Alternativen zur OpenText Suche

29. OpenText Web Solutions Anwendertagung

Mannheim, 18. Juni 2012

Sebastian Henne

elbformat content solutions

Übersicht

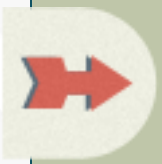
- Allgemeines zur Suche
- Die OpenText Common Search
- Indexierung ohne DeliveryServer
- Integration über die XML API
- Integration über die SearchAPI
- Fazit

Allgemeines zur Suche

elbformat content solutions

Aufbau des Suchcontents

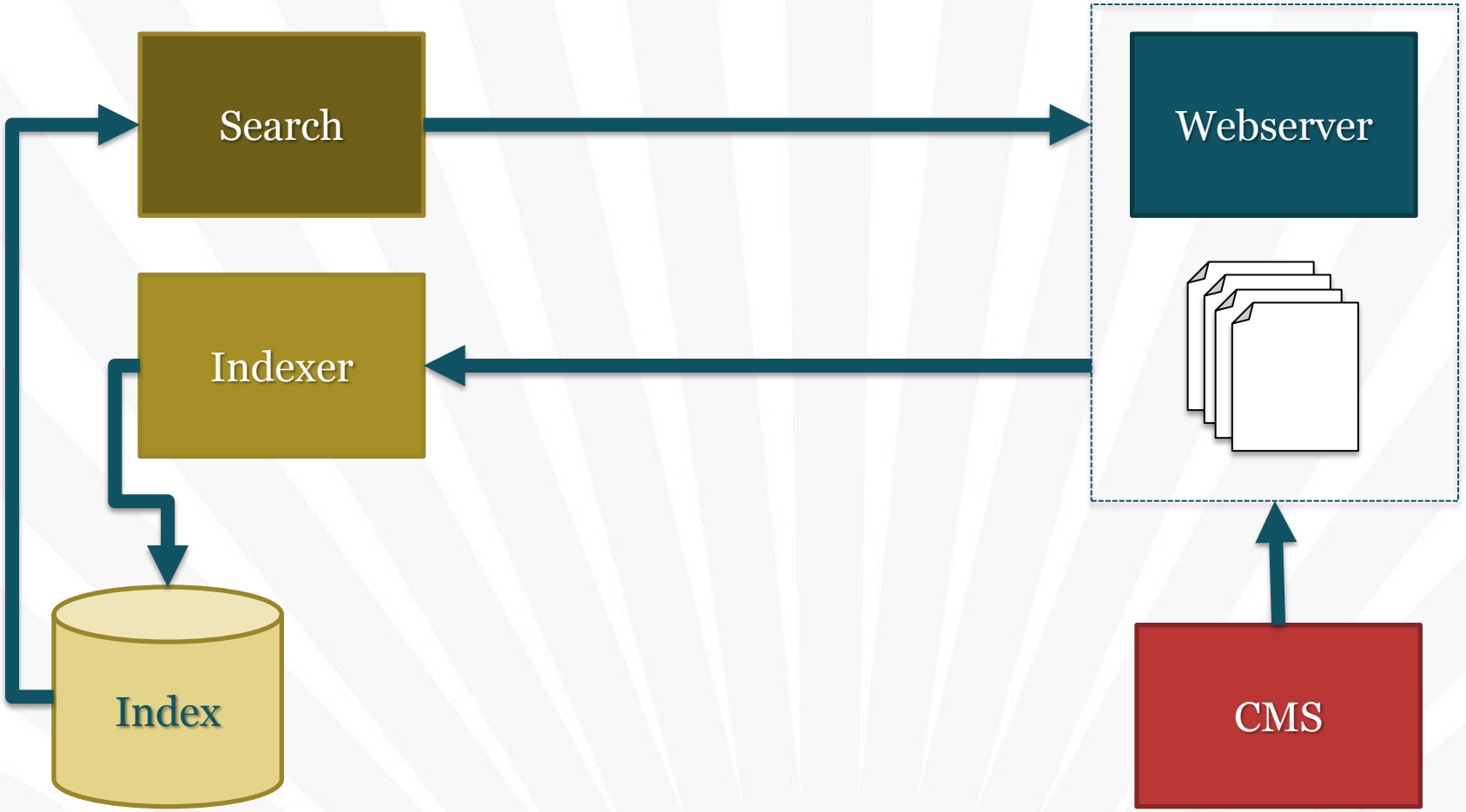
Textinhalt	Attribute	Berechtigungen	Verbindungen
• Volltext • Metadaten	• Klassifizierung • Metadaten	• Gruppen • Rollen • ACLs	• Enthaltene Dokumente • Weitere Informationen • Quellen



- Reine Attributsuche entspricht Datenbankabfrage
- Aus Performancegründen u.U. trotzdem sinnvoll
- Abwägung gegen target Dynament

