Karlsruher Institut für Technologie

Storing and Analyzing Bibliographic Metadata with ElasticSearch

Clemens Düpmeier, Christian Schmitt (KIT / IAI)

Institute of Applied Computer Science (IAI)

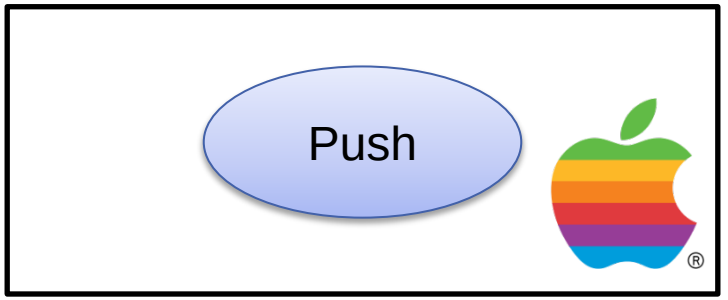


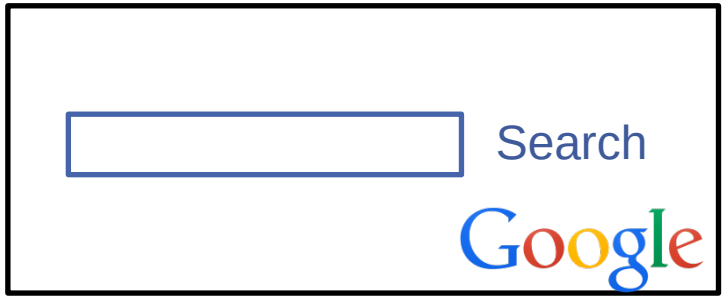
Overview

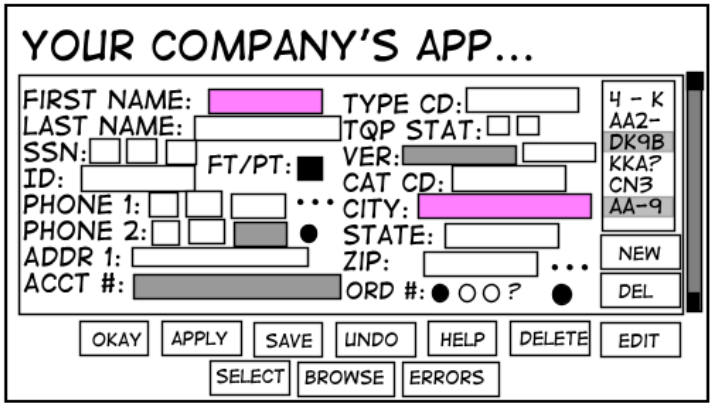
- Search based web applications and the openTA project
- A short overview on ElasticSearch and its features
- Storing and analyzing bibliographic metadata
- Summary and outlook

Modern web user interfaces

- The design principles of web user interfaces have changed
- Instead of complex forms and static hierarchical navigation
  =>
  responsive and natural language based interfaces
- Search engine technology can support these types of user interfaces well



