Indexing: Personal Cloud Search

Web Development Seminar SS 2016
Software Architecture Group

Felix Wolff
Daniel Werner

Context

Lively is used in various different ways:

- To prototype ideas
- To develop applications and tools
- To organize and access private files
- …

It provides the capability to mount several cloud file systems.

Very complex environment, would benefit greatly from a fast search.

IT Systems Engineering | Universität Potsdam

Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

?? (Challenges)

- Hard to search and find private content and code
- Cannot use public search services (Google, …) for private data
- Data is usually distributed among multiple online services (Github, Dropbox, …)

IT Systems Engineering | Universität Potsdam

Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Goals

A

We want **instant search** across **all mounted files.**

- Regardless of file system source
- Github, Server, Dropbox, (OneDrive, Google Drive)

B

We want to keep **private** data **private.**

- Do not upload data to a 3rd-party service

Approach

Ideally every cloud service provides a full text search via API.

- Github provides search API
- Dropbox only for business customers
- Lively server used grep so far

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Software Architecture Group

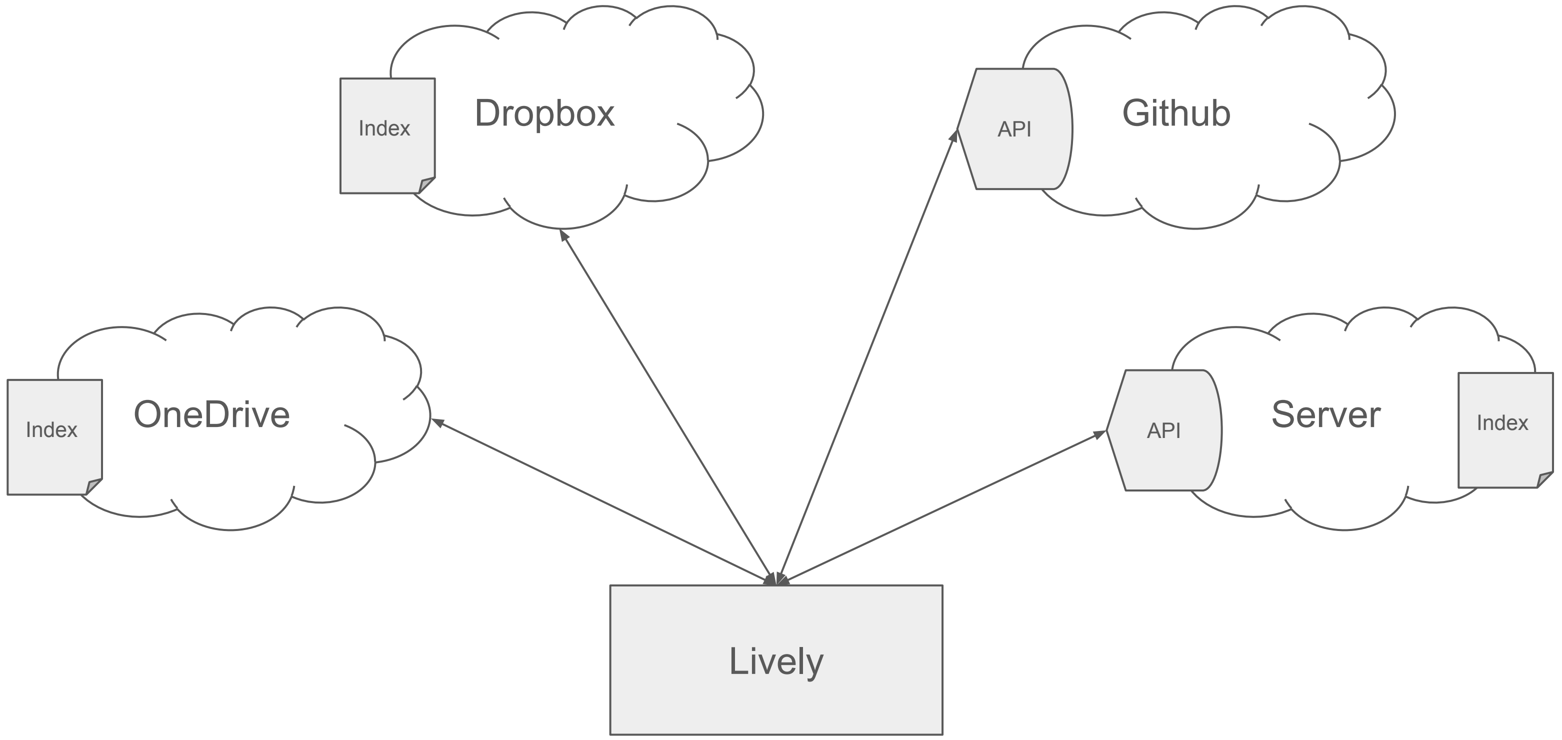
Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Architecture



Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Index Design Decision

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Create a separate index for each mount point:

- Search needs to combine results from different indexes
- Index is stored together with data
- Paths in index remain valid when re-mounting
- Shared dropbox folders can share one index

Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Search Engine

search-index (github.com/fergiemcdowall/search-index) 631★ :

- Full text (decentralised) search engine
- Depends on LevelDB backend

Lunr.js (github.com/olivernn/lunr.js) 3527★:

- Full text search engine for client-side applications
- Small and light-weight
- No need for a server-side search service

Excursion: Inverted index

document1: the brown fox jumps
document2: over the brown fence

```
{
     brown: [document1, document2],
     fence: [document2],
     fox: [document1],
     ...
}
```

- Scoring mechanism:
  - **TF-IDF:** (query frequency in document) / (#documents containing query)

Software
Architecture Group

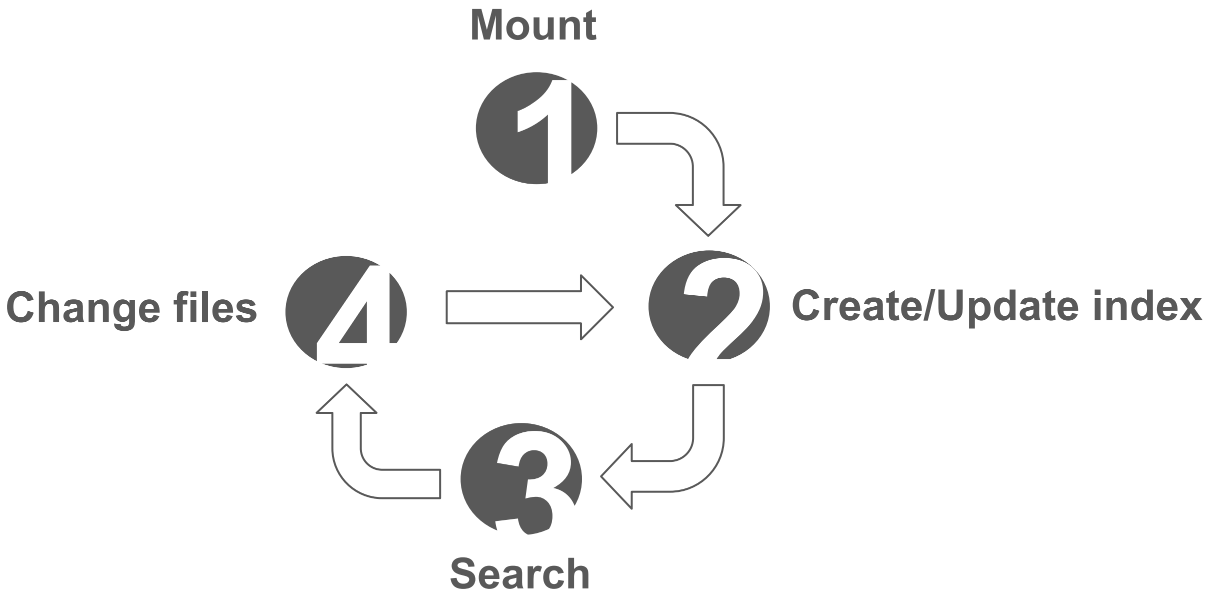
Web Development
Seminar 2016

Topic: Personal
Cloud Search

2016-07-13

Lixissimus
daniel-wer

Workflow



Software
Architecture Group

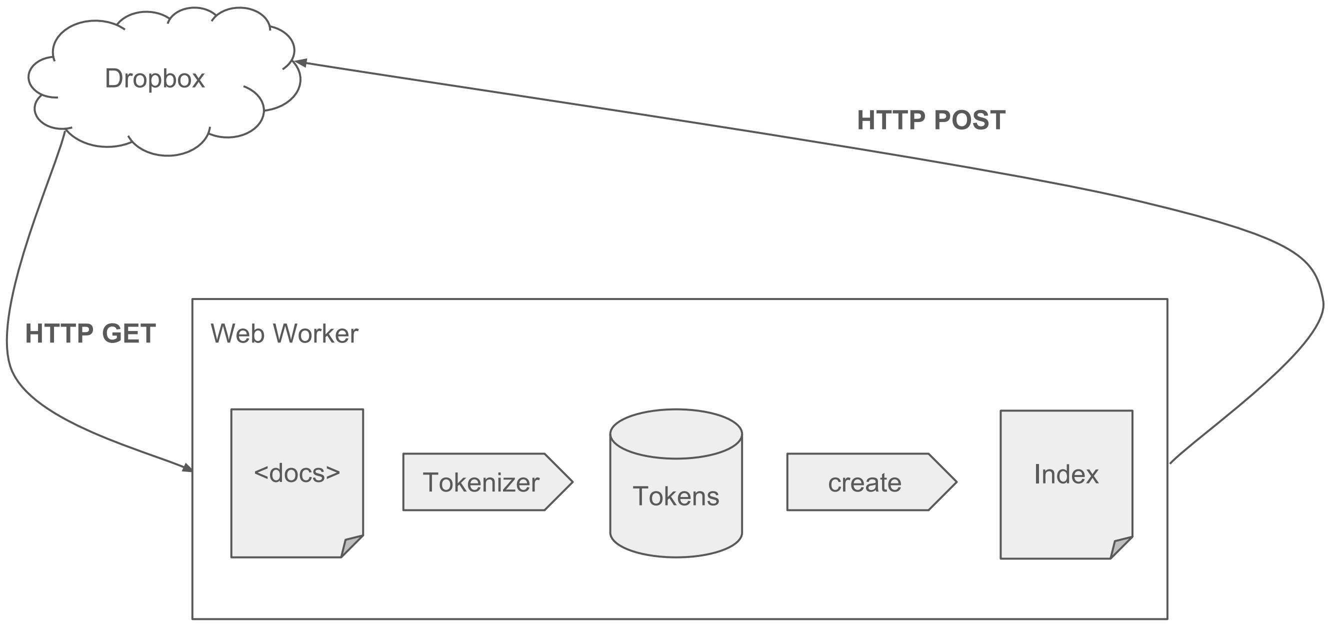
Web Development
Seminar 2016

Topic: Personal
Cloud Search

2016-07-13

Lixissimus
daniel-wer

Indexing a Dropbox



* Only index files with certain endings: .js, .html, .md, .txt
** Only index files that are smaller than 500kb

