Invenio @ HGF
JuSER

June 17, 2013 | Alexander Wagner |
Overview

- History
- Basic Usage
- Installation
- Backend
VDB and JUWEL

Publications database **VDB**:  
- Centralized database of the scientific output  
- > 62,000 entries from \( \approx 13 \) years \( (\approx +4800/a) \)  
- **Obligatory input**\(^1\) (editors at each institute, crosschecked by ZB)

\(^1\)cf. Publication Guidelines
VDB and JUWEL

Publications database **VDB:**
- Centralized database of the scientific output
- > 62,000 entries from \( \approx 13 \) years \( (\approx +4800/\text{a}) \)
- **Obligatory input\(^1\)** (editors at each institute, crosschecked by ZB)
- Basis for
  - Scientific Report
  - Evaluations
  - Publication lists (e.g., WWW)

\(^1\)cf. Publication Guidelines
Forschungszentrum Jülich - Publikationen - Iceweasel

Institut für Neurowissenschaften und Medizin
Strukturelle und funktionelle Organisation des Gehirns (INM-1)

AKTUELLES    FORSCHUNG    LEISTUNGEN    KARRIERE    ÜBER UNS

INM-1    Forschung    Publikationen

FORSCHUNG

Architekttonik und Hirnfunktion
Multimodale Bildverarbeitung
Modellierung kortikaler Systeme
Genomic Imaging

Publikationen

SERVICE

Ansprechpartner
Mitarbeiter
Publikationen
Anfahrt
Downloads

Zeitschriftenbeiträge 2012

Amunts, K.; Zilles, K.

Architecture and organizational principles of Brocas region

Bis, J. C.; et, a.

Common variants at 12q14 and 12q24 are associated with hippocampal volume
Nature genetics 44, 545 - 551 (2012) [10.1038/ng.2237]

Boutros, N. N.; Gini, K.; Eickhoff, S. B.; Urbach, H.; Pfieger, M. E.

Mapping repetition suppression of the P50 evoked response to the human cerebral cortex
Clinical neurophysiology 00, 00 (2012) [10.1016/cnph.2012.13.007]

Bzdok, D.; Laird, A.; Zilles, K.; Fox, P.; Eickhoff, S. B.

An investigation of the structural, connectional, and functional sub-specialization in the human amygdala
Human brain mapping (2012) [10.1002/hbm.22138]


Parsing the neural correlates of moral cognition: ALE meta-analysis on morality, theory of mind, and empathy
Brain structure & function 217, 763 - 796 (2012) [10.1007/s00429-012-0360-y]

PUBLIKATIONEN INM-1

Publikationen 2011
Publikationen 2010
Publikationen 2009
Publikationen 2008

→ Mehr

In Print

Download
VDB and JUWEL

Publications database VDB:
- Centralized database of the scientific output
- > 62,000 entries from ≈ 13 years (≈ +4800/a)
- Obligatory input² (editors at each institute, crosschecked by ZB)
- Basis for
  - Scientific Report
  - Evaluations
  - Publication lists (e. g. WWW)

JUWEL:
- Berlin Declaration for Open Access
- Institutional repository (≈ 4500 full text files, ≈ +560/a)

²cf. Publication Guidelines
Longitudinal deformation-based morphometry reveals spatio-temporal dynamics of brain volume changes in patients with corticobasal syndrome

Pieperhoff, Peter; Ferrea, Stefano; Krause, Holger; Groiss, Stefan Jun; Elben, Saskia; Woitecki, Lars; Zilles, Karl; Amunts, Katrin; Schnitzler, Alfons; Süddeier, M (Corresponding author)

2012
PLoS Lawrence, Kan.

Published in: PLoS one 7 (7) e41873 - [10.1371/journal.pone.0041873]
Due to reporting requirements: 3 step document flow
Due to reporting requirements: 3 step document flow

1. **Der Wissenschaftler:**
   - Einstieg mit gültigem Login (E-Mail)
   - Eingabe der Publikation (Submit)

2. **VDB-Relevant?**
   - yes / no

3. **Institutscollection (Restricted)**
   - Zugriff nur Institutsangehörige

   - submitten
   - editieren
   - für Institut freigeben

4. **Öffentlich sichtbar, keine Statistik. In Print.**

5. **Bearbeitung und Freigabe durch die Bibliothek**

6. **Öffentlich sichtbar, Statistik = bisherige VDB**

**Thanks to H. Lexis**
JuSER – Main features

- **Import interfaces** (improve data quality, ease up input)
JuSER – Main features

- Import interfaces  (improve data quality, ease up input)
- Exports to BibTeX/EndNote  (Integrate with citations management)
JuSER – Main features

