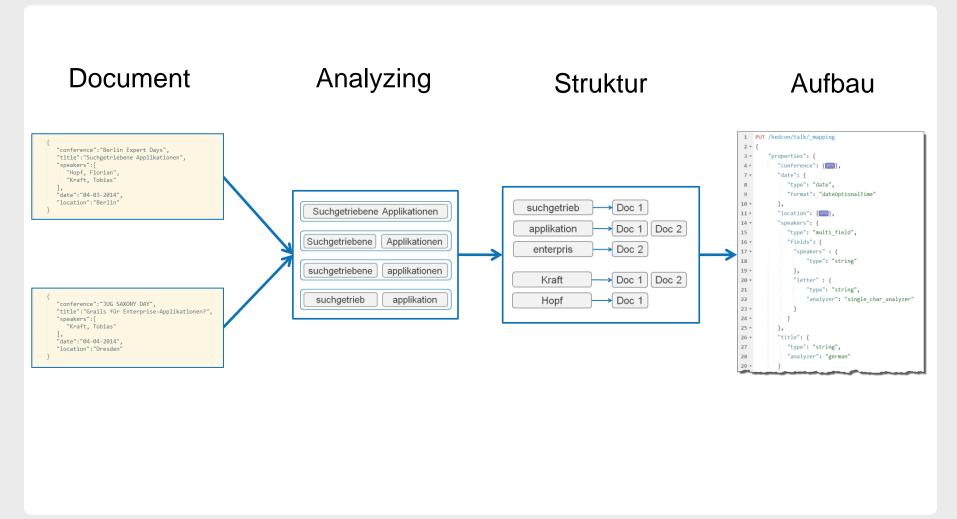
Aufbau der Such-Datenstruktur

Anwendungsfälle

Fallstricke



Vom Inhalt zum Index

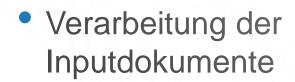




Input-Dokumente

```
"conference": "Berlin Expert Days",
"title": "Suchgetriebene Applikationen",
                                                                                                     1 PUT /bedcon/talk/_mapping
"speakers":[
                                                                                                         "properties": {
                                                                                                          "conference": { [ ],
    "Hopf, Florian",
                                                                                                           "type": "date",
                                                                                                           "format": "dateOptionalTime"
    "Kraft, Tobias"
                                                                                 Doc 1
                                                                                                          "speakers": {
                                                                                  Doc 1 Doc 2
                                                                                                           "type": "multi field",
                                                                                                           "fields": {
"date":"04-03-2014",
                                                                                 Doc 2
                                                                                                            "speakers" : {
                                                                                                              "type": "string"
"location": "Berlin"
                                                                                Doc 1 Doc 2
                                                                                                            "letter" : {
                                                                                                              "type": "string",
                                                                                                              "analyzer": "single_char_analyzer"
                                                                                 Doc 1
                                                                                                          "title": {
                                                                                                           "type": "string",
                                                                                                           "analyzer": "german"
"conference": "JUG SAXONY DAY",
"title": "Grails für Enterprise-Applikationen?"
"speakers":[
    "Kraft, Tobias"
"date": "04-04-2014",
"location": "Dresden"
```

