EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Seminar für Sprachwissenschaft

WebLicht-Batch – A Web-Based Interface for Batch Processing Large Input with the WebLicht Workflow Engine

Claus Zinn and Ben Campbell

CLARIN 2022, Prague

CLARIN

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

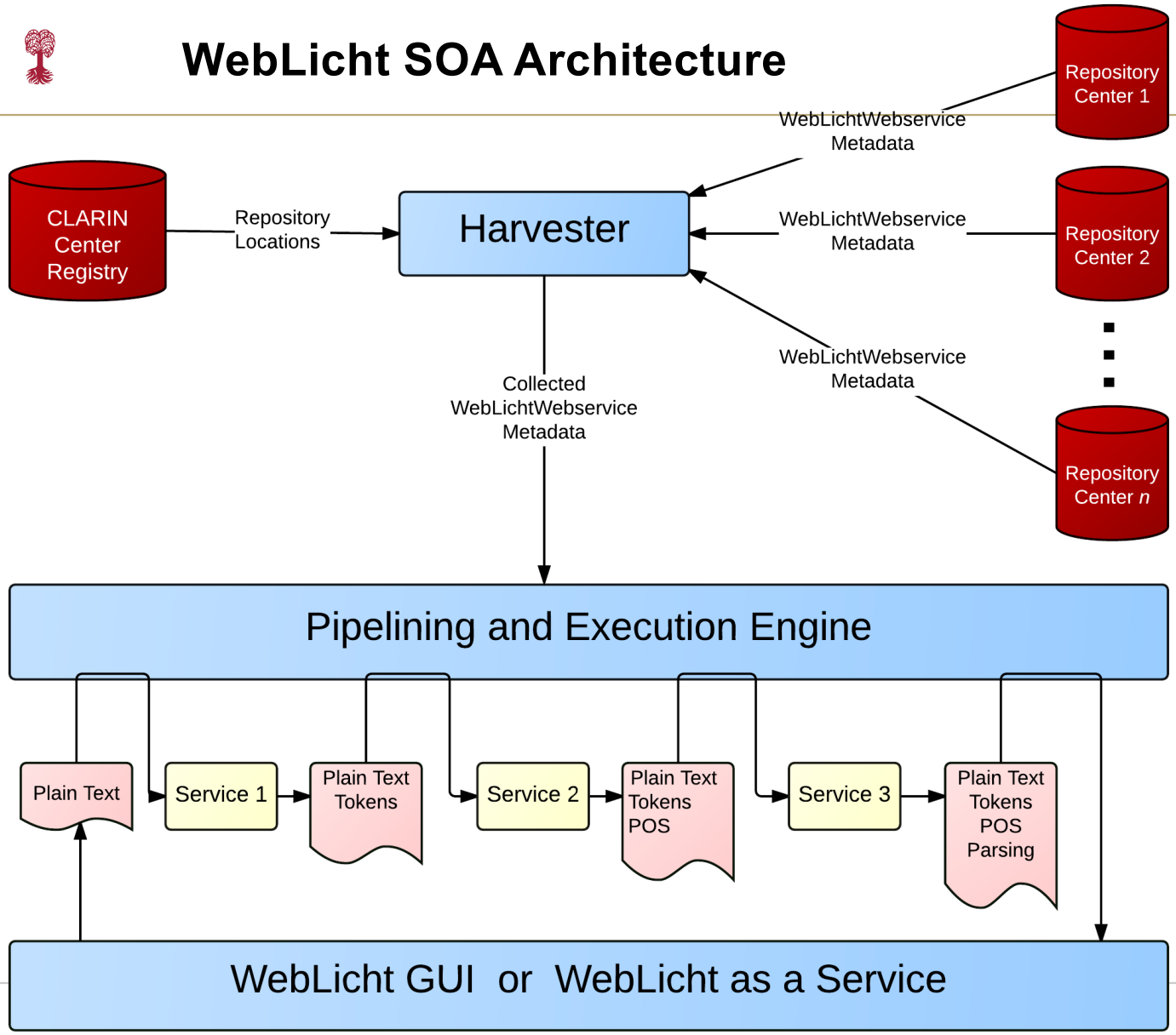
Motivation

- WebLicht, a widely-used, web-based tool to create and execute tools chains for linguistic analysis
- *Reigns over dynamic, distributed* tool space, each tool with varying capabilities to process larger bits of data
- Ensure smooth, fair use, avoid time-outs