elbformat
content solutions

Indexierung



elbformat content solutions

HTML Content

```
<body class="kompetenzen">
<!-- start wrapper -->
<div id="wrapper" class="inner-page">

<div class="wrapper container_12">
<div id="container">
<!-- start header -->
<header class="clearfix">

<section class="logo">
<a href="/index.htm "><img src="[illegible]img/logo.png" alt="plenum" title="[illegible] /></a>
</section>
<section class="header-right">
<ul>
<li class="contact-link"><a href="/Kontakt_und_Standorte.htm" title="Kontakt">Kontakt</a></li>
<li class="search">
<ul>
<li class="search-input">
<input type="text" name="search" id="search" value="SUCHE" />
</li>
<li class="serach-btn"><input type="button" value="" class="btn-search" /></li>
</ul>
</li>
</ul>
</section>
<section class="text-oben">

<h3>Den Kunden stärker in den Mittelpunkt stellen</h3>
<p>Der Erfolg von Versicherungen hängt in immer stärkerem Maße von durchgängigen, IT-gestützten Prozessen ab, die konsequent auf die Bedürfnisse des Marktes ausgerichtet sind.</p>
</section>
</header>
<!-- end header -->
<!-- start navigation -->

<nav class="clearfix" id="mainNavigation">
<ul class="nav-lavel-1 mainNavigation">

<li class="nav-1 active"><a href="/kompetenzen.htm"><span>Kompetenzen</span></a>
```

elbformat content solutions

Binärdokumente

- Extraktion von Metadaten
- Bereitstellung expliziter Attribute
- Unterstützte Formate
- Suche in/nach Bildern

elbformat content solutions

Suchanfragen

- Einfache Anfrage
  - Suche nach „Rentenversicherung“
  - Suche nach „OpenText Websolutions“
- Kombinierte Anfrage
  - Suche „Riester“ in „allen Produktflyern“
  - Suche „OpenText Websolutions“ jünger als 01.01.2012
- Attributsuche
  - Suche alle Dokumente zum Produkt „Riester Rente“
  - Suche die aktuellsten News zu „OpenText Websolutions“
- Berechtigungen
  - Filter auf Basis der aktuellen Benutzerinformationen
  - Auf Basis der Attribute oder auf Basis von indexierten Berechtigungen
- Implizite Kombinationen
  - „auch interessant“: Suchergebnisse zu den zuletzt angesehenen Produkten
  - „neu“: Neue Ergebnisse basierend auf meinen letzten Anfragen

elbformat content solutions

Optimierung Suche

Aufbereitung von Inhalten

- Linguistische Prozesse
- Extraktion von Metadaten
- Quelltextbereinigung

Clustern

- Ermitteln von Clustern (Phrasen, Metadaten, Quellen)
- Aufbau Cluster und Zuordnung Inhalte