Text-Analyzing

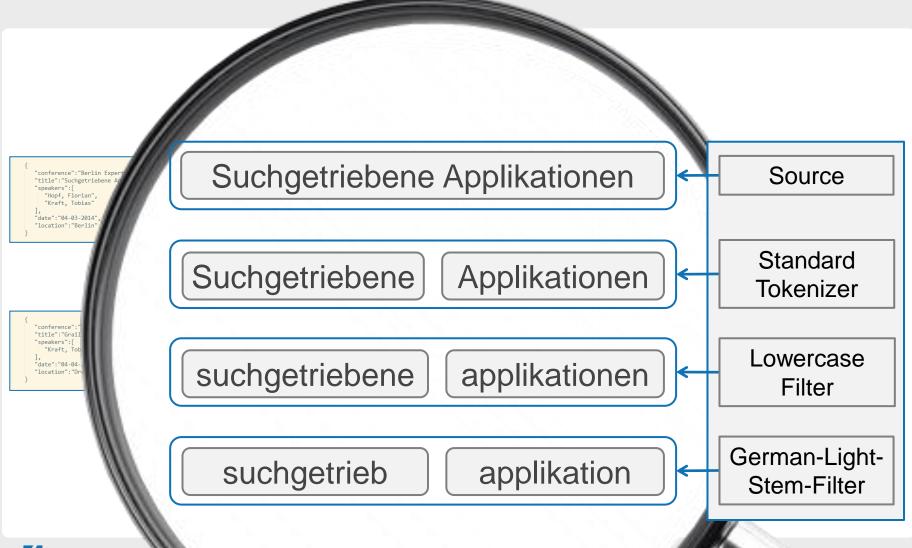


Preprocessing vor der Ablage

 Entfernen, modifizieren, hinzufügen von Termen



Beispiel Text-Analyzing





Der Indexaufbau

```
{
    "conference":"Berlin Expert Days",
    "title":"Suchgetriebene Applikationen",
    "speakers":[
    "hopf, Florian",
    "Kraft, Tobias"
],
    "date":"04-03-2014",
    "location":"Berlin"
}
```

```
"conference":"JUG SAXONY DAY",
  "title":"Grails für Enterprise-Applikatio
"speakers":[
    "Kraft, Tobias"
]
    "date":"94-94-2014",
```

Invertierter Index

 Inputfeld kann auf unterschiedliche Weise abgelegt werden

Suchterm zu Dokument zuordnen

Doc 2

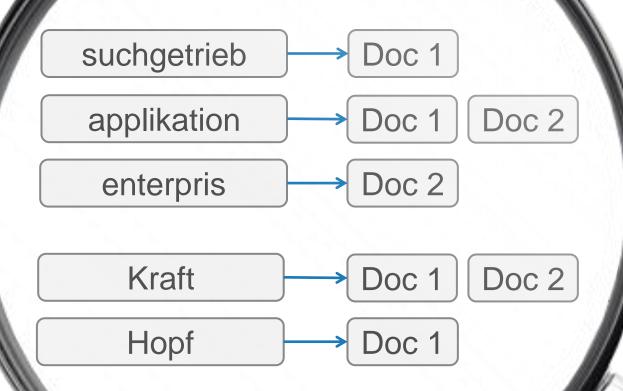
oc 2



Die Indexstruktur

```
{
  "conference":"Berlin Expert Days",
  "title":"Suchgetriebene Applikationen",
  "speakers":[
   "Hopf, Florian",
   "Kraft, Tobias"
],
  "date":"04-03-2014",
  "location":"Berlin"
}
```

```
{
  "conference":"JUG SAXONY DAY",
  "title":"Grails für Enterprise-Applikationen?",
  "speakers":[
        "Kraft, Tobias"
        ],
        "date":"04-04-2014",
        "location":"Dresden"
```





Das Mapping

```
"conference": "Berlin Expert Days",
"title": "Suchgetriebene Applikationen",
"speakers":[
"Hopf, Florian"
  "Kraft, Tobias"
"date":"04-03-2014",
"location":"Berlin'
                                                                                                            10
                                                                                                           11 -
                                                                                                           12
"conference":"JUG SAXONY DAY",
"title": "Grails für Enterprise-Applikationen?",
                                                                                                            13
"speakers":[
"Kraft, Tobias"
"date":"04-04-2014",
                                                                                                            14 -
                                                                                                            15 -
                                                                                                            16 -
                                                                                                            17 -
```

18

```
oedcon/talk/_mapping {
"properties": {
  "date": {
     "type": "date",
     "format": "dateOptionalTime"
  "speakers": {
      "type": "multi field",
     "fields": {
        "speakers" : { "type": "string" },
        "letter" : {
            "type": "string",
            "analyzer": "single_char_analyzer"
  "title": {
     "type": "string",
     "analyzer": "german"
```



Modellierung des Index

- Dokumentenmodell
- Datenablage anhand geplanter Abfragen
- Duplizierte Datenablage





Near Realtime

- Inkrementell Indexieren
- Änderungen auch im Cluster schnell verfügbar
- Onlineabfragen für Facetting / Aggregationen