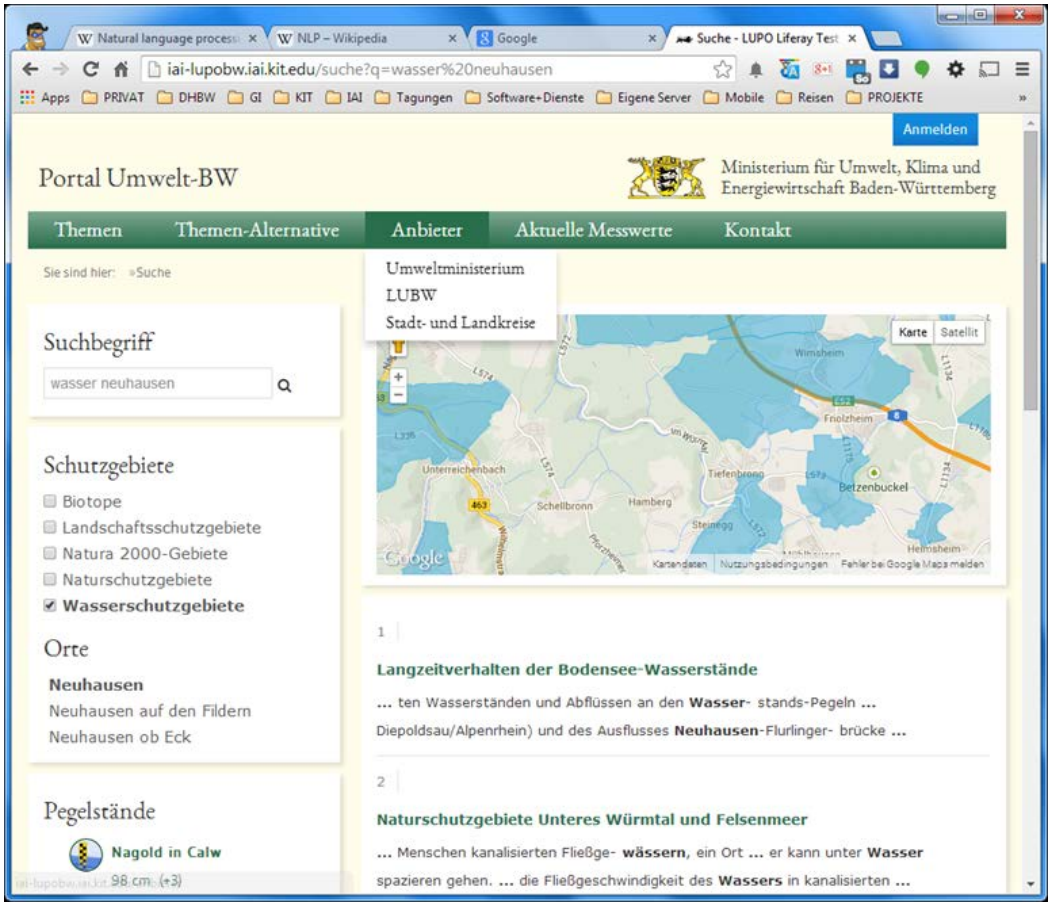




Search based web applications

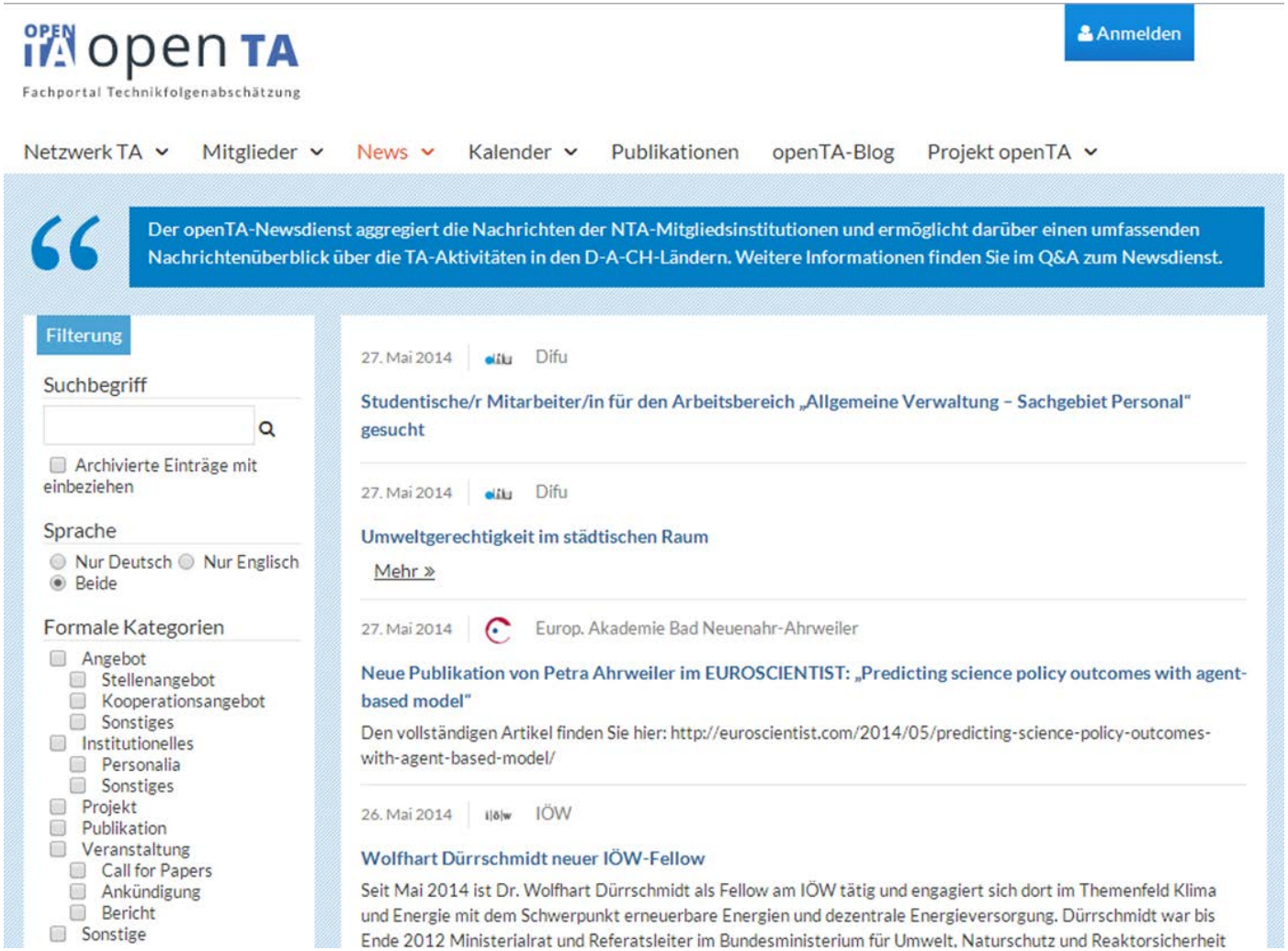
- Use search engine technology as key element for data access