- Import interfaces  (improve data quality, ease up input)
- Exports to BibTeX/EndNote  (Integrate with citations management)
- Institute collections  (collect and share documents)
JuSER – Main features

- **Import interfaces** (improve data quality, ease up input)
- **Exports to BibTeX/EndNote** (Integrate with citations management)
- **Institute collections** (collect and share documents)
- **Add full texts**
JuSER – Main features

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)
- Institute collections (collect and share documents)
- Add full texts
- Normalize as much as possible (Key: Authorities)
JuSER – Main features

- **Import interfaces** (improve data quality, ease up input)
- **Exports to BibTeX/EndNote** (Integrate with citations management)
- **Institute collections** (collect and share documents)
- **Add full texts**
- **Normalize as much as possible** (Key: Authorities)
  - **Authors** (tell apart Meier and Meier)
  - **Institutes**
  - **Journals**
  - **Projects** (POF, EU, …)
  - …
JuSER – Main features

- Import interfaces  (improve data quality, ease up input)
- Exports to BibTeX/EndNote  (Integrate with citations management)
- Institute collections  (collect and share documents)
- Add full texts
- Normalize as much as possible  (Key: Authorities)
  - Authors  (tell apart Meier and Meier)
  - Institutes
  - Journals
  - Projects  (POF, EU, . . .)
  - . . .

... Users don’t need to care about technical details ...
Collections

- **Publicationsdatabase**: Publications from Jülich
Collections

- **Publicationsdatabase**: Publications from Jülich
- **Documents in Print:**
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
Collections

- **Publicationsdatabase**: Publications from Jülich
- **Documents in Print**:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
  - already visible on institutes web pages
Collections

- **Publicationsdatabase**: Publications from Jülich
- **Documents in Print**:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
  - already visible on institutes web pages
- **OpenAccess repository JUWEL**
Collections

- **Publicationsdatabase**: Publications from Jülich
- **Documents in Print**:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
  - already visible on institutes web pages
- **OpenAccess repository** JUWEL
- **Institute Collections**:
  - Institutes private workspace
  - e. g. drafts, collected literature, journal clubs ...
  - Access for members of the institute only
- **Authorities**
Institute Collections

JuSER can hold documents beyond own publications
Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management
- document exchange at the institutes
- centralized collection of papers
- easy export to formatting tools (BibTeX, EndNote)
- commenting (individual and in group)
Institute Collections

JuSER can hold documents beyond own publications
- webbased literature management
- document exchange at the institutes
- centralized collection of papers
- easy export to formatting tools (BibTeX, EndNote)
- commenting (individual and in group)

- Institute collections require proper login
- Non-VDB relevant items do not show up on the webpage
- ZB does not care about usage
Submit

1. Log in (LDAP based)
2. Select **Submit** from the main menu ([http://juser.fz-juelich.de/submit](http://juser.fz-juelich.de/submit))
3. Select document type
Submit

1. Log in (LDAP based)
2. Select **Submit** from the main menu ([http://juser.fz-juelich.de/submit](http://juser.fz-juelich.de/submit))
3. Select document type
4. Fill in the submit form
Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form
   - import if possible (doi, pmid, arXiv, inspire, own recs...)

Permissions & workflow: quite complex collection structure
Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form
   - import if possible (doi, pmid, arXiv, inspire, own recs…)
   - take care of the authors
   - Own publication? ⇒ VDB-Relevant = yes
Submit

1 Log in  (LDAP based)
2 Select **Submit** from the main menu  (http://juser.fz-juelich.de/submit)
3 Select document type
4 Fill in the submit form
   - **import** if possible  (doi, pmid, arXiv, inspire, own recs…)
   - take care of the authors
   - Own publication?  ⇒  VDB-Relevant  =  yes
5 Add full text
6 Submit it:
Submit

1. Log in (LDAP based)
2. Select **Submit** from the main menu ([http://juser.fz-juelich.de/submit](http://juser.fz-juelich.de/submit))
3. Select document type
4. Fill in the submit form
   - **import** if possible (doi, pmid, arXiv, inspire, own recs…)
   - take care of the authors
   - Own publication? ⇒ VDB-Relevant = yes
5. Add full text
6. Submit it:
   - “Finish & Release” (proceed to next level in workflow)
   - “Postpone” (generates TEMPENTRY)
Submit

1. Log in (LDAP based)
2. Select **Submit** from the main menu ([http://juser.fz-juelich.de/submit](http://juser.fz-juelich.de/submit))
3. Select document type
4. Fill in the submit form
   - *import* if possible (doi, pmid, arXiv, inspire, own recs…)
   - *take care of the authors*
   - Own publication? ⇒ VDB-Relevant = yes
5. Add full text
6. Submit it:
   - "Finish & Release" (proceed to next level in workflow)
   - "Postpone" (generates TEMPENTRY)

Permissions & workflow: quite complex collection structure
Field designations

Red: mandatory for a full bibliographic description
Black: might not all apply
Blue: save manual work!
Field designations

Red: mandatory for a full bibliographic description
Black: might not all apply
Blue: save manual work!