Software Architecture Group

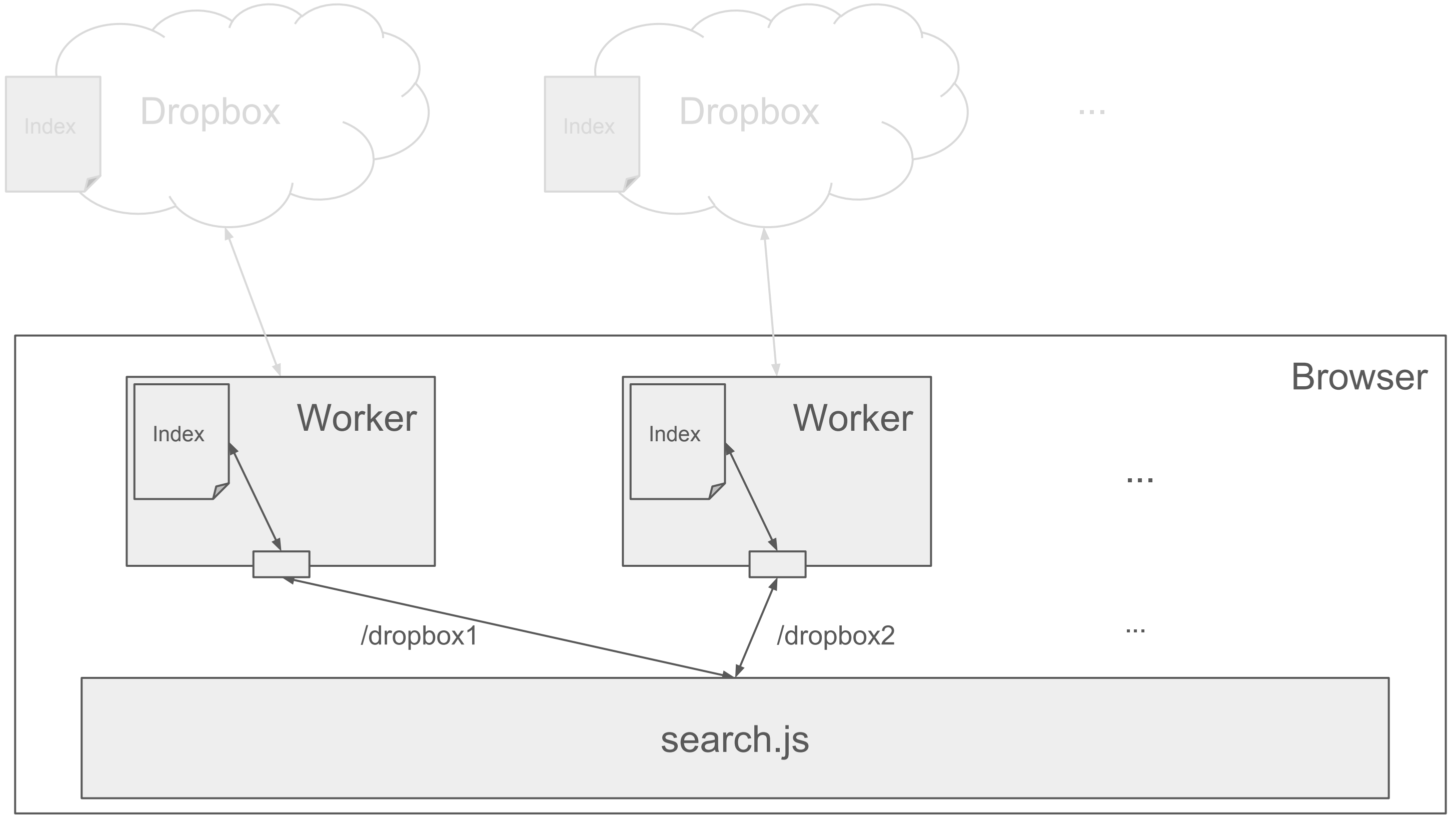
Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Searching in Dropboxes



Software
Architecture Group

Web Development
Seminar 2016

Topic: Personal
Cloud Search

2016-07-13

Lixissimus
daniel-wer

Doku: Searching in Dropboxes

IT Systems Engineering | Universität Potsdam

- User opens search bar and searches for “foo”
- The search module knows which mount points are indexed and has those indexes loaded into the browser
- Each indexed mount point has a dedicated worker thread that manages the index
- Ask each worker for search results for “foo”
- Combine search results from all workers
- Display search results