- Available data is a mixture of
  - unstructured
  - semi-structured
  - structured

  information coming from different sources
- Use semantic technologies for
  - aggregation,
  - normalization and
  - classification of data
- And a natural language approach for data access



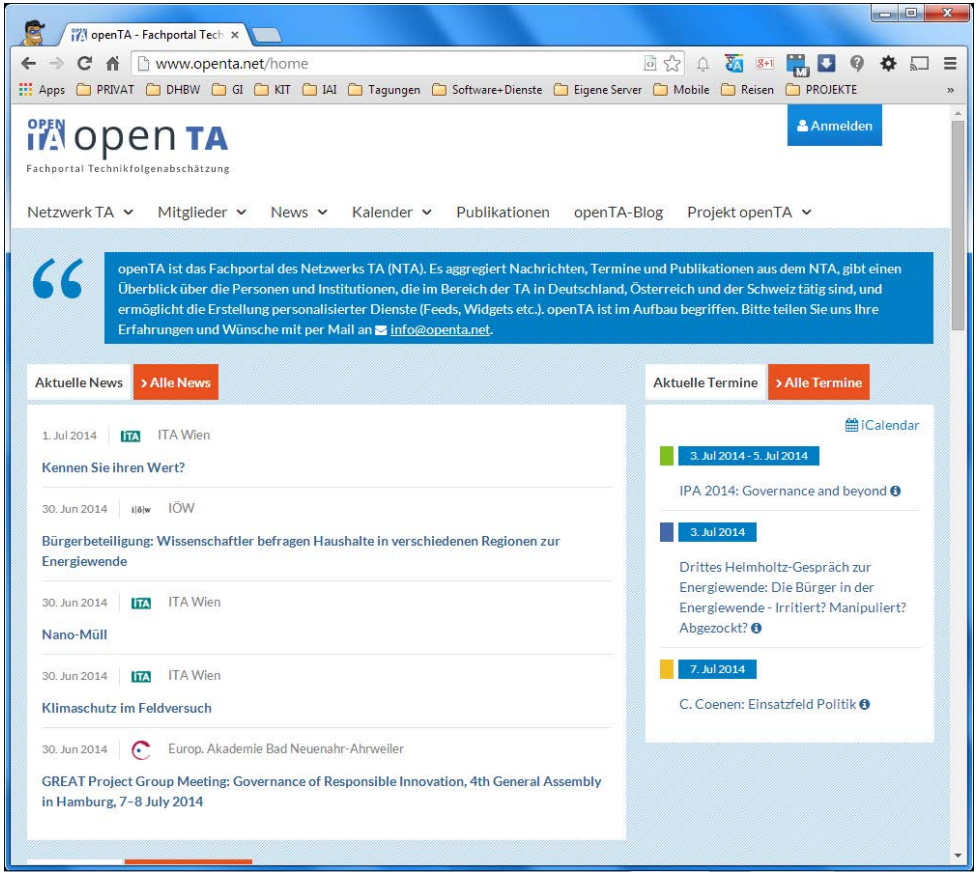
Search based environmental information portal