Agenda

Motivation

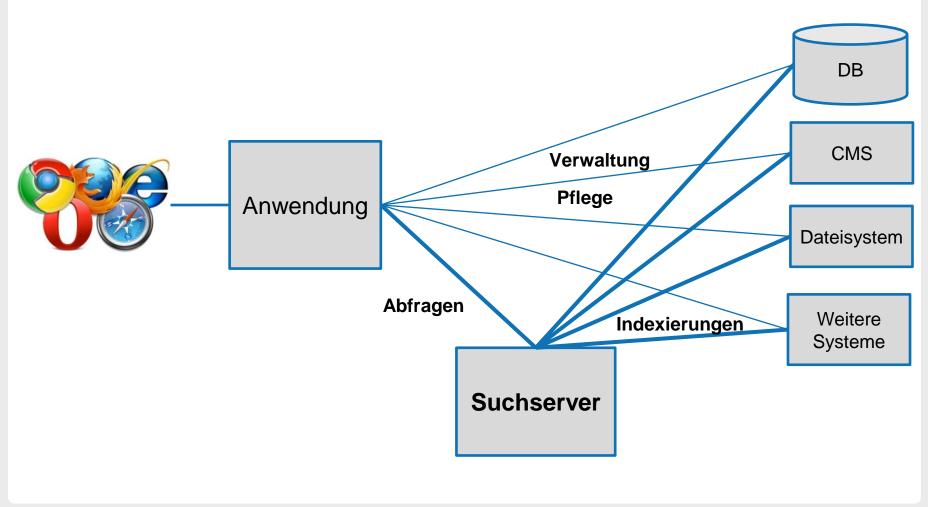
Aufbau der Such-Datenstruktur

Anwendungsfälle

Fallstricke

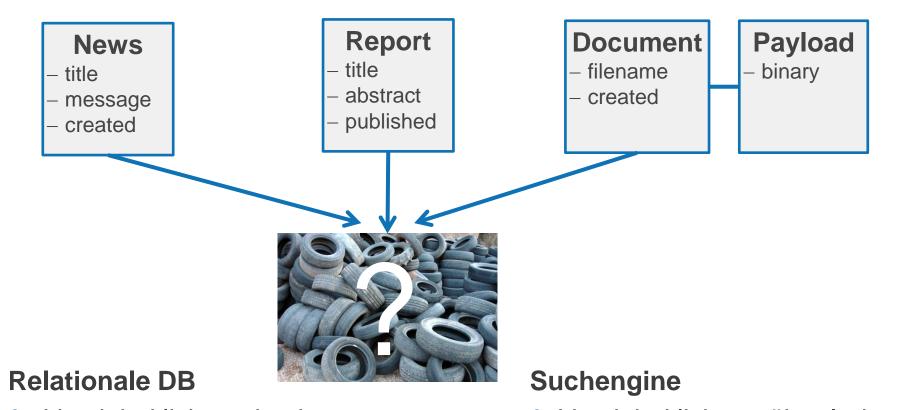


Suche ist zentraler Bestandteil der Applikation





Vereinheitlichung von verschiedenen Quellen

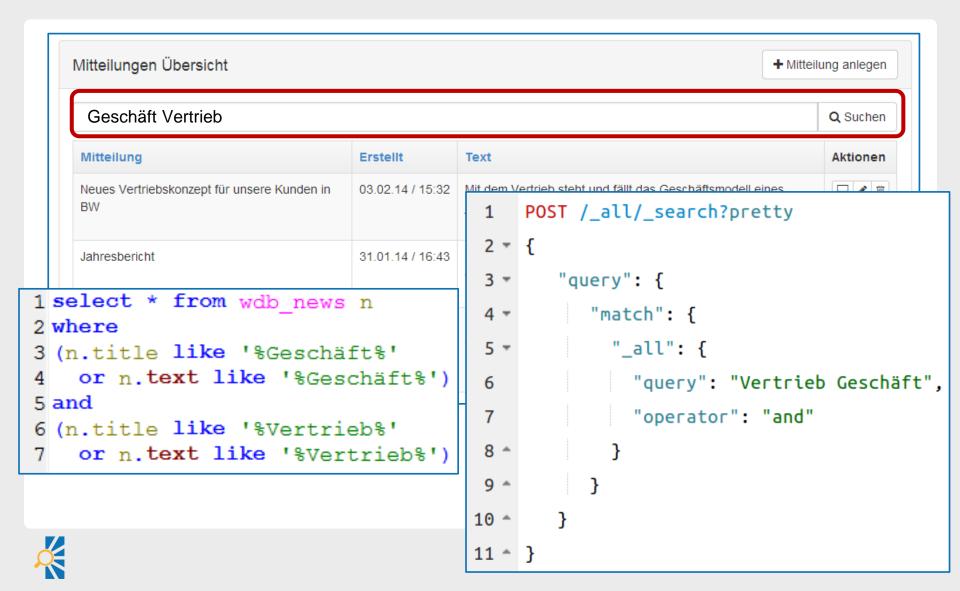


 Vereinheitlichung in einer Ansicht/Liste ist schwierig

- Vereinheitlichung über Index
- Entsprechendes Mapping Index-Prozess



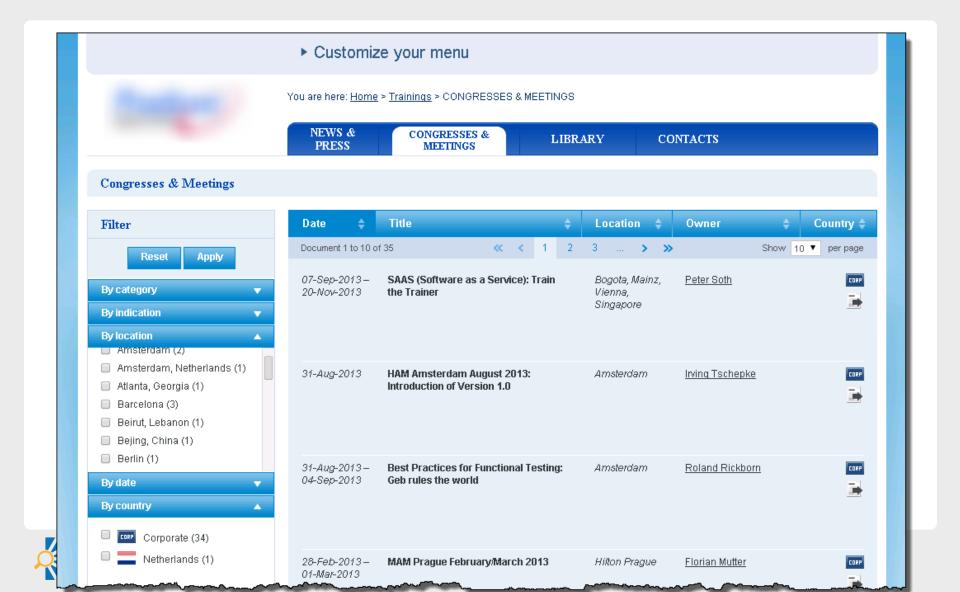
Listen für CRUD-Dialoge



Listen für CRUD-Dialoge

```
Mitteilungen Übersicht
                                                                        + Mitteilung anlegen
      Geschäft Vertrieb Aussendienst
                                                                             Q Suchen
                                 Erstellt
                                                                             Aktionen
 1 select * from wdb news n
 2 inner join ex taglink tl
                                          POST /_all/_search?pretty
     on n.id = tl.object id
                                       2 - {
 4 inner join ex tag t
                                             "query": {
                                        3 =
     on tl.tag id = t.id
 6 where
                                               "match": {
                                        4 =
 7 (n.title like '%Geschäft%'
                                              "_all": {
                                        5 =
    or n.text like '%Geschäft%'
                                                  "query": "Vertrieb Geschäft Aussendienst",
    or t.name like '%Geschäft%')
                                                  "operator": "and"
10 and
11 (n.title like '%Vertrieb%'
                                       8 *
   or n.text like '%Vertrieb%'
12
                                        9 -
     or t.name like '%Vertrieb%')
13
                                      10 -
14 and
15 (n.title like '%Aussendienst%' 11 * }
    or n.text like '%Aussendienst%'
16
     or t.name like '%Aussendienst%')
17
```