Software Architecture Group

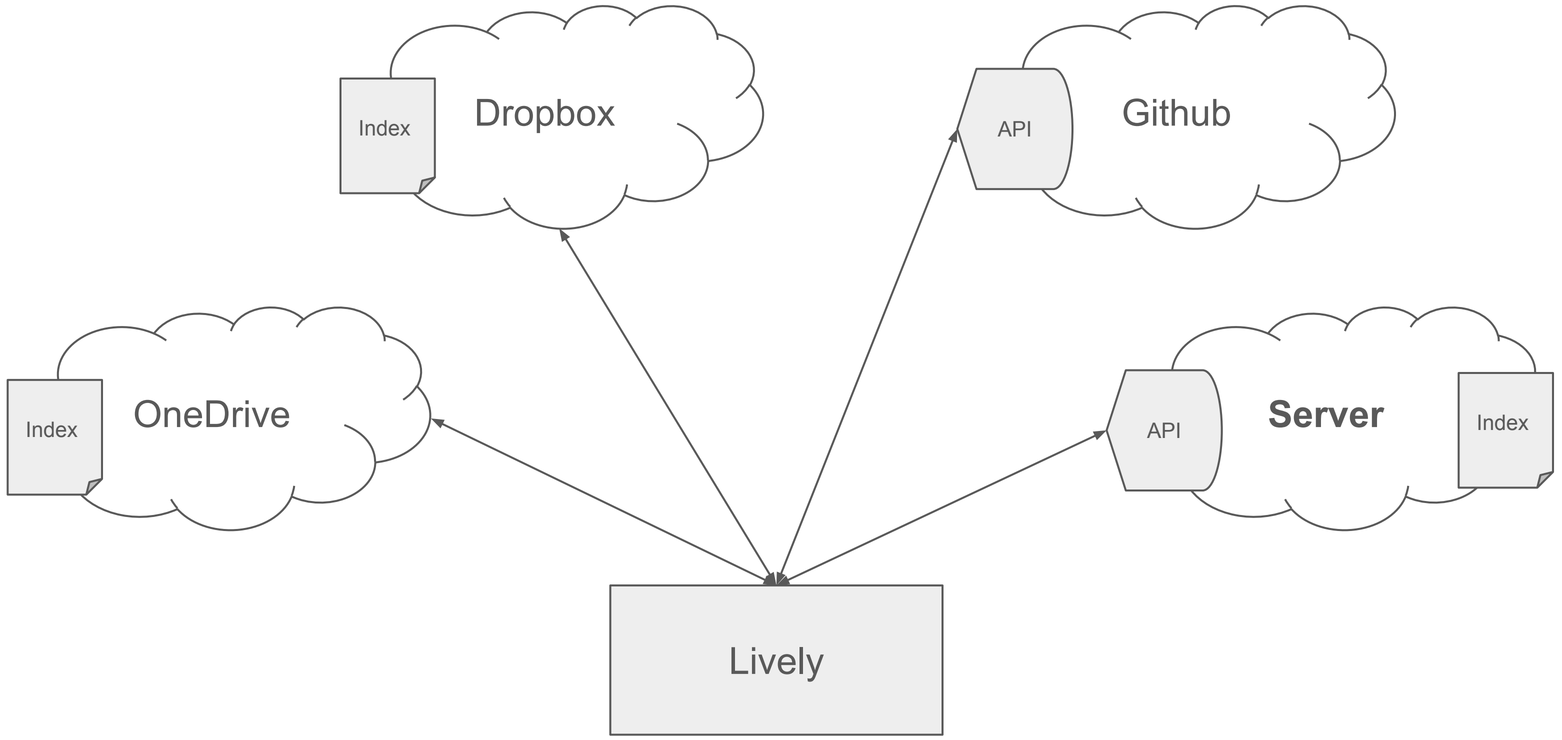
Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Indexing the Server



