zühlke
empowering ideas

Effizientes Suchen mit Apache Solr

Java Forum Stuttgart 2023

Matthias Graf | 13. Juli 2023 | Stuttgart

searching "hot dog"

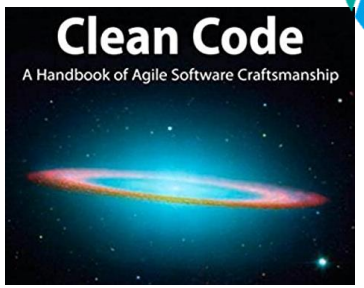
google:

bing:

from https://www.reddit.com/r/dankmemes/comments/pi2aj3/why_is_my_search_engine_called_e621/


Java


spring boot

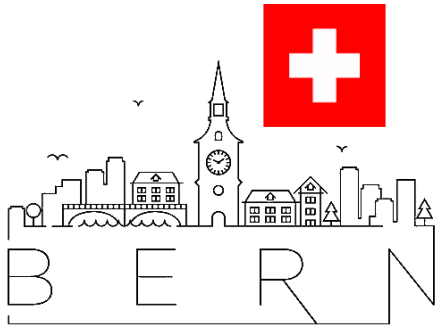

Clean Code
A Handbook of Agile Software Craftsmanship


Plan
Design
Develop
Test
Release
Feedback
AGILE


{dev}

About Me


zühlke
empowering ideas


BERN

Übersicht

Um was geht es heute?

Inhalt

- Was ist “Suche”? Und wieso ist das wichtig?
- Was ist Solr und wie funktioniert es?
- Wie kann ich Solr in eine Applikation integrieren?

Out-Of-Scope

- Suche in Binärdateien
- Schema Optimierungen
- Performance und Load Balancing
- Alternativen zu Solr
- Query-Languages

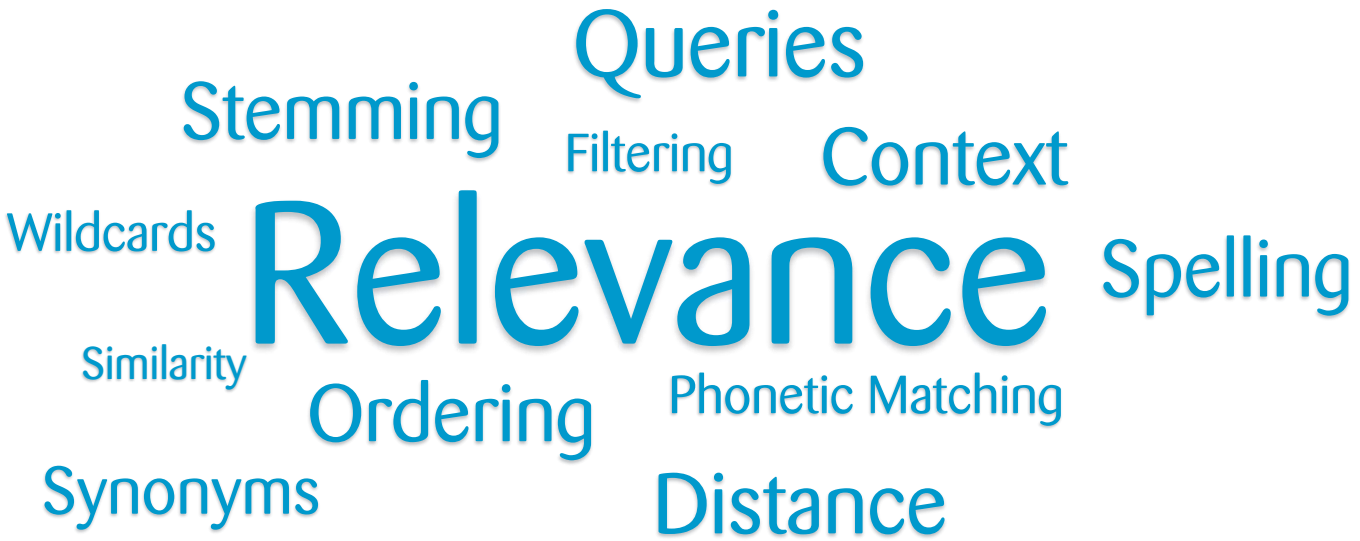
Ziel

- Entscheiden können ob Solr für die eigenen Applikation sinnvoll ist
- Wissen wie Solr Integriert werden kann

Was ist Suche?