The openTA portal is search based too



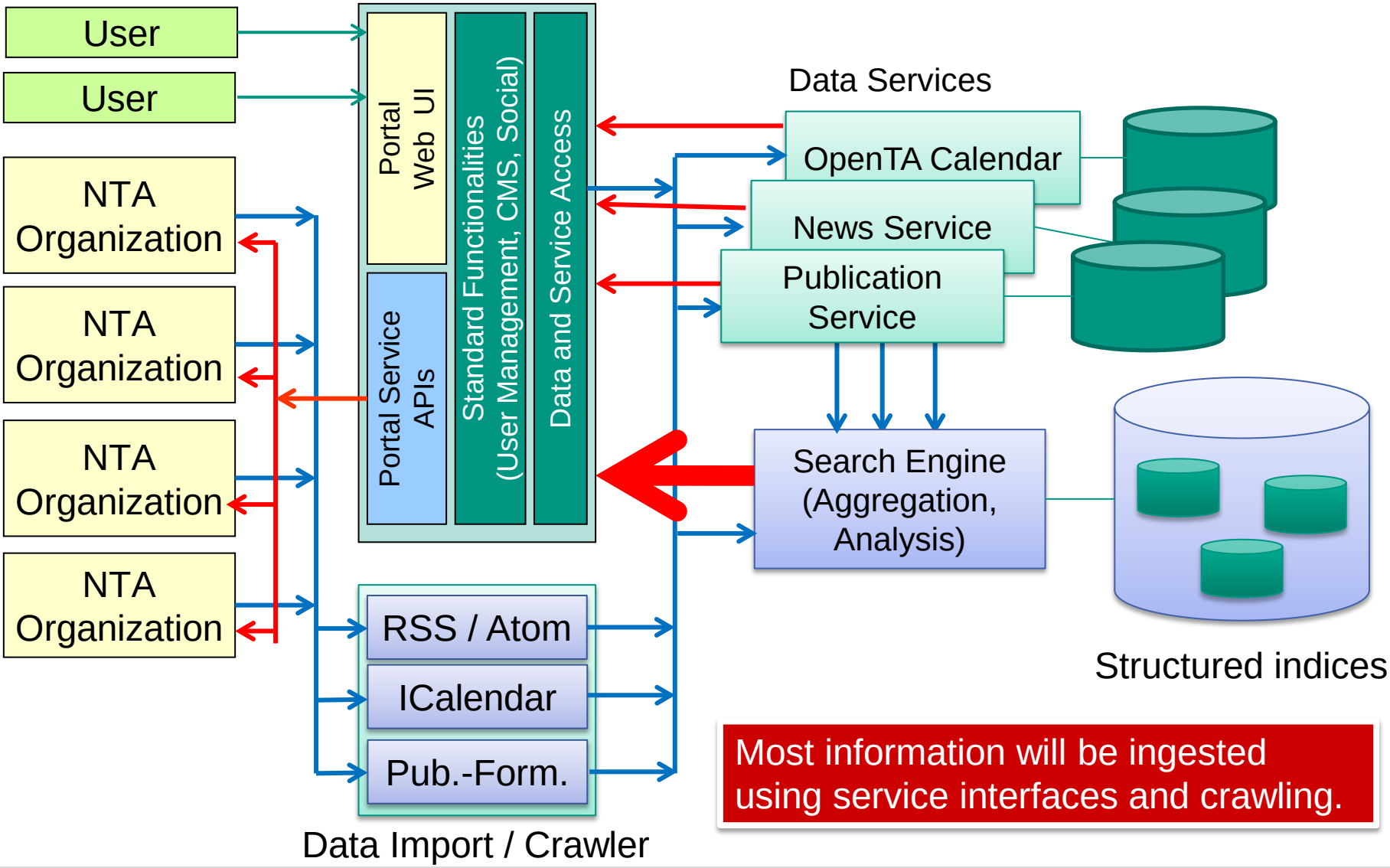
The openTA portal

- DfG project with partners (IAI, ITAS, KIT library) from KIT
- Web portal for the Network Technology Assessment (NTA)
  - Aggregates and provides information about
    - members, organizations
    - news and events
    - scientific publications
  - Provides services and web technology (widgets) to use the aggregated information remotely



Homepage of openTA with aggregated news feed and calendar

Search driven architecture of openTA



What kind of search technology?

- Well known open source products are: Solr, ElasticSearch
  - Both are based on Apache Lucene
- They both provide
  - Document oriented, structured search indices
  - Scalable architecture by distributing and processing index entries on multiple servers („Sharding“)
  - Advanced features for data aggregation, analysis and retrieval
- The openTA project uses ElasticSearch
    - Newer and better architecture (compared to Solr)
      - Completely REST and
      - JSON format based
    - Supports hierarchical document structures and parent / child relationships
    - Supports queries against multiple indices at once
    - And other interesting features explained on the next slides

Some elements of the search language (Lucine based)

Terms	apple apple iphone
Phrases	"apple iphone"
Proximity	"apple safari"~5
Fuzzy	apple~0.8
Wildcards	app* *pp*
Boosting	apple^10 safari
Range	[2011/05/01 TO 2011/05/31] [java TO json]
Boolean	apple AND NOT iphone +apple -iphone (apple OR iphone) AND NOT review
Fields	title:iphone^15 OR body:iphone published_on:[2011/05/01 TO "2011/05/27 10:00:00"]

Indexing data with ElasticSearch

- Send JSON documents to server, e.g. use REST API
  - No schema necessary => ElasticSearch determines type of attributes
  - But's possible to explicitly specify types for attributes
    - Like string, byte, short, integer, long, float, double, boolean, date
- Analysis of (text) attributes
  - Word extraction, reduction of words to their base form (stemming)
  - Stop words
  - Support for multiple languages
  - Can be extended via plugins for quite complex analysis
    - Entity recognition
    - De-duplication
- Generate identifier for data sets; duplicate recognition

