From strings to things

A Linked Open Data API for library hackers and web developers

SWIB 2013, Hamburg

November 27th, 2013

hbz

Linked Open Data

- Interoperability through common, flexible data model and common identifiers
- <Typee> <was written by> <Melville>
- <http :// lobid .org/resource/HT002189125> <http :// purl .org/dc/elements/1.1/creator> <http :// d-nb.info/gnd/118580604> .

Message

- So our message has been: Use things, not strings!
- e.g.
  http :// d–nb.info/gnd/118580604, not ‘Melville, Herman’, ‘Herman Melville’, ‘H. Melville’, etc.
- But: where to get these IDs from?

CC-SA-2.0 Infrogmation of New Orleans, Wikimedia Commons, File:WrongWayCarrolltonNOLA.JPG

Message

“Clothes are great, so please learn knitting”

CC-BY-2.0 Angela Montillon, Wikimedia Commons, `File:Colourful_wool_2.jpg`
CC-SA-2.5 Wikimedia Commons, `File:Knit4.jpg`
CC-BY-SA-3.0 Jomegat, Wikimedia Commons, `File:Knitting_dropped_stitch_5.jpg`

Response

- “OK, but can’t I just wear some clothes? Do I have to create them myself, manually?”
- Do you have to be a LOD expert to benefit from LOD?

CC-BY-2.0 Andrew Vargas, Wikimedia Commons, `File:Well-clothed_baby.jpg`

lobid.org

- lobid.org: LOD service of hbz, since 2010
- title data of union catalog (lobid-resources), authority data (lobid-organisations)
- Dumps, resolvable URIs, content negotiation, RDFa, SPARQL (triple store)
- different problems, new requirements → developed a new backend since late 2012

Problems

- General performance issues: complex queries causing triple store hang ups
- Specific performance-critical use cases: auto suggest, e.g. for authority data
- Technological obscurity: Semantic Web, cutting edge since 2001. Our goal: provide data, not just evangelize technology

Approach

- Fix performance problems: stabilize current applications and enable new use cases
- Put the web and web developers into focus
- LOD for web devs, not only for LOD experts

hbz

Approach

JSON over HTTP

API: what

- Application programming interfaces: essential for reusable software modules
- These modules communicate only via their API, they know no implementation details
- So implementations become exchangeable – without requiring changes in API clients

API: why

- Only with a stable API, modules are *actually* reusable: reuse has to work
- Triple store or search index not suitable as an API: should provide a stable abstraction over implementation details and the data

API: requests

```
GET /resource?id=0940450003
GET /resource?name=Typee

GET /organisation?id=DE-605
GET /organisation?name=hbz

GET /person?id=118580604
GET /person?name=Herman+Melville
```

hbz

API: responses

```
GET /person?name=Ernest+Hem&format=short

[
"Hemingway, Ernest (1899-1961)",
"Hemmann, Augustin Ernst Roman (1748-1820)",
"Hempel, Ernst Wilhelm (1745-1799)",
"Jamaigne, Jean Ernest de",
"Lacheman, Ernest R. (1906-1982)",
"Uthemann, Ernest W. (1953-)"
]
```

API: usage

This can be used for an auto suggest feature:

When a suggestion is selected, insert its ID:

API: from strings to things

That actually uses a different response format:

```
GET http://api.lobid.org/person?name=Ernest+Hem&format=ids

[{
    label: "Hemingway, Ernest (1899-1961)",
    value: "http://d-nb.info/gnd/118549030"
},{
    label: "Hemmann, Augustin Ernst Roman (1748-1820)",
    value: "http://d-nb.info/gnd/130030252"
},{
    label: "Hempel, Ernst Wilhelm (1745-1799)",
    value: "http://d-nb.info/gnd/100292437"
}]
```

