



Search.
Discover.
Share.

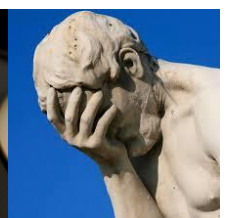


An open-source resource
discovery portal.

Mehrere VuFind-Ableitungen – Best Practice?

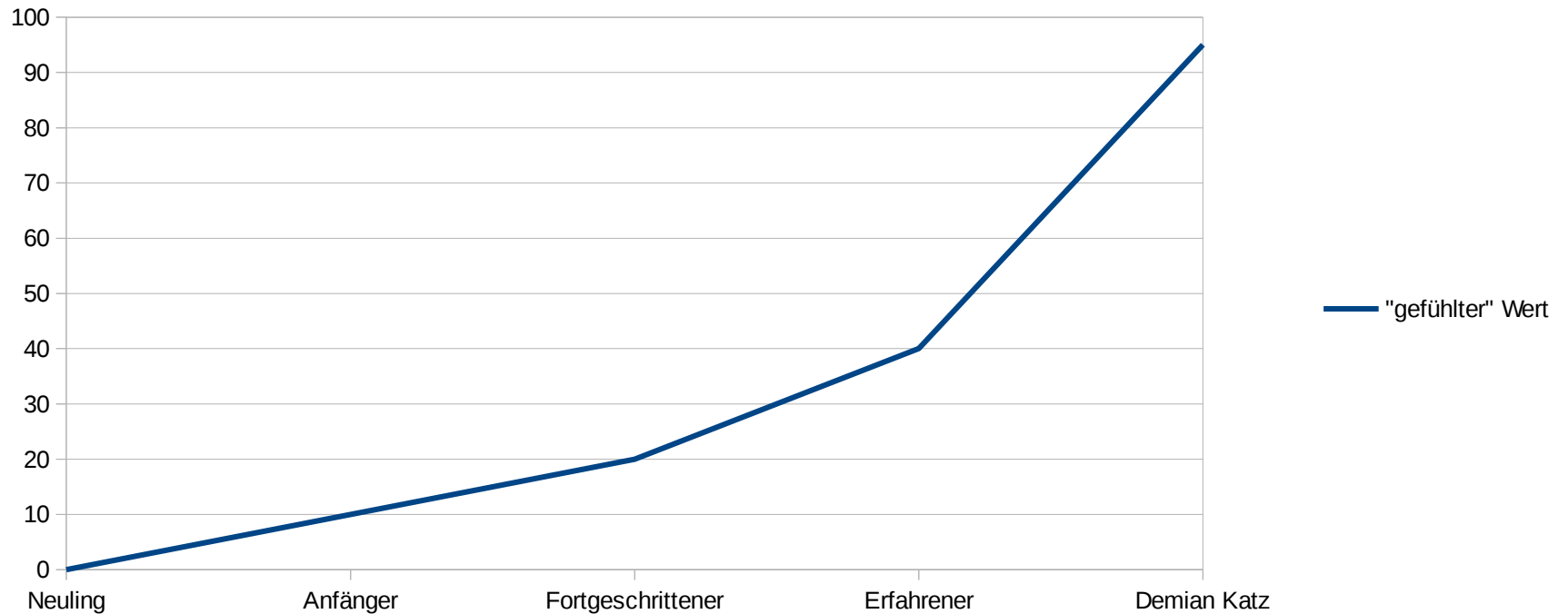
Am Beispiel KrimDok, IxTheo und RelBib

04.10.17, Mario Trojan





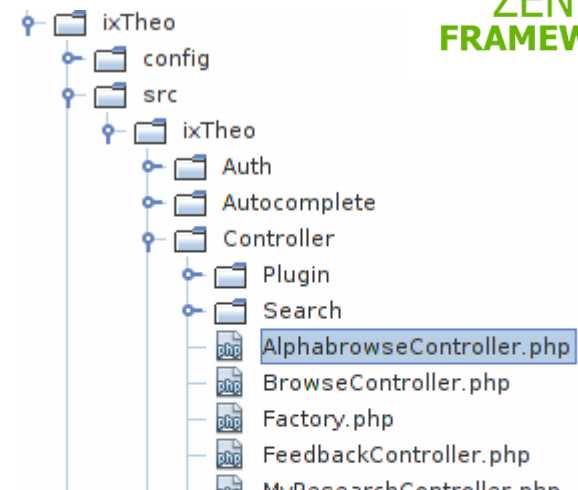
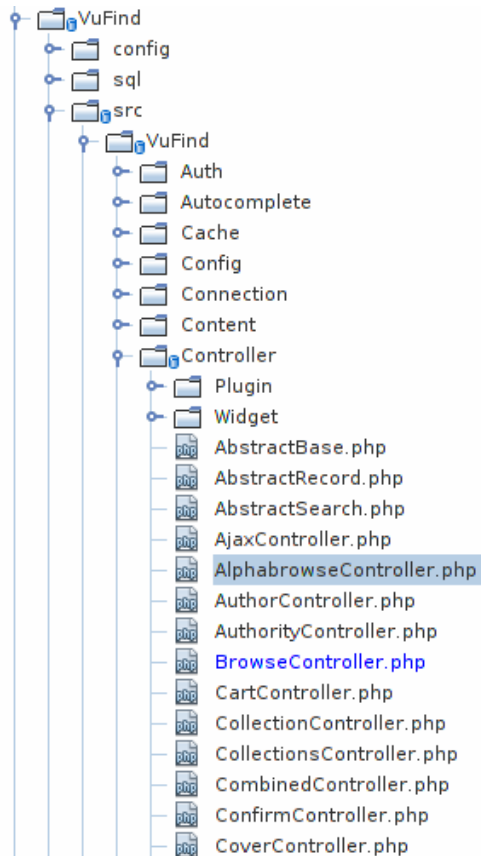
VuFind-Lernkurve



Ab hier wird Code produziert!



