Peer-to-Peer Volltextsuche

github.com/htw-projekt-p2p-volltextsuche

Boris Caspary
Emma Calewaert
Jonathan Neidel
Joscha Seelig
Leon Enzenberger
Ryan Torzynski
Simon Breiter
Stefan Sadewasser

HTW Berlin, Angewandte Informatik, Projektstudium bei Herr Prof. Hoppe

Juli 2021

Projektidee

Basiskonzepte I

Peer-to-peer

- Rechnernetz
- Gleichtberechtigte Knoten

Basiskonzepte II

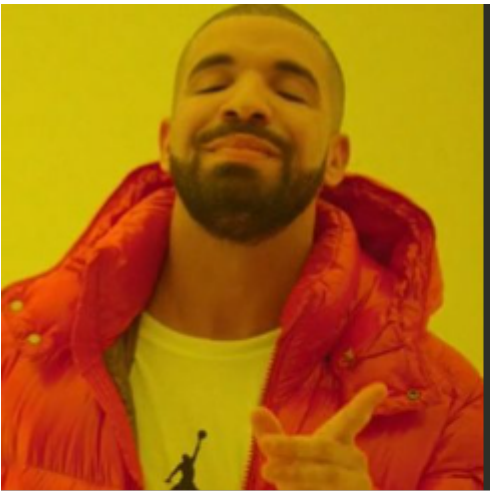
Volltextsuche

- Finden von Wörtern
- Handelt sich um Texte
- Zwei Phasen: Indexierung- und Anfragephase

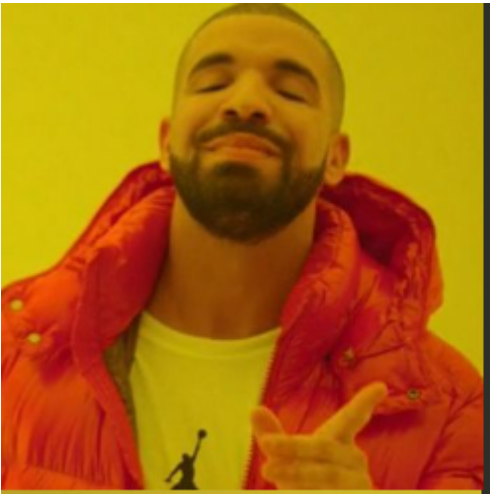
Bundestagsreden

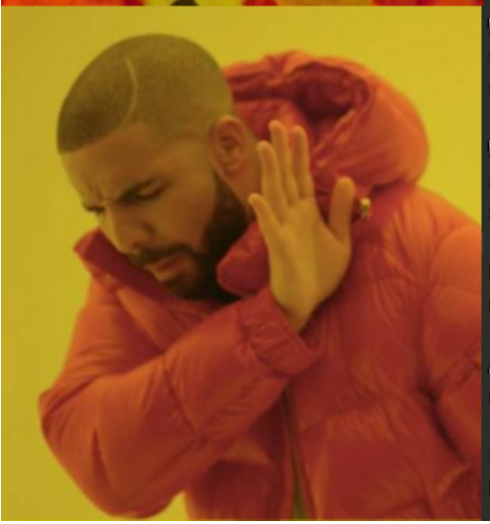
- Protokolle als Open Data verfügbar
- Großer Umfang an Daten (+/- 33 000 Reden)
- XML-Dateien*

Bildquelle: Deutscher Bundestag / Thomas Köhler/ photothek.net



```
<dbtplenarprotokoll vertrieb="Bundesanzeiger Verlagsgesellschaft mbH, Postfach 1 0 05 34, 5044
GmbH  Co. KG, Buch- und Offsetdruckerei, Bessemerstraße 83–91, 12103 Berlin, www.heenemann-dru
sitzung-datum="25.06.2021" sitzung-start-uhrzeit="9:00" sitzung-ende-uhrzeit="18:16" sitzung-n
    <vorspann>
        <kopfdaten>
            <plenarprotokoll-nummer>Plenarprotokoll <wahlperiode>19</wahlperiode>/<sitzungsnr>
            </plenarprotokoll-nummer>
            <herausgeber>Deutscher Bundestag</herausgeber>
            <berichtart>Stenografischer Bericht</berichtart>
            <sitzungstitel>
                <sitzungsnr>237</sitzungsnr>. Sitzung</sitzungstitel>
            <veranstaltungsdaten>
                <ort>Berlin</ort>, <datum date="25.06.2021">Freitag, den 25. Juni 2021</datum>
            </veranstaltungsdaten>
```



