WebLicht: Web-based LRT services for German

Overview

- Current Situation
- Service Oriented Architectures (SOA)
- WebLicht: Services, Repository and User Interface
- Data Formats
- Live Presentation
- Summary

Current Situation

- Many linguistic resources (corpora, dictionaries, …) and tools (tokenizer, tagger, parser, …) are available
- Most of them are implemented to run on local machines. This can be inconvenient and error-prone ….
- …on the user side:
  - Every potential user has to download and install them on his own machine: this may cause problems with operating systems, compiler versions, missing libraries, …
  - Keep an eye on updates, (security) patches, new versions etc.

Current Situation

- … on the developers side:
  - How to publish LRT?
  - Question of license, user permissions, …
  - Combination and comparison with other tools/resources
  - Sustainability, long term support

One Possible Solution

- Make LRT available on the web! –

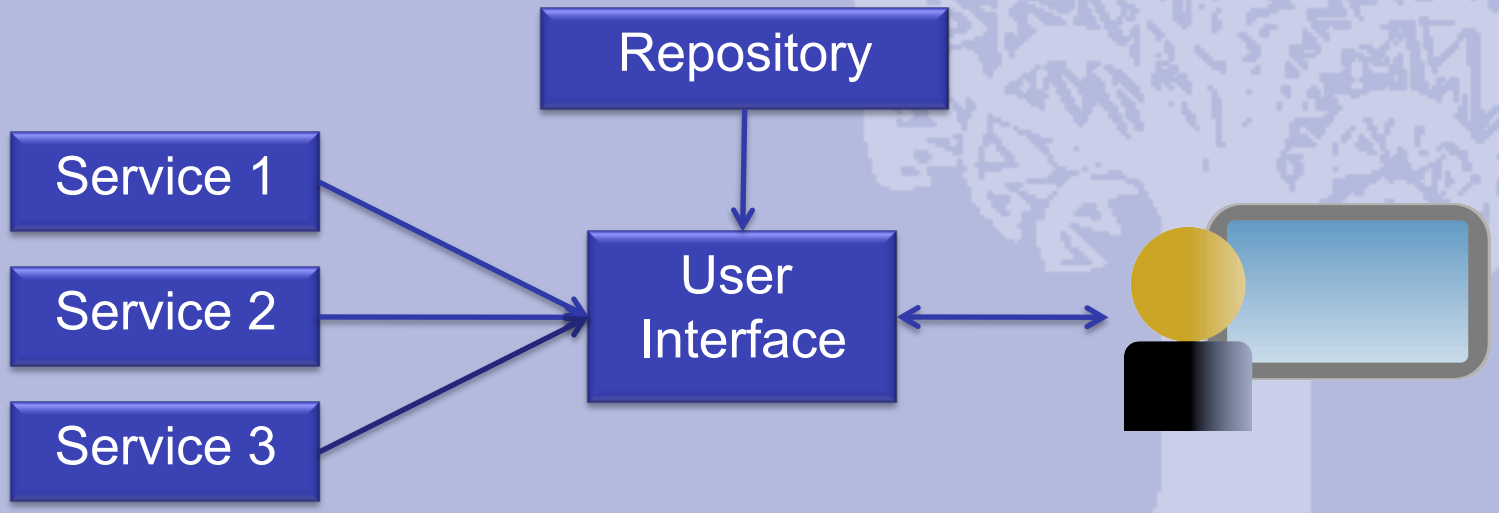
- For some kinds of LRT, its easy to put them online (make resources downloadable, offer search engines etc.)
- For other kinds, more effort is necessary (limiting access to resources, how to make tools online usable)

➔ Solution: a ***Service Oriented Architecure*** **(SOA)**

Service Oriented Architectures

- Components of a SOA
  - ***Distributed Services*:** offering functionality (resources & tools) over the (inter-)net. Mostly implemented as webservices
  - ***Repository*:** stores metadata and technical information about the services
  - ***User interface*:** interacts with the user and combines services and information from the repository

Service Oriented Architecture



WebLicht

- WebLicht: a Service Oriented Architecure for creating annotated textcorpora
- Work started in October 2008
- Participants (September 2009):
  - BBAW
  - IMS Stuttgart
  - ASV Leipzig
  - SfS Tübingen
  - IDS Mannheim

