

# MyCoRe stellt sich vor

<http://www.mycore.de/>



**Frank Lützenkirchen**

Universität Duisburg-Essen

luetzenkirchen@ub.uni-duisburg-essen.de

# MyCoRe: Entstehung

- Ursprung: **miless**, Multimedialer Lehr- und Lernserver Essen, seit Ende 1997 entwickelt
- Erste **Nachnutzer**: Jena (Urmel, DB Thüringen), Freiburg (Freimore), Leipzig/IBM (Bach Digital)
- Zusätzliche **Anforderungen** entstehen, z.B. konfigurierbares Datenmodell, Anpassbarkeit
- Gemeinsame Entwicklung eines **Software-Kerns** auf Java- und XML/XSL Basis, als Open Source (GNU GPL), der **gemeinsame Funktionen** zur schnelleren Entwicklung mächtiger lokaler Dokumenten- und Publikationsserver enthält
- MyCoRe = My **Content Repository**  
MyCoRe = Miles Community **Core**

# MyCoRe: Beteiligte

- **Sehr aktiv** an der Entwicklung beteiligt zur Zeit

RZ Leipzig, RZ/UB Jena, RZ/UB Rostock, RZ Hamburg, UB Duisburg-Essen

- **Anwender** von miles / MyCoRe, weitere, teils ehemalige **Entwickler**  
(Projekte, spezielle Module, Diplomarbeiten):

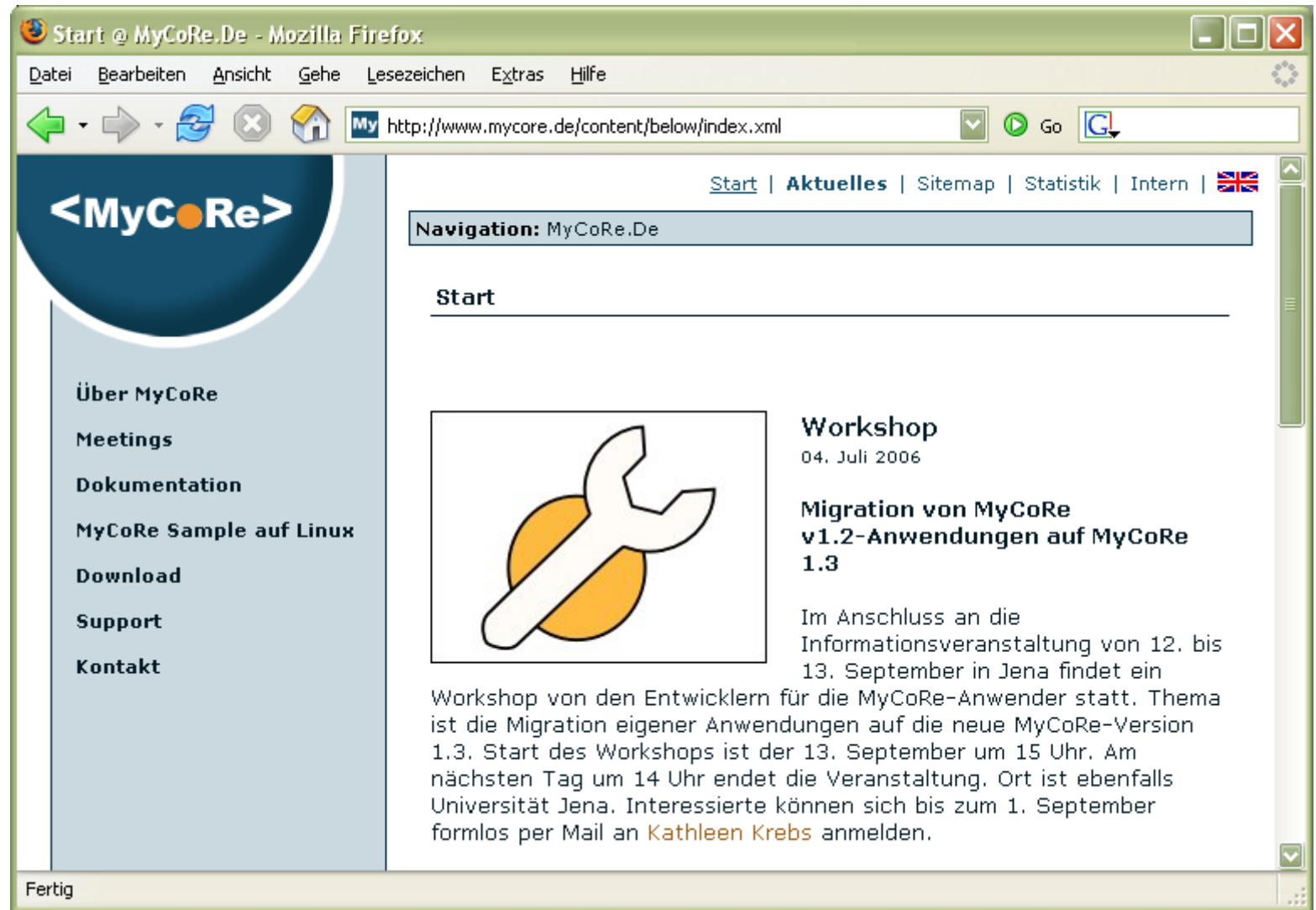
UB Münster, RZ Freiburg, UB Braunschweig, FU Berlin, RZ Bochum, UB Paderborn, UB Düsseldorf, UB Halle, TU München, RZ Bonn

- Kooperation mit **IBM**

IBM Content Manager Programmierung  
Kontakt zum Entwickler-Labor, HowTo Support

# MyCoRe: Organisation

- **Geschäftsstelle: [info@mycore.de](mailto:info@mycore.de)**  
AnsprechpartnerIn für Interessenten, Anwender und Entwickler, Koordination, Redaktionelle Arbeiten, Öffentlichkeitsarbeit, Tests
- Mailing-Listen: mycore-developer, mycore-user
- Telefonkonferenzen: ca. alle 14 Tage
- Entwicklertreffen: ca. 2-3 mal jährlich
- CVS Versionsmanagementsystem
- Webserver [www.mycore.de](http://www.mycore.de)
- Sourceforge Bug Tracking System
- Wiki: Notizen, Ideen, Entwürfe



