MPDL-Prototyp: Stand der Technik

Inhalt

1. Gesamteinordnung
2. MPDL query and indexing system with eXist/Lucene
3. Sprachtechnologie: Donatus/Pollux
4. Ausblick/Zukunft

Gesamteinordnung

Was ist das MPDL Teil-Projekt „Content based web access“

- „a content-based access mechanism for these texts that incorporates language technology, which will be built upon the MPDL infrastructure and will be publicly web-accessible“ (aus MPDL_project_desc.pdf)
- offen für neue Projektideen, wenn sie mit den vorhandenen Ressourcen machbar sind

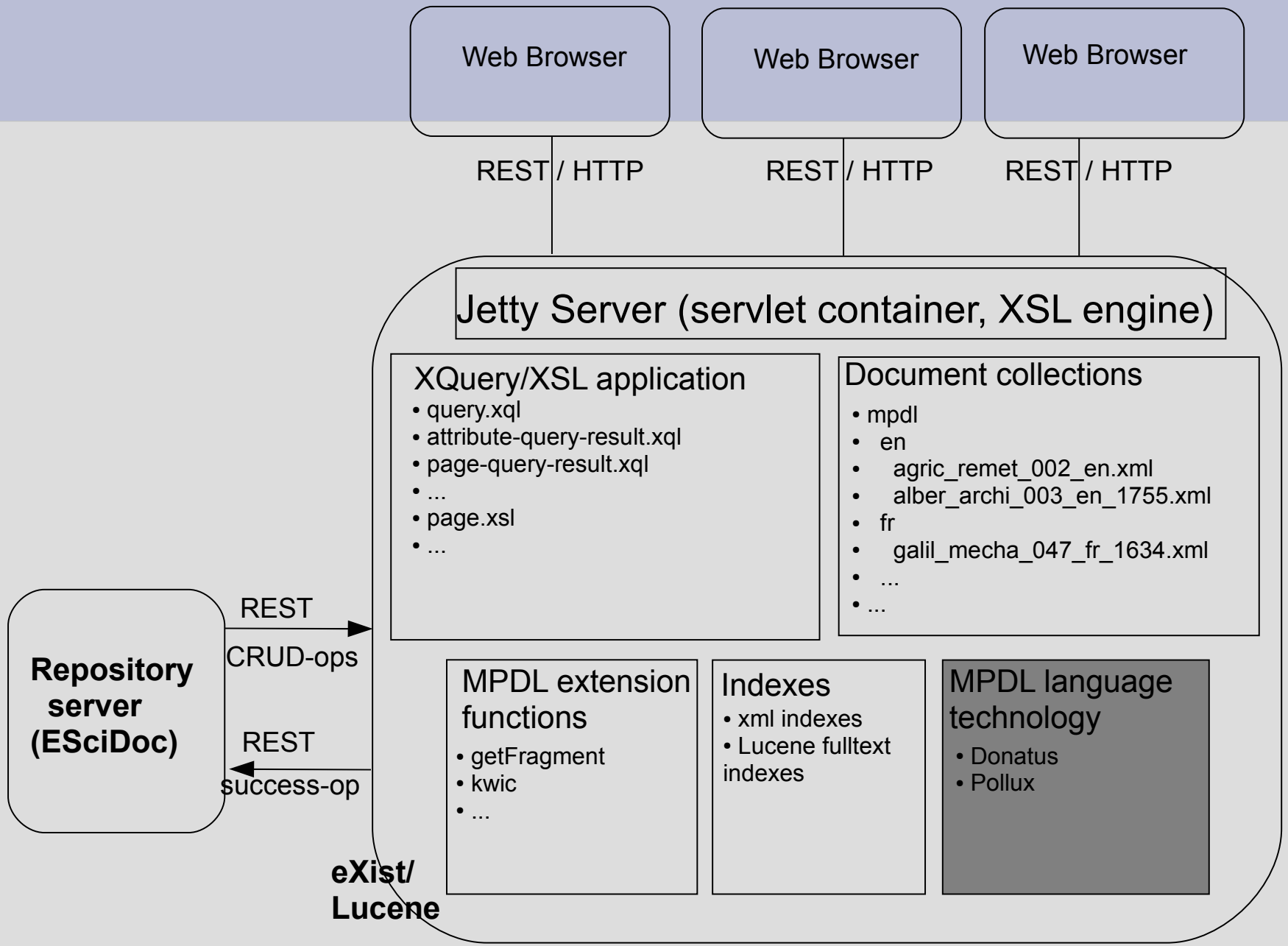
Was ist das MPDL Teil-Projekt „Content based web access“ nicht