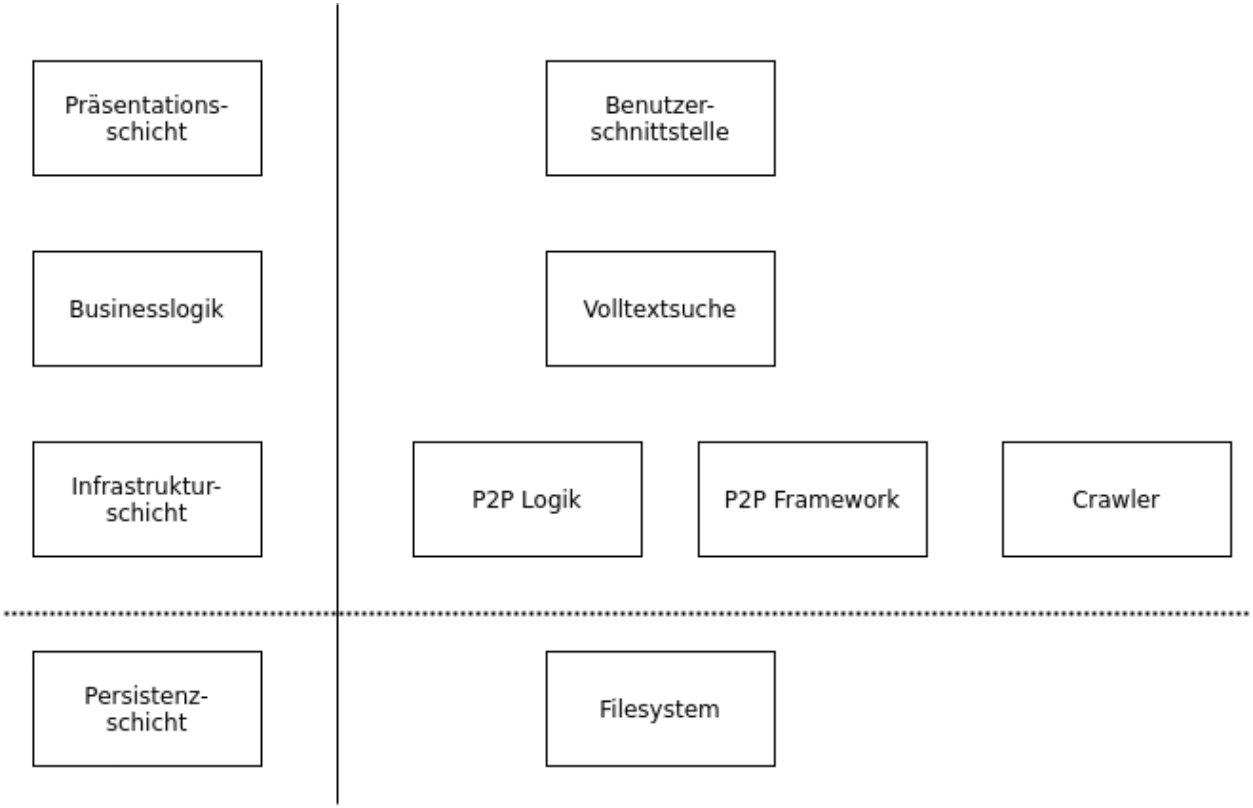


```
<dbtplenarprotokoll vertrieb="Bundesanzeiger Verlagsgesellschaft mbH, Postfach 1 0 05 34, 5044
GmbH  Co. KG, Buch- und Offsetdruckerei, Bessemerstraße 83-91, 12103 Berlin, www.heenemann-dru
sitzung-datum="25.06.2021" sitzung-start-uhrzeit="9:00" sitzung-ende-uhrzeit="18:16" sitzung-n
    <vorspann>
        <kopfdaten>
            <plenarprotokoll-nummer>Plenarprotokoll <wahlperiode>19</wahlperiode>/<sitzungsnr>
            </plenarprotokoll-nummer>
            <herausgeber>Deutscher Bundestag</herausgeber>
            <berichtart>Stenografischer Bericht</berichtart>
            <sitzungstitel>
                <sitzungsnr>237</sitzungsnr>. Sitzung</sitzungstitel>
            <veranstaltungsdaten>
                <ort>Berlin</ort>, <datum date="25.06.2021">Freitag, den 25. Juni 2021</datum>
            </veranstaltungsdaten>
```

