Google Home Final Presentation

Mehmet Kardan, Hanna Köb, Mathias Meinschad, Daniel Linter

University of Innsbruck - STI

June 22, 2020

Agenda

1. Introduction to Google Home
   - Device Types & Traits
   - Execution Path
2. Introduction to Dialogflow
   - Intents
   - Entities
   - Architecture
3. Implementation
   - Entities Extraction
   - Handling Intents
     - What Type Question
     - Difference Type Question
     - List Type Question
     - Example Type Question
     - Step Type Question
   - Problems
4. Live Demo

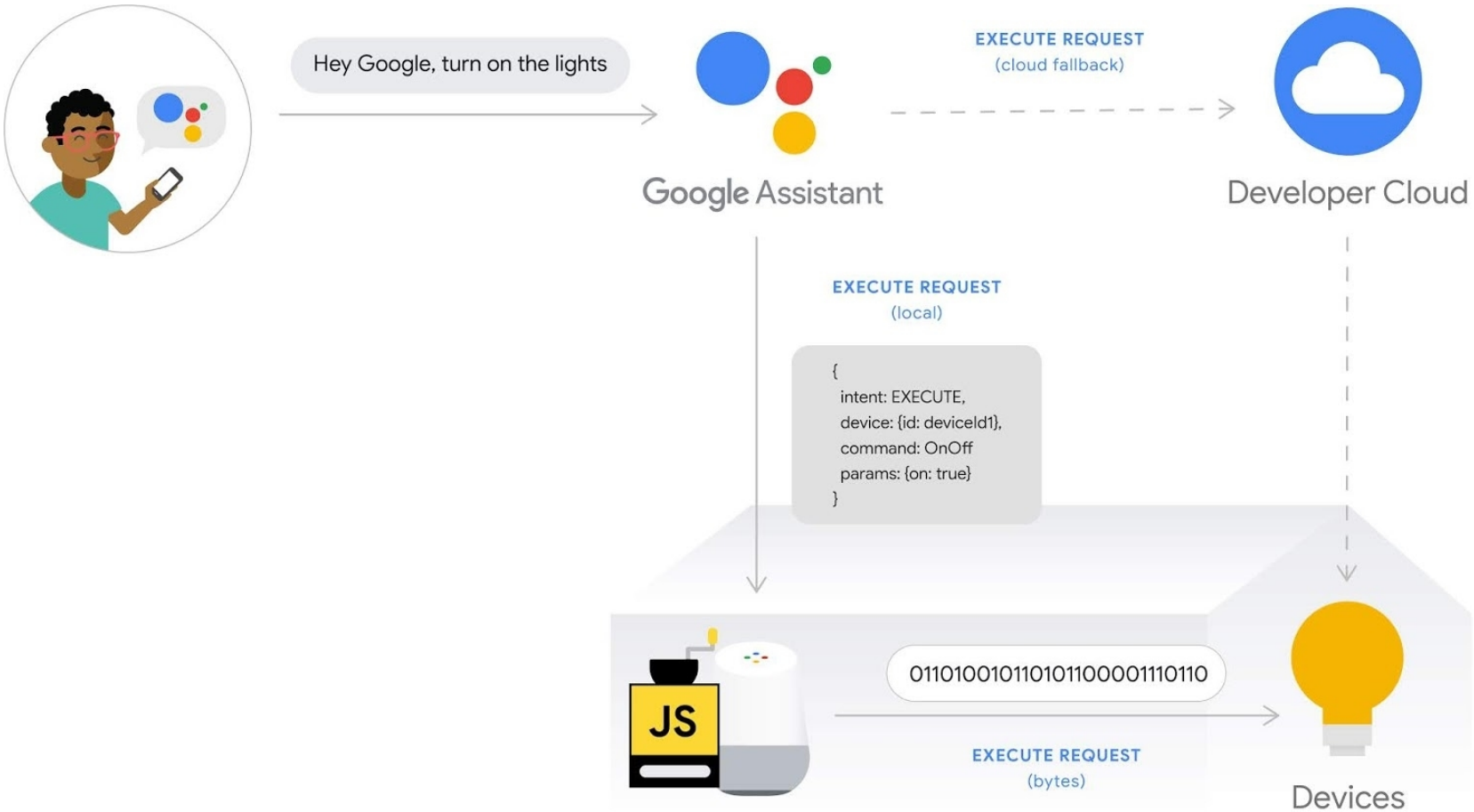
Introduction to Google Home

- Founded by Google in 2016
- Development through Googles developer console and Dialogflow
- Creating skills pretty easy
- No programming skills required

Device Types & Traits

- Various device types (air purifier to yogurt maker)
- Capabilities of a device $\Rightarrow$ traits