➢ Support required to allow use of WebLicht with “big” data
➢ and also, support for users with multiple file input (batch processing)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

WebLicht SOA Architecture



Dynamic tool space

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

TCF format (example)

```
<md:MetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:cmd="http://www.clarin.eu/cmd/"
<TextCorpus xmlns="http://www.dspin.de/data/textcorpus" lang="en">
  <tc:text xmlns:tc="http://www.dspin.de/data/textcorpus">YOU don't know about me without you have
    read a book by the name of The Adventures of Tom Sawyer; but that ain't no matter.</tc:text>
  <tc:tokens xmlns:tc="http://www.dspin.de/data/textcorpus">
    <tc:token ID="t_0">YOU</tc:token>
    <tc:token ID="t_1">do</tc:token>
    ...
    <tc:token ID="t_27">matter</tc:token>
    <tc:token ID="t_28">.</tc:token>
  </tc:tokens>
  <tc:sentences xmlns:tc="http://www.dspin.de/data/textcorpus">
    <tc:sentence tokenIDs="t_0 t_1 ... t_27 t_28"/>
  </tc:sentences>
  <tc:POStags xmlns:tc="http://www.dspin.de/data/textcorpus" tagset="penntb">
    <tc:tag tokenIDs="t_0">PRP</tc:tag>
    <tc:tag tokenIDs="t_1">VBP</tc:tag>
    ...
    <tc:tag tokenIDs="t_27">NN</tc:tag>
    <tc:tag tokenIDs="t_28">.</tc:tag>
  </tc:POStags>
```

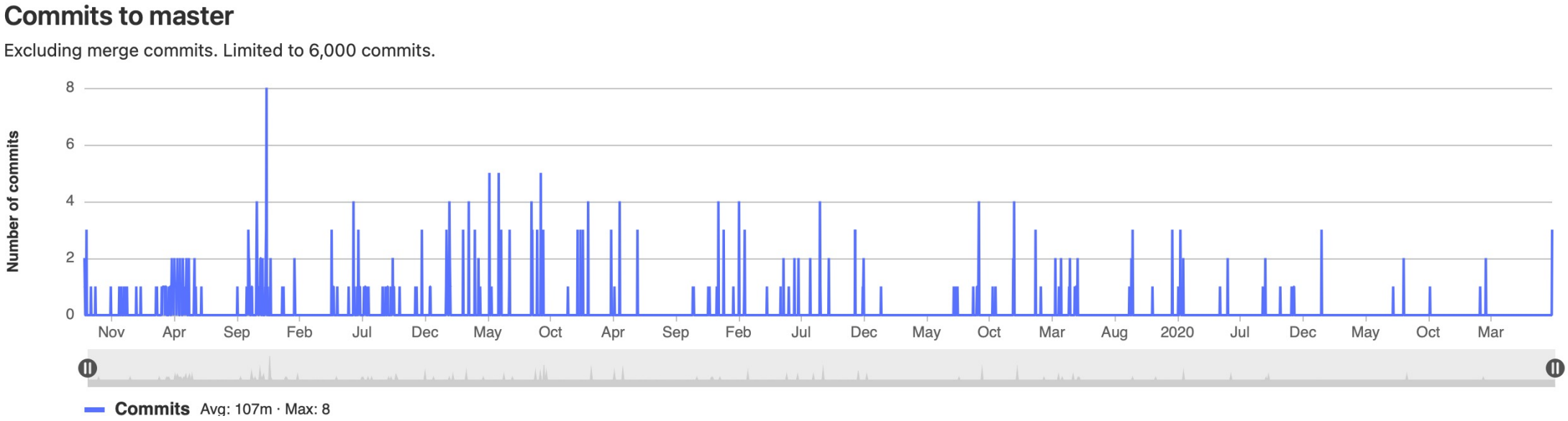
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

WebLicht History & Impact

- Since its inception, WebLicht has attracted a large user community in research & teaching
- Since 2014, WebLicht services have been invoked 2.5 million times in total,
  - with an average of 30.000 invocations per month
  - 72.000 unique addresses