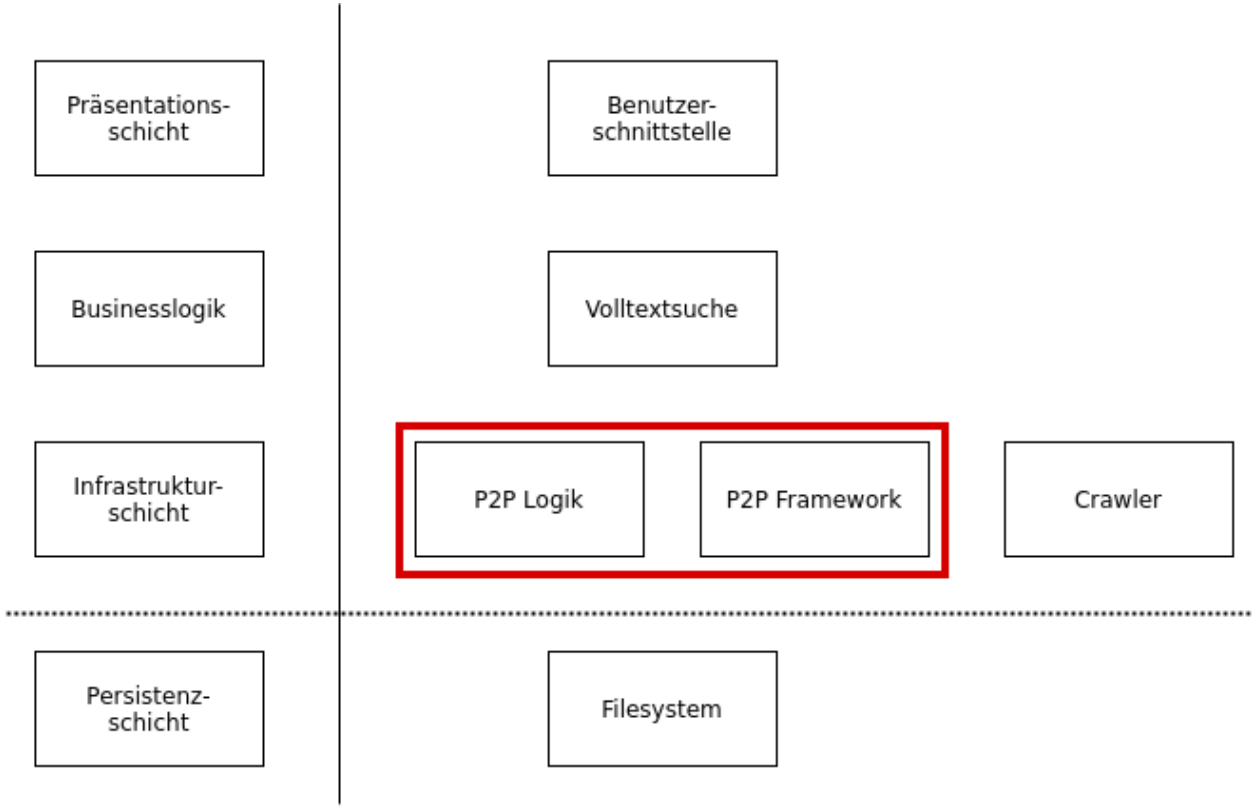
```
Wahl der Abgeordneten Nina Warken als or-
dentliches Mitglied des Gemeinsamen Aus
schusses  . . . . . 17575 A

Wahl des Abgeordneten Steffen Bilger als
ordentliches Mitglied des Vermittlungsaus
schusses  . . . . . 17575 B

Erweiterung und Abwicklung der Tagesord-
nung . . . . . 1

Absetzung der Tagesordnungspunkte 14, 15 b
und 25  . . . . . 1757

Begrüßung des Botschafters der Republik
Polen, Herrn Jerzy Jozef Marganski  . . . . . 17613 C
```