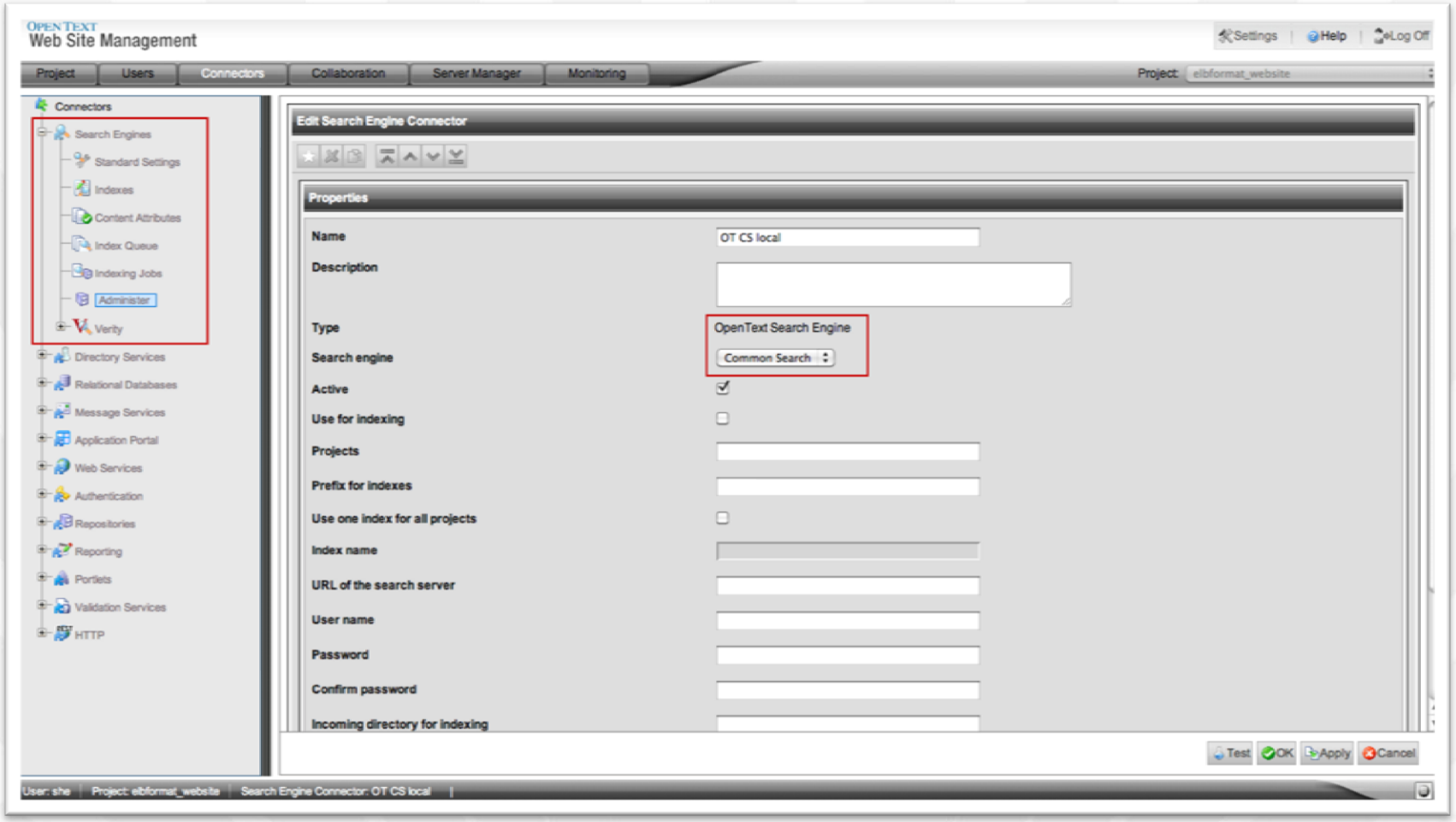
Einbeziehung von Referenzwerten

- Wörterbücher
- Thesauren
- Taxonomien

OpenText Common Search

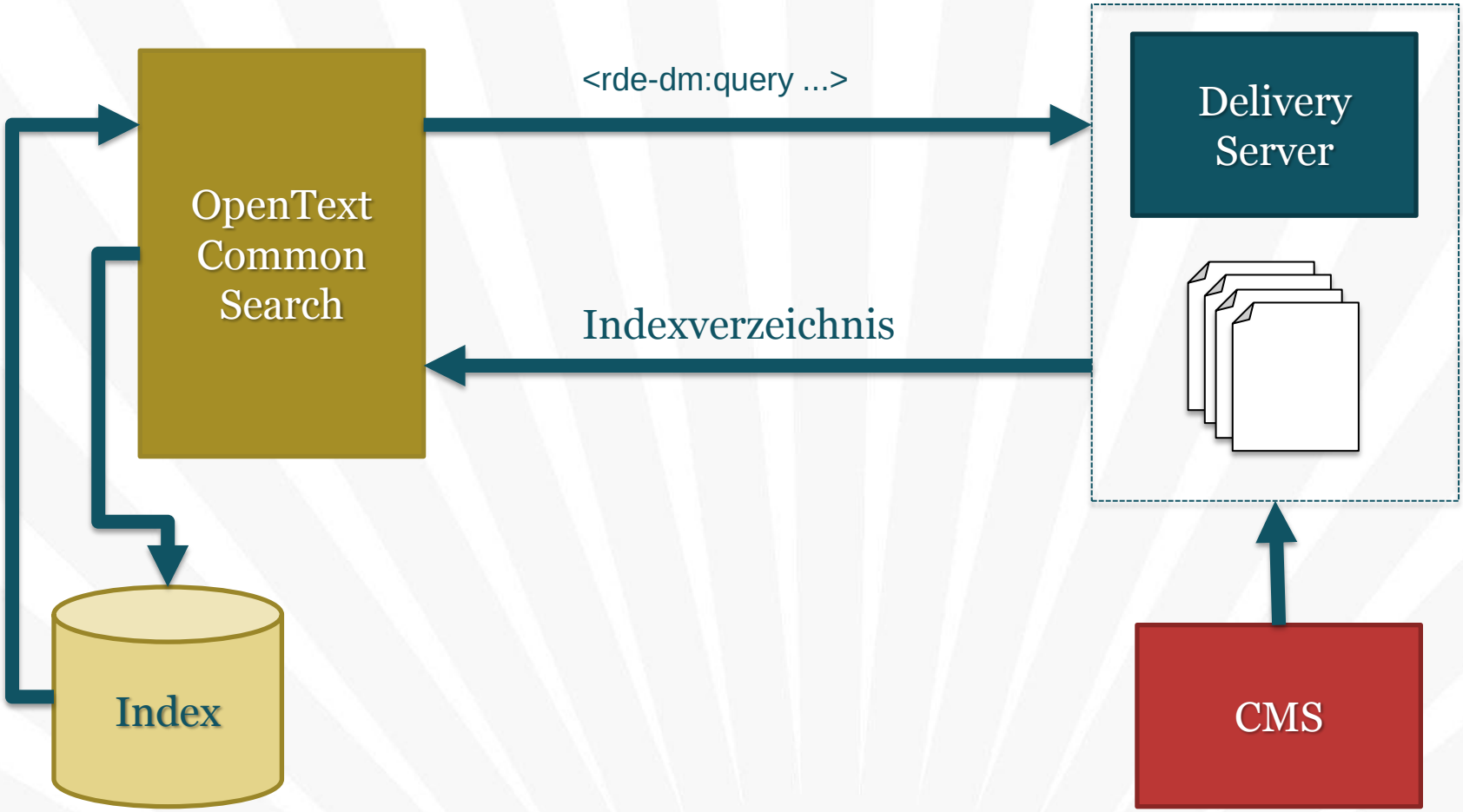
elbformat
content solutions

Der offizielle Nachfolger



elbformat content solutions

Übersicht



elbformat content solutions

Nachteile

- Komplexe Matrix für unterstützte Plattformen
  - MSSQL nur bei Installation unter Windows
  - Solaris nur mit SPARC, nicht x86
  - X Umgebung unter Linux benötigt
- Größerer IT Footprint
  - Mindestens ein Server je Stage
  - Zusätzliche Datenbanken
- Kein erweiterter Funktionsumfang
  - Lediglich Abbildung des Status quo

elbformat content solutions

Vorteile

- Gut dokumentierte Standardimplementierung
- Bessere Administrationsmöglichkeiten für Index
- Vollständig supported durch OpenText
- Professional Services verfügbar
- Migration des Status (problemlos) quo möglich

Indexierung ohne DeliveryServer

elbformat
content solutions

Herausforderungen

- Externe Indexierung
- Kein Publizierungsevent auf Livesystem
- Absicherung der Suchengine
- Index Bereinigung