Import data

…is always available and allows to fetch data from external or internal sources.

PS: is the manual (Wiki-based, german and english)
Import HowTo

- **DOI**: insert doi or dx.doi.org-url (e. g. 10.1016/j.physletb.2006.11.038)
- **pubmed**: copy as displayed (e. g. PMID: 20923669)
- **arXiv.org**: copy as displayed (e. g. arxiv:hep-ph/0610431)
- **inspire**: use URL
- **own**: `recid: + record-Id or 037__a` (e. g. recid:FZJ-2013-00499)
- **ISBN**: use the ISBN-field for this import
Import HowTo

- DOI: insert doi or dx.doi.org-url (e.g. 10.1016/j.physletb.2006.11.038)
- PubMed: copy as displayed (e.g. PMID: 20923669)
- arXiv.org: copy as displayed (e.g. arxiv:hep-ph/0610431)
- inspire: use URL
- own: recid: + record-Id or 037__a (e.g. recid:FZJ-2013-00499)
- ISBN: use the ISBN-field for this import
Import HowTo

- **DOI**: insert doi or dx.doi.org-url (e.g. 10.1016/j.physletb.2006.11.038)
- **pubmed**: copy as displayed (e.g. PMID: 20923669)
- **arXiv.org**: copy as displayed (e.g. arxiv:hep-ph/0610431)
- **inspire**: use URL
- **own**: recid: + record-Id or 037__a (e.g. recid:FZJ-2013-00499)
- **ISBN**: use the ISBN-field for this import

![Import data](image)

**Import data:**

Duplicate entries

At import via doi, pmid, arXiv... JuSER

- can identify potential duplicates
- refuses the import
- shows links to the potential dupes
E. g. DOI Import

Most red fields are filled in already!

Alexander Wagner
Authors

Upon import

authors are a mere guesses by the system. Check them!
Authors

Upon import

authors are a mere guesses by the system. Check them!

- : confirm the guess
- : correct a wrong guess
- : remove an entry
- : clear the whole list
Authors

Upon import

authors are a mere guesses by the system. Check them!

- ✓: confirm the guess
- 🧐: correct a wrong guess
- ✗: remove an entry
- 🥳: clear the whole list

- Associate only authors that show an email address/institute
- If unsure, leave them red
Author input and association
Author input and association

Note: ZB covers publication costs if the corresponding author is from Jülich.
Author input and association

Note: ZB covers publication costs if the corresponding author is from Jülich.
Author input and association

Note: ZB covers publication costs if the corresponding author is from Jülich.
Author input and association

Note: ZB covers publication costs if the corresponding author is from Jülich.
Author input and association

Note: ZB covers publication costs if the corresponding author is from Jülich.

Alexander Wagner
Own publications

Author association

allows easy extraction of individual and exact publication lists.
Own publications

Author association allows easy extraction of individual and exact publication lists.
Claim work (required only once!)

1. Log in
2. Navigate to: Authorities / People (http://juser.fz-juelich.de/collection/People)
3. Search for own name (e.g. ‘Hofmann, D’)
4. Note the IDs in question (e.g. P:(DE-Juel1)VDB63458 and P:(DE-Juel1)129471)
5. Open details and check the records found by the link
   All known publications (below the list of recent publications)
6. notify juser@fz-juelich.de
   (e.g. if the above two people are the same, we need both ids to join them)
Claim work (required only once!)

1. Log in
2. Navigate to: Authorities / People (http://juser.fz-juelich.de/collection/People)
3. Search for own name (e.g. 'Hofmann, D')
4. Note the IDs in question (e.g. P:(DE-Juel1)VDB63458 and P:(DE-Juel1)129471)
5. Open details and check the records found by the link
   All known publications (below the list of recent publications)
6. notify juser@fz-juelich.de
   (e.g. if the above two people are the same, we need both ids to join them)

To search own publications use aid: and own ID in quotes ("")

   e.g. aid:‘P:(DE-Juel1)133794’

Want only first authorships? Use fai instead of aid.
Invenio @ HGF
Part III: Installation