Zend Framework: Erweiterbarkeit



```
'controllers' => [
    'factories' => [
        'browse' => 'ixTheo\Controller\Factory::getBrowseController',
        'record' => 'ixTheo\Controller\Factory::getRecordController',
    ],
    'invokables' => [
        'alphabrowse' => 'ixTheo\Controller\AlphabrowseController',
        'BibleRangeSearch' => 'ixTheo\Controller\Search\BibleRangeSearchController',
        'feedback' => 'ixTheo\Controller\FeedbackController',
        'KeywordChainSearch' => 'ixTheo\Controller\Search\KeywordChainSearchController',
        'MyResearch' => 'ixTheo\Controller\MyResearchController',
        'Pipeline' => 'ixTheo\Controller\Pipeline',
        'search' => 'ixTheo\Controller\SearchController',
        'StaticPage' => 'ixTheo\Controller\StaticPageController',
    ],
],
```



Theme-Ableitung / Beispiel: Detailansicht



- (=> core.phtml)
- Beispiel: Zusätzliches Feld einbauen

Fanatismus

1. Verfasser:	Utsch, Michael 1960- (VerfasserIn)
Medienart:	Aufsatz
Sprache:	Deutsch
Verfügbarkeit prüfen:	HBZ Gateway
Journals Online & Print:	Verfügbarkeit prüfen
In:	Materialdienst Jahr: 2017, Jahrgang: 80, Heft: 2, Seiten: 69-72
normierte Schlagwort(-folgen):	B Fanatismus

Beschreibung	Internformat
Beschreibung:	Printed Material
ISSN:	0721-2402



Beispiel: Neuer Controller (invokable)



- module.config.php:
 - Klasse für Controller anlegen und von VuFind\Controller\AbstractBase ableiten, z.B. „MyResearchController“. Funktion hinzufügen, z.B. „homeAction“
 - In \$config[„controllers“][„invokables“] registrieren
 - „MyResearch“ => „ixTheo\Controller\MyResearchController“
 - StaticRoute hinzufügen
 - MyResearch/Home
(=Suburl um Action aufzurufen)

```
'controllers' => [
    'factories' => [
        'browse' => 'ixTheo\Controller\Factory::getBrowseController',
        'record' => 'ixTheo\Controller\Factory::getRecordController',
    ],
    'invokables' => [
        'alphabrowse' => 'ixTheo\Controller\AlphabrowseController',
        'BibleRangeSearch' => 'ixTheo\Controller\Search\BibleRangeSearchController',
        'feedback' => 'ixTheo\Controller\FeedbackController',
        'KeywordChainSearch' => 'ixTheo\Controller\Search\KeywordChainSearchController',
        'MyResearch' => 'ixTheo\Controller\MyResearchController',
        'Pipeline' => 'ixTheo\Controller\Pipeline',
        'search' => 'ixTheo\Controller\SearchController',
        'StaticPage' => 'ixTheo\Controller\StaticPageController',
    ],
],
```



Beispiel: Neuer Controller (AbstractFactory)



- Wenn z.B. eine Factory abhängig vom Parameter unterschiedliche Klassen in ähnlichen Namespaces auflösen soll.
- Beispiel: Vufind\Search\<requestedName>\Params
- Die Factory erhält als Parameter zusätzlich den Dienstnamen und sucht die passende Klasse (z.B. für die „Subscriptions“-Suche)
- Falls hier weitere PluginManager verwendet werden, kann es sehr unübersichtlich werden!

```
public function createServiceWithName(ServiceLocatorInterface $serviceLocator,
    $name, $requestedName
) {
    $options = $serviceLocator->getServiceLocator()
        ->get('VuFind\SearchOptionsPluginManager')->get($requestedName);
    $class = $this->getClassName($name, $requestedName);
    // Clone the options instance in case caller modifies it:
    return new $class(
        clone($options),
        $serviceLocator->getServiceLocator()->get('VuFind\Config')
    );
}

'search_params' => [
    'abstract_factories' => ['ixTheo\Search\Params\PluginFactory'],
],
```



... unübersichtlich, weil:



- AbstractFactory
- calls PluginFactory
- extends \VuFind\ServiceManager\AbstractPluginFactory
- uses VuFind\SearchOptionsPluginManager
- extends AbstractPluginManager
- uses VuFind\ServiceManager\Initializer
- extends Base
- uses module.config.php
- which redirects to another factory
- which finally gets the wanted object

=> sehr schwierig zu durchschauen!

Beispiel Search Options, Params und Results!