Facetten für den Drilldown von Events



7 SQL-Abfragen für den Aufbau der Facetten ...

- 1 SELECT location, COUNT(*) FROM (SELECT e.location,e.id, COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR_EVENT_TAG eventCATEGORY ON (e.id = eventCATEGORY. eventid) LEFT OUTER JOIN BR_TAG CATEGORY ON (eventCATEGORY.tagid = CATEGORY.id) AND (CATEGORY.type = 'CATEGORY')) LEFT OUTER JOIN BR_TAG ERANDSELECTED ON (e.id = eventBRANDSELECTED.tagid = BRANDSELECTED.id) AND (BRANDSELECTED.type = 'BRAND')) WHERE ((e.location IS NOT NULL) AND (category.path IN ('BR:PM/category/events/tags/corporatebrandteam')) AND (brandselected.path IN ('BR:PM/brandsfranchises/brands/pr')) AND (e.deleted = 0)) GROUP BY e.location,e.id) GROUP BY location ORDER BY location ASC
- SELECT path, COUNT(*) FROM (SELECT category.path,e.id,COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR_EVENT_TAG eventCATEGORY ON (e.id = eventCATEGORY. eventid) LEFT OUTER JOIN BR_TAG CATEGORY ON ((eventCATEGORY.tagid = CATEGORY.id) AND (CATEGORY.type = 'CATEGORY')) LEFT OUTER JOIN BR_EVENT_TAG eventBRANDSELECTED ON (e.id = eventBRANDSELECTED. eventid) LEFT OUTER JOIN BR_EVENT_TAG eventBRANDSELECTED ON (e.id = eventBRANDSELECTED. eventid) LEFT OUTER JOIN BR_EVENT_TAG eventBRANDSELECTED ON (e.id = eventBRANDSELECTED. eventid) LEFT OUTER JOIN BR_EVENT_TAG eventBRANDSELECTED ON (e.id = eventBRANDSELECTED. ON (e.id = eventBRANDSELECTED.
- SELECT path,COUNT(*) FROM (SELECT country.path,e.id,COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR_EVENT_TAG eventCATEGORY ON (e.id = eventCATEGORY. eventid) LEFT OUTER JOIN BR_TAG CATEGORY ON ((eventCATEGORY.tagid = CATEGORY.id) AND (CATEGORY.type = 'CATEGORY')) LEFT OUTER JOIN BR_TAG BRANDSELECTED ON (e.id = eventCOUNTRY ON (e.id = eventCOUNTRY.eventid) LEFT OUTER JOIN BR_TAG COUNTRY ON ((eventCOUNTRY.tagid = COUNTRY.id) AND (COUNTRY.type = 'COUNTRY')) WHERE ((country.path IS NOT NULL) AND (category.path IN ('BR:PM/category/events/tags/corporatebrandteam')) AND (brandselected.path IN ('BR:PM/brandsfranchises/brands/pr')) AND (e.deleted = 0)) GROUP BY country.path,e.id) GROUP BY path ORDER BY path ASC
- 7 SELECT path,COUNT(*) FROM (SELECT indication.path,e.id,COUNT(*) FROM BR EVENT e INNER JOIN BR USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR EVENT TAG eventCATEGORY ON (e.id = eventCATEGORY.

SELECT country.path,e.id,COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON

by path order by path ASC

- SELECT userid,COUNT(*) FROM (SELECT u.userid,e.id,COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR_EVENT_TAG eventCATEGORY ON (e.id = eventCATEGORY. eventid) LEFT OUTER JOIN BR_TAG CATEGORY ON ((eventCATEGORY.tagid = CATEGORY.id) AND (CATEGORY.type = 'CATEGORY') LEFT OUTER JOIN BR_EVENT_TAG eventBRANDSELECTED ON (e.id = eventBRANDSELECTED. eventid) LEFT OUTER JOIN BR_TAG BRANDSELECTED ON ((eventBRANDSELECTED.tagid = BRANDSELECTED.id) AND (BRANDSELECTED.type = 'BRAND')) WHERE ((u.userid IS NOT NULL) AND (category.path IN ('BR:PM/category/events/tags/corporatebrandteam')) AND (brandselected.path IN ('BR:PM/brandsfranchises/brands/pr')) AND (e.deleted = 0)) GROUP BY u.userid,e.id) GROUP BY userid ORDER BY userid ASC
- SELECT path,COUNT(*) FROM (SELECT brand.path,e.id,COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR_EVENT_TAG eventCATEGORY ON (e.id = eventCATEGORY.type = 'CATEGORY')) LEFT OUTER JOIN BR_EVENT_TAG eventBRAND ON (e.id = eventBRAND.eventid) LEFT OUTER JOIN BR_TAG BRAND ON (eventCATEGORY.type = 'BRAND') AND (BRAND.path IN ('BR:PM/brandsfranchises/brands/actilyse', 'BR:PM/brandsfranchises/brands/aggrenox', 'BR:PM/brandsfranchises/brands/apgrenox', 'BR:PM/brandsfranchises/brands/aptivus', 'BR:PM/brandsfranchises/brands/apgrenox', 'BR:PM/brandsfranchises/brands/aptivus', 'BR:PM/brandsfranchises/brands/pri', 'BR:PM/brandsfranchises/brands/spiriva', 'BR
 - SELECT path,COUNT(*) FROM (SELECT franchise.path,e.id,COUNT(*) FROM BR_EVENT e INNER JOIN BR_USER u ON (e.initiator = u.id) LEFT OUTER JOIN BR_EVENT_TAG eventCATEGORY ON (e.id = eventCATEGORY.eventid) LEFT OUTER JOIN BR_TAG CATEGORY ON ((eventCATEGORY.tagid = CATEGORY.id) AND (CATEGORY.type = 'CATEGORY')) LEFT OUTER JOIN BR_EVENT_TAG eventFRANCHISE ON (e.id = eventFRANCHISE.eventid) LEFT OUTER JOIN BR_TAG FRANCHISE ON ((eventFRANCHISE.tagid = FRANCHISE.id) AND (FRANCHISE.type = 'BRAND') AND (FRANCHISE.path IN ('BR:PM/brandsfranchises/franchises