June 17, 2013 | Alexander Wagner
Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.
Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- git: cds-invenio (centrally kept at DESY)
Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- **git**: `cds-invenio` (centrally kept at DESY)
  - CERN git at “our” commit
  - used for base system
  - applied by configure/make/make install
Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- **git: cds-invenio** (centrally kept at DESY)
  - CERN git at “our” commit
  - used for base system
  - applied by `configure/make/make install`

- **git: hgf-invenio** (centrally kept at DESY)
Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- **git: cds-invenio** (centrally kept at DESY)
  - CERN git at “our” commit
  - used for base system
  - applied by configure/make/make install

- **git: hgf-invenio** (centrally kept at DESY)
  - overlay for our instances (our additions)
  - our patches (no replacements!)
  - directory structure like /opt/invenio
  - setup routines for roll-out (e.g. collections, rolles, doctypes...)
  - instances configs
InstallInvenio

sets up the **whole instance** at each partner in the proper layout including all **global** (hgf) and **local** configs
InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
InstallInvenio

sets up the **whole instance** at each partner in the proper layout including all **global** (hgf) and **local** configs

- assumes all deps are met and database exists
- sets up **from scratch** or to a **given point**
InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or to a given point
- compiles and installs \texttt{INVENIO-src}
InstalInvenio

InstalInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or to a given point
- compiles and installs INVENIO-src
- applies hgf patches and configs
InstallInvenio

sets up the **whole instance** at each partner in the proper layout including all **global** (hgf) and **local** configs

- assumes all deps are met and database exists
- sets up from scratch or to a given point
- compiles and installs \texttt{INVENIO-src}
- applies hgf patches and configs
- applies global and local customizations
InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or to a given point
- compiles and installs \texttt{INVENIO}\textregistered-src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or to a given point
- compiles and installs `INVENIO-src`
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
- no need to mouse click in the GUI
InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or to a given point
- compiles and installs \texttt{INVENIO}-src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
- no need to mouse click in the GUI
- enables desaster recovery
InstallInvenio

sets up the **whole instance** at each partner in the proper layout including all **global** (hgf) and **local** configs

- assumes all deps are met and database exists
- sets up **from scratch** or to a **given point**
- compiles and installs **Invenio**-src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
- no need to mouse click in the GUI
- enables desaster recovery
- allows setup of an **identical** test environment
Invenio @ HGF
Part IV: Backend

June 17, 2013  |  Alexander Wagner
HGF “specifics”

Broad areas of research

some *INVENIO*-HEP-specifics just don’t work
HGF “specifics”

Broad areas of research

some INVENIO®-HEP-specifics just don’t work

- Journal handling based on IDs, not names (hook up ZDB)
HGF “specifics”

Broad areas of research

- Journal handling based on IDs, not names (hook up ZDB)
- allow for ingest of foreign data (e.g. ebook packages)

some INVENIO-HEP-specifics just don’t work
HGF “specifics”

Broad areas of research

some **INVENIO**-HEP-specifics just don’t work

- Journal handling based on IDs, not names (hook up ZDB)
- allow for ingest of foreign data (e.g. ebook packages)
  - important in context of a libraries catalogue
  - stick to MARC definitions (Library of Congress)
  - avoid specialities (exceptions: 9xx)
  - implies adoption even of default defs (e.g. internal Marc, BibTeX)
HGF “specifics”

Broad areas of research

- some INVENIO-HEP-specifics just don’t work

- Journal handling based on IDs, not names (hook up ZDB)
- allow for ingest of foreign data (e. g. ebook packages)
  - important in context of a libraries catalogue
  - stick to MARC definitions (Library of Congress)
  - avoid specialities (exceptions: 9xx)
  - implies adoption even of default defs (e. g. internal Marc, BibTeX)
- Normalize all you can get
HGF “specifics”

- Journal handling based on IDs, not names (hook up ZDB)
- allow for ingest of foreign data (e. g. ebook packages)
  - important in context of a libraries catalogue
  - stick to MARC definitions (Library of Congress)
  - avoid specialities (exceptions: 9xx)
  - implies adoption even of default defs (e. g. internal Marc, BibTeX)
- Normalize all you can get
  - to be handled by uses & librarians, not programmers
  - vitally important: Authorities

some INVENIO-HEP-specifics just don’t work
HGF-Modules

- **Importer:** (crossref, pubmed, GVK, arXiv.org, inspire...)