Übersicht Zend Framework



- Vorteile
 - Klare Trennung Original-Code / Angepasster Source Code
 - Neue Anpassungen am Original können relativ einfach eingespielt werden
 - Ziel/Strategie: Merge-Konflikte vermeiden!
 - (Bei pulls von vufind-org: release-changelog durchgehen)
- Nachteile
 - Erfordert Verständnis von (teils sehr komplexen) Softwarestrukturen
 - Was wenn nur wenig Zeit zur Verfügung steht, und das Verständnis fehlt?
 - Was wenn Änderungen eigentlich ins Original sollen, aber man mit einer alten Version arbeitet?



Beispiel: KrimDok



All Fields [Find](#) [Advanced](#)

Search Options

- [Search History](#)
- [Advanced Search](#)
- [Other catalogs](#)

Find More

- [Browse the Catalog](#)
- [Browse Alphabetically](#)
- [New Items](#)

Need Help?

- [Search Tips](#)
- [FAQs](#)

- Erste VuFind-Ableitung (ca. seit Oktober 2014)
- Diverse Zusatzfunktionen wurden eingebaut
 - z.B. Suche für Neuerwerbungen, seit 30/60 Tagen
- Der VuFind Source Code wurde direkt editiert



IxTheo
Home
About
IxTheo A-Z
Login
Language

Index Theologicus

International Bibliography of Theology and Religious Studies

All Fields
Find

Advanced Search
Browse
Need Help?

Welcome!

The new IxTheo is a comprehensive bibliography for theology and religious studies. It is now possible to search not only for articles, but also for monographs, databases and relevant Internet links.

[Show more](#)

Patron-Driven-Acquisition (PDA) new

We are now offering a Patron Driven Acquisition (PDA) system which enables us to respond to special demands for theological literature in Germany.

[Show more](#)

Discover IxTheo

Search

You can start a search on all the titles catalogued in the IxTheo by clicking on the "Find" button. You will then find over 1.6 Million titles.

[Show more](#)

Browsing

The new IxTheo offers three different ways to browse. You can access it by clicking on "Browse" in the area below the search box.

[Show more](#)

Personalized searches

Many useful functions are at your disposal when you create a personal account. Once you've registered, you may log into your account by clicking on the link called "Login" on the top right corner.

By clicking on "Your account", you come into your personalized area. You may then use the following functions:

[Show more](#)

Search for Bible passages

The search for Bible passages has been newly designed in order to cater for the needs of academic users. In order to activate it, you must select the search category "Bible passages" on the right side of the search field. The list of entries must be sorted according to their relevance.

[Show more](#)

Search Options

- Search History
- Advanced Search

Find More

- Browse by Subject
- Browse the Catalog
- Browse Alphabetically
- New Items

Need Help?

- Search Tips



RelBib

[Home](#)
[About](#)
[RelBib A-Z](#)
[Login](#)
[Language](#)

Religious Studies Bibliography

Religionswissenschaftliche Bibliographie – Bibliographie des sciences de la religion

All Fields

Find

[Advanced Search](#)
[Browse](#)
[Need Help?](#)
[List of Journals](#)

Welcome

The new "Religious Studies Bibliography" RelBib is now online! Our comprehensive bibliographical database contains journal articles of Religious Studies and related subjects. Apart from journals and articles, it is possible to search for monographs, databases and other media genres relevant to the field of Religious Studies.

Further journals are gradually being integrated into our indexing procedure for articles. The alphabetical [list of journals](#) gives the latest information on which periodicals are being processed.

Show more

Discover RelBib

Search

You can choose between two different search modes in RelBib: "Simple Search" and "Advanced Search".

Show more

More Features

Without a personal login you can see / use the following features in RelBib: Patron-Driven-Acquisition (PDA), availability check, RSS feed, citation, exporting and printing search results as well as the feedback button.

Show more

Browsing

Next to simple and advanced search RelBib offers three different ways to browse the database: by topic ("Find Topic"), according to categories ("Browse by Category") and alphabetically ("Browse Alphabetically"). You can access this function by clicking on "Browse" in the area below the search field.

Show more

My RelBib

Many useful functions are at your disposal when you create a personal account.

In order to register or log into your account, click on the link called "Login" on the top right corner. By clicking on "Your account", you come into your personalized area.

Show more

Search Options

- Search History
- Advanced Search

Find More

- Browse by Subject
- Browse by Category
- Browse Alphabetically

Need Help?

- Search Operators
- Browse by Subject
- Browse by Category
- Browse Alphabetically



Beispiel: IxTheo/RelBib



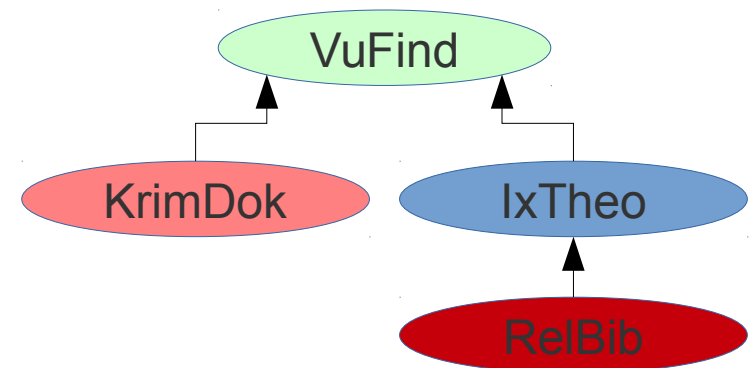
- Kam erst später (ca. März 2015)
- Auch hier: diverse Zusatzfunktionen
 - Alerting (E-Mail-Benachrichtigung für neue Einträge)
 - Verfügbarkeitsabfragen (HBZ, TAD, JOP, ...)
 - PDA (Benutzergesteuerte Erwerbung)
 - Mehrsprachigkeit (z.B. für Keywords)
 - Bibelstellensuche
 - Spezialklassifikationen
 - Aber auch: Anpassungen von fehlerhaften deutschen Hilfstexten im VuFind-Standard
- Wurde größtenteils als Modul/Theme abgeleitet (nicht alles!)
 - IxTheo/Relbib sind sehr ähnlich => gleiches Modul
 - Verschiedene Themes
 - RelBib abgeleitet von ixTheo