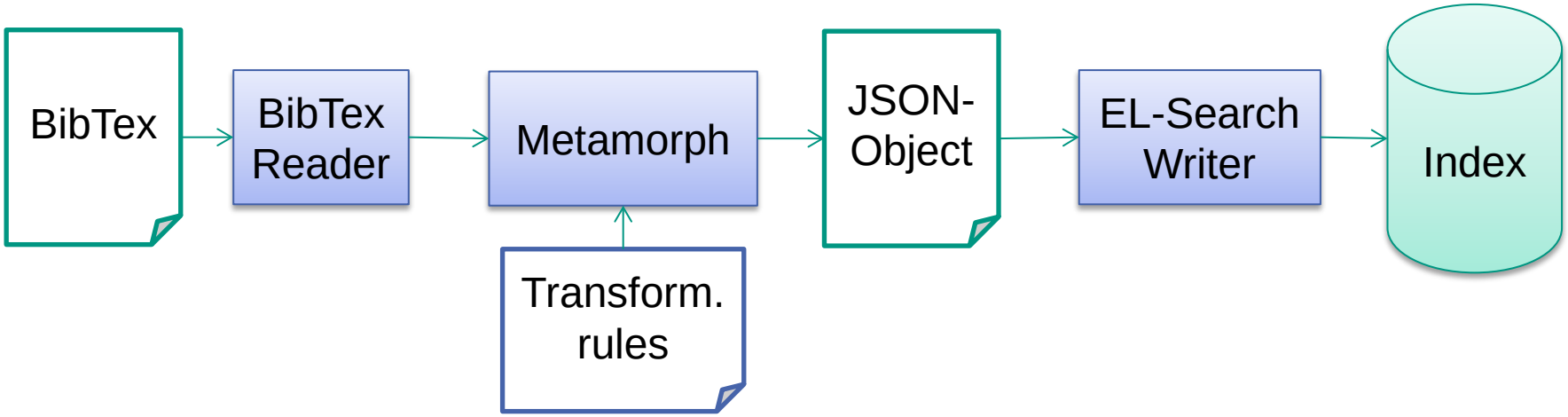
Query Possibilities

- Combined search on different attributes and different indices
  - Many possibilities for full-text search on attribute values
    - Exact, non-exact, proximity (phrases), partial match
  - Support well-known logical operators (And / or, …)
  - Range queries (i.e. date ranges)
  - …
- Control relevance and ranking of search results, sort them
  - Boost relevance while indexing
  - Boost or ignore relevance while querying
  - Different possibilities to sort search results otherwise

More advanced features

- Multi-tenant
- Spatial data queries
- Search suggestions
- Real time aggregation of search data
  - Statistical calculations (sums, mean value, max, min, …)
  - Faceting
    - By using terms
    - Statistical calculations
    - Classification (Grouping by using ranges
    - Filter rules
    - By geographical distance
- Percolators
- Warm up of indices

Importing bibliographic data in ElasticSearch



- Readers transform source format to record-oriented Metamorph format (in Java)
- Metafacture / Metamorph is a rule based transformation engine for bibliographic data (open source as part of the CultureGraph platform, DNB)
- Transformation rules applied by Metamorph transform data into internal objects, which are then serialized to JSON
- EL-Search Writer stores the JSON objects in ElasticSearch

Some features of Metamorph

- Change grouping of attributes, change name of attribute
- Split / join attributes
- Substring and pattern extraction
- Transformation between data types
- Map values to other values using lookup tables
- Build records from attributes, convert data records in attributes
- Supports recursion and conditions in transformation rules
- Allows to plugin your own code at different places of the transformation