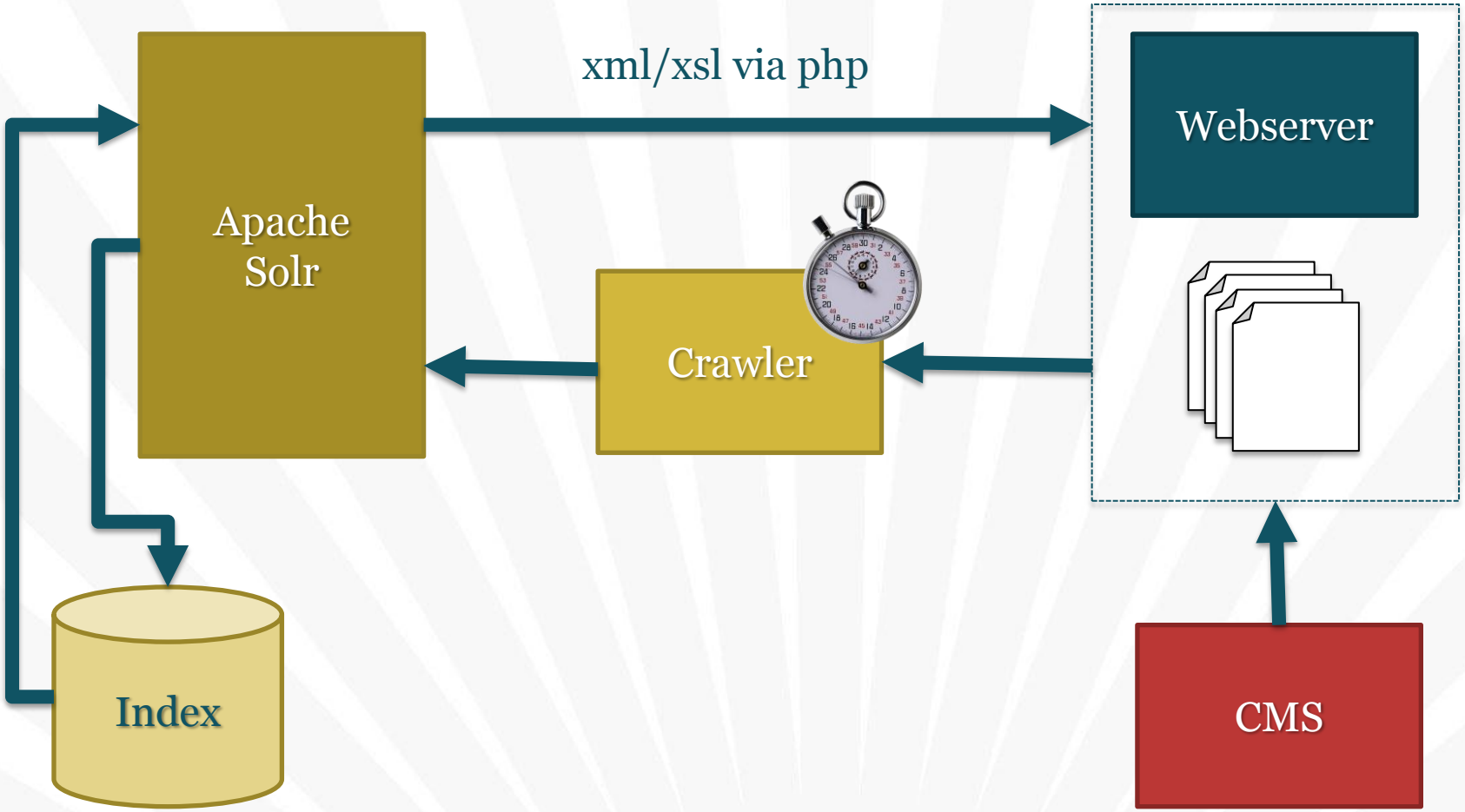
elbformat
content solutions

Mögliche Anbieter

- Kommerziell
  - Google Analytics
  - Autonomy IDOL
  - (CommonSearch)
  - Microsoft Fast Search
- OpenSource
  - Apache Solr/Lucene
  - ElasticSearch

elbformat
content solutions

Implementierung



Verwendete Features

- Volltextsuche für Webseiten und Binärdokumente
- Highlighting Suchbegriff
- Autosuggest
- Stemming für Autosuggest
- Redaktionelle Steuerung indexierbarer/nicht indexierbarer Seiten
- Einschränken zu indexierender Seitenelemente

elbformat
content solutions

Vorteile

- Einfaches Setup
- Guter Crawler verfügbar (Nutch)
- Autosuggest
- Erweiterte Optionen (Faceting, Spellcorrection)
- Kleiner IT Footprint, zusätzlicher Tomcat je Stage
- Einfache Darstellung der Suchergebnisse (xml/xsl)
- Anpassung der Crawler Logik dank OpenSource

