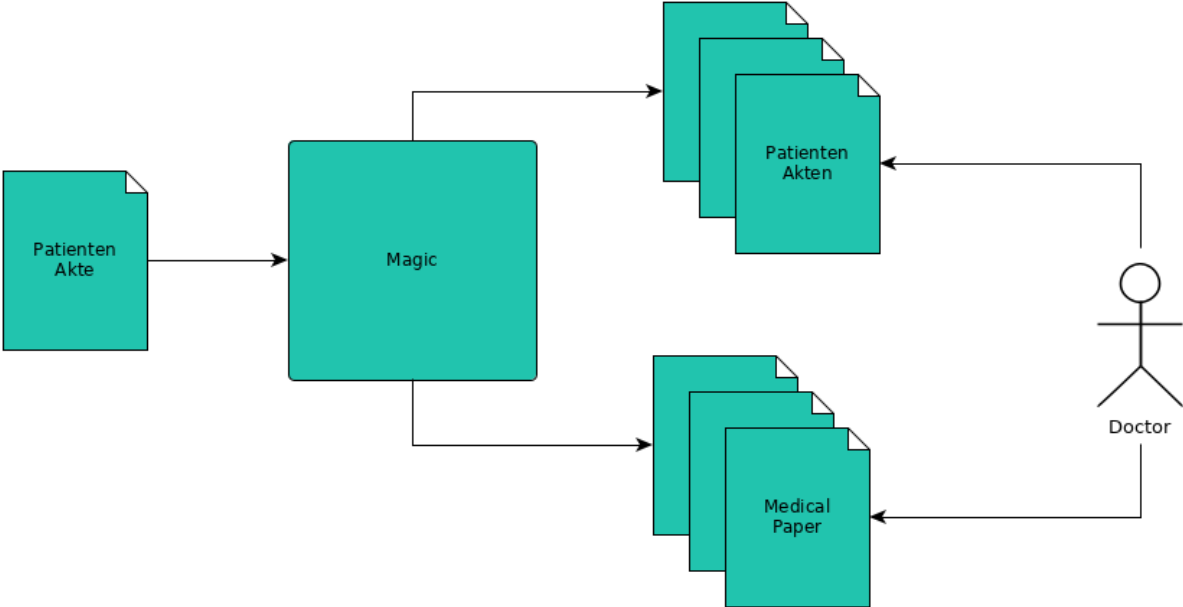


TASTY ON FHIR[®] [©]

CLINICAL NAMED ENTITY LINKING

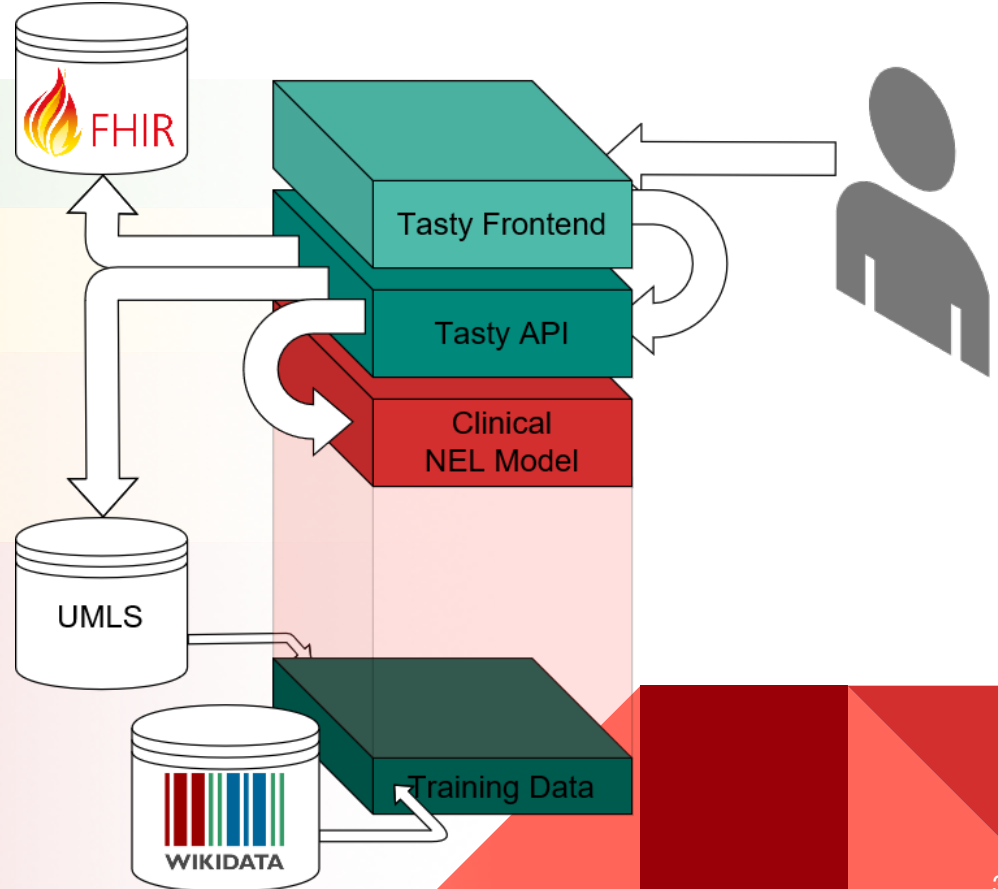
Julius Brückner
Kevin Mattutat
Arnold Schwarz