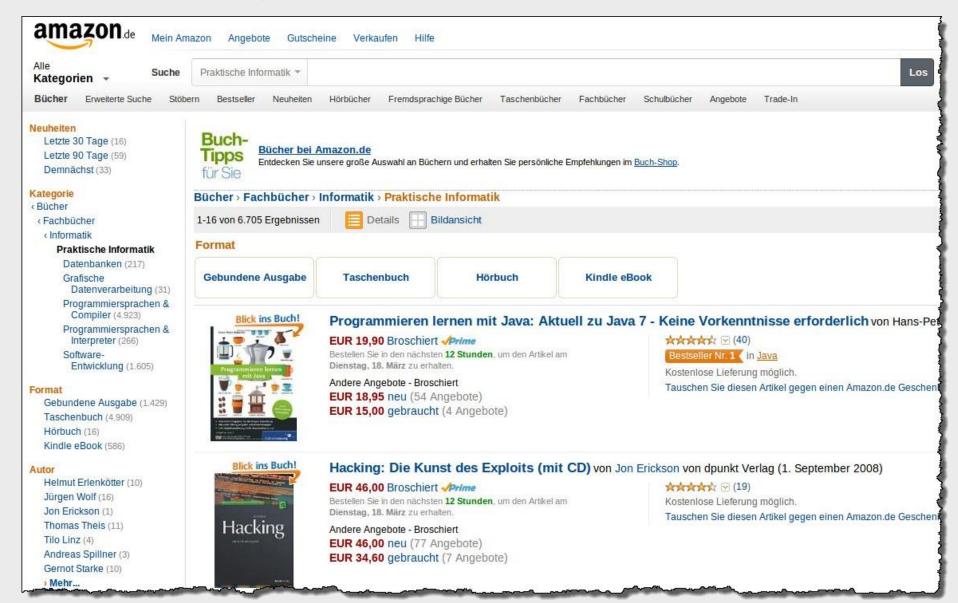


Facette mit Elasticsearch

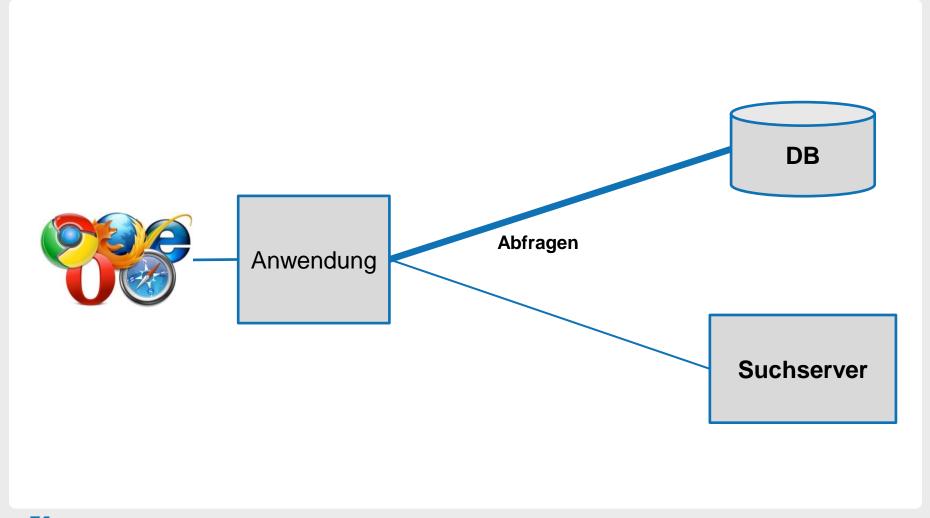
```
POST /events/_search
                                       "aggregations": {
                               15 -
                               16 -
                                          "locations": {
        "aggs" : {
                                             "buckets": [
                               17 -
         "categories" :
                               18 🔻
         "indications" : {
                                                    "key": "Berlin",
                               19
         "locations" : { "t
                                                    "doc_count": 626
                               20
         "countries" : { "t
                               21 -
          "created_range":
                               22 *
              "date range":
 9 -
                               23
                                                   "key": "Basel",
                  "field": '
10
                               24
                                                    "doc_count": 93
11
                  "format":
                               25 -
                  "ranges":
12 -
                               26
                       { "fro
                               27 -
13
                               28 -
                      { "fro
14
                                          "countries": {
                      { "fro
                               29 +
15
                               30 -
                                             "buckets": [
16 -
                               31 -
17 -
                                                    "key": "Germany",
                               32
18 -
                               33
                                                    "doc_count": 671
19 -
                               34 -
20 - }
                               35 +
```