Binary Data
Performance
Updates
Autocomplete
Imports
Features
Highlighting
Clustering
Faceting


Avoid Navigation
Simplicity
Full-Text
Suggestions
Usability
Intuitive
Fast
Mobile-Friendly


Queries
Stemming
Filtering
Context
Wildcards
Relevance
Spelling
Similarity
Ordering
Phonetic Matching
Synonyms
Distance

Was ist eine gute Suchfunktion?

- Eine gute Suchfunktion ist **wichtig** denn sie Unterstützt den **Benutzer**
  - Eine gute Suchfunktion hilft Benutzern die gewünschten Informationen schnell und einfach zu finden
  - Schneller als «klassische» Navigation
  - Funktioniert gut Mobile und für neue Benutzer
  - Benutzer sind daran gewöhnt und erwarten, dass eine Suchfunktion in Applikationen verfügbar ist

- Eine gute Suchfunktion ist **schwierig** denn es ist unklar was der **Benutzer will**
  - Suchergebnisse müssen Relevant für den Benutzer sein, Dies ist kontextabhängig!
  - Die Suche soll nicht genau, sondern fuzzy sein (Stemming, Spellcheck, Synonyms, Distance etc.)
  - Die Suche soll alle relevanten, aber nur relevante Ergebnisse finden, sortiert nach Relevanz

Wie machen wir dies?

Wie bauen wir eine gute Suchfunktion?

- Für eine gute Suche braucht **spezielle Tools**
  - Eine „LIKE %$1%“ SQL-Query reicht nicht

  => Solche Tool existieren!

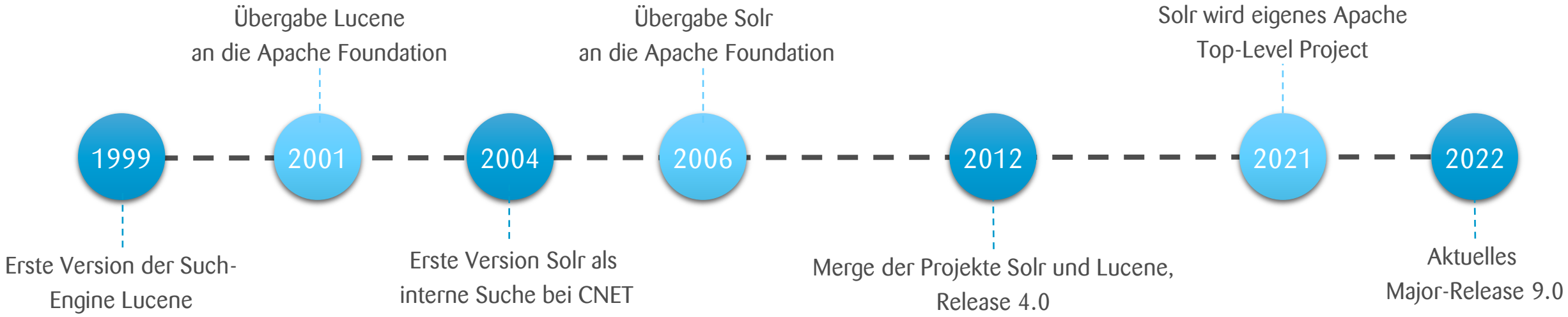
- Eine wirklich gute Suche **braucht kontinuierlichen Aufwand**
  - Nach was sucht der Benutzer? Findet er die richtigen Resultate?
  - Wie benutzt der Benutzer die Suchfunktion?
  - Was für Daten werden durchsucht?

Was ist Solr?