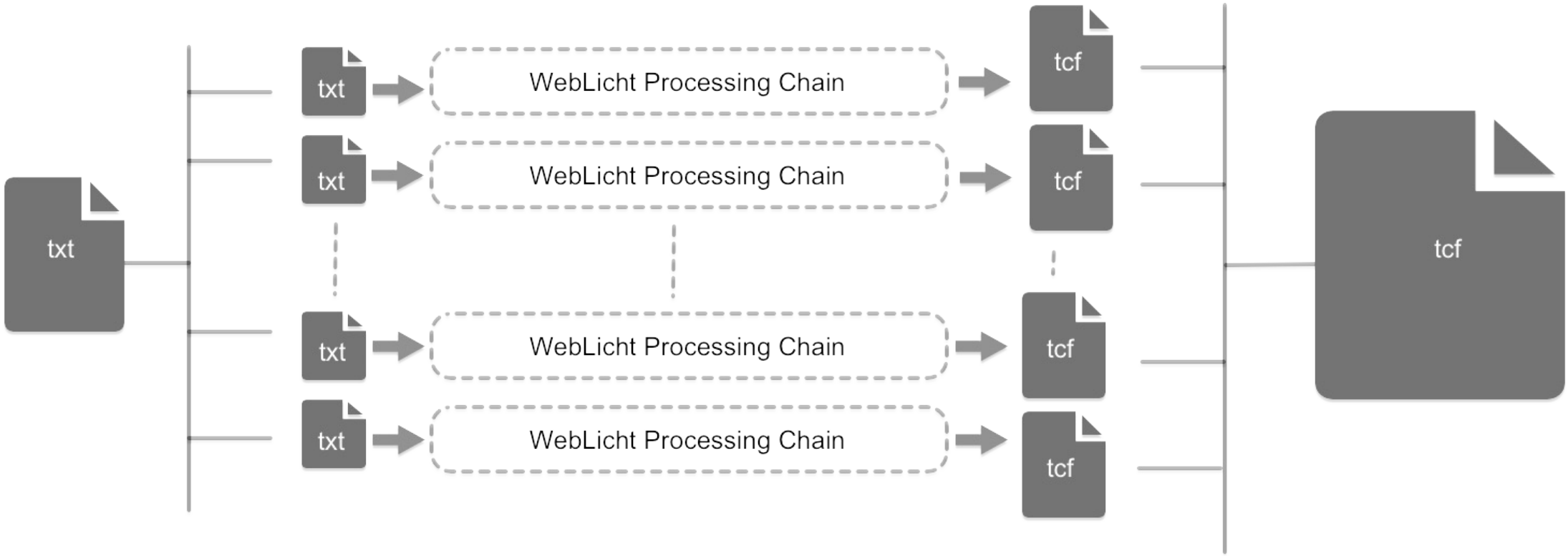
➢ Usability and performance matter, dynamic & distributed tool space makes it hard(er)

  - not all tools cope well with large input



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Underlying Idea: Divide and conquer



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Technical Details (back-end)

- split input file into chunks of 100kb
- splitting at sentence boundaries creates chicken-and-egg problem
- ➢ use of UDPipe tokeniser and sentence splitter
  - send 100kb to UDPipe
  - assume that last sentence is incomplete sentence
  - add it to the beginning of next chunk