Das große Ganze



Architektur & Prozess

1. Trainingsdaten erzeugen
2. NEL-Model trainieren
3. Tasty API mit FHIR
4. Tasty Frontend mit FHIR



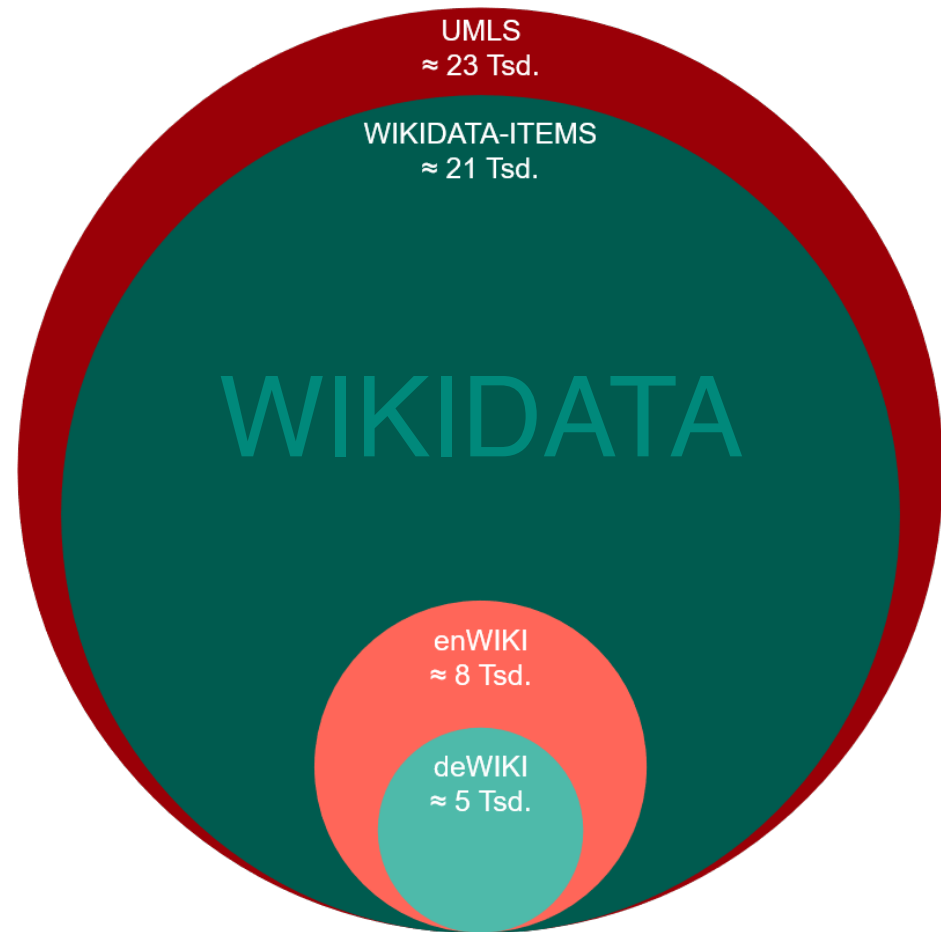
Wie wir vorgehen

- UMLS Daten zugänglich machen
- WIKIDATA Daten extrahieren
- Erzeugung von Trainingsdaten aus UMLS und WIKIDATA
- Training des neuronalen Netzes
- Implementierung der FHIR API