Ausgangssituation



- Auswirkungen dieser Struktur
 - Keine Wiederverwertung (Mehrfache Pflege von neuem Code)
 - Inkonsistenz (Bugs in einem System aber nicht im anderen trotz selber Features)
 - Mehr Aufwand bei neuen VuFind-Versionen
- Absehbar: Weitere VuFind-Ableitungen!





Lösungsansatz

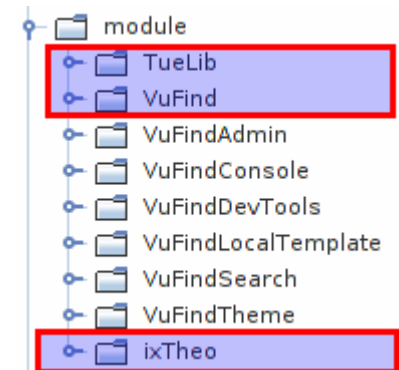


- Code modularisieren
 - VuFind-Code nicht verändern, alle bestehenden VuFind-Veränderungen auslagern
 - => Diff des kompletten Baumes gegen den VuFind-Originalcode!
(diff -rq <vufind-3.1.3-orig> <vufind-3.1.3-custom>)
 - Zwischenschichten einführen, die von allen Ableitungen gleichermaßen genutzt werden können
 - => TueLib-Modul, (ub)TueTheme



Lösungsansatz (Git)

- Fernziel: Gemeinsames Repository!
- Übergangslösung: Submodule
 - Repositories bleiben zunächst getrennt, + je ein separates Repository für Modul + Theme
 - Vorteilhaft, da Refactoring gemacht werden kann bis die Basis für alle Ableitungen gleich ist
 - Nicht auf Dauer sinnvoll, da hoher Pflegeaufwand bei neuen Versionen





Lösungsansatz (Git) => Submodule



- Problem: eigentlich gedacht, um Drittbibliotheken einzubinden!
- Separates Repository pro submodul
 - Entwickeln direkt im separaten Submodul (+commit/push)
- Repository im VuFind-Repository einrichten
 - submodul registrieren (.gitmodules)
 - submodul initialisieren (git submodule init)
- submodul auschecken (git submodule update --remote)
 - Hier wird nur die aktuelle Version verlinkt

```
root@MB055:/usr/local/vufind/module/TueLib# git status
HEAD losgelöst bei efb56ce
nichts zu committen, Arbeitsverzeichnis unverändert
```




Lösungsansatz (Git) => Submodule



- Bei Änderungen am Submodul muss auch das VuFind-repository aktualisiert werden:
 - Git submodule update –remote
 - Änderung am VuFind-Repository committen, damit git die neue HEAD Version des submodule kennt
 - Auf dem Produktivserver reicht git pull nicht mehr aus, es muss danach immer ein „git submodule update“ ausgeführt werden (!!! OHNE „–remote“!!!)



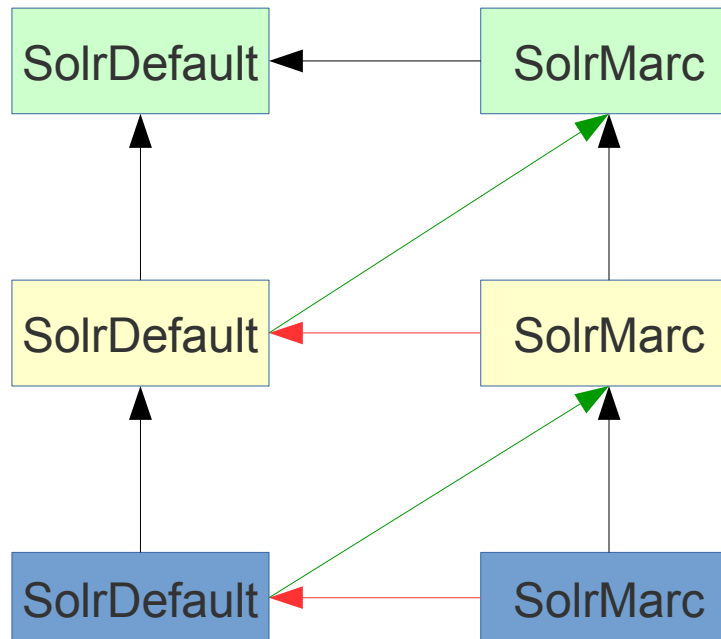
Beispiel: RecordDriver

Workaround für Mehrfachvererbung

VuFind

TueLib

Custom
(IxTheo, KrimDok)