elbformat
content solutions

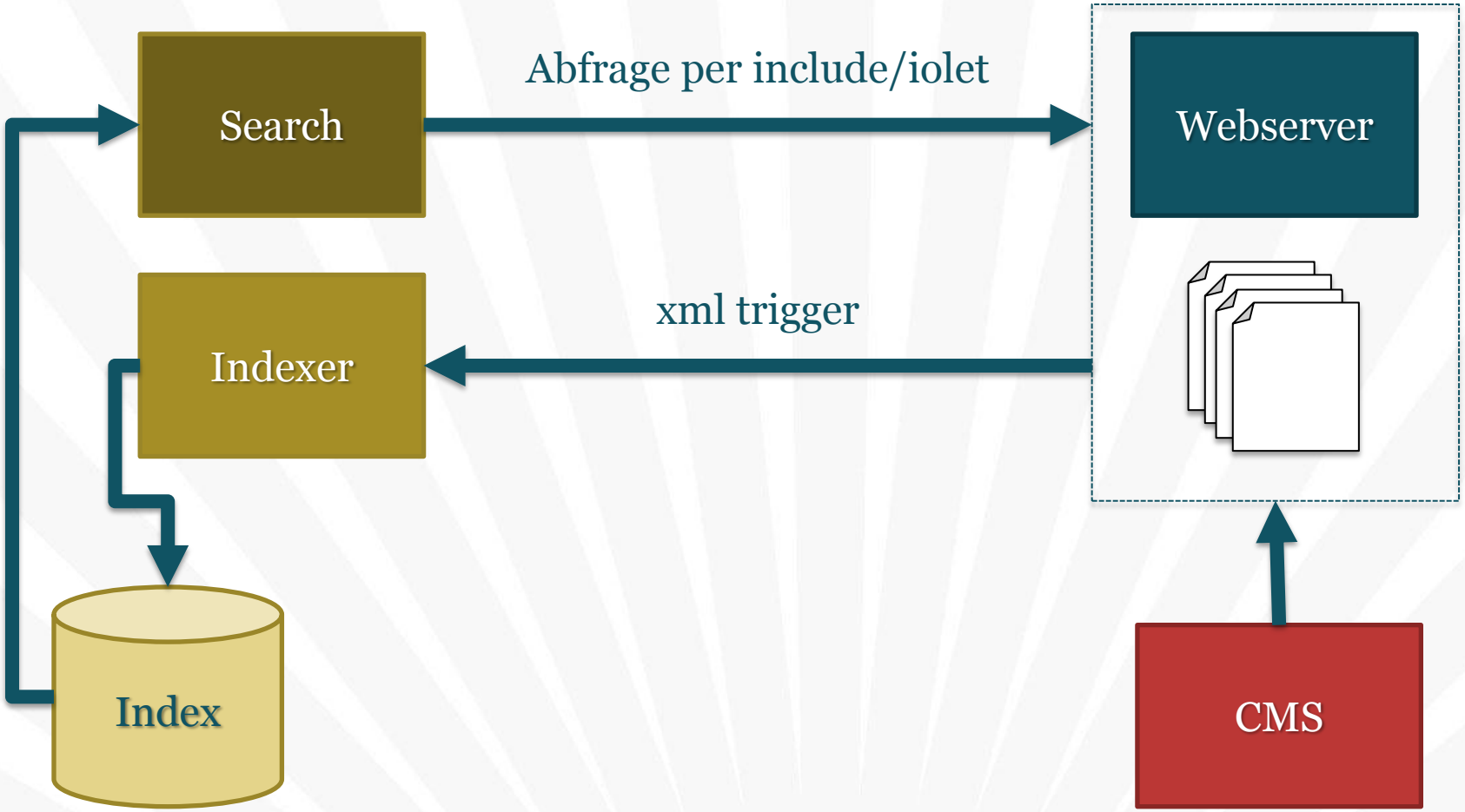
Nachteile

- Entfernen von Suchergebnissen nicht trivial
- Keine explizite Attribute möglich
- Berechtigungen schwierig
- Multi-Indexfähigkeit nicht einfach

