

# Expertensuche

## Einleitung

Die Expertensuche ist der Forschungsanteil im Bereich Search & Retrieval. Es soll ein SQL Wizzard entstehen, der die Inhalte der Datenbank in geeigneter Form in Elementen darstellt. Diese Elemente sollen per Drag&Drop auf einer Arbeitsfläche angeordnet und verknüpft werden. Anhand der Visualisierung wird ein QSQL Statement generiert, welches an die Datenbank geschickt wird. Die Experten Suche ist damit ausschließlich für eine Recherche in der Datenbank vorgesehen. Darüber hinaus, können SQL / QSQL Befehle auch direkt abgesetzt werden.

## PROPhEcy Framework

Das PROPhEcy Framework wurde durch Sebastian Lehrack entwickelt und in der Masterarbeit von Christian Winkel implementiert. Es ermöglicht die Verwendung von probabilistischen Anfragen auf einer normalen relationalen Datenbank. Für OpenInfRA ist der probabilistische Ansatz vorerst nicht vorgesehen. Hier wird ausschließlich auf die Möglichkeit der vagen Anfrageformulierung zurückgegriffen. Diese ermöglicht eine Ähnlichkeitssuche auf einer Datenbank, ähnlich wie in einem indexbasierten Information Retrieval System. Dazu gehört ebenso ein Ranking bei den Ergebnistreffern.

~~Das [PROPhEcy Framework](#) wird durch Christian Winkel als Open Source Projekt weiterentwickelt.~~

**Update [11.12.2014]:** Die Entwicklung durch Christian Winkel wurde eingestellt. Das zwingt die Entwicklungen zum Rückfall auf das Vorgängersystem ProQua. Dort muss zunächst geprüft werden, wie die Anbindung an die Experten Suche erfolgen kann.

## GUI-Konzept<sup>1</sup>

## Anforderungen

Es wurden bereits, neben den im Anforderungskatalog<sup>2</sup> aus dem Grobkonzept spezifizierten, spezielle Anforderungen für die Experten Suche<sup>3</sup> entworfen. Diese können im Laufe der Zeit durchaus erweitert werden.

## Konzeptionierung

Der Aufbau der Suche ist wie folgt geplant:

---

<sup>1</sup> Expertensuche - GUI Konzept.pdf

<sup>2</sup> Anhang-B Anforderungskatalog.xls

<sup>3</sup> Expertensuche - Anforderungen.pdf

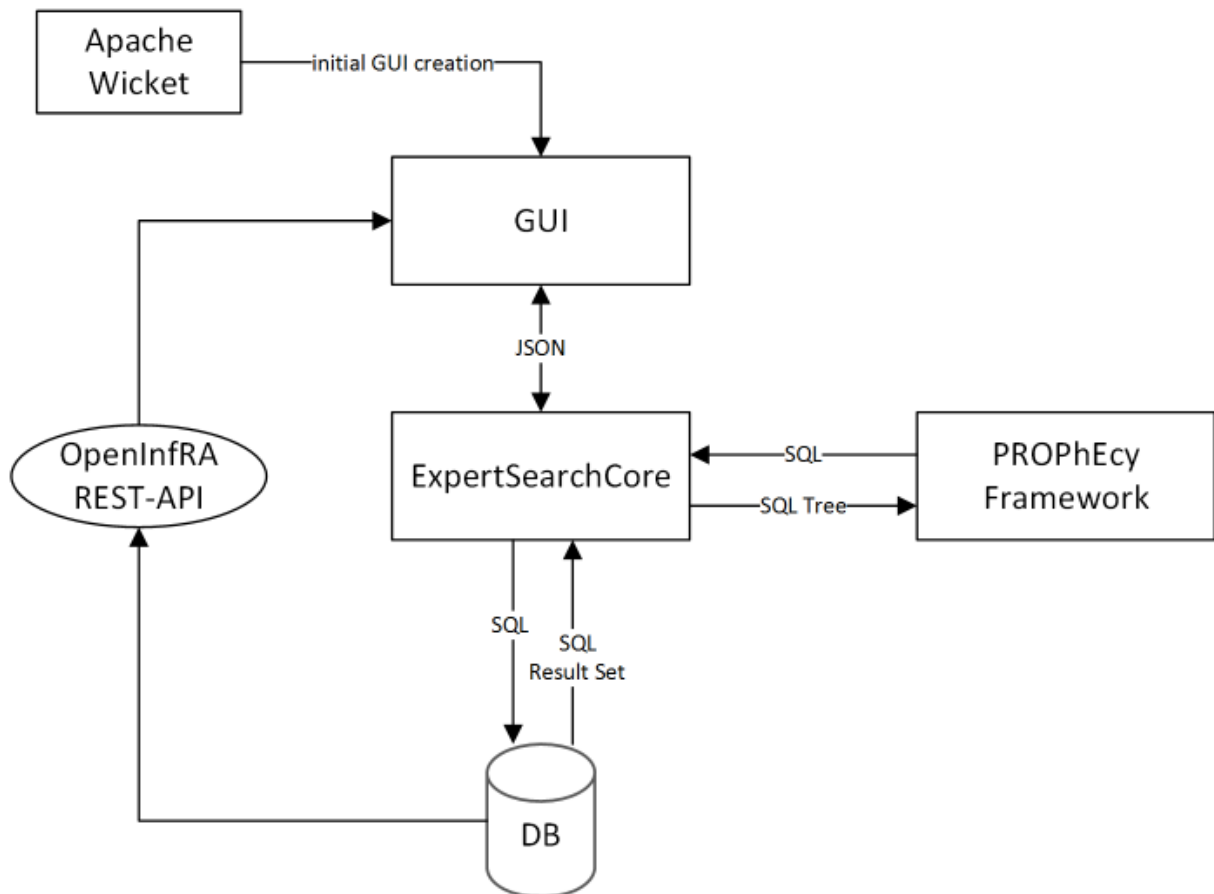


Abbildung 1: Konzept

1. Bereitstellen der GUI über Apache Wicket
2. Nachladen von Inhalten über REST Schnittstelle nach der Auslieferung der GUI durch Wicket
3. Anfrageerzeugung in GUI und Übermittlung als JSON Objekt an ExpertSearchCore
4. Transformation des JSON Objekts in SQL Tree
5. Übergabe des SQL Tree an PROPhEcy Framework
6. Generierung eines SQL Statements durch PROPhEcy Framework
7. Absetzen der SQL Anfrage an Datenbank und Verarbeitung des daraus resultierenden Result Sets
8. Rückgabe des Ergebnisses an GUI und Darstellung dieser

### **Anforderungen an REST Schnittstelle**

Unter Beachtung des Rollen / Rechte Konzepts werden Inhalte aus folgenden Datenbanktabellen der Projektdatenbank benötigt:

#### Lokalisierungstabellen

- value\_list
- value\_list\_values
- project
- topic\_characteristic
- attribute\_type
- attribute\_type\_to\_topic\_characteristic

- attribute\_value\_domain
- attribute\_value\_geom
- attribute\_value\_geomz
- attribute\_value\_value

## Implementierung

Die Implementierung der GUI erfolgt aller Wahrscheinlichkeit nach mit JQuery. Dabei sollen folgende Plugins zusätzlich verwendet werden

- <http://jaukia.github.io/zoomooz/> - ermöglicht Zoom Effekte, welche für das Zeichenfenster notwendig sind
- <http://gridster.net/> - shiny drag Effekte, welche bei den Elementen im Zeichenfenster genutzt werden könnte
- <http://i18next.com> - Mehrsprachigkeit
- <http://bseth99.github.io/jquery-ui-extensions/tests/visual/waitbutton/base.html> - Signalisierung, dass System beschäftigt ist (für Suchbutton sinnvoll?)

### ***JSON Objekt***

Die gesamte inhaltliche Darstellung auf der GUI soll in einem JSON Objekt hinterlegt werden. Die Logik der GUI ändert also nur das JSON Objekt und zeigt der GUI an, dass dieses verändert wurde. Daraufhin lädt die GUI das JSON Objekt neu. Um das Aktualisieren der GUI zu optimieren, sollte nicht das gesamte JSON Objekt neu geladen werden, sondern nur der Bereich, der verändert wurde. Die Kommunikation zwischen der GUI und dem Java basierten SearchCore soll ebenfalls über JSON erfolgen (siehe Abbildung 1). Daraus resultiert, dass die inhaltliche Darstellung auf der GUI und die Kommunikation zwischen GUI und SearchCore, ein und dasselbe JSON Objekt verwendet werden kann.