Apache Solr

Facts and Figures

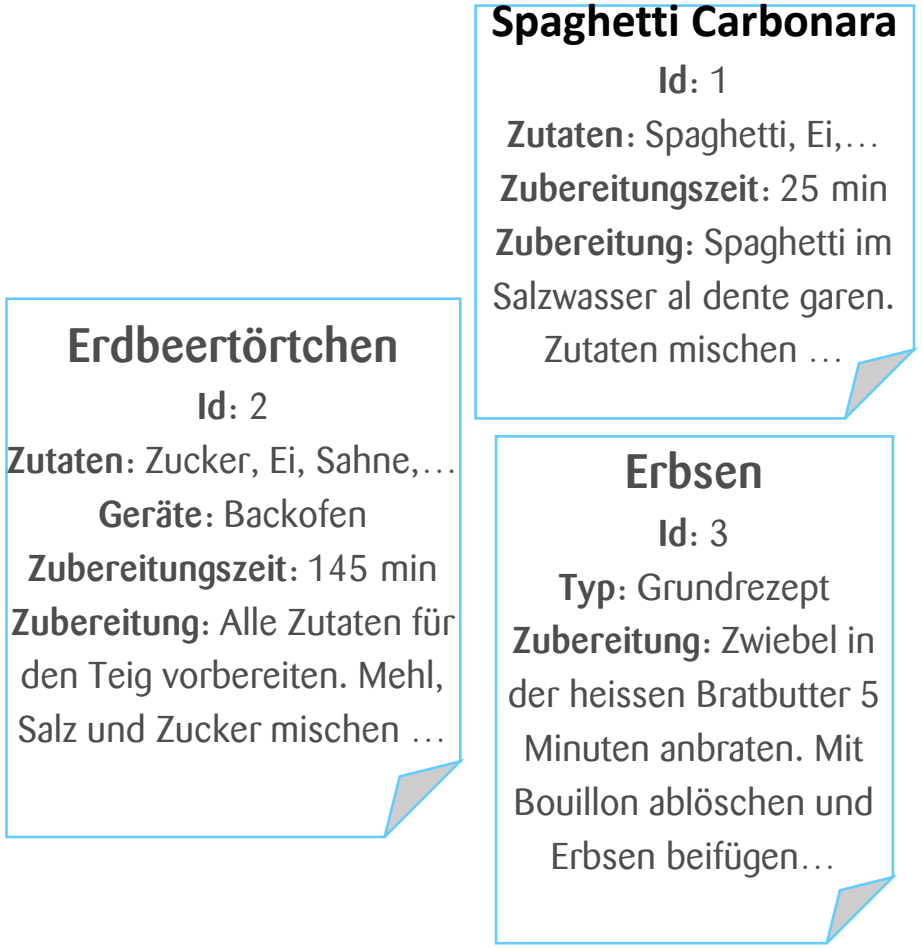
- Open-Source-Enterprise-Suchplattform
- Entwickelt in Java, basiert auf Apache Lucene
- Index-Suche
- Wichtigste Features: Volltextsuche, Highlighting, Faceting und Clustering, Replikation und Balancing, Binärdokumente etc.
- Weltweit in Verwendung unter anderem von DuckDuckGo, Adobe, Instagram, eBay, Netflix, Disney