  - *hook up with web services*
  - *reuse old code basically in perl (to be rewritten to Python as time allows)*
  - *add new stuff in Python*
  - *HGFImport.py (call externals but allow for permission checking)*
  - *HandleNames.py (author guessing)*
  - *JSGetAllChildren.py (recurse to the end of tree like MARC structures)*
  - *PubExporter.py (handle specifics needed for web export)*

  - Other functions
    - *Always try to use*
    - *functions like alerts, baskets, ...*

  - *Every unwritten line of code is a good line of code!*

Alexander Wagner 29
HGF-Modules

- **Importer:** (crossref, PubMed, GVK, arXiv.org, inspire...)
  - hook up with web services
  - reuse old code basically in perl *(to be rewritten to Python as time allows)*
  - add new stuff in Python
HGF-Modules

- **Importer:** (crossref, pubmed, GVK, arXiv.org, inspire...)
  - hook up with web services
  - reuse old code basically in perl (to be rewritten to Python as time allows)
  - add new stuff in Python

- **HGFImport.py** (call externals but allow for permission checking)
HGF-Modules

- **Importer**: (crossref, pubmed, GVK, arXiv.org, inspire...)
  - hook up with web services
  - reuse old code basically in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- **HGFImport.py** (call externals but allow for permission checking)
- **HandleNames.py** (author guessing)
HGF-Modules

- **Importer:** (crossref, PubMed, GVK, arXiv.org, inspire...)
  - hook up with web services
  - reuse old code basically in perl (to be rewritten to Python as time allows)
  - add new stuff in Python

- **HGFImport.py** (call externals but allow for permission checking)
- **HandleNames.py** (author guessing)
- **JSGetAllChildren.py** (recurse to the end of tree like MARC structures)
HGF-Modules

- **Importer:** (crossref, pubmed, GVK, arXiv.org, inspire...)
  - hook up with web services
  - reuse old code basically in perl (to be rewritten to Python as time allows)
  - add new stuff in Python

- **HGFImport.py** (call externals but allow for permission checking)
- **HandleNames.py** (author guessing)
- **JSGGetAllChildren.py** (recurse to the end of tree like MARC structures)
- **PubExporter.py** (handle specifics needed for web export)
HGF-Modules

- **Importer**: (crossref, pubmed, GVK, arXiv.org, inspire...)
  - hook up with web services
  - reuse old code basically in perl (to be rewritten to Python as time allows)
  - add new stuff in Python

- **HGFImport.py** (call externals but allow for permission checking)
- **HandleNames.py** (author guessing)
- **JS GetAllChildren.py** (recurse to the end of tree like MARC structures)
- **PubExporter.py** (handle specifics needed for web export)

**Other functions**

Always try to use INVENIO functions like alerts, baskets, ...
HGF-Modules

- **Importer**: (crossref, PubMed, GVK, arXiv.org, inspire…)
  - hook up with web services
  - reuse old code basically in perl (to be rewritten to Python as time allows)
  - add new stuff in Python

- **HGFImport.py**: (call externals but allow for permission checking)
- **HandleNames.py**: (author guessing)
- **JSGetAllChildren.py**: (recurse to the end of tree like MARC structures)
- **PubExporter.py**: (handle specifics needed for web export)

Other functions

Always try to use **INVENIO** functions like alerts, baskets, …

Every unwritten line of code is a good line of code!
Authorities: Types of Records

- **People** e.g. authors

  - individualize local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e.g. Jülich with RWTH Aachen and MLZ)

- **Institutes** (people-like aggregates)
  - keep track of history (renaming, merging, splitting)
  - horizonal and vertical linkage
  - complex topography in names (e.g. loops)

- **Grants** (money-like aggregates)
  - horizonal as well as vertical connections
  - different types (POF (HGF-specific), EU, DFG, ...)
  - flexibly extensible by librarians
  - visibility in websubmit depending on instance
Authorities: Types of Records

- **People** e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)
Authorities: Types of Records

- **People** e.g. authors
  - *individualize(!) local authors*
  - *entries are real people, not names* (distinguish Wagner, A. and Wagner, A.)
  - *connect to local institutes* (incl. history)
  - *allow interchange of records* (e.g. Jülich with RWTH Aachen and MLZ)

- **Institutes** (people-like aggregates)
Authorities: Types of Records

- **People** e. g. authors
  - individualize(!) local authors
  - entries are real people, not names  (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)

- **Institutes** (people-like aggregates)
  - keep track of history  (renaming, merging, splitting)
  - horizontal and vertical linkage
  - complex topography in names  (e. g. loops)
Authorities: Types of Records

- **People** e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)

- **Institutes** (people-like aggregates)
  - keep track of history (renaming, merging, splitting)
  - horizontal and vertical linkage
  - complex topography in names (e. g. loops)

- **Grants** (money-like aggregates)
Authorities: Types of Records

- **People**  e. g. authors
  - individualize(!) local authors
  - entries are real people, not names  (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes  (incl. history)
  - allow interchange of records  (e. g. Jülich with RWTH Aachen and MLZ)