Software Architecture Group

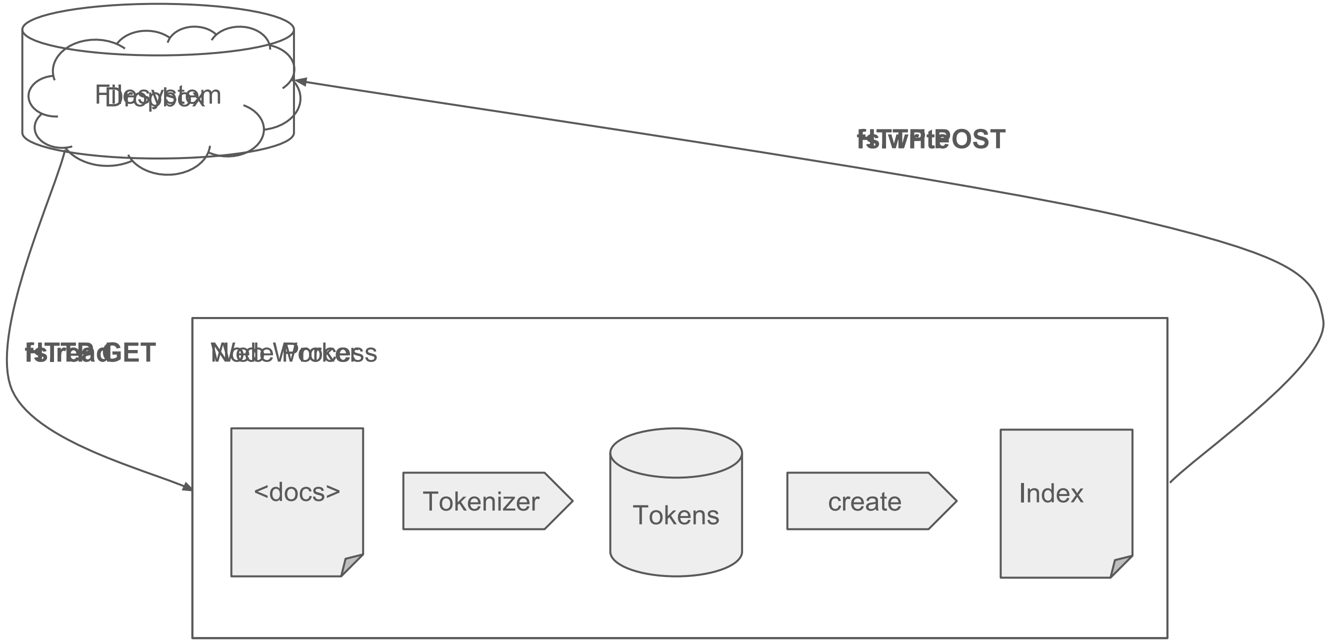
Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Indexing the Server



* Only index files with certain endings: .js, .html, .md, .txt
** Only index files that are smaller than 500kb

