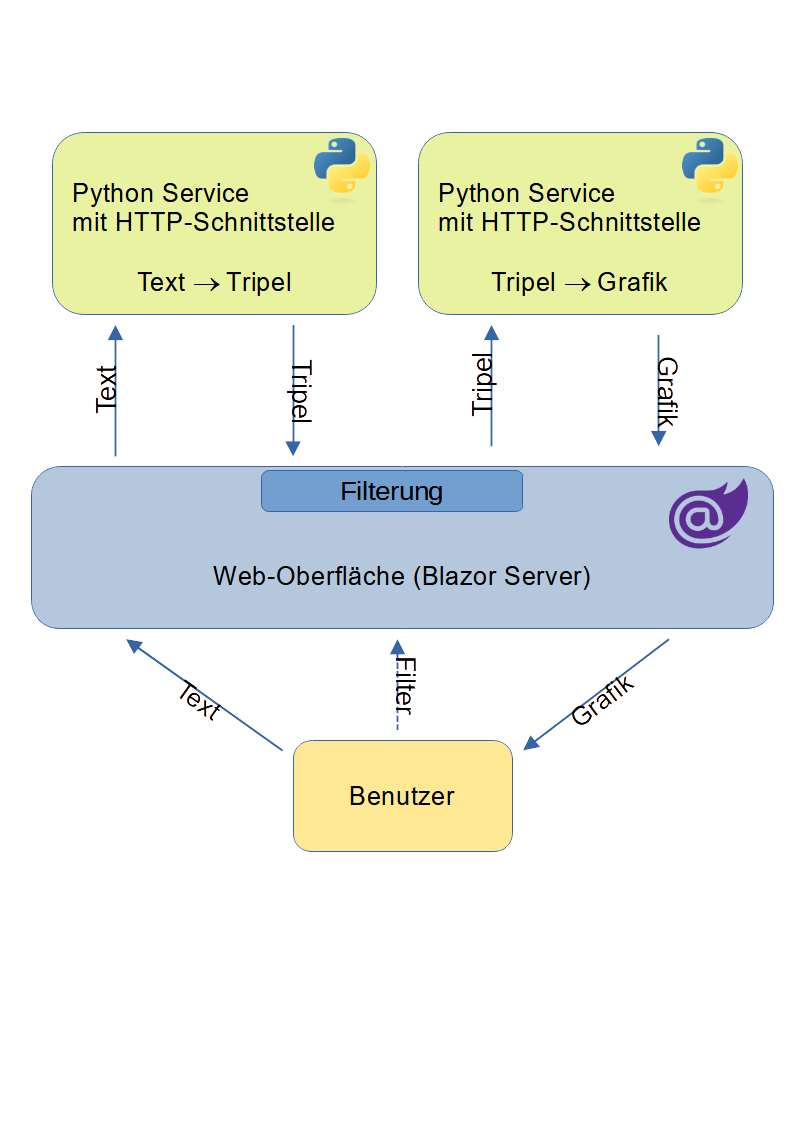
1. **Grobe Lösungsskizze (Sven)**

Es soll eine Anwendung zur automatischen Generierung von *Concept Maps* aus einem vom Benutzer eingegebenen Text entwickelt werden.

Die Benutzeroberfläche bildet eine **Blazor** (Server) Webanwendung, welche in C# geschrieben wird. Dies erleichtert die Entwicklung enorm, weil kein JavaScript-Code und keine separaten APIs für die Schnittstelle *Browser → Server* implementiert werden müssen.

Um die *Concept Map* zu generieren werden die Python-Services im Hintergrund aufgerufen:

* Zunächst werden aus dem Text die Tripel extrahiert. Jedes Tripel stellt die Beziehung von Wörtern untereinander dar. Dazu wird der Service auf der linken Seite aufgerufen.
* Anhand diesen Tripel kann dann die *Concept Map* erstellt werden. Der Benutzer soll jedoch die Möglichkeit haben, die Erstellung mittels Filtern zu beeinflussen. Die vorher ermittelten Tripel könnten beispielsweise mittels Blocklist, Allowlist o.ä. gefiltert werden, bevor sie an den zweiten Service an der rechten Seite übergeben werden.