Aufgabenteilung

Schichtenmodell



Peer-to-Peer

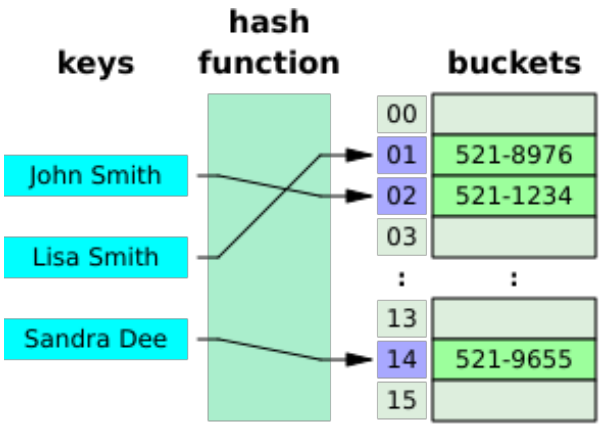

Präsentations-
schicht
Benutzer-
schnittstelle
Businesslogik
Volltextsuche
Infrastruktur-
schicht
P2P Logik
P2P Framework
Crawler
Persistenz-
schicht
Filesystem

DHT

- Distributed
- Hash Table

Hash Table

- key-value store



Hash table

Type	Unordered associative array
Invented	1953

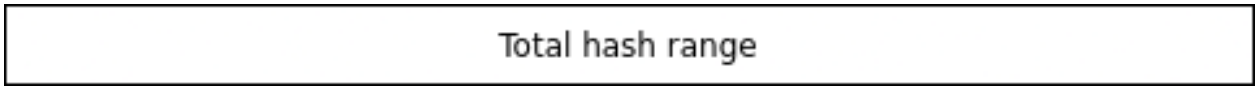
Time complexity in big O notation

Algorithm	Average	Worst case
Space	O(*n*)[1]	O(*n*)
Search	O(1)	O(*n*)
Insert	O(1)	O(*n*)
Delete	O(1)	O(*n*)

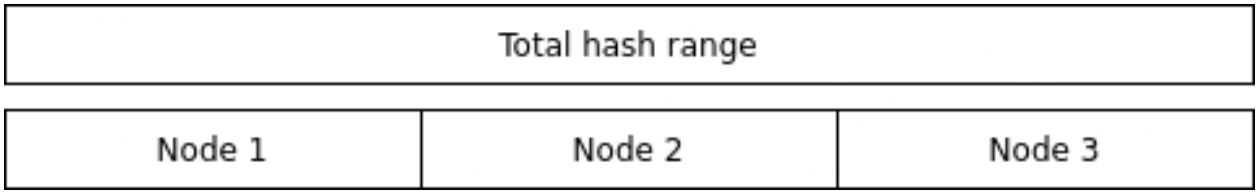
Bildquelle: https://en.wikipedia.org/wiki/Hash_table