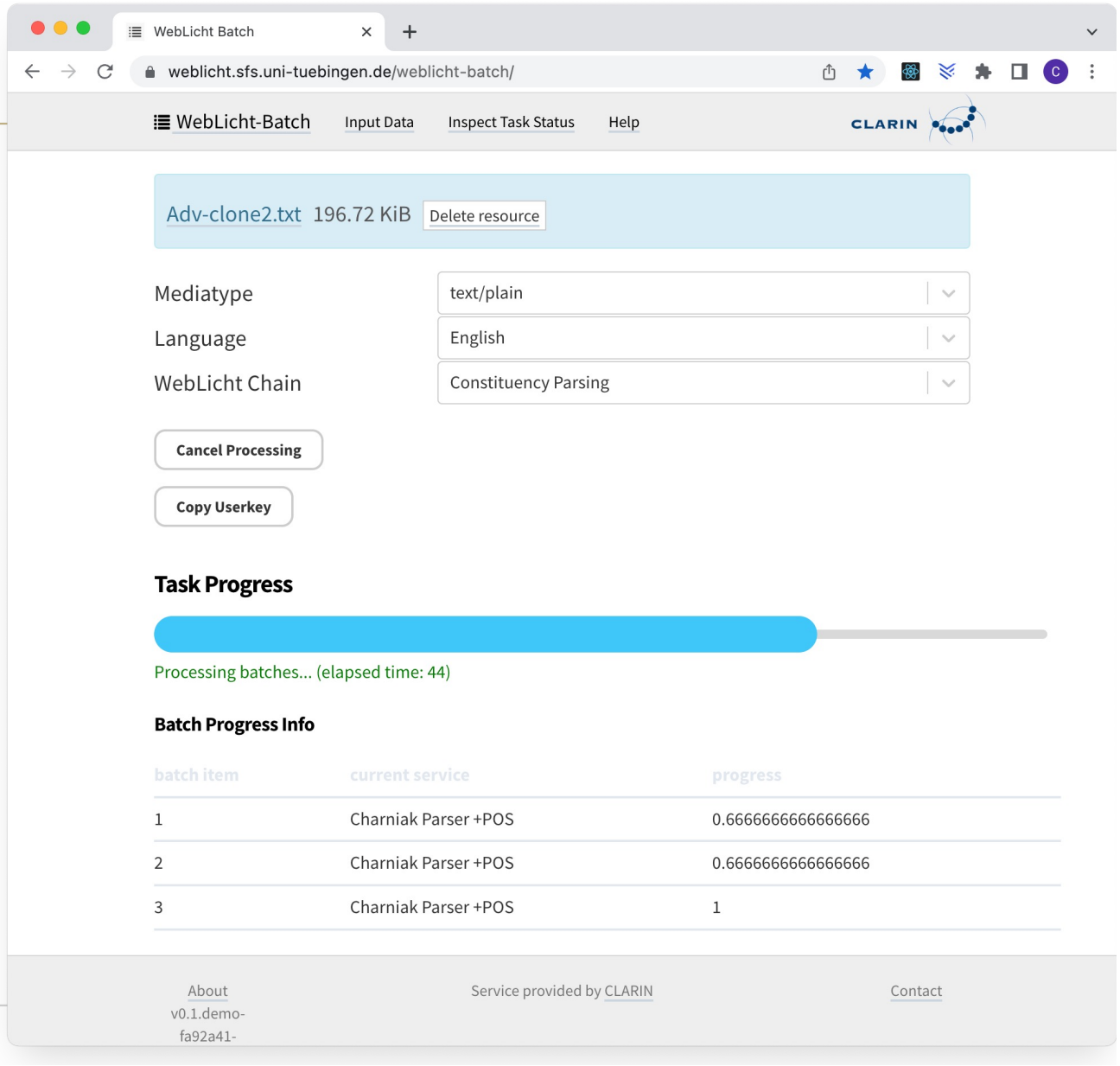
- for all chunks obtained, batch-process 4 of them “simultaneously”:
  - if processing of a chunk failed, 2 other attempts are made
  - three failures in a row, batch fails
- if all batches succeed, resulting output TCFs are recombined into complex TCF
  - TCF of individual runs must be manipulated to ensure all token id *etc*. are correct.
- for a zip file of text files, this process is repeated for each file

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Front-end

1. Upload file
2. Select language
3. Select easy-chain
    - or upload your own chain

- 200kb file split into batch of three
- Processing information updated
- User key for copy-paste
    - to check progress later
    - even after closed browser



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Discussion

- Use of WaaS rather than WebLicht-GUI
  - for sets of small files, some users (or use cases) may prefer WaaS
  - for a large file, WaaS delegates file splitting at sentence boundaries and combination of individual TCF files into a compound TCF to its users
  - this is a non-trivial task
- WaaS does not help when a tool is crashing on larger input
- **Divide and conquer** seems like the only feasible approach
  - given the dynamic and distributed tool space

- EUDAT-Generic Execution Framework
  - move tools to the data rather than the data to the tools
  - configure GEF environment of tools with direct access to the data

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Future Work

- most services that are part of WebLicht's easy-chains are installed locally at institutional servers using Docker technology.
- ➢ use Docker to spawn new workers of a given service *on the fly* giving a rising demand from WebLicht-Batch users

- a large WebLicht-Batch process could block regular WebLicht GUI users from getting their (smallish) input processed in time.
- ➢ batch processing may want to postpone heavy processing to a point in time where Docker-based services are idle
- ➢ scheduling option for users? Give users time slots where processing is faster

- often one bottleneck service per easychain
- ➢ give complex services more CPU time than simpler ones

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

WebLicht-Batch -- Demonstration

- https://weblicht.sfs.uni-tuebingen.de/weblicht-batch/
- Feedback welcome, email claus.zinn@uni-tuebingen.de for feedback

Questions?