Begriffe

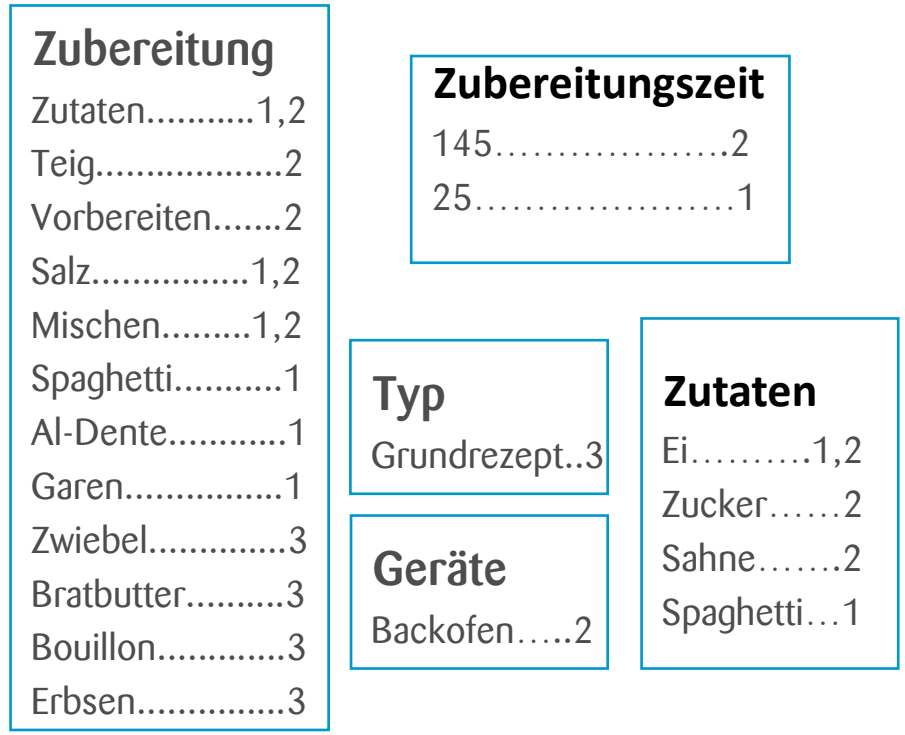
Document	Eine Menge von Daten, die etwas beschreiben. Grundlegende Informationseinheit von Solr. Im Dokumente setzen sich aus Feldern zusammen (Semi-Strukturiert). Sollten über eine ID verfügen.
Field	Teil eines Document, können verschiedene Arten von Daten enthalten, z.B. Text, Fließkommazahlen, Listen etc.
Collection	Dokumente, die in einem einzigen logischen Index mit einer einzigen Konfiguration gruppiert sind.
Core	Eine einzelne Solr-Instanz / logischer Knoten. Mehrere Cores können auf einem einzigen physikalischen Knoten laufen.
Shard	Dokumente werden Shards zugewiesen. So kann eine Collection wenn nötigt auf mehrere Cores verteilt werden.

Was ist ein Index?



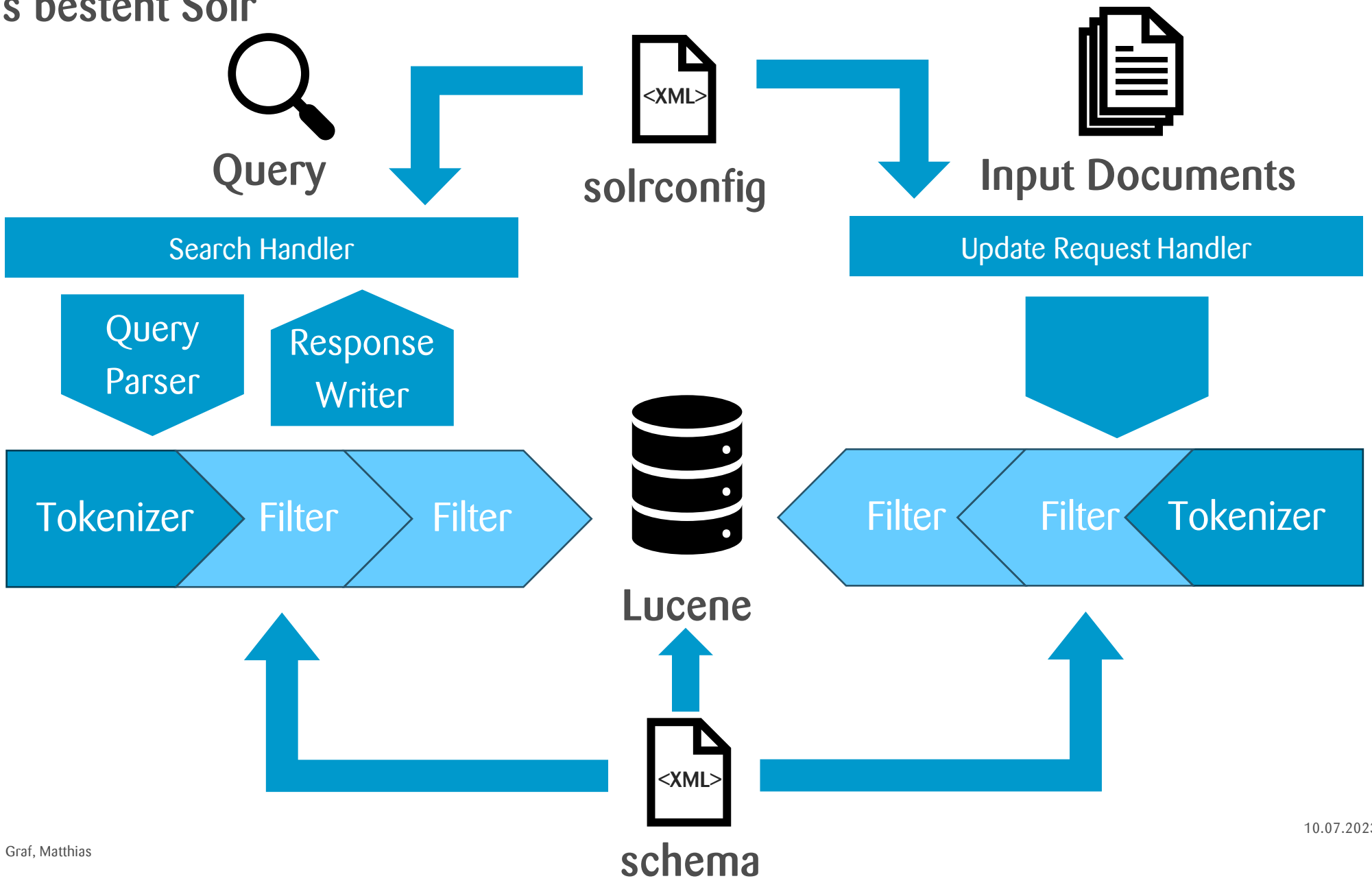
Semistrukturierte Daten (Dokumente)