Hinweis: Diese Struktur könnte sich in VuFind 5 ändern, siehe JIRA (VUFIND-1240)

Beispiel: RecordDriver + Templates

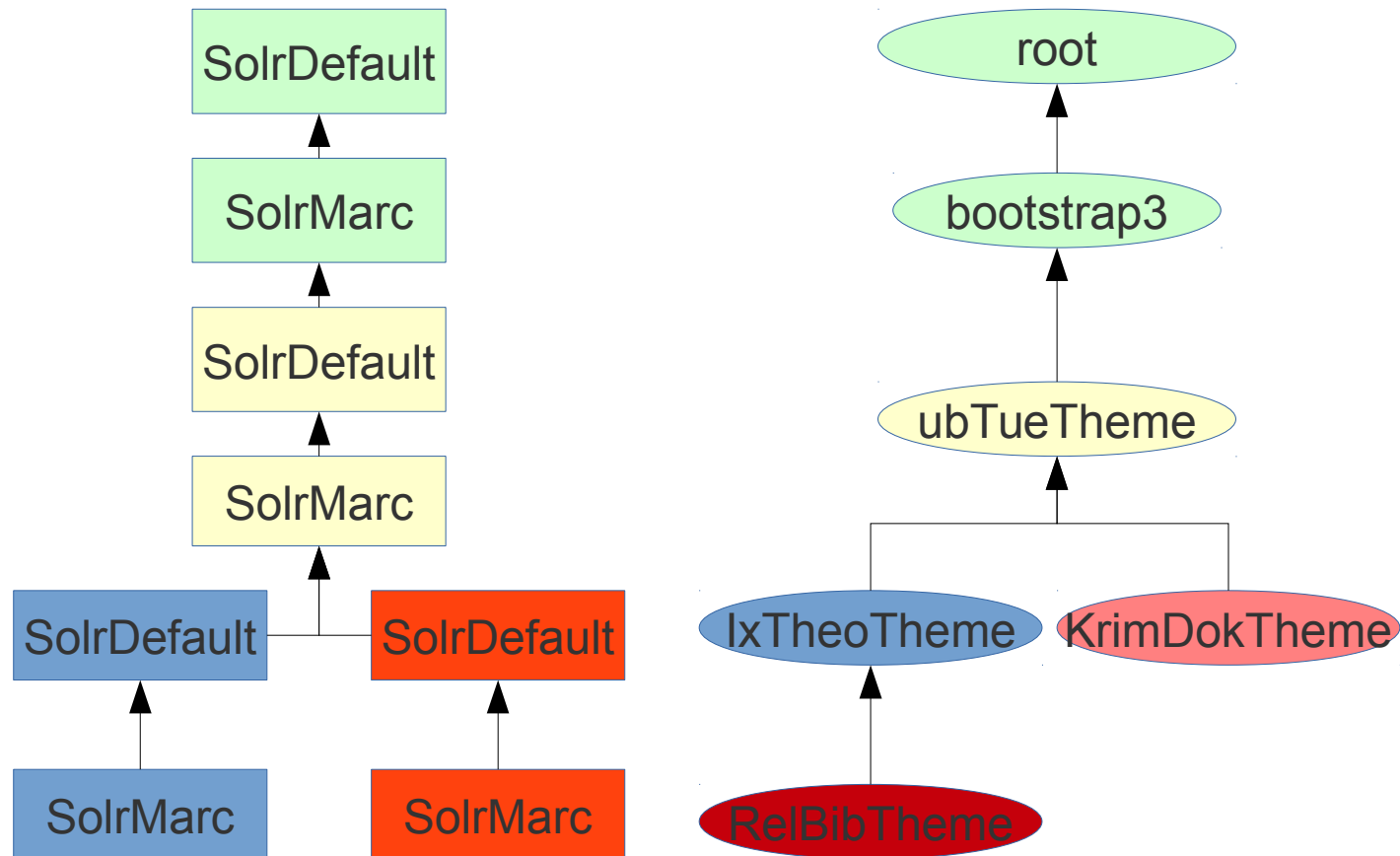


VuFind

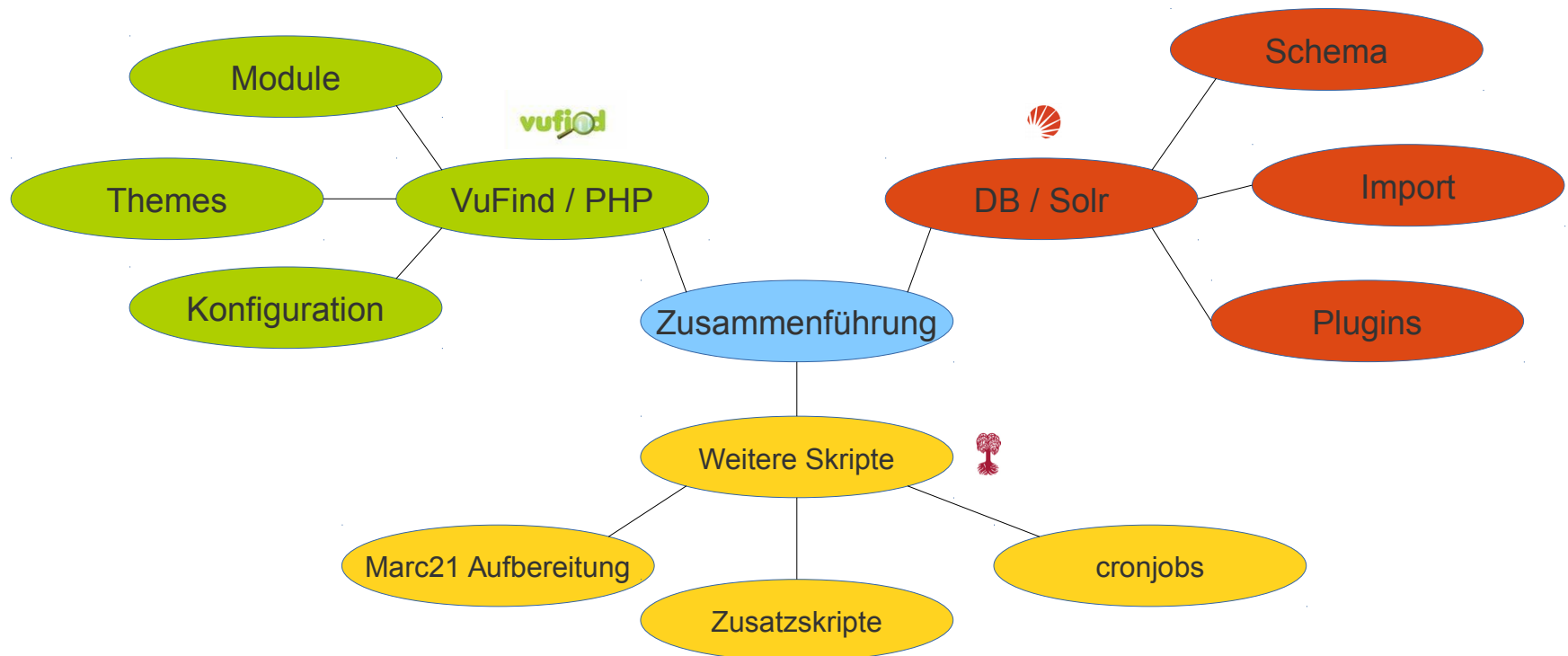
TueLib

IxTheo

KrimDok



Repositories zusammenführen: „TuFind“



+ komplett neues Git Repository!



Verzeichnisstruktur: vufind\local

- tufind

- instances

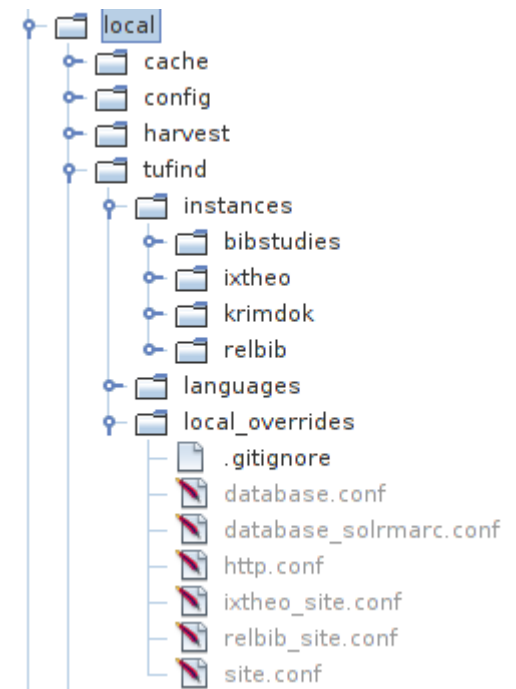
- Home-Verzeichnis für Apache Config (VUFIND_LOCAL_DIR)
 - ixtheo/relbib (+bibstudies) auf selber Maschine, da gleiche DB

- languages

- Einheitliche Displaytexte für alle Ableitungen
 - aktiv über symlink aus allen „instances“ z.B. von tufind/instances/ixtheo/languages

- local_overrides

- Nicht in git verwaltet
 - z.B. Datenbank-Zugangsdaten für lokale Maschine
 - Aktiv über symlink aus allen „instances“, z.B. von tufind/instances/ixtheo/config/vufind/local_overrides





Installation per Skript

- `tufind.sh <ixtheo/krimdok>`
 - Dateien
 - `solrconfig.xml`
 - `schema.xml`
 - `marc_local.properties`
 - `git update-index --assume-unchanged`
 - z.B. „git status“ => zeigt keine `marc_local.properties` Änderungen an
 - Bei neuem `vufind-org/master` werden trotzdem Merge-Konflikte ausgelöst
 - Außerhalb von Solr:
 - Weitere Schritte, abhängig vom Installationstyp
 - z.B. cronjobs einrichten
 - z.B. `index-alphabetic-browse_ixtheo.sh` (symlink)



Solr-Konfiguration

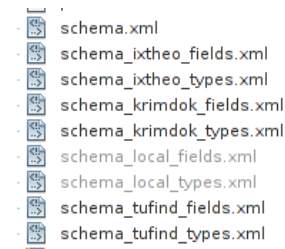


- schema.xml aufteilen
 - xi:include => schema local.xml

```
<!-- ubtue custom field types and fields -->
<xi:include href="schema_local_types.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="schema_local_fields.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
```