# CVS Repository

CVS log for mycore/sources/org/mycore/services/fieldquery/MCRData2Fields.java - Mozilla Firefox

Up to [\[MyCoRe CVS Repository\]](#) / [mycore](#) / [sources](#) / [org](#) / [mycore](#) / [services](#) / [fieldquery](#)

[Request diff between arbitrary revisions](#)

---

Default branch: MAIN  
Bookmark a link to: [HEAD](#) / ([download](#))

---

Revision [1.26](#) / ([view](#)) - [annotate](#) - [\[select for diffs\]](#), *Wed May 17 08:36:44 2006 UTC* (3 months, 3 weeks ago) by *mcrfluet*  
Branch: [MAIN](#)  
CVS Tags: [release 1.3 fixes](#), [release 1.3 beta](#), [HEAD](#)  
Changes since [1.25](#): +10 -26 lines  
Diff to [previous 1.25](#)

Fixed indexing of date fields, we now use MCRMetaISO8601Date for parsing

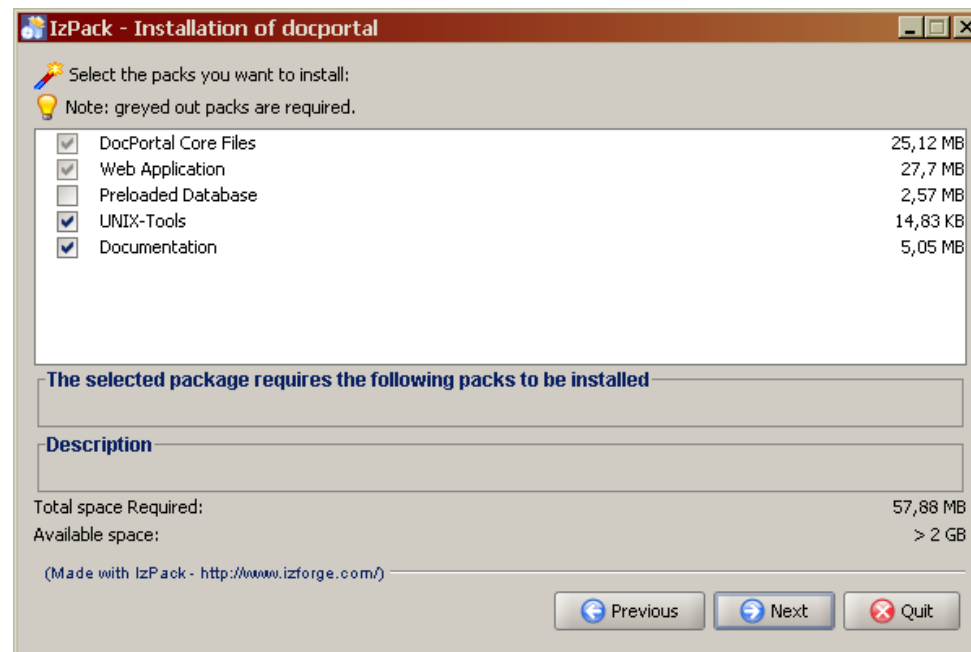
---

Revision [1.25](#) / ([view](#)) - [annotate](#) - [\[select for diffs\]](#), *Thu May 11 11:12:59 2006 UTC* (4 months ago) by *mcrfluet*  
Branch: [MAIN](#)  
Changes since [1.24](#): +137 -86 lines  
Diff to [previous 1.24](#)

For better performance, there is now just one stylesheet per index and source type

# MyCoRe Binärdistribution

- Kennenlernen von MyCoRe & DocPortal über einfaches Installationsprogramm (IzPack)
- Vollständige DocPortal Anwendung mit bereits geladenen Inhalten, Dateien und Benutzern
- Integrierte Datenbank (HSQLDB), integrierter Web Application Server (Jetty)