- Zentrale für alle mögliche Softwareideen, „die mal eben nebenher entwickelt werden sollen“
- kein Zentrum für Repository-Systementwicklung und -administration
- kein Zentrum für komplexe Redaktionssysteme für Dokumente (Editierclients)

Zwangsweise und kurzfristige Änderung des zentralen Arbeitschwerpunkts Sprachtechnologie

- Sprachtechnologie von Malcolm Hyman/Mark Schiefsky unter archimedes.fas.harvard.edu kann nicht mehr gepflegt und eingesetzt werden
- Sprachtechnologie wird zentral in Berlin als Service vorgehalten (einsetzbar in Clients und Servern, Java-basiert)

Architecture



MPDL prototype
What works (1)

- **present documents**
  - present single book pages (XML fragment)
    - different modes: Text, Text/Pollux, Image, XML
    - fast: special extension of eXist
  - navigate in document: page down/up, goto page
  - integration of digilib
  - enrichment of pages
- **query and indexing system**
  - document bases: archimedes, echo, tei
  - fulltext queries in document bases
    - boolean metadata queries
    - fulltext queries
    - morphological fulltext queries
  - fulltext queries within documents
    - fulltext and morphological queries
    - highlight query results (in pages and sentences)
    - show morphological query expansion (Donatus)
    - jump from sentence hit to page hit
    - browse document index (enriched with links to dictionaries: Pollux, Echo, Wikipedia)
  - full Xpath and Xqueries within documents

MPDL prototype
What works (2)

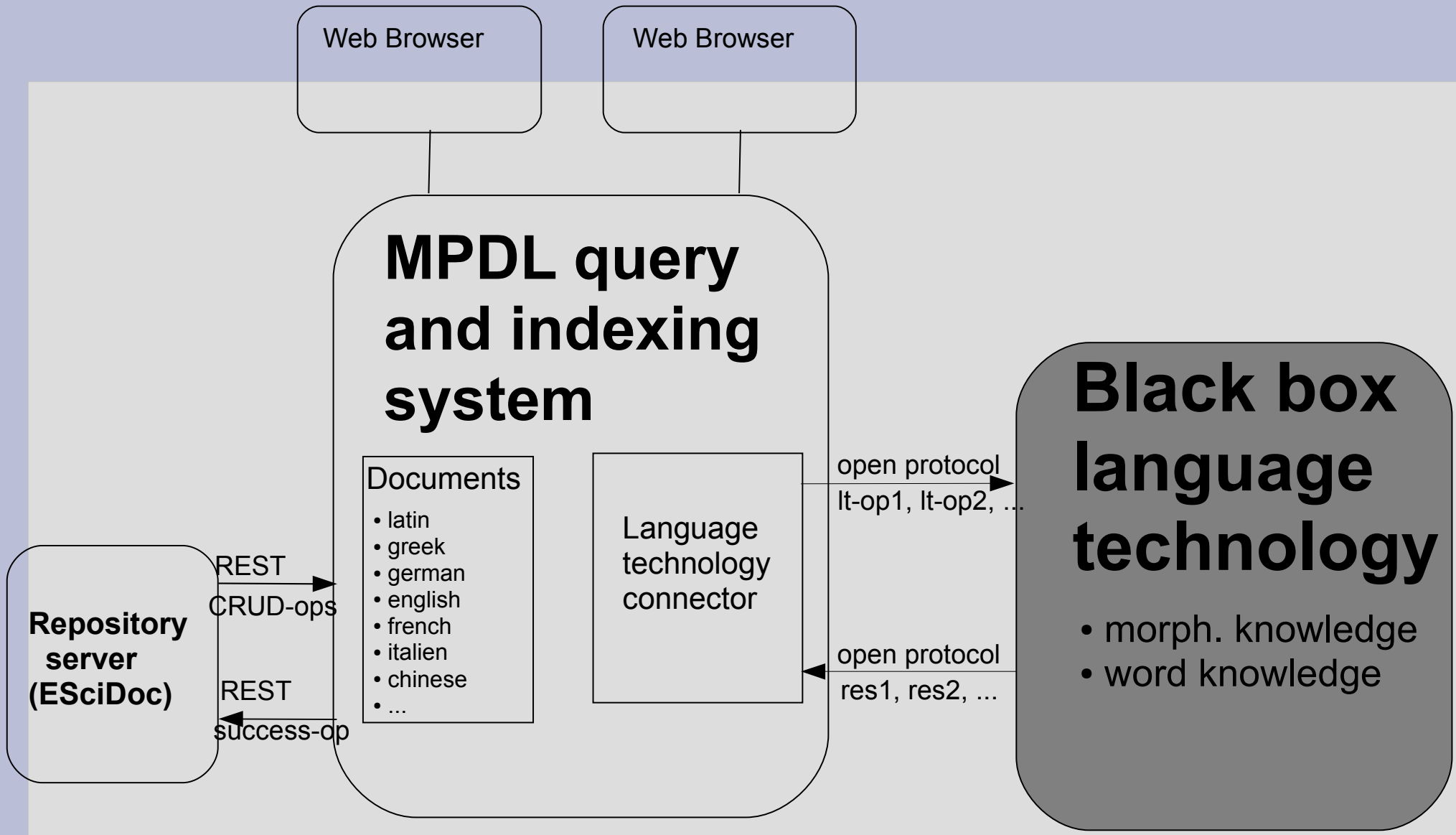
- **Language technology**
  - see later: Donatus and Pollux integration
- **CRUD interface: create, read, update, delete documents**
  - XML-RPC interface
  - REST interface to eSciDoc: first ideas, preparation phase