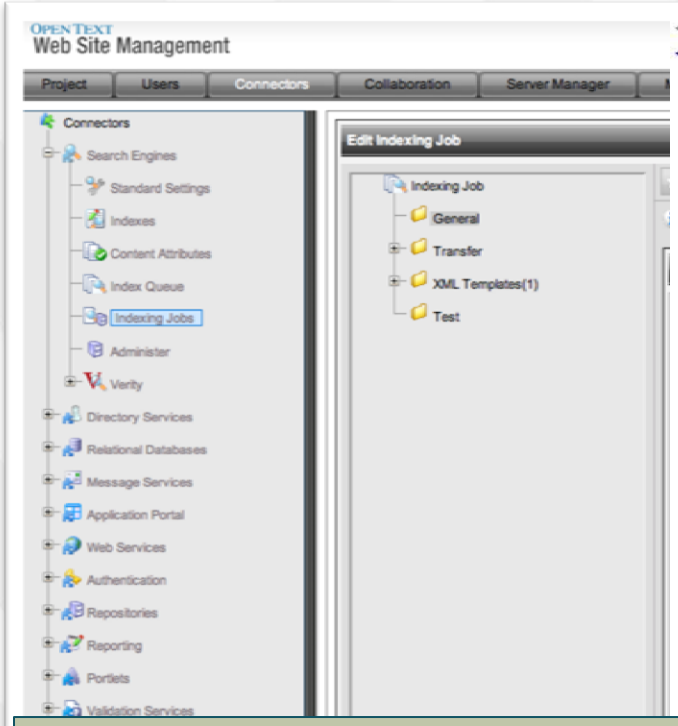
Integration über XML API

elbformat
content solutions

Prinzip



Trigger




```
<?xml version="1.0" encoding="UTF-8"?>
<rde-rd:indexitems xmlns:rde-rd="http://www.reddot.de/2000/rde/rd">
  <@index-item>
  <rde-rd:indexitem mode="[#context:mode#]" content="[#context:content-id#]"
   locale="[#context:language#]" project="[#context:project#]">
    <rde-rd:metadata>
      <rde-rd:keywords>
        <@keyword>
          <rde-rd:keyword name="[#context:keyword-name#]" type="[#context:keyword-type#]">
            <![CDATA[[#context:keyword-value#]]]>
          </rde-rd:keyword>
        </@keyword>
      </rde-rd:keywords>
      <rde-rd:constraints>
        <@constraint>
        <rde-rd:constraint mode="[#context:constraint-mode#]">
          <![CDATA[[#context:constraint-value#]]]>
        </rde-rd:constraint>
        </@constraint>
      </rde-rd:constraints>
    </rde-rd:metadata>
    <rde-rd:contenturl>
      <![CDATA[[#context:content-url#]]]>
    </rde-rd:contenturl>
    <rde-rd:task-guid>
      [#context:task-guid#]
    </rde-rd:task-guid>
    <rde-rd:mime-type>
```

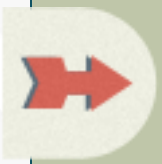
- Versenden des Triggers an eigenes Weblet ermöglicht zusätzliche Transformation
- Einfaches Deployment im DeliveryServer möglich

Indexierung

- Bereitstellen per xsearchengine Weblet statt rde Weblet
- Abfrage ohne Berechtigungen möglich
- Benötigt gültige Task Guid
- Automatisches Löschen aus Index Queue nach erfolgreicher Auslieferung
- Bereiche mit <rde-dm:query searchable=„false“> werden automatisch ausgeblendet

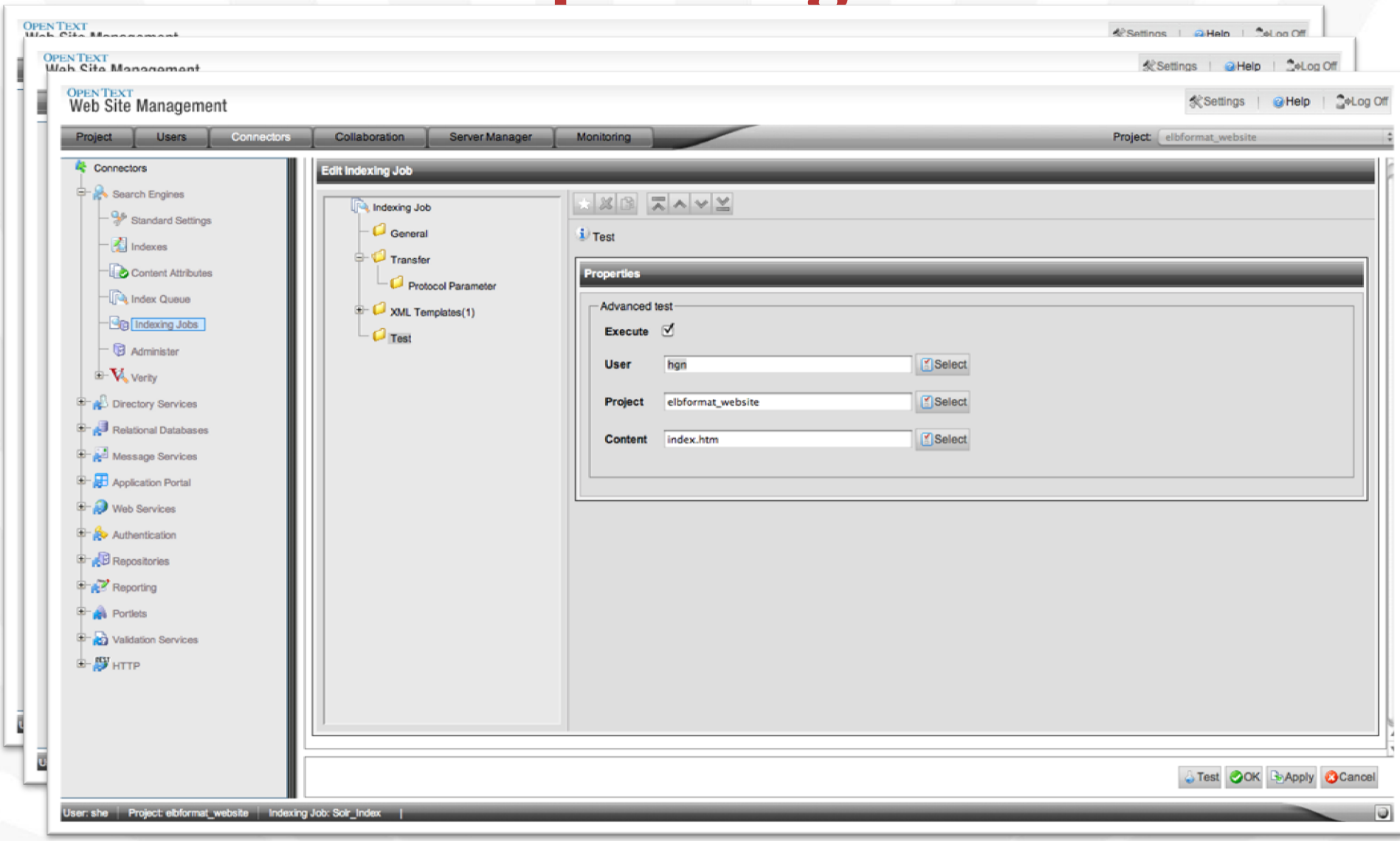
Suchanfrage

```
<rde-dm:include content=„http://localhost:8080/solr/select?q=Rentenversicherung“
stylesheet=„search.xsl“ />

<rde-dm:iolet name=„MySearch“ method=„GetResults“>
  <query>Rentenversicherung</query>
  <contentGroup>Produkte</contentGroup>
  <type>News</type>
</rde-dm:iolet>

<rde-dm:webservice name=„SearchWS“>
  <rde-dm:soap-message type=„prepared“ prepared-
```