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

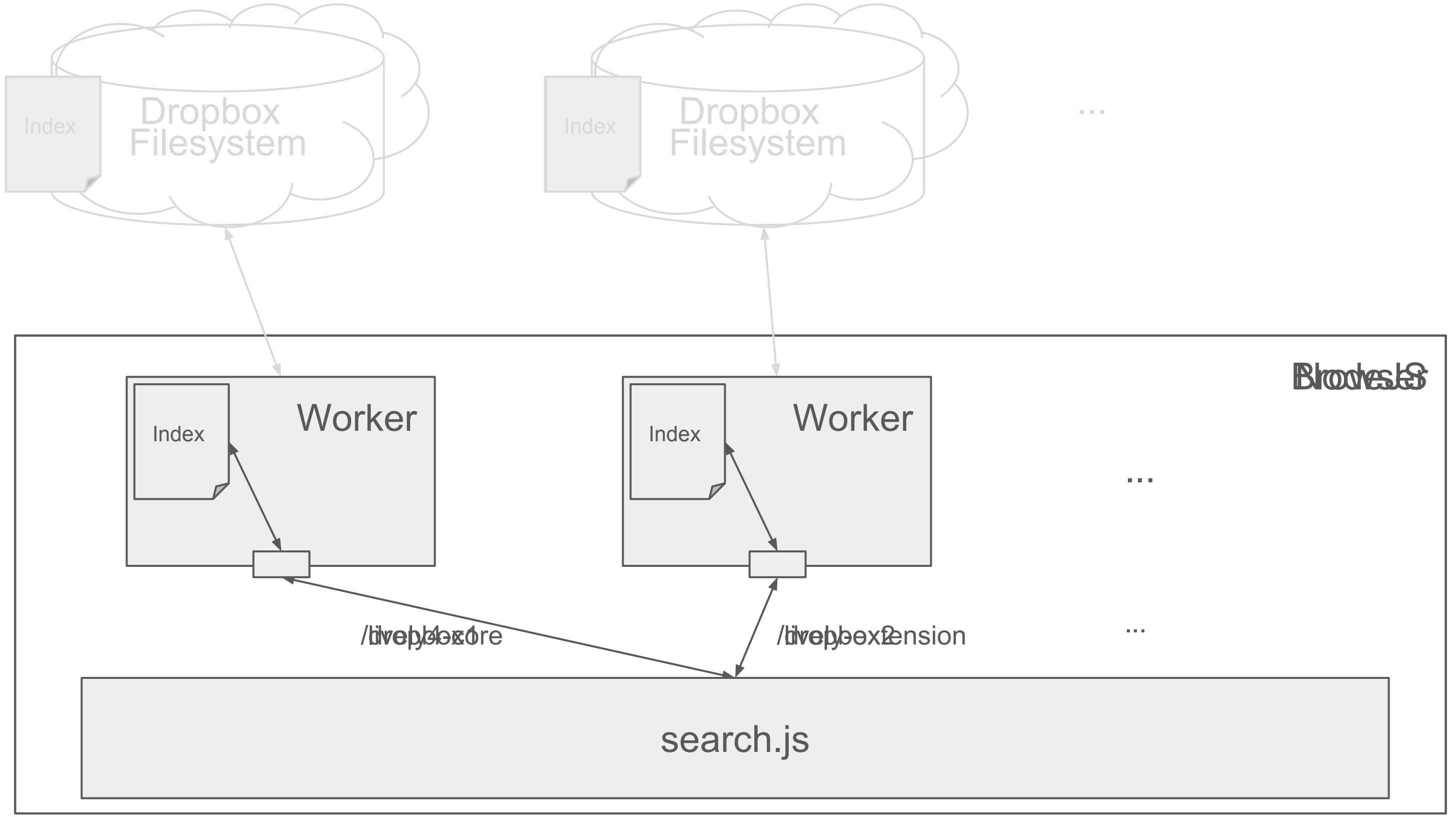
Lixissimus
daniel-wer

Searching on Server

- API request to **`httpServer.js`**
- **`/api/search?query=foo&location=/lively4-core`**

- **Server uses** internally the **same code as the browser** when searching through dropboxes
- **Dropbox mounts ≙ Repos on server** (e.g. lively4-core)

Searching on Server



Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Update an index

- User changes file that is one the server
    - File is persisted on the server
    - A hook in the file save function notifies the search module
    - Search module sends a message to the dedicated worker
    - Worker removes the file from the index and re-indexes it

- User changes file that is in the dropbox
    - We don’t notice right now
    - Could use HTTP long polling API to get notified of file changes
    - BUT what if the dropbox is changed while Lively is not running?

⇒ Persist file revision in index - Future Work!

Software
Architecture Group

Web Development
Seminar 2016

Topic: Personal
Cloud Search

2016-07-13

Lixissimus
daniel-wer

Doku: Content Provider Implementation

Content Providers are service specific (Dropbox, Server).

They have to implement **six** methods:

- `isIndexable(filepath)`
- `loadIndexJson(filename, options)`
- `saveIndexJson(indexJson, filename, options)`
- `checkIndexFile(filename, options)`
- `getFilepaths(options)`
  - `Return filepaths of all files that should be indexed.`
- `*FileReader(filepaths, options)`