Donatus / Pollux

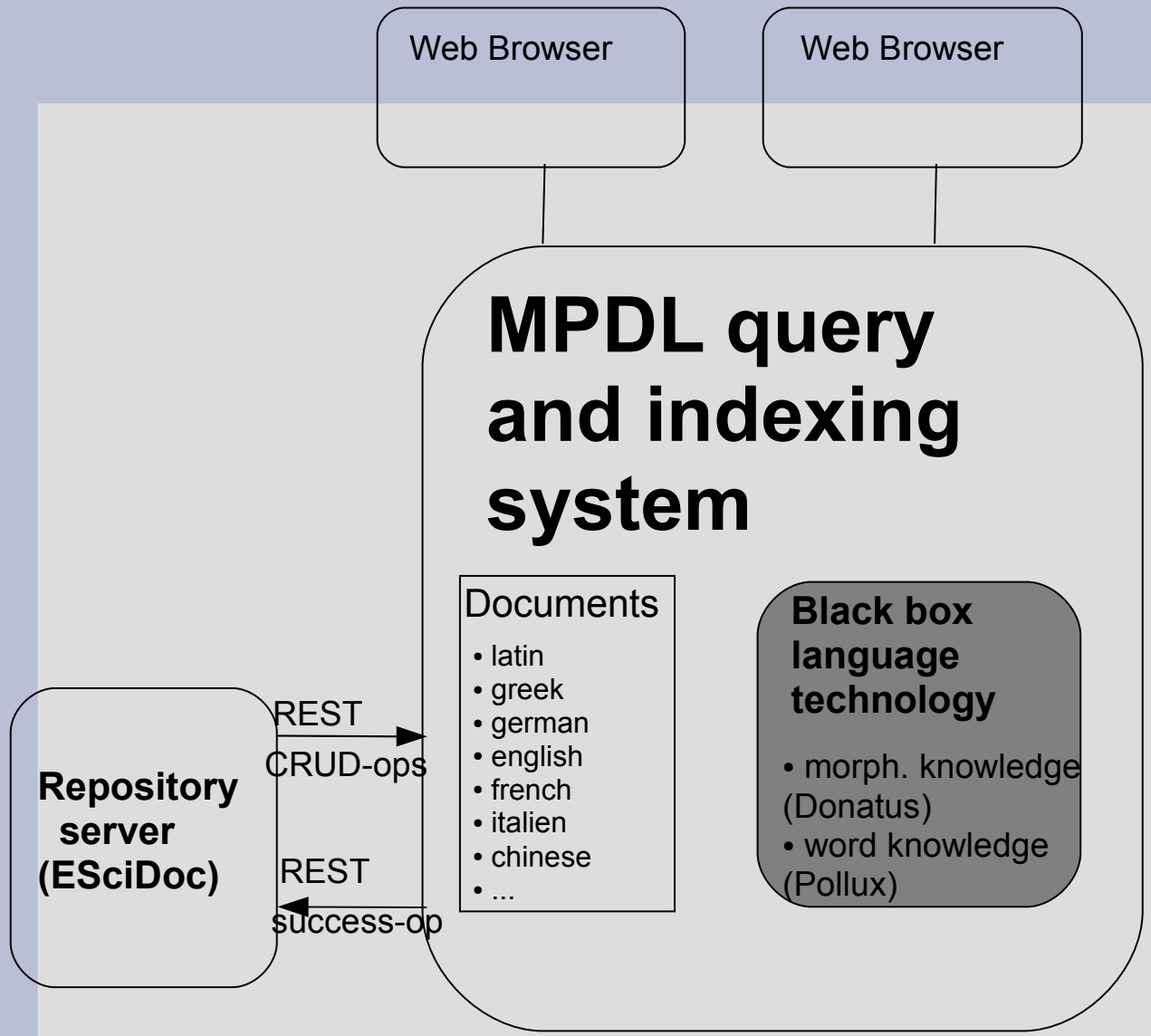
Vorwort

- Dank an Mark Schiefsky (!), ohne den diese Migration nicht möglich gewesen wäre

Architecture old



Architecture new



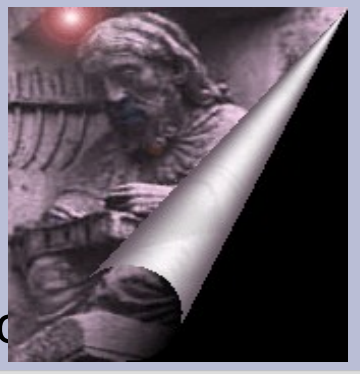
Remark: Language technology is pluggable into clients or servers and is Java based

Donatus

Was ist Malcolms/Marks Donatus Sprachtechnologie (unter: archimedes.fas.harvard.edu)

- morphologische Datenbasis für die Sprachen: arabisch, deutsch, englisch, französisch, griechisch, italienisch, lateinisch, niederländisch
- insgesamt ca. 3 Mio Wordformen mit ihren Lemmas und grammatikalischen Angaben
- Daten können mit einheitlicher Schnittstelle abgefragt und hinzugefügt werden

Donatus



Was ist die MPDL Donatus Sprachtechnologie

- die morphologische Datenbasis wurde komplett migriert
  - Persus: arabisch, griechisch, lateinisch; Celex: deutsch, englisc
    niederländisch; Speziell: französisch, italienisch
  - alle Daten wurden zunächst in ein MPDL XML Zwischenformat konvertiert
  - diese wurden dann in eine BerkeleyDB geladen
  - Betacode und Buckwalter Konvertierung nach Unicode fertig (spezielle Zeichensatzrestprobleme auch fast fertig)
  - Stand insgesamt: weit fertig, best. Bereinigungen fehlen noch (orthographische Normalisierung, Stoppworte, Umlaute, franz. Accents, grammatische Angaben noch nicht für alle Sprachen, ...)
- das Modul ist vollständig und bereits performanceoptimiert für den Einsatz bei der morphologischen Indexierung von Dokumenten und der Suche nach morph. Einträgen implementiert (pure Java) und Indexierungen des Archimedes-Bestands durchgeführt