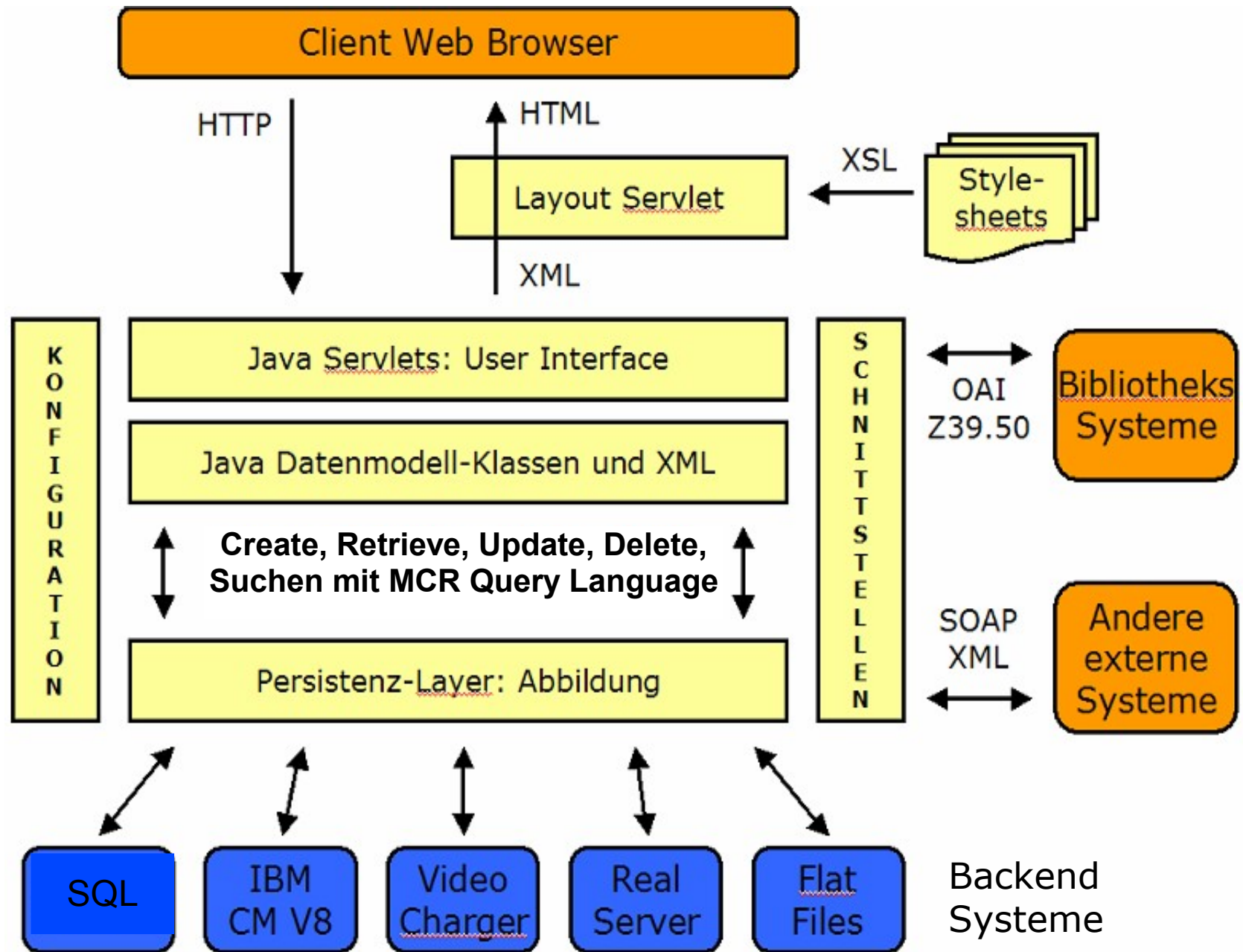
# MyCoRe 1.3 Beta Download

- Gepackt als ZIP- oder tar.gz-Distribution,  
Download über SourceForge oder mycore.de
- Alternativ direkt über einen CVS Client  
aus dem Versionsmanagementsystem  

```
cvs -d anoncvs@server.mycore.de/cvs co mycore
```
- **mycore**            MyCoRe Kernsystem  
  **docportal**        Beispielanwendung, typischer  
                      Dublin Core Dokumentenserver  
  **content**         Zusätzliche Beispielinhalte
- **Overview.pdf**  
  Überblick Funktionalität und Leistungsumfang  
  **UserGuide.pdf**  
  Installation und Konfiguration von DocPortal  
  **ProgGuide.pdf**  
  Dokumentation für Anwendungsentwickler



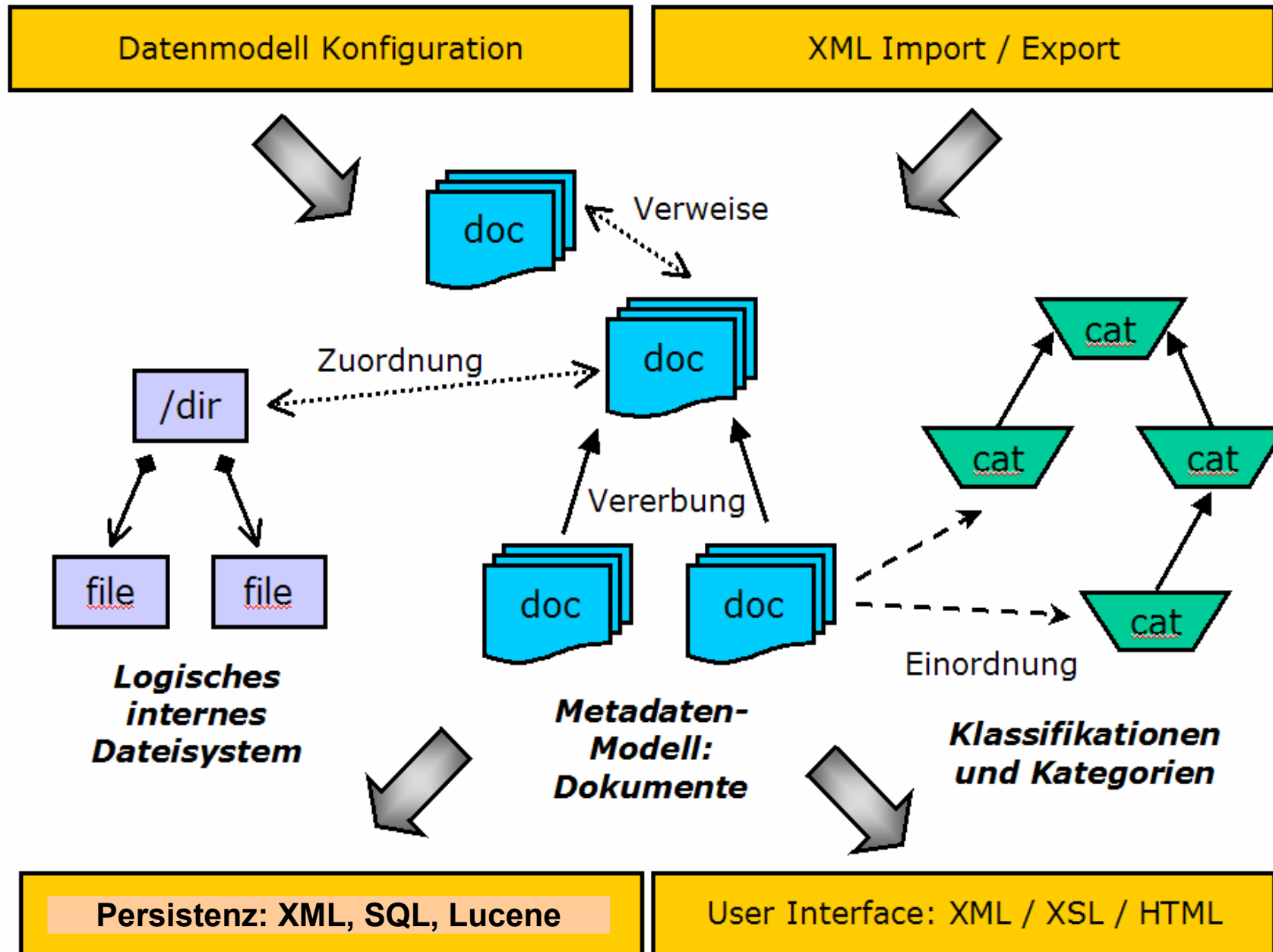
# MyCoRe Architektur



# Backend-Systeme: Datenhaltung

- Erforderlich: **Relationale Datenbank**
  - a) direkte Anbindung über Java JDBC / SQL:  
MySQL, IBM DB2, *HSQLDB (bereits integriert)*
  - b) indirekte Anbindung über *Hibernate*:  
unterstützt nahezu **alle** gängigen RDBMS
- Erforderlich: Speicherort für **Dateien und Metadaten**:  
IBM Content Manager, SQL BLOBs, *lokales Dateisystem*,  
entferntes Dateisystem (FTP, SFTP, WebDAV, CIFS)
- Optional: Streaming Server (Helix, VideoCharger)
- Weitere Komponenten werden mitgeliefert, z.B.  
**Suchmaschine**: Apache Lucene / Hibernate / SQL
- Erforderlich: Java **Web Application Server**  
z.B. Tomcat, IBM WebSphere, Jetty (integriert)  
zusätzlich empfohlen: Web Server (z.B. Apache)
- **Java** ab Version 1.4, Apache Ant, CVS Client
- OS: Linux, Windows, Solaris, AIX, Mac OS