Anwendungsfälle für E-Commerce

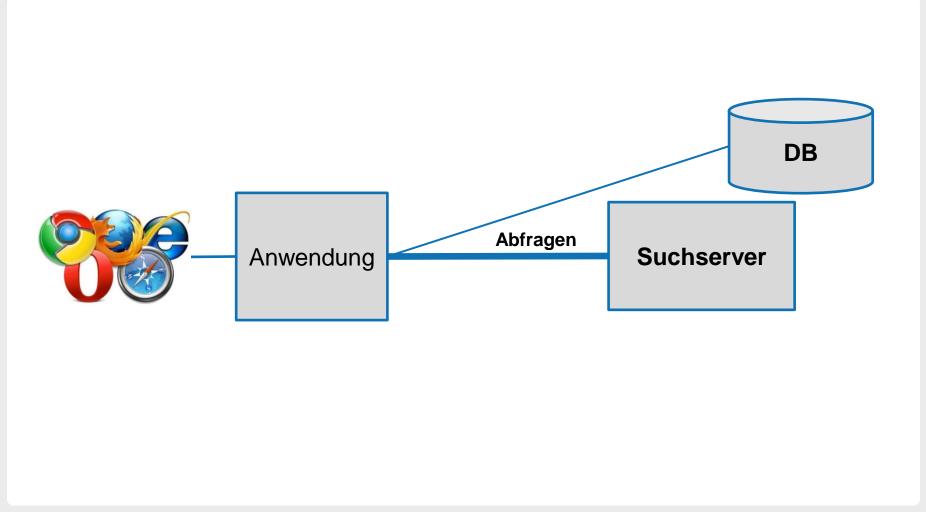


Der Suchserver rückt in den Mittelpunkt



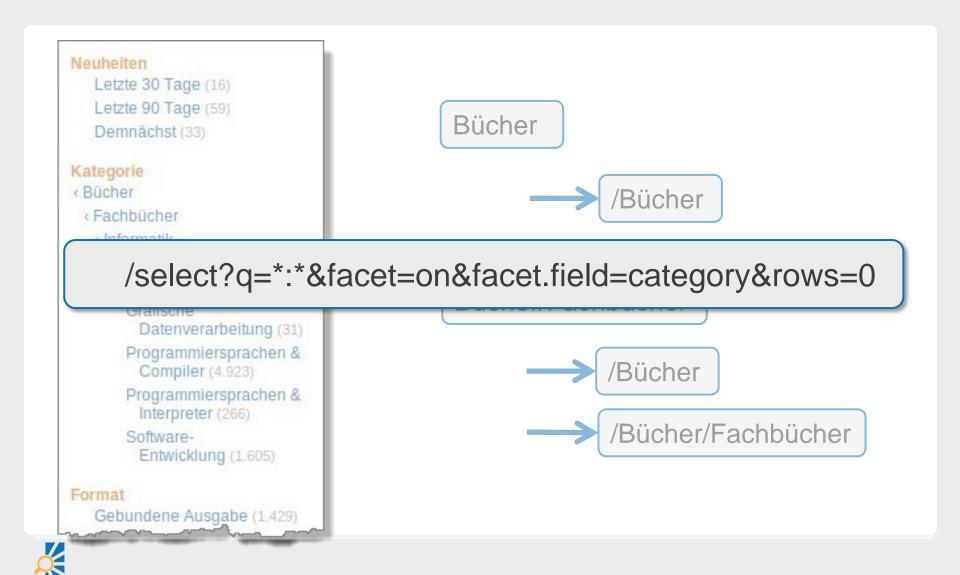


Der Suchserver rückt in den Mittelpunkt

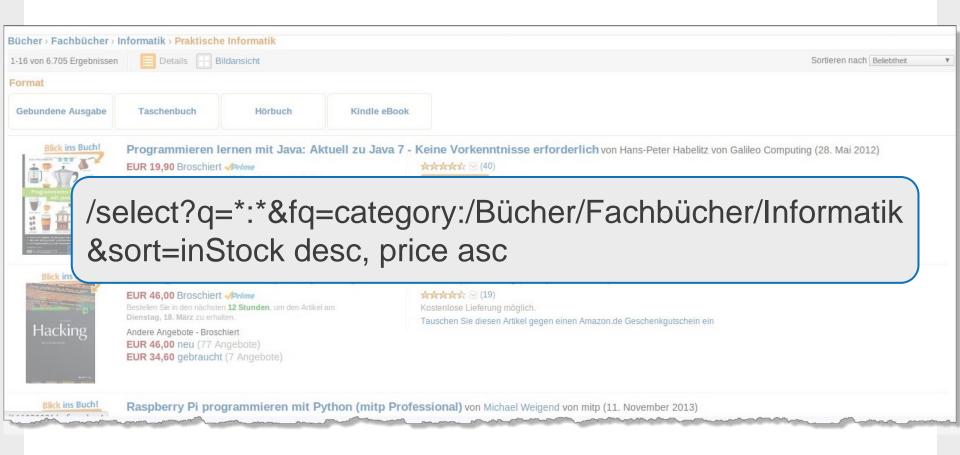




Kategorien-Navigation



Produktlisting





Agenda

Motivation

Aufbau der Such-Datenstruktur

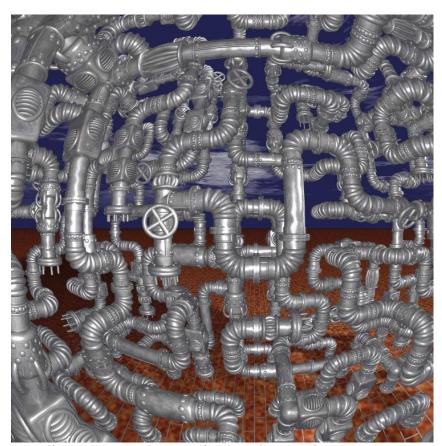
Anwendungsfälle

Fallstricke



Komplexität der Architektur steigt

- SynchronisierungSysteme
- Fehlersuche
 - Mehrere `Verursacher`
 - NachstellenProduktionsprobleme
- Entwicklungssetup



http://www.morguefile.com/archive/display/591261



Transaktionen

```
if (study.validate() && study.validateTagHierarchy() && study.save(flush: true)) {
    // index process
    searchService.indexStudy(study)
    result.message = tagLibService.validationTagLib.message(code: 'default.info.update')
else {
   // reset
    study.stepDone = stepBefore
    study.discard() // necessary to discard changes on errors