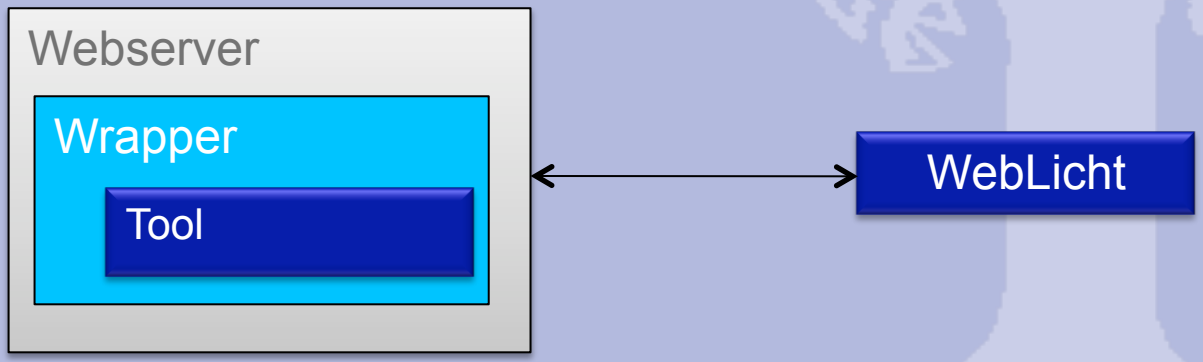
WebLicht: Services

- Services are implemented as REST style webservices
- HTTPs POST method is used to send data from the UI to the services
- As client, *anything* which is able to use the HTTP protocol, can be used:
  - Browser
  - Commandline tools (wget, curl)
  - Programming Languages

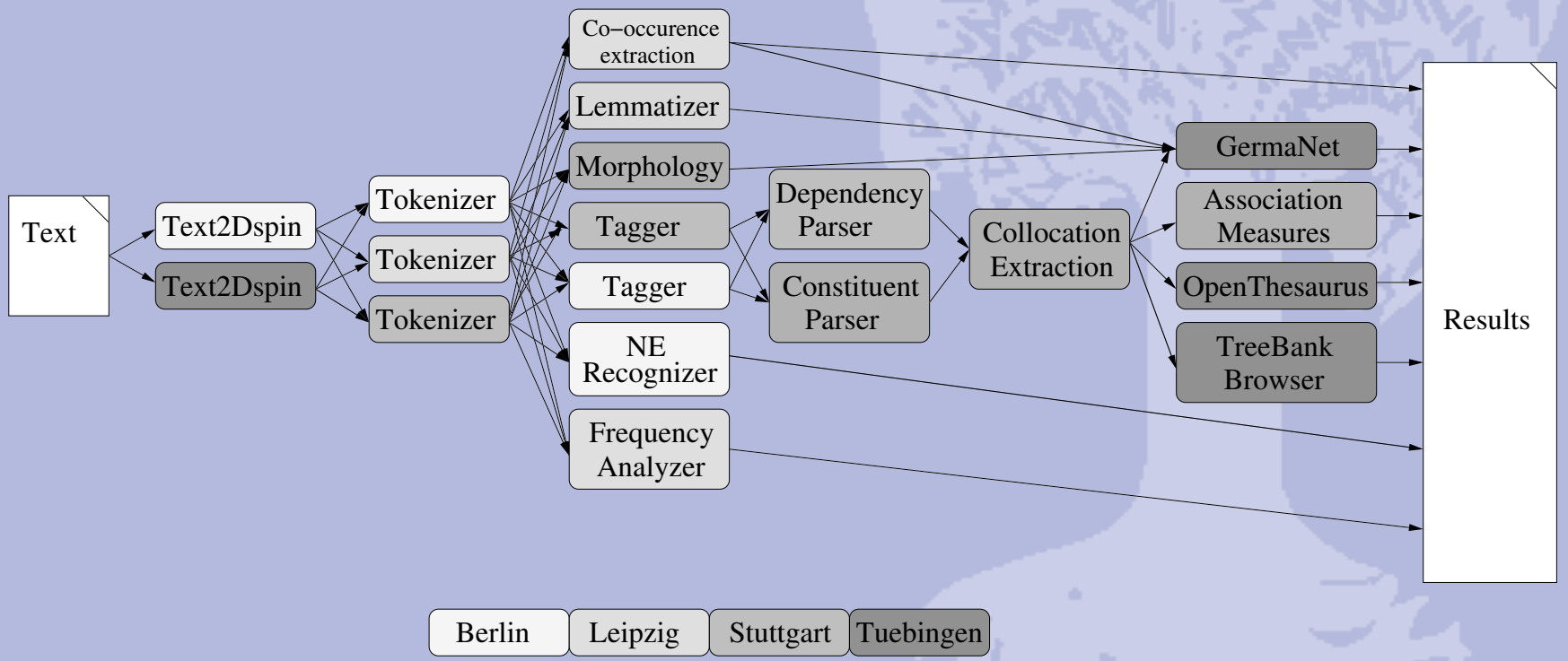
➔ Anyone can implement his/her own interface to WebLicht

WebLicht: Services

- Existing commandline applications are encapsulated into wrappers



WebLicht: Available Services



WebLicht: The Repository

- The repository is implemented at the ASV Leipzig
- It offers information and a query engine for the services:
  - Which services are available?
  - How can I combine them?
  - Which input/output format does a service accept/produce?
- Example: a tokenizer is already applied to a plain text, which services can be used next?