- Erweiterung des MPDL-Prototypen
  - Anzeige detailierter morph. Information bei der morph. Suche (Trennung nach Datenanbieter wie Perseus, Celex, Snowball)
  - Anzeige von morph. Information in den Wortindizes
  - Web-Schnittstelle mit einheitlicher URL
    - z.B.: http://mpdl-proto.mpiwg-berlin.mpg.de/mpdl/lt/lemma.xql?language=la&form=accedo

Indexing with eXist and the Donatus lemmatizer
Old technology

1. add document to an eXist document collection
2. get Donatus analyzer class for that document
   - defined in an .xconf-file in a document collection
   - language specific: different Java analyzer classes:

   Example:

```
<lucene>
<analyzer class = "de.mpg.mpiwg-berlin.mpdl.lt.DonatusGermanAnalyzer"/>
<text match="//text//*"/>
</lucene>
```

3. analyzer class lemmatizes each word in each text node or text attribute of that document
   - e.g. class DonatusGermanAnalyzer
   - method „stem(String word)“
     - call of DonatusHandler (only one time for performance reasons)
       - prepare the document for Donatus
       - open xml-rpc connection to Donatus Server in Berlin
       - call method „donatus.analyze“ for that document
       - receive the result (all lemmatized words) as an XML document and caches it as <lemma, variant, language> pairs
     - get each lemmatized word via the cached result of the DonatusHandler for that document
4. add index entries: for each lemmatized word: at key add value
   - key: lemmatized word: e.g. professor
   - value: xml documentId + xml nodeId: e.g. 137 3.6.7.4/1