    return fail(code:"default.update.failure")
documentService.store(doc)
// success
return result
```



Daten

- Analyzing ist verlustbehaftet
- Update einzelner Felder schwierig
- Originalinhalt für Anpassungen weiter zugreifbar

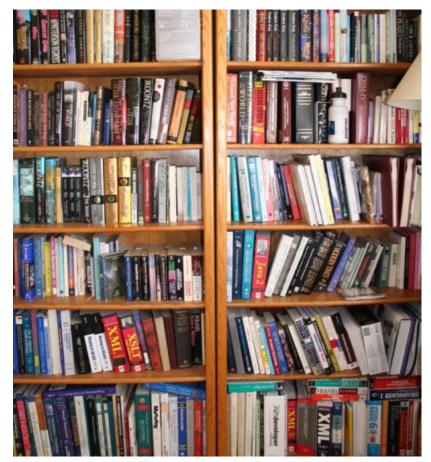


http://www.morguefile.com/archive/display/183870



Testbarkeit

- Testunterstützung ist vorhanden
 - SolrtestCaseJ4
 - Elasticsearch Test Framework
- Relevanz ist subjektiv
- Analyzing-Prozess kann komplexer werden
- Rollback in Tests

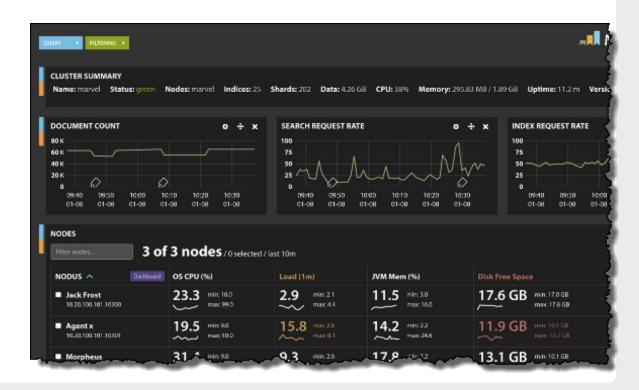


http://www.morguefile.com/archive/display/87396



Betrieb

- Security
- Separate Installation
- Verfügbarkeit
- Verteilte Logfiles
- Monitoring





Fazit

Was bringt die Umsetzung von Applikations-Logik mit Suchmaschinen?

Abfragen sind teilweise einfacher

- Flexibler für Erweiterungen
- Performance-Vorteile
- Neue Anwendungsszenarien





Mehr Interesse an Suche?



@KASearch

Vielen Dank!

Florian Hopf, @fhopf Tobias Kraft, @tokraft







1. OG DIREKT ÜBER DEM EMPFANG