WebLicht: The User Interface

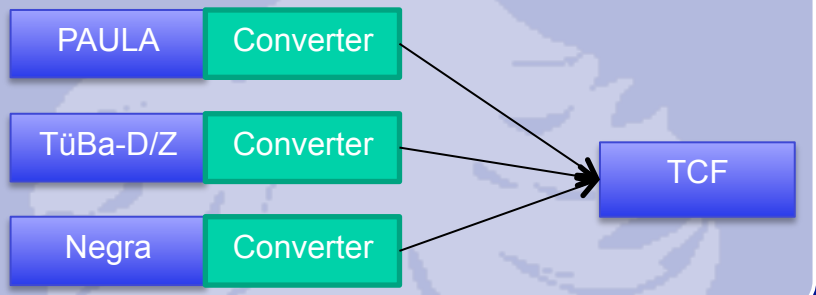
- Implemented at the SfS Tübingen
- Webapplication in Java, deployed in Apache Tomcat
- Allows the user to
  - upload a plain text (soon: also upload of MS Word, RTF and PDF files)
  - construct a text from corpora in Leipzig
  - use some hardwired example texts
- Executes the tool chain and presents the results
- During the chaining process, it queries the repository for available services

WebLicht: The D-Spin Dataformats

- Developed by the D-Spin team
- Stand-off, different annotation layers are stored in one file
- UTF-8 encoded XML format, validated with XML Schema, NG Relax and Schematron
- Different formats (TextCorpus, Lexicon and Metadata)
- WebLicht makes use of TextCorpus format (TCF)

WebLicht: The D-Spin Dataformats

- TCF strives to be compatible with established standards, especially the dataformats of the ISO/TC 37-SC4 group:
  - LAF: Linguistic Annotation Framework
  - LMF: Lexical Markup Framework
  - MAF: Morpho-Syntactic Annotation Framework
- At the moment, converters are available for:
  - PAULA
  - Negra
  - TüBa-D/Z



WebLicht: The D-Spin Dataformats

- Disadvantages of using XML formats: high demand on computer memory and CPU power
- With the help of *Data Binding Frameworks* (JibX), the D-Spin dataformats can be converted back to *Plain Old Java Objects* or objects other programming languages

```
aus dem Hause jagte. Nachdem sie lange unstet in der Welt umhergeirrt, und da niemand eine
Person, die Schlangen und Kröten sprach, bei sich aufnehmen wollte, ging sie im wilden Walde
jämmerlich zugrunde.</text>
<tokens>
 <token ID="t2">Es</token>
 <token ID="t3">war</token>
 <token ID="t4">einmal</token>
 <token ID="t5">eine</token>
 <token ID="t6">Witwe</token>
 <token ID="t7">,</token>
 <token ID="t8">die</token>
 <token ID="t9">hatte</token>
 <token ID="t10">zwei</token>
 <token ID="t11">Töchter</token>
 <token ID="t12">.</token>
 <token ID="t15">Die</token>
 <token ID="t16">ältere</token>
 <token ID="t17">glich</token>
 <token ID="t18">an</token>
 <token ID="t19">Gesicht</token>
 <token ID="t20">und</token>
 <token ID="t21">Charakter</token>
 <token ID="t22">so</token>
 <token ID="t23">sehr</token>
 <token ID="t24">ihrer</token>
 <token ID="t25">Mutter</token>
 <token ID="t26">,</token>
 <token ID="t27">daß</token>
 <token ID="t28">man</token>
 <token ID="t29">sogleich</token>
```

Live Presentation ....

Summary

- Pros and cons of a webservice based SOA:

Pros	Cons
+ Flexible architecture for making resources and tools available over the net	- Server/services can be offline
+ Easy integration of further services	- Not applicable for huge amounts of data
+ User friendly: no local installation etc.	
+ Central administration of services	
+ Combination with PID services and authentification (DRM)	
+ Compatible with other applications by the use of standards / converters	

Future Work

- Add more services (in Tübingen: implementation of a webservice for an online HPSG grammar)
- A webcrawler to compile dynamic textcorpora direct from the internet
- Already available in an early stage: a platform for (quantitative) analysis of the new created textcorpora (DCA)

Links

- WebLicht: http://clarin.sfs.uni-tuebingen.de:8080/WebLicht0/
- D-Spin webpage: http://www.sfs.uni-tuebingen.de/dspin/

Thank you!