- **Institutes**  (people-like aggregates)
  - keep track of history  (renaming, merging, splitting)
  - horizontal and vertical linkage
  - complex topography in names  (e. g. loops)

- **Grants**  (money-like aggregates)
  - horizontal as well as vertical connections
  - different types  (POF (HGF-specific), EU, DFG, . . .)
  - flexibly extensible by librarians
  - visibility in websubmit depending on instance
Authorities: Types of Records

- Periodicals, Journals

- need ≈ 30,000 journals (links to german ZDB and EZB)
- complex external tool to build up records
- autogenerated/updated once a year (trigger: new JCR)
- allow harvesting of records generated
- contain statistics keys for evaluation

- print and online edition are one journal (ISSN does not work as key)

- Statistics keys
  - database coverage (journal listed in Scopus or Web of Science, JCR or pubmed...)
  - OpenAccess designation
  - license identifier (Allianz-licensing)

- Vocabulary
  - allow for classifications (e.g. POF level)
Authorities: Types of Records

- **Periodicals, Journals**
  - need \( \approx 30,000 \) journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain **statistics keys** for evaluation
  - print and online edition are one journal (ISSN does not work as key)
Authorities: Types of Records

- Periodicals, Journals
  - need \( \approx 30,000 \) journals (links to German ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - print and online edition are one journal (ISSN does not work as key)

- Statistics keys
Authorities: Types of Records

- **Periodicals, Journals**
  - need ≈ 30,000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain **statistics keys** for evaluation
  - print and online edition are one journal (ISSN does not work as key)

- **Statistics keys**
  - database coverage (journal listed in Scopus or Web of Science, JCR or pubmed …)
  - OpenAccess designation
  - license identifier (Allianz-licensing)
Authorities: Types of Records

- Periodicals, Journals
  - need ≈ 30,000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - print and online edition are one journal (ISSN does not work as key)

- Statistics keys
  - database coverage (journal listed in Scopus or Web of Science, JCR or pubmed...)
  - OpenAccess designation
  - license identifier (Allianz-licensing)

- Vocabulary
Authorities: Types of Records

- Periodicals, Journals
  - need ≈ 30,000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - print and online edition are one journal (ISSN does not work as key)

- Statistics keys
  - database coverage (journal listed in Scopus or Web of Science, JCR or pubmed ...)
  - OpenAccess designation
  - license identifier (Allianz-licensing)

- Vocabulary
  - allow for classifications (e.g. POF level)
Frontend

- rich, browserbased GUI: (AJAX and CSS)
Frontend

- rich, browserbased GUI: (AJAX and CSS)
- JavaScript (JQuery + x)
Frontend

- rich, browser-based GUI: (AJAX and CSS)
- JavaScript (jQuery + x)
- use few different elements (mainly tokeninput, autocomplete, datepicker)

Tokeninput (heavily extended and debugged(!) version of jQuery.tokeninput)
Frontend

- rich, browserbased GUI: (AJAX and CSS)
- JavaScript (JQuery + x)
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
Frontend

- **rich, browserbased GUI:** (AJAX and CSS)
- **JavaScript** (JQuery + x)
- **use few different elements** (mainly tokeninput, autocomplete, datepicker)
- **browser independent** (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- **Tokeninput** (heavily extended and debugged(!) version of jquery.tokeninput)
Frontend

- rich, browserbased GUI: (AJAX and CSS)
- JavaScript (JQuery + x)
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
Frontend

- **rich, browser based GUI**: (AJAX and CSS)
- **JavaScript** (JQuery + x)
- **use few different elements** (mainly tokeninput, autocomplete, datepicker)
- **browser independent** (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- **Tokeninput** (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- **JSON-returns from the backend**
Frontend

- rich, browser-based GUI: (AJAX and CSS)
- JavaScript (jQuery + x)
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jQuery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)
Frontend

- rich, browserbased GUI: (AJAX and CSS)
- JavaScript (JQuery + x)
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)
- use encoding scheme for returns (allow processing in loops)
Frontend

- **rich, browserbased GUI:** (AJAX and CSS)
- **JavaScript** (JQuery + x)
- **use few different elements** (mainly tokeninput, autocomplete, datepicker)
- **browser independent** (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- **Tokeninput** (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- **JSON-returns from the backend**
- **avoid program logic** (backend has to deliver ready to use data)
- **use encoding scheme for returns** (allow processing in loops)
- **define hidden subfields to keep structures** (e. g. repeatable fields)
Frontend

- rich, browserbased GUI: (AJAX and CSS)
- JavaScript (JQuery + x)
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)
- use encoding scheme for returns (allow processing in loops)
- define hidden subfields to keep structures (e. g. repeatable fields)
- invisibly enrich data from the backend (e. g. IDs, DDC, statistics keys...)
New stuff for Websubmit

- repeatable fields (tokeninputs)
New stuff for Websubmit

- repeatable fields *(tokeninputs)*
  - authors
  - grants
  - institutes
  - ...
  