# MyCoRe Datenmodell

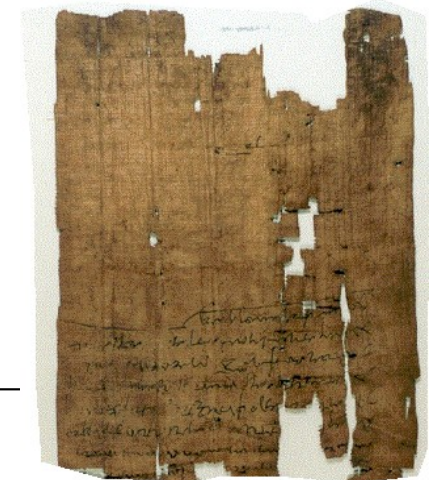
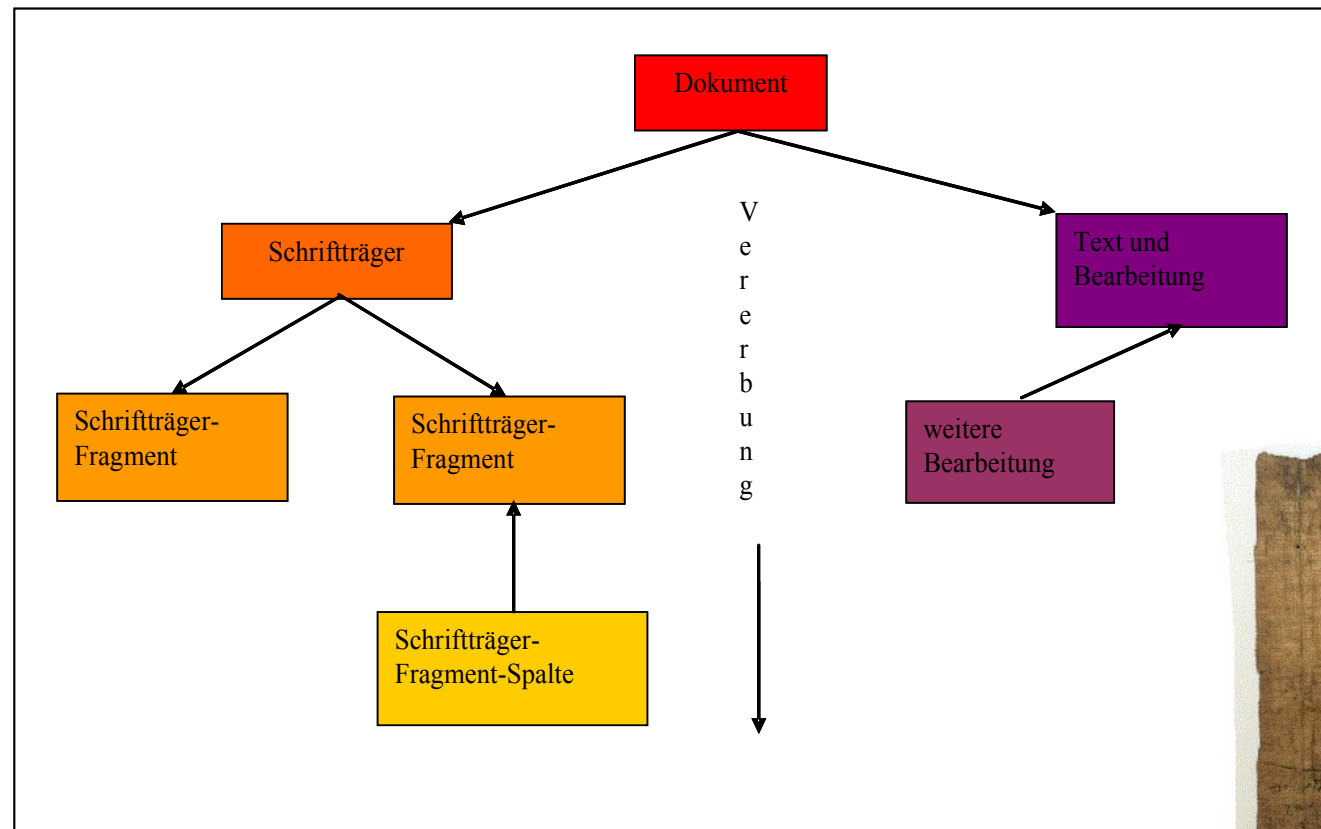


# Konfigurierbares Datenmodell

- **Metadatenmodell** ist frei konfigurierbar
- **Objekttypen** sind frei definierbar, z.B. *Dokument, Person, Institution*
- Je Objekttyp: Konfiguration der **Felder**:  
*Name, Typ, Wiederholbarkeit*  
z.B. *Titel, MCRMetaLangText, 1-n mal*
- Vordefinierte **Basisdatentypen**, z.B. *LangText, ISO8601Date, Number, Boolean, Link, Classification, ...*
- Erweiterung um eigene Datentypen möglich (z.B. GIS-Koordinaten, ...)