Example of Metamorph transformation rules

Karlsruher Institut für Technologie

Transformation rules

Split authors field at character |

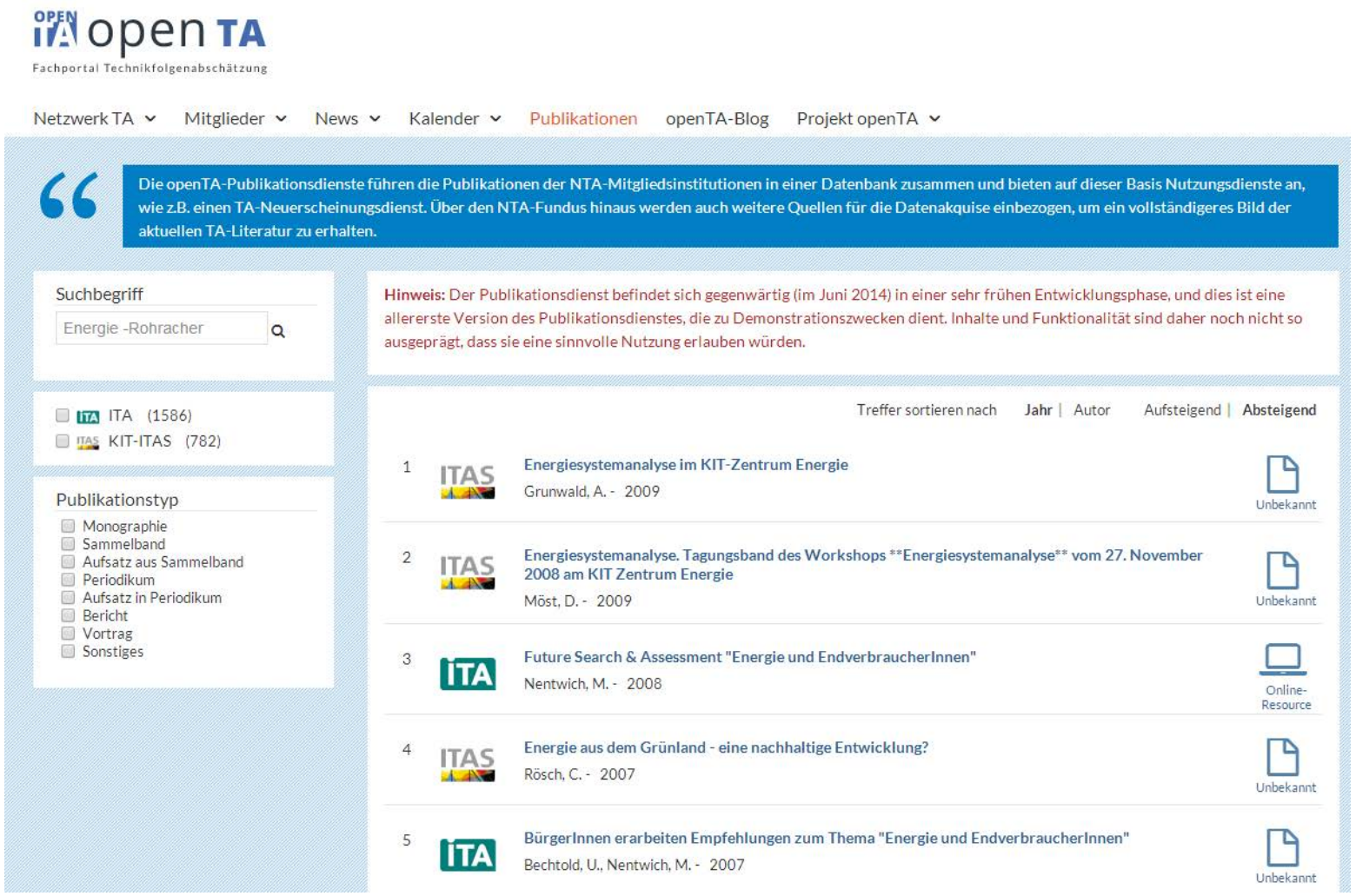
Convert into lower case and map type by using lookup table

```
<?xml version="1.0" encoding="UTF-8"?>
<metamorph>
    <!-- Transformation rules -->
    <rules>
        <data name="foreignId" source="_id" />

        <data name="authors" source="author">
            <split delimiter="[|]" />
            <trim />
        </data>

        <data name="titles" source="title" />

        <data name="pubYear" source="year" />

        <data name="pubType" source="_type">
            <case to="lower" />
            <lookup in="pubtypes" />
        </data>

        <data name="contributorId" source="_contributor" />

    </rules>
</metamorph>
```

Role of ElasticSearch

- While storing
  - Generate unique identifier und keys to find duplicates
  - Group documents which seem to describe same entities
  - Further analysis of attributes (especially attributes containing text)
    - Entity recognition
    - Keyword extraction / Classification
- Querying in realtime
  - Search across multiple fields, calculate ranking
    - Hits in different fields can be given different ranking factors (i.e. finding an author name in the author field)
    - Or matching the publication date
  - Aggregation of data; calculations for faceted search
    - Publication year
    - Top 10 authors (with most publications)
    - Type of publication