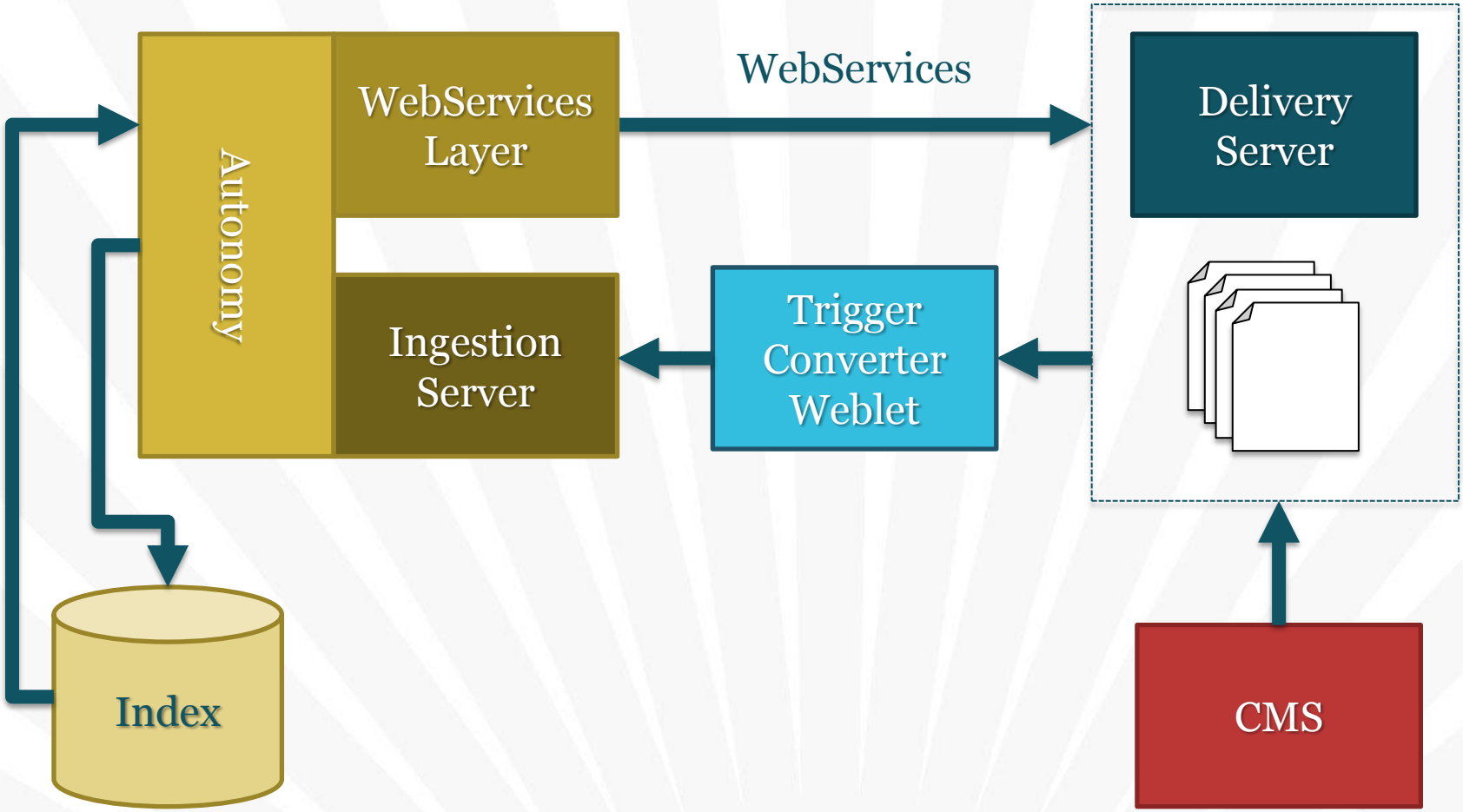
- Berechtigungen in Suchanfrage einbauen ist performanter
- Neuer Constraint: ACL-Constraint

Anpassungen



elbformat content solutions

Implementierungsbeispiel



elbformat content solutions

Herausforderungen

- Abarbeitung Index Queue/Löschen von Einträgen
- Berechtigungen
- Datensicherheit bei Transport ungesicherter Inhalte über internes Netz
- Performance (Clustering)
- Testen der Trigger (besser in Version 11)
- Aufbau Indexierungs-URL

elbformat content solutions

Vorteile

- Leichtgewichtige, schnelle Implementierung
- Flexible Anpassung durch freie Trigger-xml
- Anpassung Index URL möglich
- Einfaches Staging und Deployment
- Automatische Übernahme der Attribut-Mappings
- Neue Suchfeatures möglich
  - Spellcorrection
  - Autosuggest
  - Faceted Search
  - Sponsored Links

elbformat content solutions

Nachteile

- Keine direkte Unterstützung von Dynaments
- Eigene Implementierung der Berechtigungen notwendig
- Umständliche Suchanfragen bei komplexeren Anforderungen
- Nur bis zum Versenden des Triggers supported