# Datenmodell: Beziehungen

- Hierarchien mit Metadaten-Vererbung
- Verweise zwischen Objekttypen
- Beispiel: Datenmodell Papyri-Anwendung



# Datenmodell: Klassifikationen





















- **Klassifikation:** Hierarchie von Kategorien oder flache Liste, z.B. Dewey Dezimal-Klassifikation, Publikationstyp, Organisationsstruktur der Universität, Fachklassifikationen wie PACS, MSC
- Objekte (Dokumente, Personen) werden Klassifikationskategorien zugeordnet
- Einfacher **Import** über XML-Dokumente
- **Suche** in Kategorien oder Hierarchien
- **Browsen** über Kategorien
- **Online-Klassifikations-Editor**

# Datenmodell: Klassifikationen

- ☞ [ \_\_22 Dok.] [Allgemeines](#)
  - ♦ [ \_\_4 Dok.] [Wissenschaft und Kultur allgemein](#)
  - ♦ [ \_\_17 Dok.] [Schrift, Buch, Bibliothek, Information und Dokumentation](#)
  - ♦ [ \_\_0 Dok.] [Nachschlagewerke, Bibliographien](#)
  - ♦ [ \_\_0 Dok.] [Adreßbücher, Telefonbücher](#)
  - ♦ [ \_\_0 Dok.] [Kalender](#)
  - ♦ [ \_\_0 Dok.] [Publizistik](#)
  - ♦ [ \_\_0 Dok.] [Kinder- und Jugendliteratur](#)
  - ♦ [ \_\_0 Dok.] [Comics, Cartoons, Karikaturen](#)
- ☞ [ \_\_18 Dok.] [Philosophie, Psychologie](#)
  - ♦ [ \_\_4 Dok.] [Grenzgebiete der Wissenschaft und Esoterik](#)
  - ♦ [ \_\_6 Dok.] [Philosophie](#)
  - ♦ [ \_\_16 Dok.] [Psychologie](#)
- ☞ [ \_\_5 Dok.] [Religion, Theologie](#)
- ☞ [ \_\_28 Dok.] [Sozialwissenschaften](#)
- ☞ [ \_405 Dok.] [Mathematik, Naturwissenschaften](#)
  - ♦ [ \_\_19 Dok.] [Natur, Naturwissenschaften allgemein](#)
  - ♦ [ \_\_26 Dok.] [Mathematik](#)
  - ♦ [ \_337 Dok.] [Informatik, Datenverarbeitung](#)
  - ♦ [ \_\_24 Dok.] [Physik, Astronomie](#)
  - ♦ [ \_\_11 Dok.] [Chemie](#)
  - ♦ [ \_\_6 Dok.] [Geowissenschaften](#)
  - ♦ [ \_\_24 Dok.] [Biologie](#)
- ☞ [ \_180 Dok.] [Angewandte Wissenschaften, Medizin, Technik](#)
- ☞ [ \_\_7 Dok.] [Kunst, Kunstgewerbe, Photographie, Musik, Spiel, Sport](#)
- ☞ [ \_\_10 Dok.] [Sprach- und Literaturwissenschaften, Belletristik](#)
- ☞ [ \_\_20 Dok.] [Geographie, Geschichte](#)
- ☞ [ \_\_0 Dok.] [Zusätzliche Gruppen](#)

# Datenmodell: Klassifikationen

Dewey Dezimal-Klassifikation (Insgesamt 526 Dokumente nach dieser Klassifikation eingeordnet)

- +  [\_21 Dok.] 000 Informatik, Informationswissenschaft, allgemeine Werke
- +  [\_\_6 Dok.] 100 Philosophie und Psychologie
- +  [\_\_1 Dok.] 200 Religion
- +  [\_42 Dok.] 300 Sozialwissenschaften
- +  [\_\_4 Dok.] 400 Sprache
-  [299 Dok.] 500 Naturwissenschaften und Mathematik
  - +  [\_\_0 Dok.] 500 Naturwissenschaften
  - +  [\_41 Dok.] 510 Mathematik
  - +  [\_\_0 Dok.] 520 Astronomie
  - +  [\_39 Dok.] 530 Physik
  - +  [119 Dok.] 540 Chemie
  - +  [\_\_8 Dok.] 550 Geowissenschaften
  - +  [\_\_0 Dok.] 560 Paläontologie
  - +  [\_91 Dok.] 570 Biowissenschaften; Biologie
  - +  [\_\_1 Dok.] 580 Pflanzen (Botanik)
  - +  [\_\_2 Dok.] 590 Tiere (Zoologie)
- +  [185 Dok.] 600 Technik, Medizin, angewandte Wissenschaften
- +  [\_12 Dok.] 700 Künste und Unterhaltung
- +  [\_\_2 Dok.] 800 Literatur
- +  [\_\_1 Dok.] 900 Geschichte und Geografie



# Internal Filesystem (IFS)

- Dokumente können nicht nur „flache“ Dateien, sondern auch **ganze Verzeichnisstrukturen** enthalten (auch mehrere 1000, > 1 GByte)
- IFS verwaltet Datei- und Verzeichnisdaten, z.B. Dateityp, Größe, **MD5 Prüfsumme**, ... in eigenen Datenbanktabellen
- FileContentStores speichern Dateiinhalte:
  - Beliebige Dateisysteme: lokal oder entfernt angebunden über **FTP, SCP/SFTP, CIFS** / Samba, **WebDAV** (Apache Commons VFS), **IBM Content Manager 8**
- Upload: Über HTML Formular oder Applet
- Download: **ZIP-Generierung** on-the-fly
- Integrierte Volltextsuche, Image Viewer

# Datei-Upload mit Java-Applet

MILESS: Derivat anzeigen - Microsoft Internet Explorer

Adresse <http://miless.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-13758.xml?XSL.Cc> Wechseln zu Google db-thueringen

HOME

IM GESAMTBESTAND  
übergreifend suchen  
Dokumententypen  
Medientypen  
nen von A-Z  
ne Suche  
al: Suche in MILESS  
ACHBEREICHEN  
TERAPPARATE  
TATIONEN  
LLE SAMMLUNGEN  
BLOGGRAPHIE

OKUMENTENKORB

FÜR AUTOREN  
/ PLUGINS  
ILESS

um Hilfe von A-Z

In den Korb Statistik

Allgemeine Angaben:		Dateiformat:	PlugIn / Viewer:
Ort der Dateien:	Dateien sind intern in MILESS abgelegt	SMIL Synchronized Multimedia	PlugIn / Viewer herunterladen
Größe:	246,93 MB in 3 Dateien	Real Video	PlugIn / Viewer herunterladen

ZIP-

Eing

Letz am:

Dok

Hau

Liste

Date

dkf\_6b04

Kam b39d

Streaming Video

216,07 KB pro Sekunde  
Dauer bisher 3 Sek. / geschätzt noch 31 Sek.