Indexing with eXist and the Donatus lemmatizer
New technology

1. add document to an eXist document collection
2. get Donatus analyzer class for that document
   - defined in an .xconf-file in a document collection
   - language specific: different Java analyzer classes:

   Example:

```
<lucene>
<analyzer class = "de.mpg.mpiwg.berlin.mpdl.lt.analyzer.MpdlAnalyzerDE"/>
<text match="//text//*"/>
</lucene>
```

3. analyzer class lemmatizes each word in each text node or text attribute of that document
   - e.g. class MpdlAnalyzerDE
   - method „stem(String word)“
     - call of MorphologyCache
       - get the stem directly out of BerkeleyDB
       - better performance: forms and lemmas are cached
4. add index entries: for each lemmatized word: at key add value
   - key: lemmatized word: e.g. professor
   - value: xml documentId + xml nodeId: e.g. 137 3.6.7.4/1

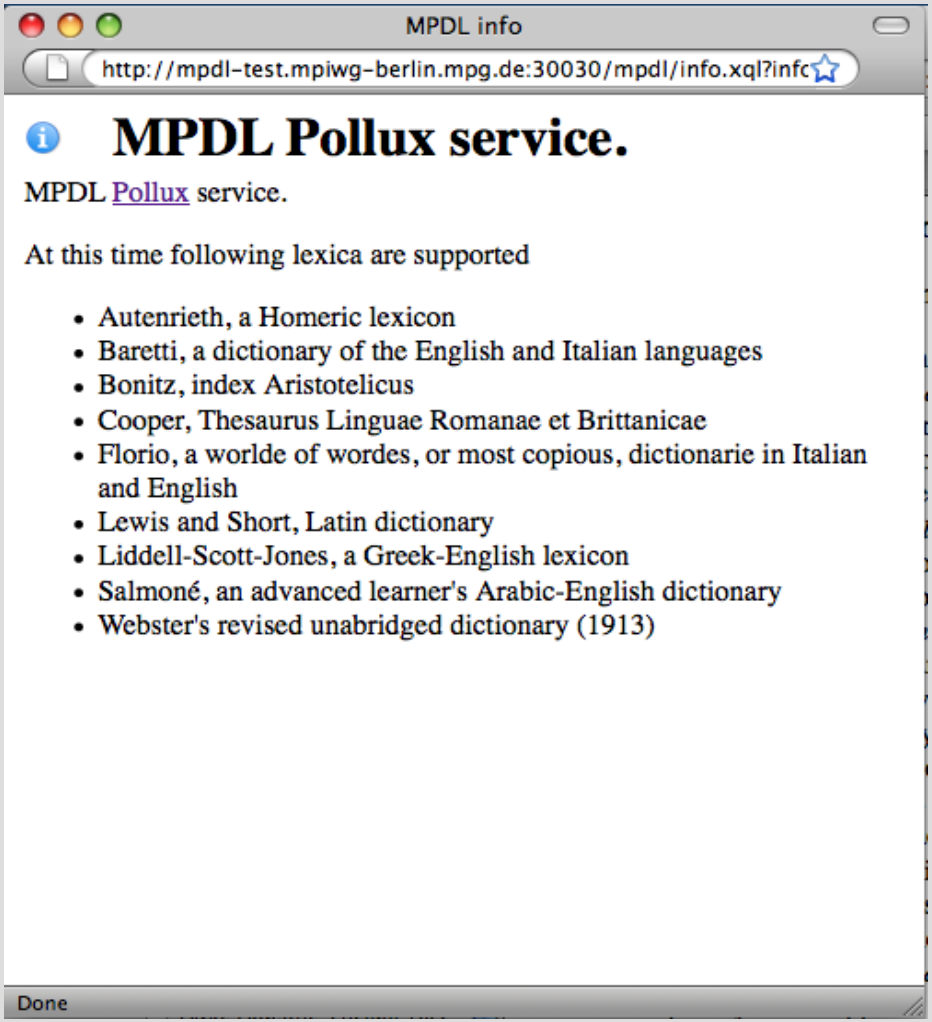
Advantages with Donatus

- no dependency to the network connection and stability of Donatus at archimedes.fas.harvard.edu any more: there were many inconsisties and errors in the indexing process
  - stability old: it needed more than 5 manual try and errors for indexing the archimedes document collection (with intermediate data inconsistencies !)
  - stability new: it is stable, no errors and data inconsistencies are detected any more
- better performance: the fetching of the morphological data is done directly on the same computer without network overhead and parsing etc.
  - performance old (indexing of archimedes collection): needed 5 hours
  - performance new (indexing of archimedes collection): needs 50 minutes
  - performance old (query one entry): needed 3 sec. (estimated)
  - performance new (query one entry): needs 0.01 sec.