  Identifiers
  Statistics keys

Handling
Add hidden fields for all visible fields and do the fancy stuff there.
Pass on string encoded JSON structures.
New stuff for Websubmit

- repeatable fields (tokeninputs)
  - authors
  - grants
  - institutes
  - ...

- structured fields (mostly hidden)
New stuff for Websubmit

- repeatable fields \((\text{tokeninputs})\)
  - authors
  - grants
  - institutes
  - ...

- structured fields \((\text{mostly hidden})\)
  - Identifiers
  - statistics keys
New stuff for Websubmit

- repeatable fields *(tokeninputs)*
  - authors
  - grants
  - institutes
  - ...

- structured fields *(mostly hidden)*
  - Identifiers
  - statistics keys

Handling

Add hidden fields for all visible fields and do the fancy stuff there. Pass on string encoded JSON structures.
Backend

Exposé records in **JSON** structures to **JavaScript**
Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
Backend

Expose records in **JSON** structures to JavaScript

- no repeatable **subfields** simplifies JSON
- define new **output format**: **JS**
Exposé records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
Exposé records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i.e. escape substructures to strings)
Exposé records in **JSON** structures to **JavaScript**

- no repeatable **subfields** simplifies **JSON**
- define new **output format**: **JS**
- define **format templates** for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i.e. escape substructures to strings)
- return hashes keyed as **MARC**:
Backend

Expose records in **JSON** structures to **JavaScript**

- no repeatable **subfields** simplifies JSON
- define new **output format**: **JS**
- define **format templates** for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i.e. escape substructures to strings)
- return hashes keyed as MARC:
  1. Field/Subfield: I245__a
  2. Structure: I536__

Alexander Wagner
Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i.e. escape substructures to strings)
- return hashes keyed as MARC:
  1. Field/Subfield: I245__a
  2. Structure: I536__
- innerrecordlinks: return from the backend (e.g. 536__ and 913__)
Generate People Records

Use the local phone book (=LDAP, avoid privacy issues)
Generate People Records

Use the local phone book (=LDAP, avoid privacy issues)

- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs
- allow for multiple IDs per entity (0247 vs. 035__a)
- allow usage of IDs from several sources
Generate People Records

Use the local phone book (=LDAP, avoid privacy issues)

- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs
- allow for multiple IDs per entity (0247__ vs. 035__a)
- allow usage of IDs from several sources

Goal
Implement ORCiD linkup with automagic registration/fetching
Statistics

Use Marc 915__

Upon journal association: keys → bibliographic records
Statistics

Use Marc 915__

Upon journal association: keys → bibliographic records

- Pythonic bean counters (use intbitsets)
Statistics

Use Marc 915__

Upon journal association: keys $\rightarrow$ bibliographic records

- Pythonic bean counters (use intbitsets)
- allow live Web query (if fast enough)
Statistics

**Use Marc 915**

Upon journal association: keys → bibliographic records

- Pythonic bean counters (use intbitsets)
- allow live Web query (if fast enough)
- allow complex queries
Statistics

Use Marc 915

Upon journal association: keys → bibliographic records

- Pythonic beancounters (use intbitsets)
- allow live Web query (if fast enough)
- allow complex queries, e.g.
  - output per institute and/or program
  - all JCR covered journal articles with external authors
  - all Scopus listed articles from John Smith (director of Inst X)
  - ...

Alexander Wagner
Statistics

Use Marc 915__

Upon journal association: keys → bibliographic records

- Pythonic beancounters *(use intbitsets)*
- allow live Web query *(if fast enough)*
- allow complex queries, e. g.
  - output per institute and/or program
  - all JCR covered journal articles with external authors
  - all Scopus listed articles from John Smith *(director of Inst X)*
  - ...
- CLI is enough, but should be easily adoptable
Statistics

Use Marc 915__

Upon journal association: keys → bibliographic records

- Pythonic beancounters (use intbitsets)
- allow live Web query (if fast enough)
- allow complex queries, e. g.
  - output per institute and/or program
  - all JCR covered journal articles with external authors
  - all Scopus listed articles from John Smith (director of Inst X)
  - ... 
- CLI is enough, but should be easily adoptable
- generate LaTeX-based reports
Statistics