- Sinnvoll für den VuFind-Standard?
- schema_local wird generiert

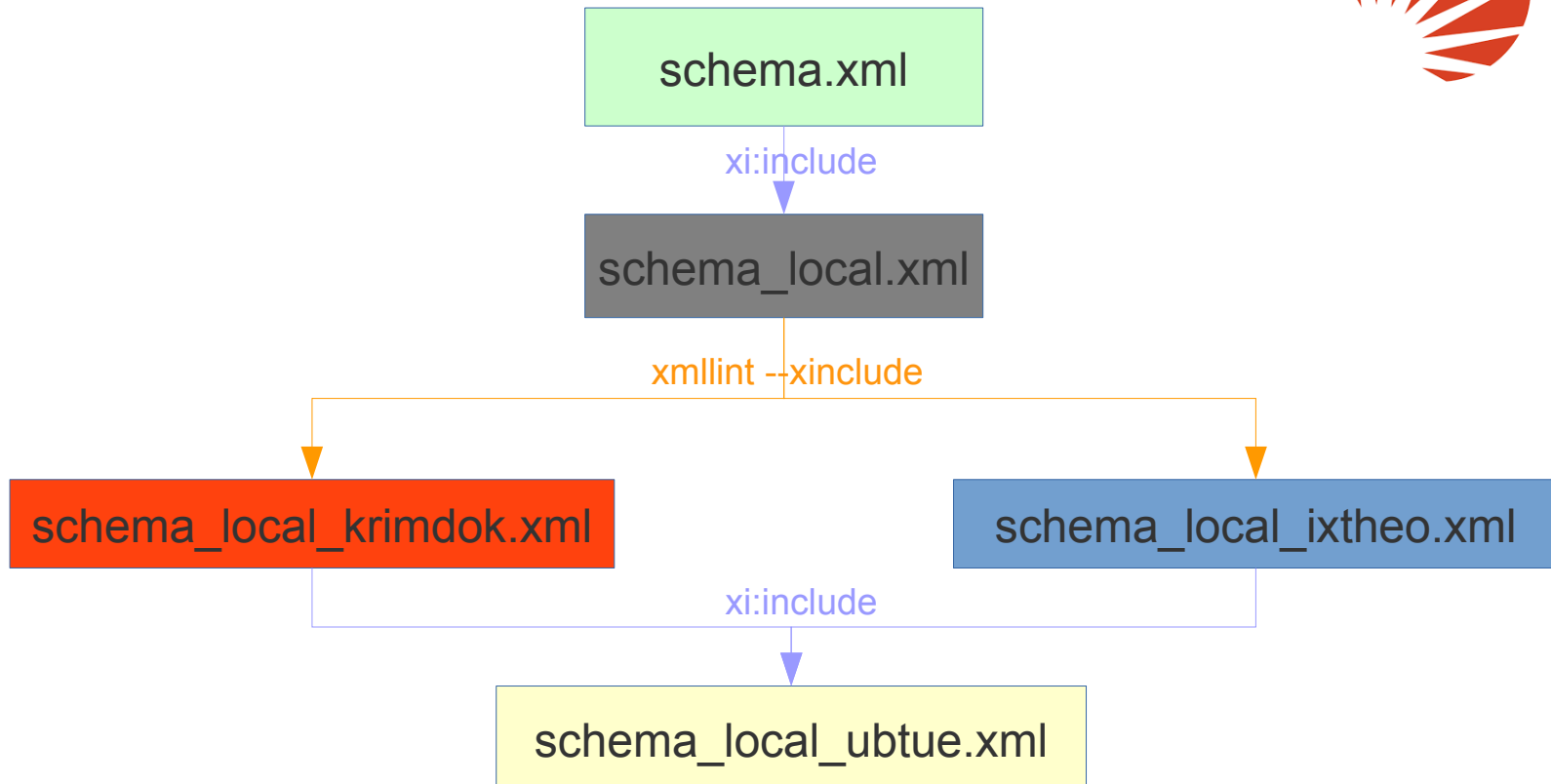
- schema_local_fields.xml
 - schema_tufind_types.xml
 - schema_ixtheo_types.xml
 - schema_krimdok_types.xml



- Generieren notwendig
 - Einhalten der Solr-Struktur beim Generieren über mehrere Ebenen nur mit xpointer möglich
 - xpointer in solr nicht verfügbar (getestet mit solr 6.6.1)



Solr-Konfiguration (Hierarchie)





Solr-Konfiguration (Beispiel)