UMLS

Unified Medical Language System

- Kompendium vieler kontrollierter Vokabulare in der Medizin
- Mapping-Struktur zwischen diesen Vokabularen



```
SELECT (COUNT(DISTINCT ?item) AS ?itemCount)
```

- alle UMLS Identifier die gesetzt wurden
- alle Wikidata Items die die Eigenschaft “UMLS CUI” besitzen

```
?item wdt:P2892 ?umls .
```

- alle Englische und Deutsche Wikipedia Artikel die ein Wikidata Item behandeln

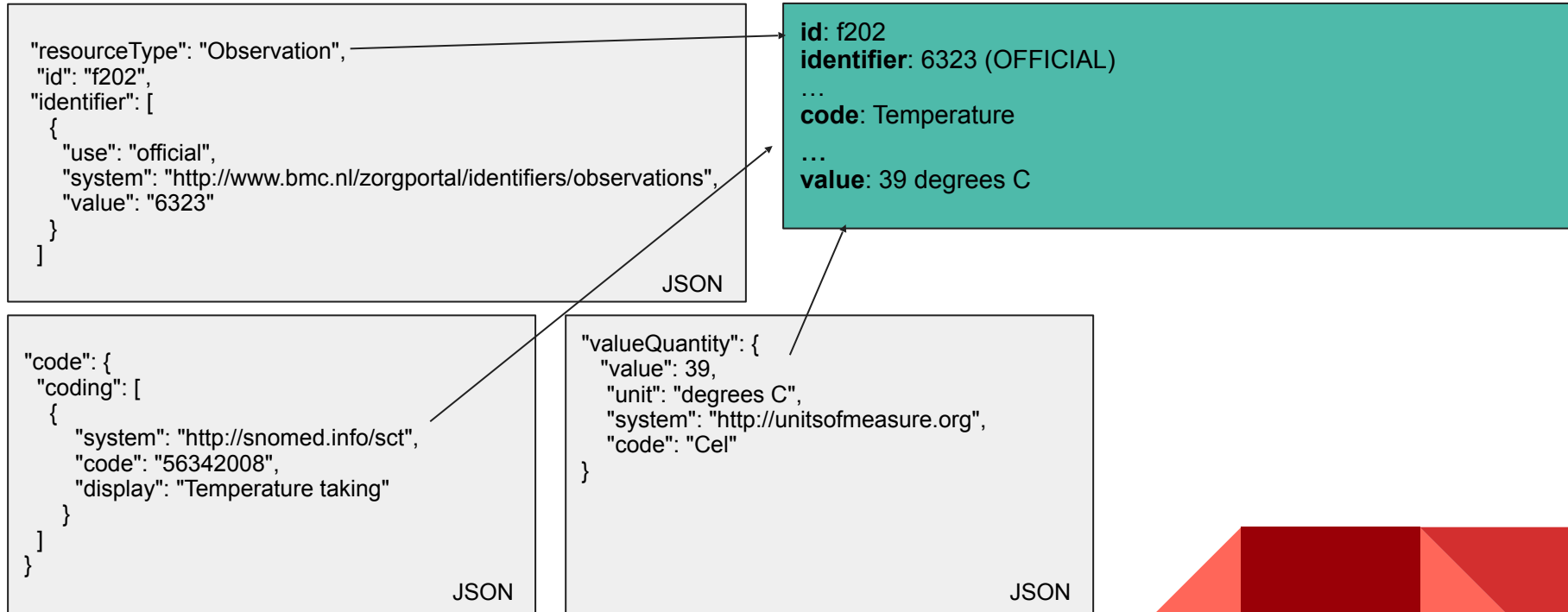
```
OPTIONAL {  
  ?articleEN schema:about ?item .  
  ?articleEN schema:isPartOf <https://en.wikipedia.org/> .  
}
```

FHIR

Fast Healthcare Interoperability Resources

- Funktionserweiterungen durch Third-Parties
- Ressourcen sind nur teilweise implementierbar
- JSON Format
- Erweiterung der bisherigen Schnittstelle

Beispiel



Warum wir scheitern

- Trainingsdaten aus UMLS und WIKIDATA sind unrealistisch
- Patientenakten sind immer unterschiedlich geschrieben und enthalten mehrere Unregelmäßigkeiten
- Patientenakten sind nicht gelabelt

Wie wir unseren Erfolg messen

- Ziel: Precision = 60 %
- Recall muss noch definiert werden
- Kommunikation mit einem FHIR kompatiblen Gerät