Distributed HT I

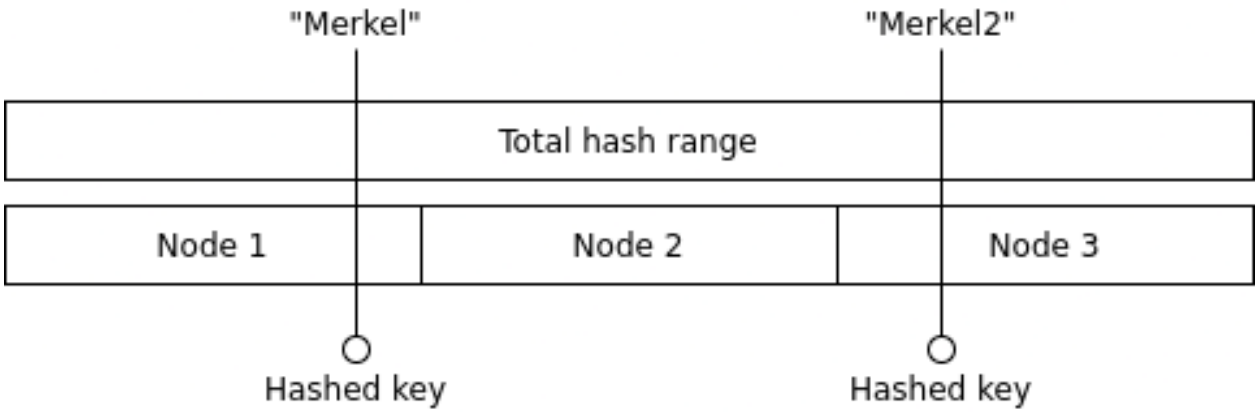
Basierend auf Kademlia



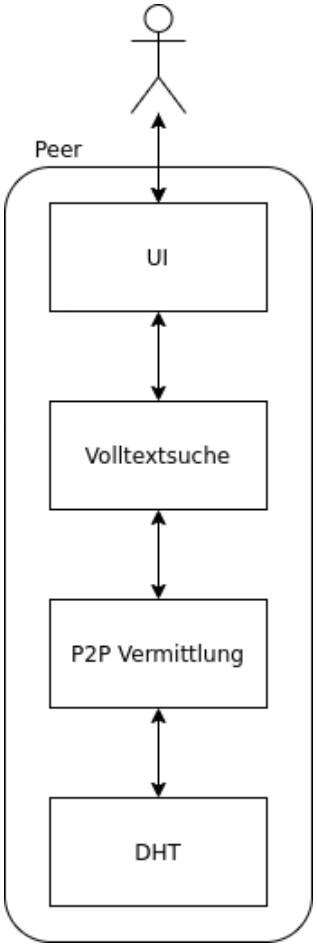
Distributed HT II



Distributed HT III



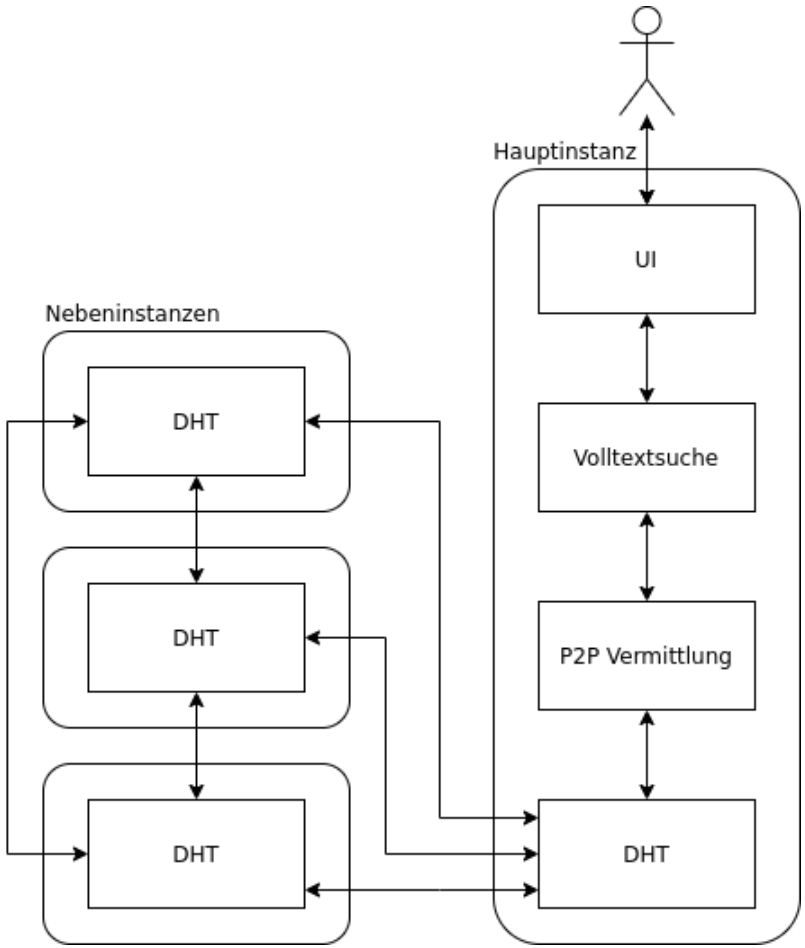
Schichten