API: from strings to things

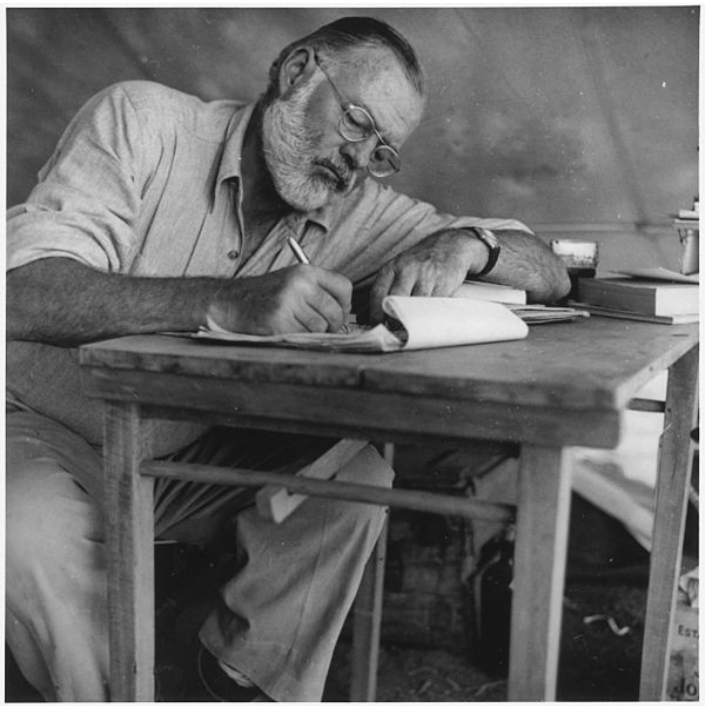
```
GET http://api.lobid.org/person?id=118549030&format=full

[{
    @id: "http://d-nb.info/gnd/118549030",
    preferredNameForThePerson: "Hemingway, Ernest",
    dateOfBirth: "1899",
    dateOfDeath: "1961",
    variantNameForThePerson: [
        "Heminguej, E.", ...
    ],
    placeOfBirth: "http://d-nb.info/gnd/4461931-5",
    sameAs: "http://dbpedia.org/resource/Ernest_Hemingway",
    wikipedia: "http://de.wikipedia.org/wiki/Ernest_Hemingway",
    ...
    @context: "http://api.lobid.org/context/gnd.json"
}]
```

API: from strings to things

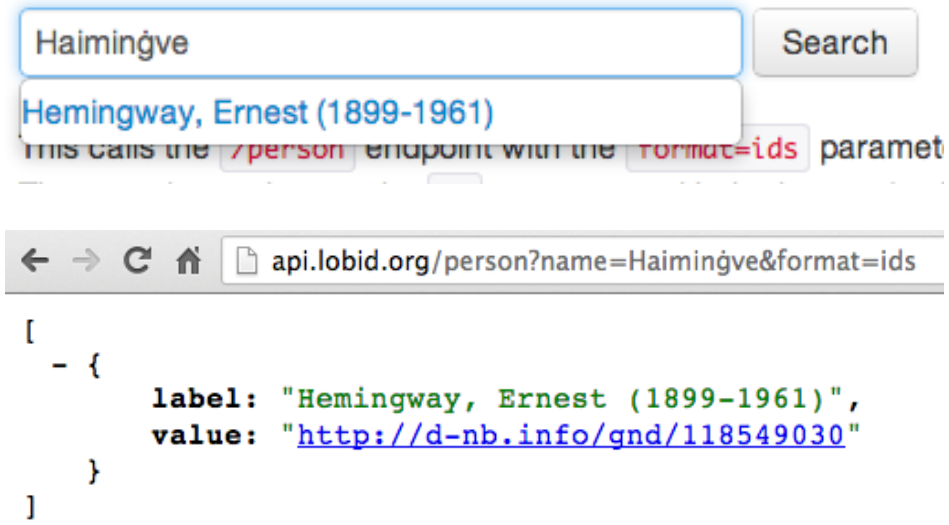
All alternative names:

```
variantNameForThePerson: [
    "Heminguej, E.",
    "Hemingvye, Ernești",
    "Cheminhuej, Ernest",
    "Heminguei, E.",
    "KheminguėĬ, Ėrnest",
    "Cheminguėj, Ėrnest",
    "Hai-ming-wei, ...",
    "Haimingvye",
    "Hemingvejs, Ernests",
    "Hemingway, Ernest Miller",
    "Heminguei, Ernest",
    "Hemingvej, Ernest",
    "Haminġwāy, Arnist",
    "Haminghwāy, Arnist",
    "Himinghwāy",
    "HĬminjwāy, Arnist",
    "Hemingwei, ...",
    "Hemingway, E.",
    "Himinghwāy, Arnist",
    "Hemingway",
    "Hayminghwāy, Arnist",
    "Hemingvey, Ernești",
    "Haminġwāy, Irnist",
    "Chemingouaiē, Ernest",
    "Hemingway, Ernest M.",
    "海明威"
],
```