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <types>
3   <xi:include href="schema_tufind_types.xml" xmlns:xi="http://www.w3.org/2001/XInclude" xpointer="xpointer(//fieldType)">
4     <xi:fallback>
5       <!-- no error if no shared fieldType exists -->
6     </xi:fallback>
7   </xi:include>
8
9   <fieldType name="textFacetRaw" class="solr.TextField" sortMissingLast="true" omitNorms="true">
10     <analyzer>
11       <tokenizer class="solr.KeywordTokenizerFactory"/>
12       <!-- this is just like textFacet but without punctuation stripping. -->
13     </analyzer>
14   </fieldType>
15   <!-- Search signatures; ignore cases, remove whitespace -->
16   <fieldType name="signaturestring" class="solr.TextField" sortMissingLast="true">
17     <analyzer>
18       <tokenizer class="solr.KeywordTokenizerFactory"/>
19       <filter class="solr.PatternReplaceFilterFactory" pattern="^[\\w]+" replacement="" replace="all"/>
20       <filter class="solr.LowerCaseFilterFactory" />
21     </analyzer>
22   </fieldType>
23
24 </types>
25
26 <?xml version="1.0" encoding="UTF-8"?>
27 <fields>
28   <!-- COMMON -->
29   <xi:include href="schema_tufind_fields.xml" xmlns:xi="http://www.w3.org/2001/XInclude" parse="xml" xpointer="xpointer(//field)" />
30
31   <!-- CUSTOM - only KrimDok -->
32   <field name="abr" type="string" indexed="true" stored="true" multiValued="true"/>
33   <field name="article_page_range" type="string" indexed="true" stored="true"/>
34   <field name="available_in_tubingen" type="boolean" indexed="false" stored="true" multiValued="false"/>
35   <field name="content" type="string" indexed="true" stored="true" multiValued="true"/>
36   <field name="fid_systematik" type="textFacetRaw" indexed="false" stored="true" multiValued="true"/>
37 </fields>

```



Solrmarc

- marc.properties / marc_local.properties
- marc_local.properties aufteilen
 - marc_tufind.properties
 - marc_custom.properties
- Jeweils gegliedert in Abschnitte
 - Überschriebene Felder
 - Zusätzlich definierte Felder

-  marc_ixtheo.properties
-  marc_krimdok.properties
-  marc_local.properties
-  marc_tufind.properties

```
#####
# marc_tufind.properties -- local overrides to standard VuFind MARC mappings. #
# All settings in this file override equivalent #
# settings in marc.properties. #
# This file will be merged with other files like #
# marc_<ixtheo|krimdok>.properties #
#####

# VuFind overrides - All TuFind instances
format = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getFormatsWithoutElectronic, format_map.properties
publishDate = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getDatesBasedOnRecordType
publishDateSort = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getPublicationSortDate
title = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getMainTitle
title_auth = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getMainTitle
title_short = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getTitleSubfield("a")
title_sub = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getTitleSubfield("b")

# additional common fields - All TuFind instances (see schema_tufind.fields.xml)
container_ids_and_titles = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getContainerIdsWithTitles
is_superior_work = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), isSuperiorWork
issue = 936a
journal_issue = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getJournalIssue
pages = 936h
superior_ppn = 800w:810w:830w:773w:776w, (pattern_map.superior_ppn)
pattern_map.superior_ppn.pattern_0 = .DE-576.(.*)=>$1
urls_and_material_types = custom(de.uniduebingen.ub.ubtools.solrmarcMixin.TuelibMixin), getUrlsAndMaterialTypes
volume = 936d
year = 936j
```



Solr-Konfiguration – offene Punkte

- schema.xml
 - Solr-Defaulttypen ändern
 - Stopwords entfernen („to be or not to be“)
 - Sortierung anpassen, z.B. „German“ => „English“ im Text
 - Solr Defaultfelder andere Typen zuweisen
 - Je nach Feld wären zu viele Ableitungen erforderlich!
z.B. beim Typ „text“ => 25 Felder doppelt pflegen ist nicht praktikabel!
- Solrconfig wird nicht generiert
 - (krimdok verwendet default, ixtheo verwendet komplett andere Version mit Plugins, zusätzlichen alphabrowse indices usw)

```
<!-- CUSTOM queryParsers -->
<queryParser class="de.uni_tuebingen.ub.ixTheo.bibleRangeSearch.BibleRangeParserPlugin" name="bibleRangeParser" />
<queryParser class="de.uni_tuebingen.ub.ixTheo.keywordChainSearch.KeywordChainParserPlugin" name="keywordChainParser" />
<queryParser class="de.uni_tuebingen.ub.ixTheo.multiLanguageQuery.MultiLanguageQueryParserPlugin" name="multiLanguageQueryParser" />
<transformer class="de.uni_tuebingen.ub.ixTheo.MultiLanguageDocTransformerFactory" name="multiLanguageDocTransformer">
  <bool name="enabled">true</bool>
  <str name="fields">topic_standardized,topic_non_standardized</str>
  <str name="lang_extensions">_de,_en,_fr,_it,_es,_hant,_hans</str>
</transformer>
```



Tübinger Ableitungen: Was kommt noch?



- TuFind fertigstellen
- Mehr Features verallgemeinern (IxTheo => TuFind)
 - Per Konfiguration aktivierbar/deaktivierbar!



Adressen

- Webseiten
 - <https://ixtheo.de>
 - <https://relbib.de>
 - <https://krimdok.uni-tuebingen.de>
- Git
 - <https://github.com/ubtue/tufind>



...Fragen?

- Zur Diskussion:
 - Grundsätzlich: andere Ansätze?
 - Mehrere DB-Instanzen pro Maschine?
 - Bessere Übergangslösung statt submodule?
 - Bessere Lösungen für Mehrfachvererbung?
 - Besserer Umgang mit Solr-Feldern?
 - Weitere praktische Probleme?
 - (z.B. Solrmrc-Mixins)



Vielen Dank!

Kontakt:

Universitätsbibliothek Tübingen

Wilhelmstraße 32, 72074 Tübingen

Telefon: +49 7071 29-72820

mario.trojan@uni-tuebingen.de