Indexing



Inverted Index

Aus was besteht Solr



Wie kann ich Solr benutzen?

Docker-Setup

Solr kann als Docker-Container gestartet werden

```
docker run -p 8983:8983 -t solr
```

Erweitertes Docker-Compose-Setup

```
solr:
    image: 'solr:9.2.0'
    ports:
      - "8983:8983"
    command:
      - solr-precreate
      - COLLECTION_NAME
    volumes:
      - ./src/main/solr:/opt/solr/server/solr/configsets/_default/conf
      - solr-data:/var/solr
```

Live Demo

Zusammenfassung Live Demo

- Solr-Server starten (Container oder Binary)
- Eigene Collection erstellen (precreate-script oder über UI)
- ManagementUI localhost:8983/solr ausprobieren
- Integration in eigene Applikation mittels Client-Library (SolrJ) oder Rest-API
- Dokumente hinzufügen (Indexing)
- Suchen ausführen (Querying)

Tipps und Tricks

- Solr ist einfach zu integrieren, kann aber schwer zu meisten sein. Start als MVP und langsames Aufbauen
- Suche ist immer Kontext-Sensitiv: Was und wie sucht der Benutzer?
  - Eigene Abfragen vs. Solr-Queries?
  - Schema-Optimierungen
  - Faceting / Clustering
  - Konstantes Monitoring
- Best-Practice
  - Eine Collection für unterschiedliche Daten
  - Möglichst alle Daten aufnehmen, Schema-Änderungen und Reindex ist teuer
- Pain-Points
  - Berechtigungsprüfungen aus den Suchresultaten?
  - Wann Index updaten? Hohe Last oder veraltete Daten?

Takeaways

- Eine Suchfunktion ist wichtig, eine gute Suchfunktion zu implementieren ist aber schwer
- Apache Solr ist eine Open-Source Lösung mit guter Performance, vielen Features, einfach zu integrieren aber schwer zu meistern
- Start Slow and Improve

Fragen?

github.com/lizzyTheLizard/solr-jfs2023

Folien, Code-Beispiele und Referenzen

Weitere Informationen

solr.apache.org/guide

Ofizielle Solr Dokumentation; Tutorials, Guides für Deployment und Konfiguration, Dokumentation der Query-Parser und vieles mehr

hub.docker.com/_/solr

Offizielles Docker-Image von Solr

baeldung.com/apache-solrj

SolrJ-Tutorial