Integration über OpenText Search Engine API

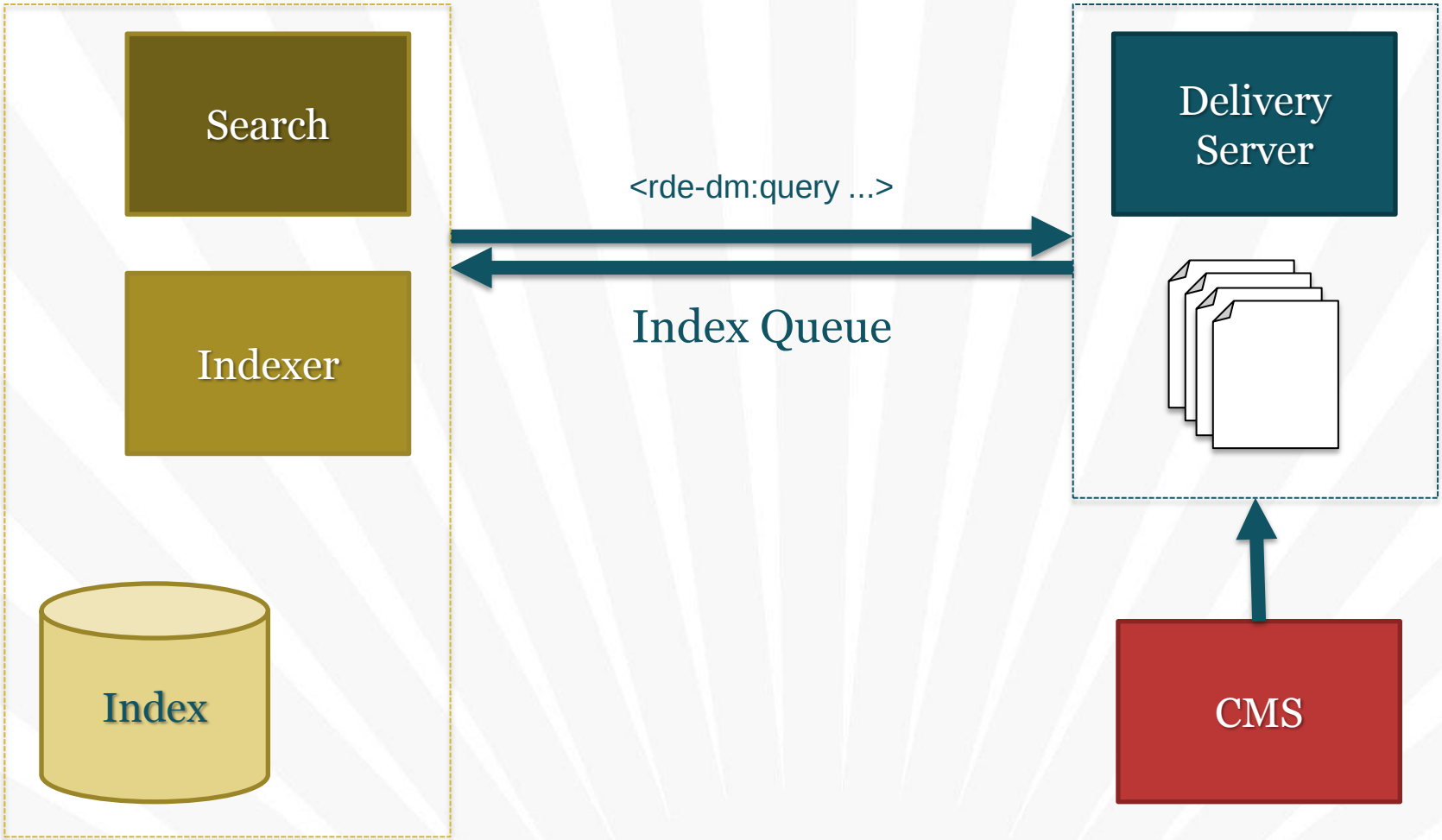
elbformat content solutions

Prinzip

- Entwicklung eigener Suchintegration in den DeliveryServer auf Java Basis
- Benötigt Installation der „Search Engine API“
- Vollständige Kontrolle über Searchengine
  - Server starten/stoppen
  - Index anlegen/löschen
  - Suchanfrage stellen
  - Zusätzliche Funktionen

elbformat content solutions

Übersicht



elbformat content solutions

Vorteile

- Vollständige Integration der Suche in den DeliveryServer
- Verwendung von Dynaments und Standard Queries
- Prototypische Integration für Lucene vorhanden
- Ergänzung um eigene Funktionen möglich per nativem Query String

elbformat
content solutions

Nachteile

- Aufwändigere Implementierung
- Weg zum ersten Ergebnis länger
- Implementierung muss vollständig sein
- Proprietäre Entwicklung notwendig

Fazit

elbformat content solutions

Fazit

- Anbindung externer Searchengines ist kein Hexenwerk und auch keine experimentelle Aufgabe
- Werden keine/wenige explizite(n) Attribute benötigt, ist eine Integration über einen Crawler einfach und sinnvoll
- Die XML API ermöglicht schnelle Ergebnisse, setzt aber voraus, dass die Searchengine Ergebnisse selber abruft
- Eine Integration über die Search Engine API ist für große, komplexe Umgebungen der beste Weg
- Alternative Searchengines können neue Funktionen für den DeliveryServer zur Verfügung stellen

Vielen Dank

Sebastian Henne
Geschäftsführer
sebastian.henne@elbformat.de
+49 (40) 209 3104 0