  - `Generator that yields the content of each file in filepaths.`

IT Systems Engineering | Universität Potsdam

Demo

Software
Architecture Group

Web Development
Seminar 2016

Topic: Personal
Cloud Search

2016-07-13

Lixissimus
daniel-wer

Evaluation

- Indexing time:
    - lively4-core repo on the server: ~**30s**
    - lively4-core repo in the dropbox: ~**8min**
- Search time: **<2s**
- Search result scores from Github and lunr.js are not compatible
- Fuzzy search is hard - not a feature of lunr.js
    - Maybe like this: https://github.com/olivernn/lunr.js/issues/70

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Summary

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

A We have instant search across mounted files.

B We keep private data private.

Software
Architecture Group

Web Development
Seminar 2016

Topic: Personal
Cloud Search

2016-07-13

Lixissimus
daniel-wer

Future Work

- Content Provider for Google Drive/OneDrive
- Detect changes in mounted file systems and update index for changed files
  - Use Dropbox `delta` API to get list of changed files
- Cache indexes in local storage
- Add index configuration files
  - Configure which file endings are indexed
  - Blacklist directories

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Software Architecture Group

Web Development Seminar 2016

Topic: Personal Cloud Search

2016-07-13

Lixissimus
daniel-wer

Indexing: Personal Cloud Search

Web Development Seminar SS 2016
Software Architecture Group

Felix Wolff
Daniel Werner