Verteilung des Systems

Design Entscheidung:
Wie soll die Funktionalität des Systems verteilt werden?

1. Zentralisierung I



1. Zentralisierung II

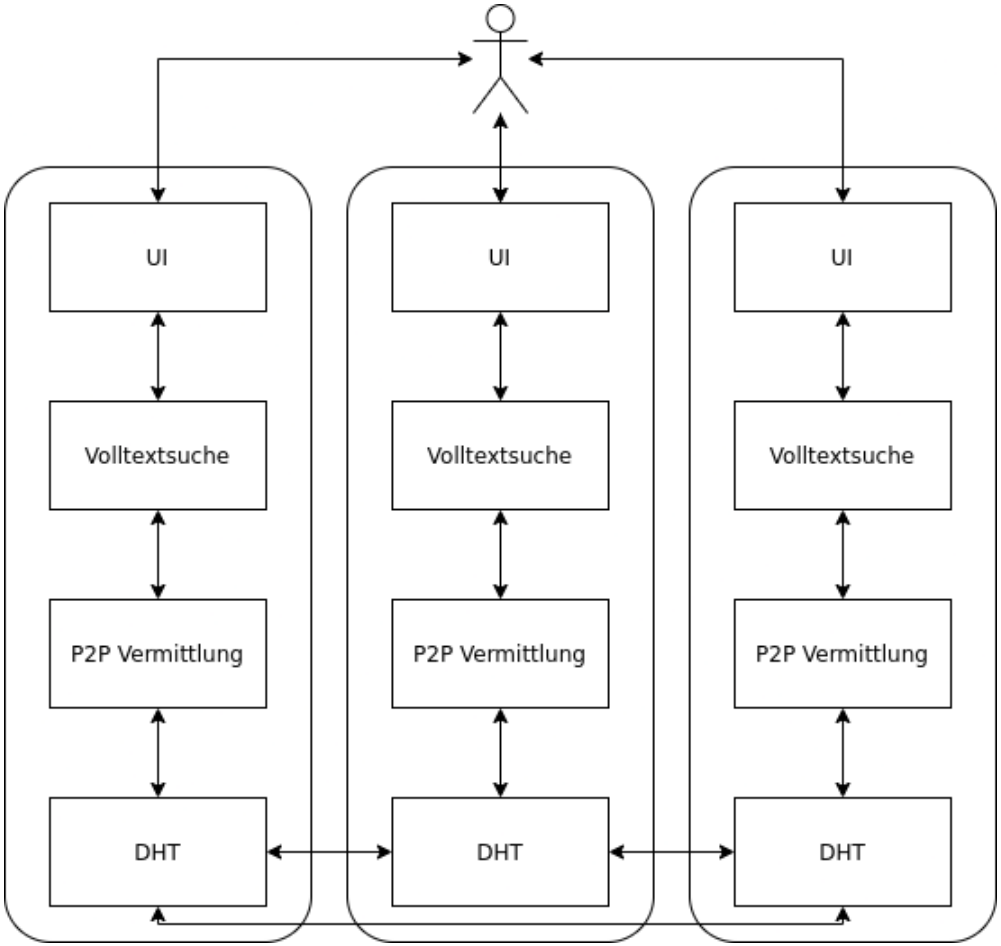
Pro:

- zentrale Anlaufstelle (Nutzerfreundlich)