For: http://d-nb.info/gnd/118549030



API: from strings to things



LOD and Semantic Web technology enable that.
But we shouldn't expect anyone to learn RDF, SPARQL, etc for such a simple use case

API: but where's the LOD

- "But where are the unified IDs in the keys of the JSON response? It's just strings!"
- Enter JSON-LD: @context maps plain JSON keys to URIs → API as abstraction
- JSON-LD also enables RDF serialization, available from API via content negotiation

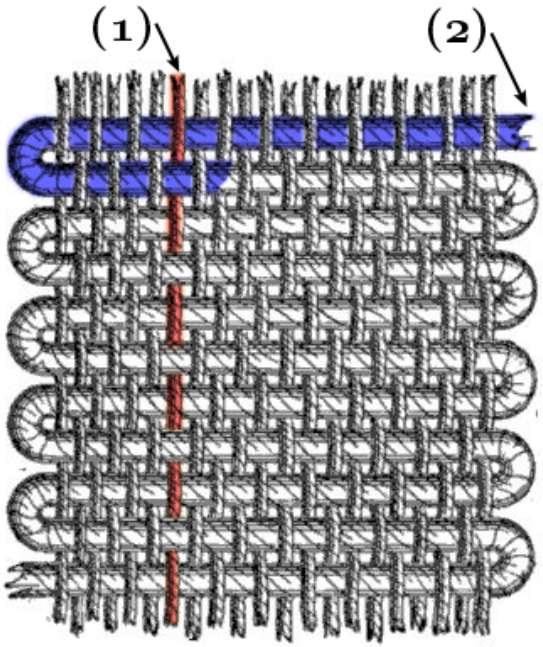
API: documentation

Sample queries, documentation on parameters and content negotiation, auto suggest samples with Javascript code, etc:

`http://api.lobid.org/`

Technology

Community needs to build and share know-how:

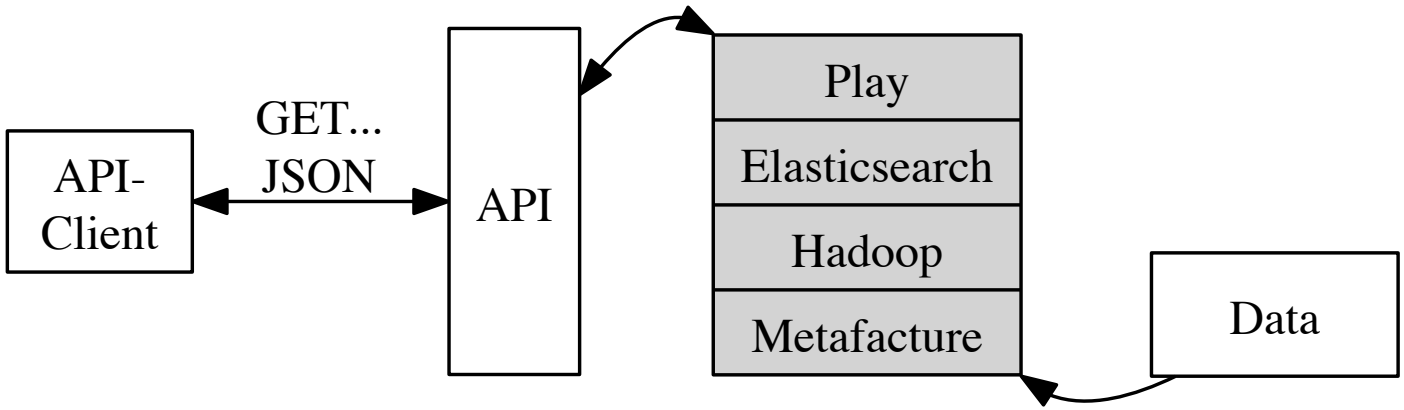


CC-BY-2.0 Angela Montillon, Wikimedia Commons, File:Colourful_wool_2.jpg
CC-BY-SA-3.0 Ryj, derivative: Derwok, Wikimedia Commons, File:Kette_und_Schuß_num_col.png
CC-BY-2.0 Tony Hisgett, Wikimedia Commons, File:Coloured_cloth_2_(3539454254).jpg