Übertrage Datei 2 von 15  
648,21 KB von 7,21 MB insgesamt übertragen  
9%

Datei ant.jar  
640,00 KB von 936,38 KB übertragen  
68%

Screen\_25012006.rm 147,61 MB 600 kBit / Sek. Real Video 25.01.2006 14:45:40 als ZIP-Datei  
5d0a79e3a3233973a19319c13c354c9a  
Streaming Video

Sek. mit 25 Bildern / Sek.  
34 Min. 15 Sek. mit 15 Bildern / Sek.

# Audio-/Video-Streaming

- **Helix Universal Server, IBM VideoCharger**
- Streaming von AVI, MPEG 1, MPEG2, Real, Windows WMV, Apple Quicktime, WAV, MP3
- Audio-/Video-Streaming-Server sind für den Endnutzer **transparent** integriert
  - Benutzer lädt Video in MyCoRe hoch
  - MyCoRe erkennt anhand des Dateityps, dass das Video auf dem Streaming Server (File Content Store) gespeichert werden soll
  - Benutzer klickt auf Videodatei, Player startet
- Einzelne Szenen eines Videos lassen sich gezielt einzeln über URL wiedergeben

# Audio-/Video-Streaming

- Übernahme technischer Metadaten (Bitrate, Framerate, Dauer etc.) vom Streaming Server

Dateiformat:	Größe der Datei:		Dateityp:
encodings/tracheostomie-DSL-350-kBit.rm Streaming Video	21,49 350 kBit	MB / Sek.	Real Video 8 Min. 30 Sek. mit 25 Bildern / Sek.
encodings/tracheostomie-Dual-ISDN-106-kBit.rm Streaming Video	6,57 106 kBit	MB / Sek.	Real Video 8 Min. 30 Sek. mit 25 Bildern / Sek.
encodings/tracheostomie-Modem-034-kBit.rm Streaming Video	2,14 34 kBit	MB / Sek.	Real Video 8 Min. 30 Sek. mit 25 Bildern / Sek.
encodings/tracheostomie-MPEG1-1300-kBit.mpeg Streaming Video	82,71 1,3 MBit	MB / Sek.	MPEG 1 Video 8 Min. 29 Sek. mit 25 Bildern / Sek.



# Audio-/Video-Streaming

**Frank Lützenkirchen**  
 Uni Duisburg/Essen

**Query Service**

**MyCoRe Entwicklertreffen Jena**  
 FSU Jena URZ/Multimediazentrum Friedrich-Schiller-Universität Jena  
 22. Februar 2006

**Datenmodell der Suche**

Es wird nicht direkt in den XML-Strukturen der Metadaten bzw. den Dateien gesucht, sondern in daraus abgeleiteten Suchfeldern. Die Definition dieser Suchfelder wird für jede MyCoRe Applikation in der Konfigurationsdatei `searchfields.xml` vorgenommen (vgl. `docportal/config/searchfields.xml`). Das XML Schema dieser Datei ist in `searchfields.xsd` zu finden.

In jeder Applikation gibt es ein oder mehrere Indizes. Ein **Index** ist eine Menge von Suchfeldern, die logisch zusammengehören. Alle Felder eines Index sind durch die gleiche Instanz eines **Searchers** durchsuchbar, der diese Felder mit Hilfe eines zugrundeliegenden Datenbanksystems (SQL, IBM Content Manager) oder einer Suchmaschine (z. B. **Lucene**) durchsuchbar macht.

```

graph LR
    JavaClass -- 1 --> Searcher
    Searcher -- 0..* --> Index
    Index -- 1 --> Field
    Field -- 1..* --> Index
    
```

Jeder Index besteht aus ein oder mehreren **Feldern**, die jeweils einen global eindeutigen Namen besitzen. Jeder Index besitzt eine eindeutige ID.

```

<searchfields>
  <index id="metadata">
    <field name="title" ... />
    <field name="dateCreated" ... />
  </index>
  <index id="content">
    ...
  </index>
</searchfields>
    
```

Vorgemerkte Clips | MyCoRe Entwicklertreffen Jena Februar 2006 | 500 Kbit/s | 0:49 / 1:30:47

Real Guide | Musik und Meine Bibliothek

# MyCoRe Query Service

- Abbildung von Daten auf Suchfelder
- Wahlmöglichkeit zwischen mehreren **Implementierungen**, basierend auf
  - Apache Lucene (Open Source, Volltextsuche)
  - SQL (Relationale Datenbank)
  - XSL im Hauptspeicher (zu Testzwecken)
  - IBM Content Manager (in Entwicklung)
- **Verteilte Suche** über mehrere Server via *WebServices* (SOAP) Schnittstelle oder Java-Schnittstelle
- Sortierung der Trefferliste, Suche verfeinern, „Sie haben gesucht nach...“

# Was ist durchsuchbar?

- **Metadaten** von Dokumenten, Personen:  
z.B. Titel, Autor, Datum
- Metadaten von gespeicherten Dateien  
z.B. Dateiname, Typ, Größe
- **Volltexte** von gespeicherten Dateien  
z.B. PDF, OpenOffice, HTML
- XML **innerhalb** gespeicherter Dateien  
z.B. Excel-Tabelle als XML-Dokument  
manifest.xml aus E-Learning Scorm Modul
- Über **Erweiterungen** (Event Handler)  
extrahierte technische Metadaten, z.B.  
EXIF-Daten von Bildern, ID3 von MP3s

# Suchmasken: frei konfigurierbar

Expertensuche nach Dokumenten:

Titel:  enthält

Autor:  enthält

Identifikation: wie (\*.?)

Herkunft: (bitte wählen)

Subjekt: (bitte wählen)

Format: (bitte wählen)

Sprache: (bitte wählen)

Typ: (bitte wählen)

Erstellungsdatum: =

Suche nach enthaltenen Dateien:

Volltext: enthält

Dateiname: wie (\*.?)

Größe in Bytes: =

Endung: wie (\*.?)