Prototype: Search with „Energie –Rohracher“



Summary and outlook

- The search based architecture of openTA works really well
- First prototype of publication service looks promising too
  - Implemented import pipeline and usage of Metamorph allows for a modular extension to support more formats
  - ElasticSearch already showed its great potential but only the basic functionality is used at the moment
- Publication service will be continually enhanced
  - New sources and formats will be added
  - More facets and filter functionality will be implemented in the UI
  - Explore more advanced features and extensions of ElasticSearch to solve some challenges

openTA website

http://www.openta.net

Metamorph: Map Publication Types

```
<!-- Data maps -->
<maps>

    <map name="pubtypes">
        <entry name="article" value="PERIODICAL_ARTICLE" />
        <entry name="book" value="MONOGRAPH" />
        <entry name="booklet" value="MISC" />
        <entry name="conference" value="MISC" />
        <entry name="inbook" value="COLLECTIVE_VOLUME_ARTICLE" />
        <entry name="incollection" value="COLLECTIVE_VOLUME_ARTICLE" />
        <entry name="inproceedings" value="COLLECTIVE_VOLUME_ARTICLE" />
        <entry name="manual" value="REPORT" />
        <entry name="mastersthesis" value="MISC" />
        <entry name="misc" value="MISC" />
        <entry name="phdthesis" value="MISC" />
        <entry name="proceedings" value="COLLECTIVE_VOLUME" />
        <entry name="techreport" value="REPORT" />
        <entry name="unpublished" value="MISC" />
    </map>

</maps>
```

- Metamorph allows to transform values by map lookup

BibTex -> JSON Mapping

```
@article{ITAS-ID5582 ,
author = "Nentwich, M.| Riehm, U. ",
title = "Internationale Fachportale
für Technikfolgenabschätzung. Brauchen
wir eines oder sogar mehrere?",
journal = "Technikfolgenabschätzung -
Theorie und Praxis ",
year  = "2012",
pages = "76-80",
volume = "21",
number = "3",
evastar_pdf = "neri12a.pdf",
ISSN = "16197623",
note= "language = de"
}
```




```
{
  "id": "47e26700-90ab-38fb-9fb0-5b72fad74610",
  "foreignId": "ITAS-ID5582",
  "authors": [
    "Nentwich, M.",
    "Riehm, U."
  ],
  "editors": null,
  "pubType": "PERIODICAL_ARTICLE",
  "titles": [
    "Internationale Fachportale f\u00fcr Technikfolgenabsch\u00e4tzung.
    Brauchen wir eines oder sogar mehrere?"
  ],
  "pubYear": 2012,
  "contributorId": 12481,
  "contentType": "UNKNOWN",
  "mediaType": "UNKNOWN",
  "fullTextReference": "http:\/\/www.itas.kit.edu\/pub\/v\/2012\/neri12a.pdf",
  "valid": true,
  "created": "2014-05-28T16:23:14+0200",
  "lastModification": "2014-05-28T16:23:14+0200"
}
```


- Text based format -> Object based format
- Split and rename fields
- Normalization of data type notions (e.g. date)

BibTex Format

```
@article{ITAS-ID5582 ,
author = "Nentwich, M.| Riehm, U. ",
title = "Internationale Fachportale für Technikfolgenabschätzung.
Brauchen wir eines oder sogar mehrere?",
journal = "Technikfolgenabschätzung - Theorie und Praxis ",
year  = "2012",
pages = "76-80",
volume = "21",
number = "3",
evastar_pdf = "neri12a.pdf",
ISSN = "16197623",
note= "language = de"
}
```

- Text based format consisting of several text lines wrapped in @article{} construct

Example of metadata as JSON


```
{
  "id": "47e26700-90ab-38fb-9fb0-5b72fad74610",
  "foreignId": "ITAS-ID5582",
  "authors": [
    "Nentwich, M.",
    "Riehm, U."
  ],
  "editors": null,
  "pubType": "PERIODICAL_ARTICLE",
  "titles": [
    "Internationale Fachportale f\u00fcr Technikfolgenabsch\u00e4tzung.
    Brauchen wir eines oder sogar mehrere?"
  ],
  "pubYear": 2012,
  "contributorId": 12481,
  "contentType": "UNKNOWN",
  "mediaType": "UNKNOWN",
  "fullTextReference": "http:\/\/www.itas.kit.edu\/pub\/v\/2012\/neri12a.pdf",
  "valid": true,
  "created": "2014-05-28T16:23:14+0200",
  "lastModification": "2014-05-28T16:23:14+0200"
}
```


- Text version of JSON format used by openTA