hbz

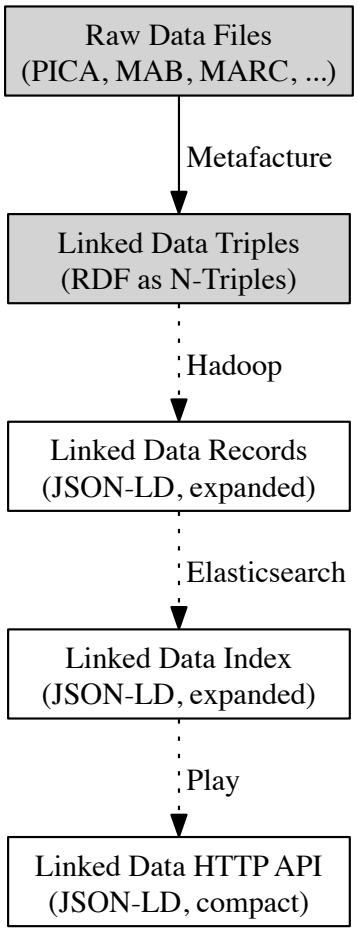
Technology

Our technology stack:
Metafacture, Hadoop, Elasticsearch, Play



Technology

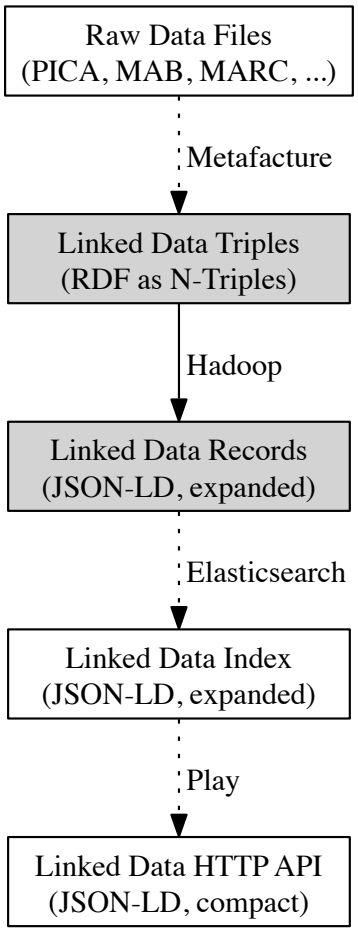
- Raw data to N-Triples: Metafacture
- N-Triples to JSON-LD records: Hadoop
- Indexing JSON-LD: Elasticsearch
- HTTP API: Play-Framework



hbz

Technology

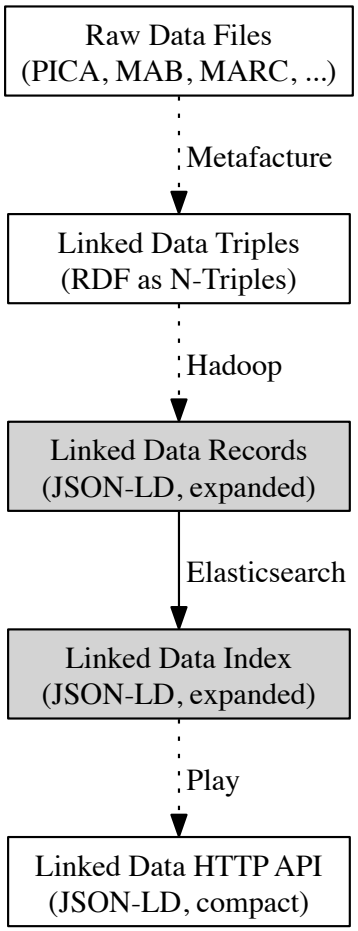
- Raw data to N-Triples: Metafacture
- **N-Triples to JSON-LD records: Hadoop**
- Indexing JSON-LD: Elasticsearch
- HTTP API: Play-Framework