**Use Marc 915**

**Upon journal association: keys → bibliographic records**

- Pythonic beancounters *(use intbitsets)*
- allow live Web query *(if fast enough)*
- allow complex queries, e. g.
  - output per institute and/or program
  - all JCR covered journal articles with external authors
  - all Scopus listed articles from John Smith *(director of Inst X)*
  - …
- CLI is enough, but should be easily adoptable
- generate LaTeX-based reports
- generate structured outputs *(for spreadsheet, literature management)*
Statistics/2: Bibliometrics

Mainly in Jülich: JuSER is source for bibliometric evaluations
Statistics/2: Bibliometrics

Mainly in Jülich: JuSER is source for bibliometric evaluations

- link up with Web of Science (just another ID \(\Rightarrow 0247_\) )
- semiautomatic Web of Science-ID assignments (doi2ut)
- specific exports
- planned: direct link to workflow tools
Statistics/2: Bibliometrics

Mainly in Jülich: JuSER is source for bibliometric evaluations

- link up with Web of Science (just another ID ⇒ 0247_-)
- semiautomatic Web of Science-ID assignments (doi2ut)
- specific exports
- planned: direct link to workflow tools

In principle

... all queries can be done in the Webfrontend
OpenAccess

- Aquire more fulltexts
OpenAccess

- Acquire more fulltexts
- At least give a stable link (e.g. pubmed-Central, OA-Journals)
OpenAccess

- Aquire more fulltexts
- At least give a stable link (e.g. pubmed-Central, OA-Journals)
  - Use statistics key for OA-Journals (for DOAJ-listed journals)
OpenAccess

- Aquire more fulltexts
- At least give a stable link *(e.g. PubMed-Central, OA-Journals)*
  - Use statistics key for OA-Journals *(for DOAJ-listed journals)*
  - enrich upon ingest
OpenAccess

- Aquire more fulltexts
- At least give a stable link (e.g. PubMed-Central, OA-Journals)
  - Use statistics key for OA-Journals (for DOAJ-listed journals)
  - enrich upon ingest
- Implement “Allianz-Licensing” (ingest publishers versions)
OpenAccess

- Aquire more fulltexts
- At least give a stable link (e.g. PubMed-Central, OA-Journals)
  - Use statistics key for OA-Journals (for DOAJ-listed journals)
  - enrich upon ingest
- Implement “Allianz-Licensing” (ingest publishers versions)
- Make OA-Articles more prominent (e.g. logo, collection)
OpenAccess

- Acquire more fulltexts
- At least give a stable link (e.g. PubMed-Central, OA-Journals)
  - Use statistics key for OA-Journals (for DOAJ-listed journals)
  - Enrich upon ingest
- Implement “Allianz-Licensing” (ingest publishers versions)
- Make OA-Articles more prominent (e.g. logo, collection)
- OpenAIRE compliance
OpenAccess

- **Aquire more fulltexts**
- **At least give a stable link** (e. g. PubMed-Central, OA-Journals)
  - Use statistics key for OA-Journals (for DOAJ-listed journals)
  - enrich upon ingest
- **Implement “Allianz-Licensing”** (ingest publishers versions)
- **Make OA-Articles more prominent** (e. g. logo, collection)
- **OpenAIRE compliance**
- **OAI-Harvesting for BASE and friends**
Next Steps

- Upgrade from 1.0 to 1.1 or, more likely, 1.2 (fix OAI Server!)
- Feed Google Scholar (how?)
- Get pending instances online:
  - RWTH Aachen
  - MLZ
- At Jülich: implement crossharvesting (RWTH ↔ FZJ ↔ MLZ)
- Add workflow for “vita” (workflows for special collections)
Next Steps

- Upgrade from 1.0 to 1.1 or, more likely, 1.2 *(fix OAI Server!)*
- Feed Google Scholar *(how?)*
- Get pending instances online:
  - RWTH Aachen
  - MLZ
- At Jülich: implement crossharvesting *(RWTH ↔ FZJ ↔ MLZ)*
- Add workflow for “vita” *(workflows for special collections)*

Continually refactor/check code to get ready to give back
Project partners

Deutsches Elektronensynchrotron, Zentralbibliothek

Forschungszentrum Jülich, Zentralbibliothek

GSI Helmholtzzentrum für Schwerionenforschung, Bibliothek + Kern-IT

RWTH Aachen, Hochschulbibliothek

Maier-Leibniz-Zentrum, Garching
Questions?

Alexander Wagner
Zentralbibliothek
Fachinformation, wiss. Publizieren
Tel.: 1586
a.wagner@fz-juelich.de

http://www.fz-juelich.de/zb

This document is available as FZJ-2013-02764