Pollux

Was ist Malcolms/Marks Pollux Sprachtechnologie

- neun digitale Wörterbücher/Thesauri als Datenbasis für die Sprachen: arabisch, englisch, griechisch, italienisch, lateinisch



Pollux

Was ist Malcolms/Marks Pollux Sprachtechnologie

- vier Wörterbücher als Schnittstellen zu Wörterbüchern
  - deutsch
    - Wörterbuch der deutschen Gegenwartssprache
    - Deutsches Woerterbuch von Jacob und Wilhelm Grimm
  - französisch
    - Dictionnaire de l'Académie francaise, 6e éd
  - sumerisch
    - ePSD (Pennsylvania Sumerian Dictionary)
- Daten können mit einheitlicher Schnittstelle abgefragt werden

Pollux

Was ist die MPDL Pollux Sprachtechnologie

- die statische Lexika-Datenbasis (neun Lexika: BerkeleyDB) wurde komplett migriert
  - BerkeleyDB-C Datenbank nach BerkeleyDB-Java Datenbank konvertiert
  - Betacode und Buckwalter Konvertierung nach Unicode: Skripte von Malcolm/Mark nach Java gebracht und erweitert, Lauf über alle Daten wird jetzt durchgeführt
- Online-Schnittstelle (vier Lexika)
  - Stand: ToDo
- das Modul ist vollständig und performanceoptimiert für den Einsatz bei der Suche nach lexikalischen Einträgen implementiert (pure Java)
- Erweiterung des MPDL-Prototypen
  - Dokument: Seitendarstellung im Polluxmodus mit Links zu den Pollux-Einträgen (gewonnen über die morph. Datenbasis)
  - Dokument: Index: Link zum Pollux-Eintrag
  - Bei Fehlern in den Originaldaten (HTML-Fehler) wird automatisch die Textversion der Daten angezeigt
  - Web-Schnittstelle mit einheitlicher URL
    - z.B.: http://mpdl-proto.mpiwg-berlin.mpg.de/mpdl/lt/lex.xql?language=la&form=accedo

Advantages with Pollux

- no dependency to the network connection to Donatus at archimedes.fas.harvard.edu any more, better stability
- better performance: the fetching of the lexical entry data is done directly on the same computer without network overhead and parsing etc.
  - performance old: query one entry: needed 1 sec. (estimated)
  - performance new: query one entry: needs 0.03 sec.

Donatus / Pollux

Ausblick / ToDo

Donatus

- Donatus liefert für eine Wortform mehr als ein Lemma und dessen Einsatz bei der Indexierung und bei der Anzeige des morph. Indexes
- Anzeige der gramm. Daten beim morph. Eintrag
- Links zwischen den Einträgen
- ...

Pollux

- Klärung best. Copyrights (DWDS, Grimm, ...)
- Links zwischen den Einträgen
- ...

Ausblick / Zukunft

- **Stand insgesamt**
  - weit fortgeschritten (nach Projektantrag/beschreibung)
  - freie Software
  - Sprachtechnologie: weltweit vorne dran, erweiterbar für die Zukunft
  - bis jetzt stabil (muss aber noch im Dauereinsatz getestet werden)

- **Was fehlt/kommt**

- Fertigstellung und Stabilisierung der Donatus und Pollux Entwicklung
- Anforderungsdefinition: Version 1 des Dokuments: was ist fertig und was soll noch fertig werden (zur Projektlaufzeit)
- Entwicklung des eSciDoc interface (REST)
- Import aller beabsichtigten Dokumente (mit Seitenbildern und Abbildungen)
  - letzte Bereinigung der Archimedes-Dokumente und Bilder
  - MPDL-Dokumente im neuen MPDL-Format „Echo“
- grosse Dokumentenbasis (> 10.000): Test: auf neue Version von eXist warten
- ...
- Verlängerung/Institutionalisierung des Projekts
  - Diskussion und Definition der beabsichtigten Workflows: Publish, ...
  - Scholary workbench (Client): Dokumenteditor, weitere Werkzeuge