Änderungsdatum: =

Suche auf: ☒ diesem Server ☐ allen Servern ☐ diesem und ausgewählten Servern:

☐ Lokal via Webservice ☐ DocPortal an der Universität Leipzig

sortiert nach (bitte wählen)  aufsteigend

Alle Felder verknüpfen mit  und

Maximal  100  Treffer,  10  pro Seite anzeigen



# MyCoRe Query Language: MCRQL

- **Einfache** Suchausdrücke:  
`title contains 'Optik'`
- **Zusammengesetzte** Suchausdrücke:  
Kombination mit UND/ODER/NICHT:  
`(objectType = "document") and  
((title contains "Regenbogen") or  
(fileSize > 500000))`
- **Suchoperatoren:** =, <, >, <=, >=, contains, phrase (Phrasensuche), like (Suche mit Platzhaltern \*,?)
- **Suchfeldtypen:** text, name, identifier, date, time, timestamp, boolean, integer, decimal

# Expertensuche

Suchausdruck in MCRQL eingeben:

`(objectType = "document") and ((title contains "Regenbogen") or (author contains "Kupferschmidt"))`

Maximal  Treffer,  pro Seite anzeigen

Suchen...

## Beispiele:

title contains Regenbogen

`(objectType = "document") and ((title contains "Regenbogen") or (author contains "Kupferschmidt"))`

## Syntax:

field operator value

`(field1 operator1 value1) and (field2 operator2 value2) [and (...) ...]`

`(field1 operator1 value1) or (field2 operator2 value2) [or (...) ...]`

Field:	Type:	Index:
id	identifier	metadata
objectType	identifier	metadata
link	identifier	metadata
parent	identifier	metadata
modified	timestamp	metadata
allMeta	text	metadata

# Such-Plug-in für Webbrowser

- für Firefox, Netscape 6, Opera, SeaMonkey

MyCoRe-Sample - Mozilla Firefox

Zeichen Extras Hilfe

http://localhost:8291/servlets/MCRSearchServlet?mode=results&id=-sxo68vos5v5jes02pgz Go My Kupferschmidt

MyCoRe/Organisation/LaufendeProjekte - ...

Start Sitemap WCMS Nutzer

Navigation: DocPortal - The MyCoRe-Sample > Suche > nach Dokumenten einfach

**Suchergebnisse**

Suche verfeinern | Neue Suche Es wurden 3 Objekte gefunden

**Die 99 582 am Lokschruppen in Schönheide.** Jens U. Kupferschmidt >>  
 Dieses Bild wurde Pfingsten 2000 aufgenommen. Der Zug mit der Lok 99 582 steht in Schönheide zur Abfahrt...  
 image , Bild , DocPortal\_document\_00410901 , zuletzt geändert am: 31.05.2006

**Die 99 729 (099 729-6) in Bertsdorf beim Wasserfassen.** Jens U. Kupferschmidt >>  
 Dieses Bild wurde im Januar 1995 aufgenommen.  
 image , Bild , DocPortal\_document\_00410902 , zuletzt geändert am: 31.05.2006

**Ein Modellbaubasteltipp - Umbau eines Wagens zum Hühnerstall.** Jens U. Kupferschmidt >>  
 Dies ist eine Kurzanleitung für einen Modellumbau.  
 text , Artikel , DocPortal\_document\_00410903 , zuletzt geändert am: 31.05.2006

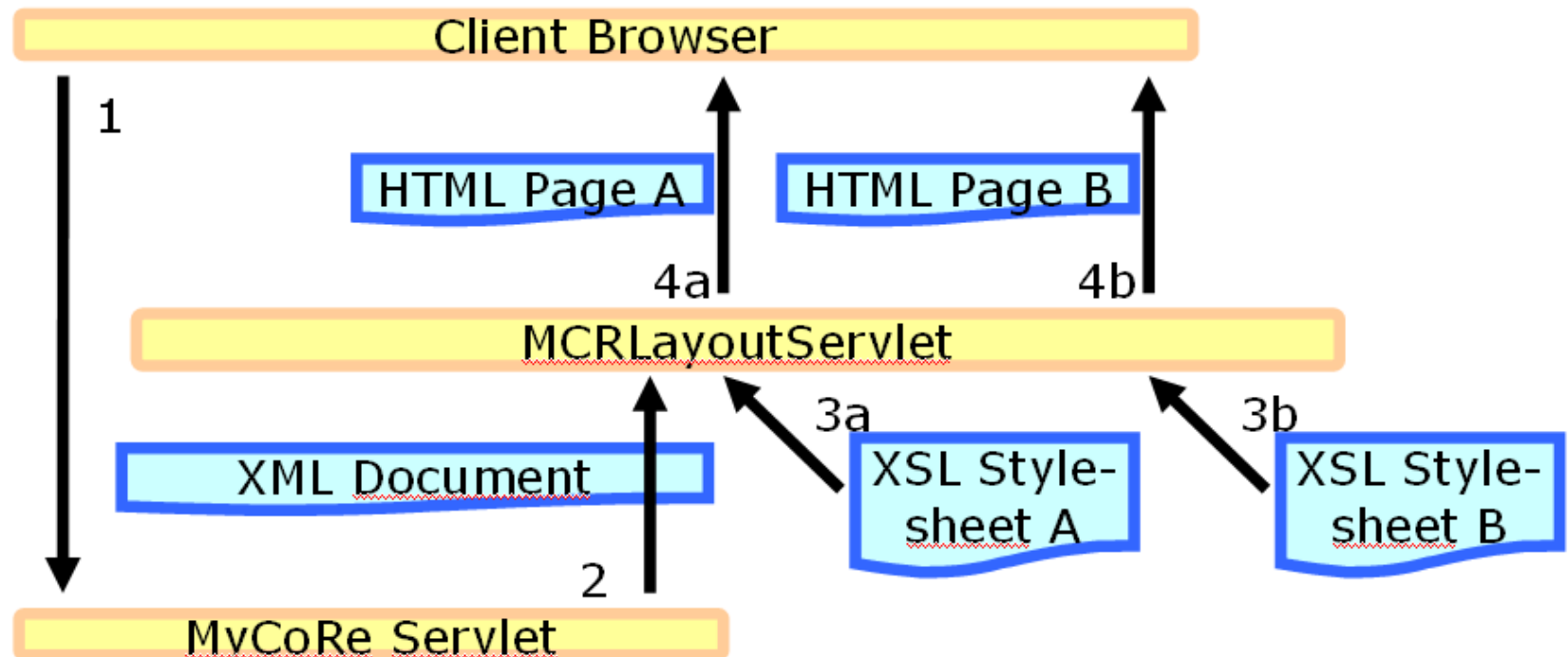
Anzahl der Trefferseiten[1]