hbz

Technology

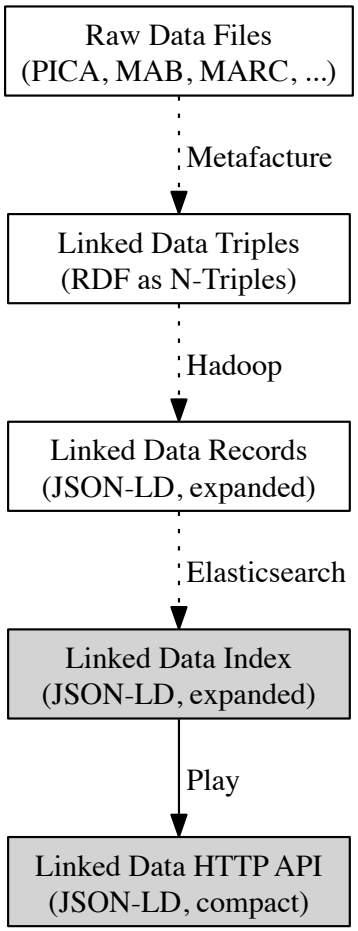
- Raw data to N-Triples: Metafacture
- N-Triples to JSON-LD records: Hadoop
- **Indexing JSON-LD: Elasticsearch**
- HTTP API: Play-Framework



hbz

Technology

- Raw data to N-Triples: Metafacture
- N-Triples to JSON-LD records: Hadoop
- Indexing JSON-LD: Elasticsearch
- HTTP API: Play-Framework