Con:

- zentrale Anlaufstelle (single point of failure)
- Load nicht verteilt

2. Pur I


UI
UI
UI
Volltextsuche
Volltextsuche
Volltextsuche
P2P Vermittlung
P2P Vermittlung
P2P Vermittlung
DHT
DHT
DHT

2. Pur II

Pro:

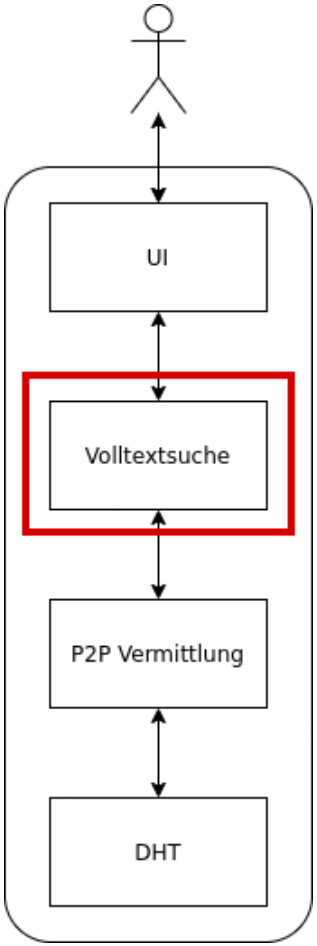
- Load verteilt
- mehrere Anlaufstellen (kein single point of failure)

Con:

- mehrere Anlaufstellen (verwirrte Nutzer $\rightarrow$ Loadbalancer)
- Administrationsaufwand ($\rightarrow$ OPS)

Unsere Wahl

Volltextsuche



Volltextsuche

Modi:

1. Indexierungsphase
   a. Aufbereitung der Daten
   b. Einfügen in Invertierten Index
2. Anfragephase

1a. Aufbereiten der Daten I

Beispiel:
Klaus-Rüdiger steht vor dem Hause der CDU.

1a. Aufbereiten der Daten II

Tokenisierung:

- Segmentierung in einzelne Wörter
- Lowercase
- Satzzeichen entfernen

'klaus-rüdiger' 'steht' 'vor' 'dem' 'hause' 'der' 'cdu'

1a. Aufbereiten der Daten III

Stoppwörter entfernen:

- Pronomen (ich, ihre, ..)
- Konjunktionen (und, aber, ..)
- Präposition (auf, bis, ..)
- Artikel (der, die, ..)

'klaus-rüdiger' 'steht' ~~'vor'~~ ~~'dem'~~ 'hause' ~~'der'~~ 'cdu'

1a. Aufbereiten der Daten IV

Stemming:

- Zurückführung auf den Wortstamm
- Snowball Stemming

'klaus-rüdiger' $\rightarrow$ **'klaus-rudig'** 'steht' **'hause'** $\rightarrow$ **'haus'** 'cdu'