# MyCoRe Benutzer-Interface

- **Command Line Interface (CLI):**  
auf dem Server, zu administrativen Zwecken,  
z.B. Batch-Import/Export vieler Daten via XML
- **Web Interface:** Ajax-Ready!
  - Java Servlets erzeugen XML-Ausgaben
  - Serverseitige Generierung von HTML-Seiten aus diesen Ausgaben über **XSL** Stylesheets
  - Dadurch Trennung von Daten, Layout und Logik, einfachere/getrennte Anpassbarkeit
- Alternative Implementierung:
  - Java Server Pages (JSP)
  - HTML-Seiten, die Java-Code enthalten

# Web Content Management System

- WYSIWYG-Editor für statische Webseiten erlaubt einfaches Bearbeiten von HTML
- Navigationsmenü online bearbeitbar
- Wechsel von Layout-Templates und CSS



# I18N: Mehrsprachigkeit

- Mehrsprachige Web-Oberfläche
- Bezeichner sind in sprachabhängigen Dateien abgelegt (Java Resource Bundle):

```
messages_de.properties, messages_en.properties
```

```
metaData.document.title = Titel  
metaData.edit = Bearbeiten
```

- Statische Webseiten enthalten sprachabhängige Abschnitte:

```
<section xml:lang="de" title="Daten bearbeiten">  
...  
</section>  
<section xml:lang="en" title="Edit data">  
...  
</section>
```

# MyCoRe Editor Framework

- Konfigurierbares Datenmodell erfordert konfigurierbare Eingabemasken, Suchmasken
- Such- und Eingabemasken werden über eine XML-Definitionsdatei erstellt
- MyCoRe Editor Framework erzeugt ein XML-Dokument aus den Eingaben im HTML-Formular bzw. stellt die XML-Quelldaten im Formular da
- Einfache Abbildung zwischen **XML** und Formular
- **Eingabevalidierung**, z.B. required, min/max, syntaktische Korrektheit von Email-Adressen...
- Einfache **Wiederholbarkeit** von Feldern oder ganzer Formularbereiche

# MyCoRe Editor: Beispiel

```
<repeater min="1" max="3" lines="off">
  <textfield width="60" maxlength="250" id="tf.email"
    autofill="@uni-duisburg-essen.de">
    <condition id="cond.email" regexp="\S+@\S+\.\S+">
      <label xml:lang='de'>Bitte geben Sie eine
        syntaktisch korrekte EMail-Adresse ein!</label>
      <label xml:lang='en'>Please enter a valid email address
        </label>
    </condition>
  </textfield>
</repeater>
```

E-Mail:	<input type="text" value="luetzenkirchen@bibl.uni-essen.de"/>	<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="↓"/>
	<input type="text" value="luetzenkirchen*ub.uni-duisburg-essen.de"/>	<input type="button" value="!"/> <input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="↑"/>
Homepage:	<input type="text" value="http://www."/>	<input type="button" value="Bitte geben Sie eine syntaktisch korrekte EMail-Adresse ein!"/>



# Benutzer- und Rechteverwaltung

- **Benutzer** (Login-User)
- Benutzer gehören **Gruppen** an
- **Zugriffsrechte** auf Dokumente  
(lesen, bearbeiten)  
können auf Benutzer oder Gruppen, oder  
auf IP-Adressbereiche beschränkt  
werden
- **Privilegien** regeln z.B. das Recht, ein  
neues Dokument einstellen zu dürfen

# URNs: Persistent Identifier

- Unterstützung von URNs (urn:nbn:de:...)  
vgl. [www.persistent-identifier.de](http://www.persistent-identifier.de)
- **Erzeugung**: wählbarer Algorithmus -  
Zeitstempel, fortlaufende Nummer, ...

```
MCR.URN.SubNamespace.Essen.Prefix =
```

```
    urn:nbn:de:hbz:465-
```

```
MCR.URN.SubNamespace.Essen.NISSBuilder =
```

```
    org.mycore.services.urn.MCRNISSBuilderDateCounter
```

```
MCR.URN.SubNamespace.Essen.NISSPattern =
```

```
    yyyyMMdd-HHmms-
```

```
erzeugt z.B. urn:nbn:de:hbz:465-20060912-102358-7
```

- Lokaler **Resolver** (URN -> Dokument)
- Registrierung bei DNB über OAI (**Epicur**)

# Modul: OAI-Schnittstelle

- OAI Data Provider Implementierung:  
**OAI-PMH 2.0** Protocol for Metadata Harvesting
- Abbildung beliebiger Klassifikation auf OAI Sets
- Unterstützung verschiedener Metadatenformate  
**OAI\_DC**  
Dublin Core  
**XMetaDiss**  
Automatische Meldung von Dissertationen  
und Hochschulschriften an die  
Deutsche Nationalbibliothek  
**Epicur**  
Automatische Registrierung von URNs  
(urn:nbn:de, Persistent Identifier) bei der  
Deutschen Nationalbibliothek
- Datenformate und Ausgaben anpassbar (XSL)



# Modul: Z39.50-Server

- Implementierung eines Z39.50 Servers auf Basis von **JZKit** (Open Source)
- Implementierung: FU Berlin, Duisburg-Essen
- Abbildung von Z39.50 Anfragen in **PQF** (Prefix Query Format) auf MyCoRe Query Language
- Derzeit wird nur ein Teil der Z39.50 Abfragesprache unterstützt
- Abbildung von Z39.50 Attributen auf MyCoRe Query Suchfelder über Konfigurationsdatei
- Transformation von MyCoRe Metadaten ins MAB2 Endlosformat über ein XSL Stylesheet

# Modul: Image Viewer

- Bildbetrachter im Browser mit serverseitiger Transformation der Bilder
- Verwendet Java Advanced Imaging API
- Geeignet z.B. für hochauflösende Digitalisate
- Größenanpassungen: **Skalieren**
- Bereitstellen von **Ausschnitten** (Sie wählen im Übersichtsbild einen anzuzeigenden Bereich)
- Generieren von **Thumbnails**
- Caching generierter Bilder
- Skalierung on-the-fly bei Bedarf

Vielen Dank für Ihre Aufmerksamkeit!