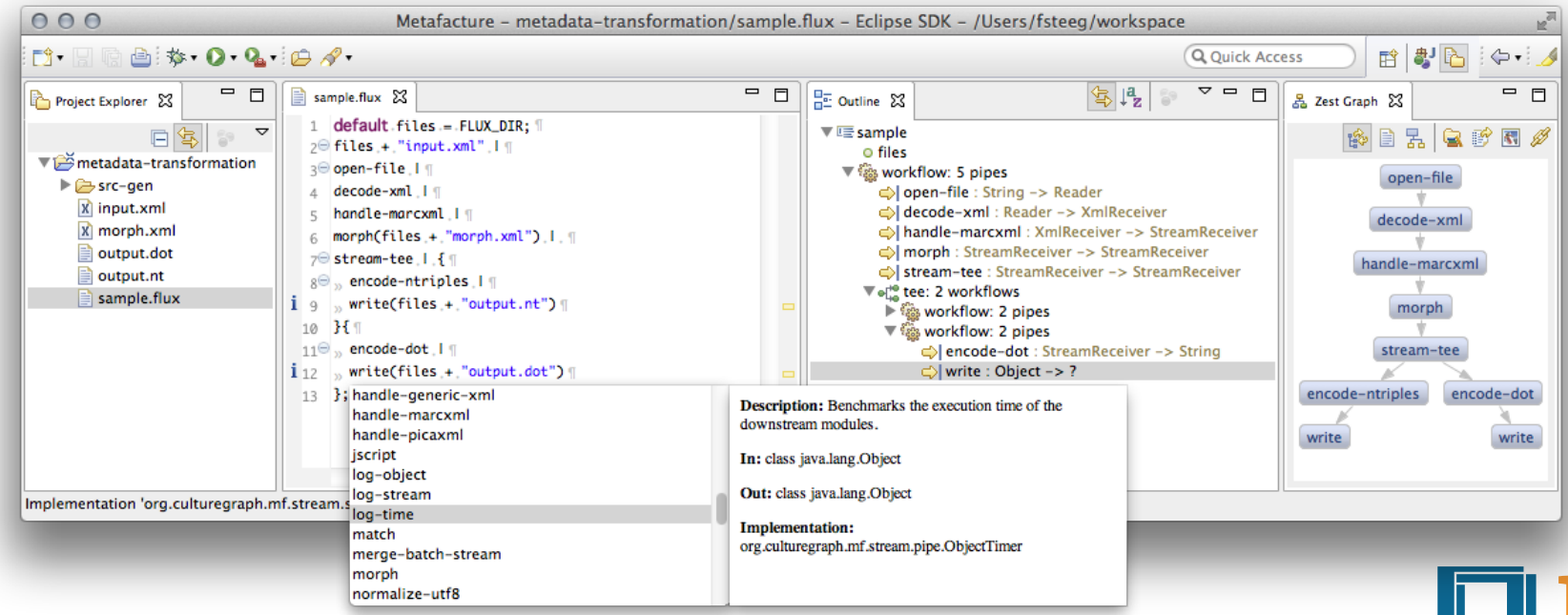


hbz

Metafacture: tools

A tool suite for metadata processing

https://github.com/culturegraph/metafacture-core/wiki
https://github.com/culturegraph/metafacture-ide/wiki



Hadoop: configuration

Config of properties for JSON-LD records:

```
resolve = \
  http://purl.org/dc/elements/1.1/creator;\
  http://purl.org/dc/elements/1.1/contributor;\
  http://purl.org/dc/terms/subject;\
  http://www.w3.org/2003/01/geo/wgs84_pos#location;\
  http://www.w3.org/2006/vcard/ns#adr;\
  http://purl.org/lobid/lv#fundertype;\
  http://purl.org/lobid/lv#stocksize

predicates = \
  http://www.w3.org/1999/02/22-rdf-syntax-ns#type;\
  http://d-nb.info/standards/elementset/gnd#preferredNameForThePerson;\
  http://d-nb.info/standards/elementset/gnd#dateOfBirth;\
  http://d-nb.info/standards/elementset/gnd#dateOfDeath;\
  http://www.w3.org/2004/02/skos/core#prefLabel;\
  http://www.w3.org/2003/01/geo/wgs84_pos#lat;\
  http://www.w3.org/2003/01/geo/wgs84_pos#long;\
```

hbz

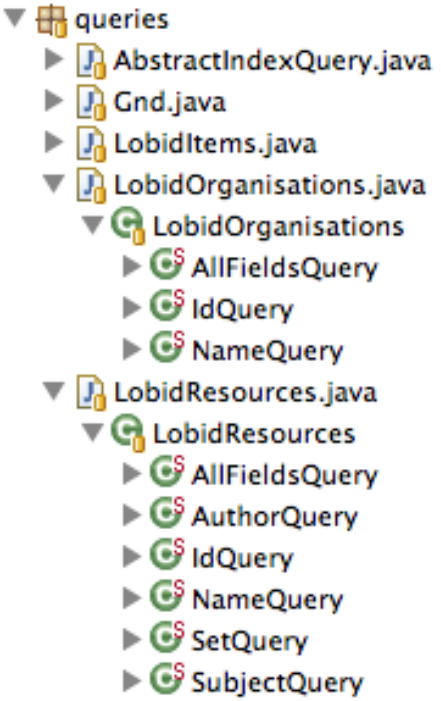
Elasticsearch: indexes

Index overview in Elasticsearch-Head-Plugin:

Play: queries

Elasticsearch queries from Play controllers:

```
public static class NameQuery extends AbstractIndexQuery {

    @Override
    public List<String> fields() {
        return Arrays.asList(
                "@graph.http://xmlns.com/foaf/0.1/name.@value",
                "@graph.http://www.w3.org/2004/02/skos/core#prefLabel.@value");
    }

    @Override
    public QueryBuilder build(final String queryString) {
        return multiMatchQuery(queryString, fields().toArray(new String[] {}))
                .operator(Operator.AND);
    }
}
```



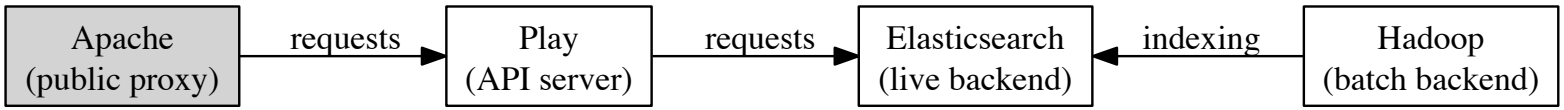
hbz

Technology: documentation

Details on how this works, the actual code and workflows, collaboration infrastructure, etc:

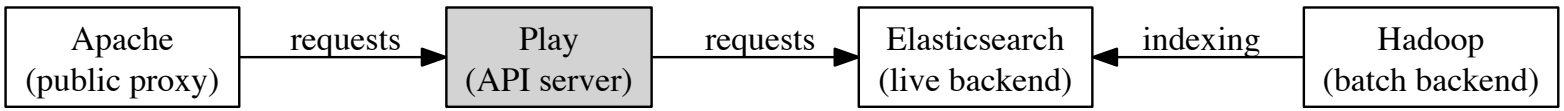
`http://github.com/lobid/lodmill/`

Operations: overview



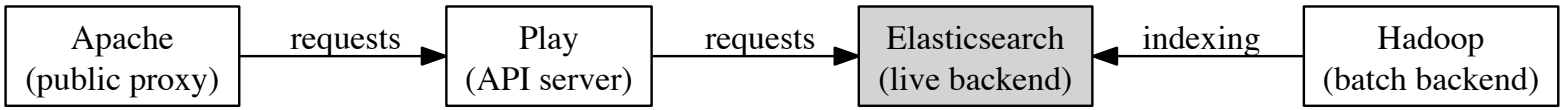
- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet

Operations: overview



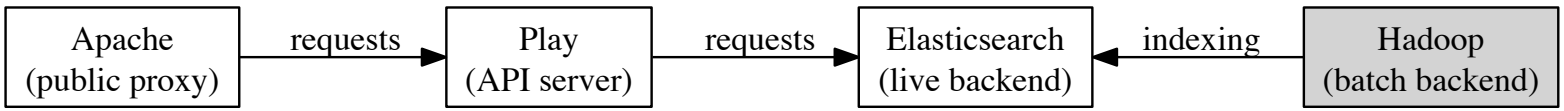
- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet

Operations: overview



- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet

Operations: overview



- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet

Operations: what we like

- Technology stack: config of transformations, queries, views
- JSON-LD, @context
- Data updates without affecting production
- Elasticsearch performance

CC-0, Wikimedia Commons,
File:Expression_of_the_Emotions_Figure_17.png

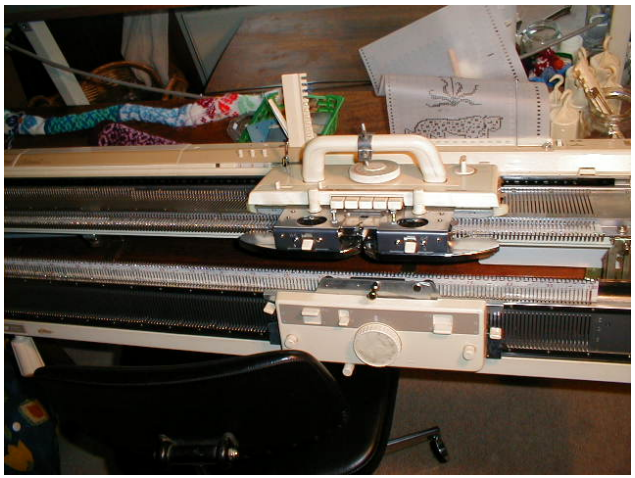
Operations: what we don't like

- Manual deployment, proxy and index switching
- Long feedback cycle for full transformation
- Goal: automation and faster indexing

CC-0, Wikimedia Commons,
File:Expression_of_the_Emotions_Figure_20.png

Operations: summary

So not completely there yet, still some manual work involved, but much more than just the yarn



hbz

Usage

- For progress, usage and feedback is key
- Internal users: e.g. lobid.org, repository cataloging, regional bibliography in 2014
- External users: in contact with various libraries and related institutions

Feedback

- Had early internal reviews, early external beta, got important feedback
- Feedback & iteration crucial: can't guess what's useful, have to find out with users

CC-SA-2.0 lumaxart, Wikimedia Commons, File:Working_Together_Teamwork_Puzzle_Concept.jpg

Openness

- Code, but also processes open: issues, CI, code reviews, wiki on GitHub

  `http://github.com/lobid/`
- Open API:

  `http://api.lobid.org/`
- We're very happy about usage, feedback, contributions on all levels

hbz

Contact

steeg@hbz-nrw.de, @fsteeg
christoph@hbz-nrw.de, @dr0ide

hbz