1. **Definition Work Breakdown Structure** (Korvin)
2. Analyse der Anforderungen (Klärung welcher Algorithmus)
3. Datenbeschaffung und -vorbereitung
4. Definieren der Schnittstellen zwischen Python und der Webanwendung
5. Entwicklung der Anwendungen
   1. Entwicklung der Python-Anwendung
   2. Entwicklung der Webanwendung
6. Integration der Python-Anwendung in die Webanwendung unter Nutzung der Schnittstellen
7. Implementierung von Funktionen zur Bearbeitung und Visualisierung von Concept Maps
8. Testing und Fehlerbehebung
9. Dokumentation

* WBS als Grafik und Detailgrad klären

1. **Team (Adriana)**

**3.1 Wer kümmert sich um was?**

Sammeln von relevanten Texten/Informationen für Testcase

* welche Algorithmen gibt es (Vor/Nachteile etc.)

**NLTK (Hauptverantwortlich: Adriana Mikuteit)**

Paket 1 – Erstellung von Triplets

Paket 2 – Generierung der Grafik (**save fig as svg)**

**PyTorch (Hauptverantwortlich: Korvin Walter)**

Paket 1 – Erstellung von Triplets

Paket 2 – Generierung der Grafik (**save fig as svg)**

**Webanwendung (Hauptverantwortlich: Sven Nicolai)**

Paket 1 – Entwicklung der Webanwendung

Paket 2 - Integration von Concept Maps in die Webanwendung

Nachträgliche Anpassung der Concept Map (Nice-to-Have)

Evaluierung und Test – alle?

* Auswertung (Quantitativ/Qualitativ)
* Katalog zur Bewertung der erstellten Concept Map
* Testcases

**3.2 Way-of-Working (WAS MUSS DA ALLES REIN? WIE WOLLEN WIR DIE AUFGABEN TRACKEN?)**

* Agile Entwicklungsmethode mit zweiwöchigen Statusreports (1 Seite)
* verwendet Tools wie GitHub (DISCORD/ARBEITSTEILUNG?)
* Die Webanwendungs-Entwicklung erfolgt unter Verwendung von Blazor.

1. **Grobe Zeitplanung (auf Milestones achten) -> alle**

WO/WAS WÄREN Milestones?

Datenextraktion - X Wochen Datum:

Datenvorverarbeitung - X Wochen Datum:

Modellierung - X Wochen Datum:

Visualisierung - X Wochen Datum:

Evaluierung und Test - X Woche Datum:

Webanwendungs-Entwicklung - Wochen Datum:

Integration von Concept Maps in die Webanwendung - Datum:

Benutzerinteraktion mit Concept Maps - Datum:

Milestone 1 – DemoVersion

1. **Beschreibung Datensammlung** (Was genau versteht er hier runter?) -> alle, Frage per Mail stellen

* Datensammlung aus einer bestimmten Domäne (z.B. Medizin)
* ~~Verwendung von RDF-Daten zur Generierung der Concept Maps (WIE?, ich kenne nur die Art über Texte und dann Extraktion etc.)~~
  1. Umfang (nur Datensammlung oder das Projekt?)
* Generierung von Concept Maps aus Text-Datei (bzw. RDF)
* User kann Concept Maps in der Webanwendung anzeigen und bearbeiten

5.2 Charakteristik (nur Datensammlung oder das Projekt?)

- englisch

5.3. Methode: (nur Datensammlung oder das Projekt?)

- Algorithmen

1. Evaluation & Test -> alle

Auswertung der Concept Maps (Quantitativ/Qualitativ)

Katalog zur Bewertung der erstellten Concept Map

Testcases werden definiert und auf Basis der Bewertungsmatrix als Erfolgreich/nicht erfolgreich klassifiziert.

Die Concept Maps werden auf ihre Genauigkeit und Relevanz in Bezug auf die Texte, aus denen sie generiert wurden, überprüft.

Die Webanwendung wird auf ihre Benutzerfreundlichkeit und Zuverlässigkeit getestet.

1. Aufbereitung & Visualisierung ->alle

Showcase vorbereiten (Democase)

Dokumentation wird parallel gepflegt (Readme)

Präsentation der Ergebnisse

Aufbereitung der Testergebnisse