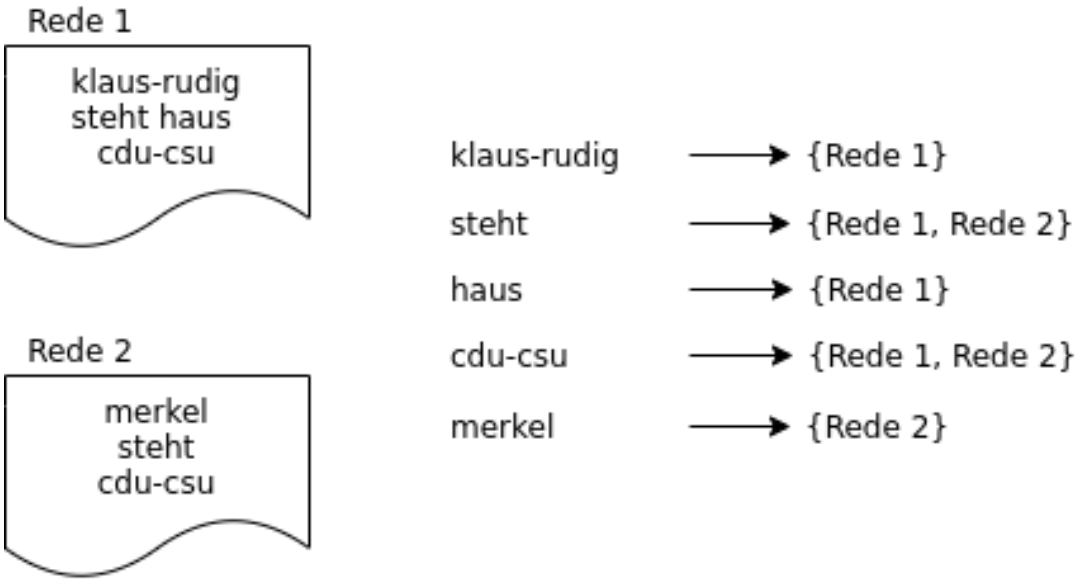
1a. Aufbereiten der Daten V

Normalisierung:

```
private val affiliationNorms: Map[String, String] =
  Set(
    AffiliationNormalization(
      norm = "cdu-csu",
      variations = Set("cdu", "csu")
    ),
    AffiliationNormalization(
      norm = "die-linke",
      variations = Set("linke")
    ),
    AffiliationNormalization(
      norm = "bündnis-90-die-grünen",
      variations = Set("die-grünen", "bündnis-90", "grüne")
    )
  ) flatMap (_ mappings) toMap
```

'klaus-rudig' 'steht' 'haus' **'cdu'** $\rightarrow$ **'cdu-csu'**

1b. Inverted Index



1b. Distributed Inverted Index

Einfügen in den DHT

cdu-csu =
027066e01f461f0a61e6cc19fd9a793a22a575bf ⟶ {Rede 1, Rede 2}

merkel =
969d9304a6dd3eb214241e0671c92659f835e08a ⟶ {Rede 2}

1b. Aufteilung der Daten I

Design Entscheidung:
Wie sollen die Daten aufgeteilt werden?

1b. Aufteilung der Daten II

1. Partition by Keyword

- Gleichmäßige Verteilung der Keywords über DHT
- Speicherung in Distributed Inverted Index

Pro:

- Schnell beim Abrufen von Keywords

Con:

- Langsam beim Einfügen
  - → Relevanz für den Use Case
  - → bei uns: 2-3 min pro Protokoll
  - Optimierungen möglich

1b. Aufteilung der Daten III

2. Partition by Document

- Gleichmäßige Verteilung der Dokumente über Peers
- Speicherung in lokalem Invertiertem Index

Pro:

- Schnell beim Einfügen (lokales Speichern in Sekunden)

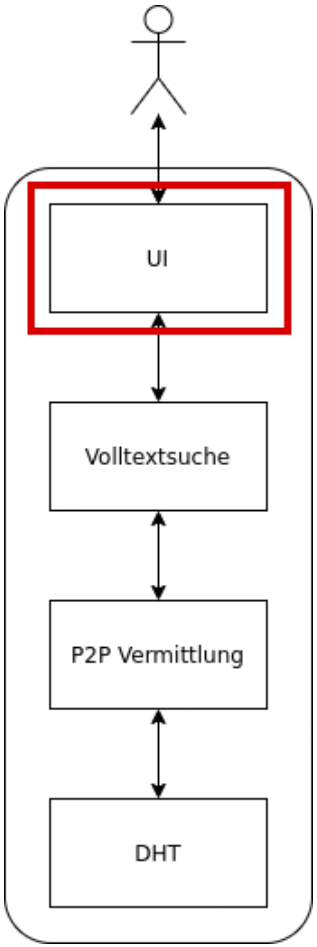
Con:

- Abruf langsam (Fluten des Netzwerks)
- Schlechte Skalierbarkeit

2. Anfragephase

- Aufbereiten der Suchbegriffe (Tokenisierung, etc.)
- Abrufen der Daten aus Distributed Inverted Index
- Bewertung der Relevanz (sortieren)

UI Demo



Folien online auf:

- github.com/htw-projekt-p2p-volltextsuche