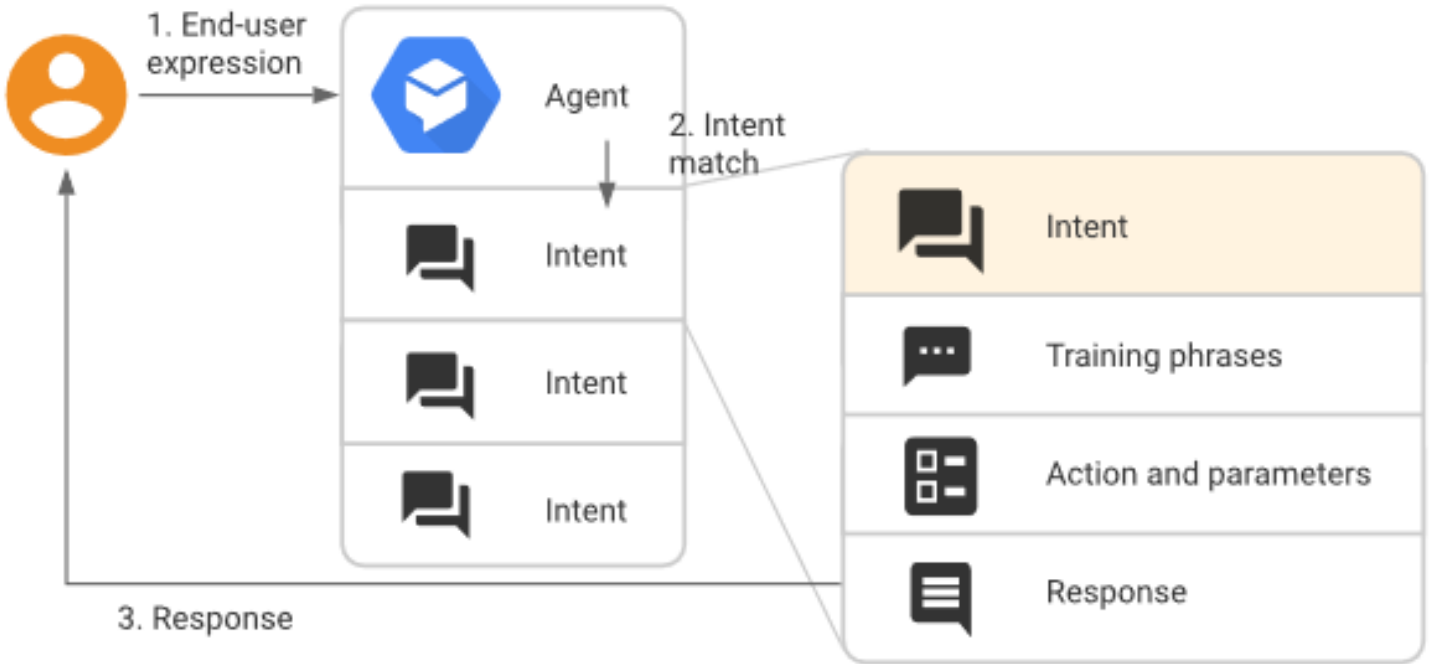
Execution Path



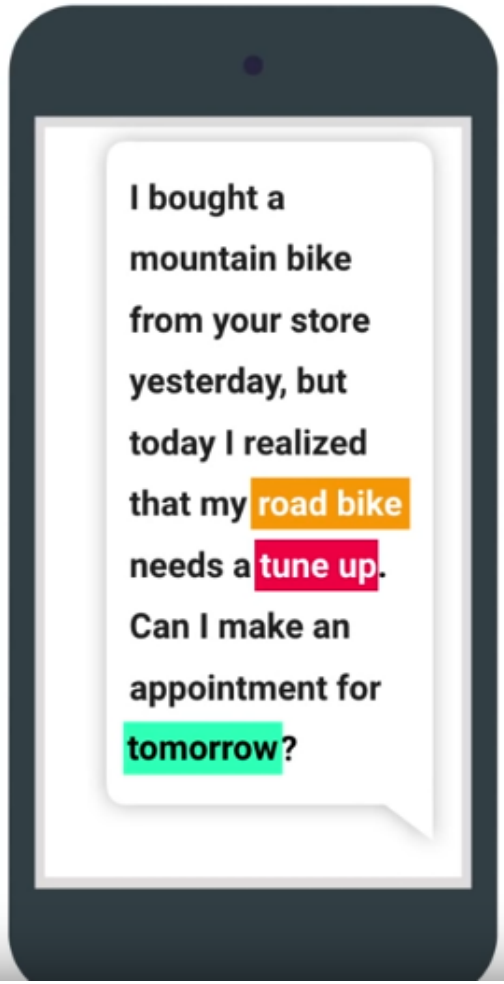
Introduction to Dialogflow

- Service developed and provided by Google
- Natural language tool to create conversational user interfaces for apps, chatbots, etc.
- By adding 'Training phrases' Dialogflow automatically trains the machine learning model

Intents


1. End-user expression
Agent
2. Intent match
Intent
Intent
Intent
Intent
Training phrases
Action and parameters
Response
3. Response

Entities



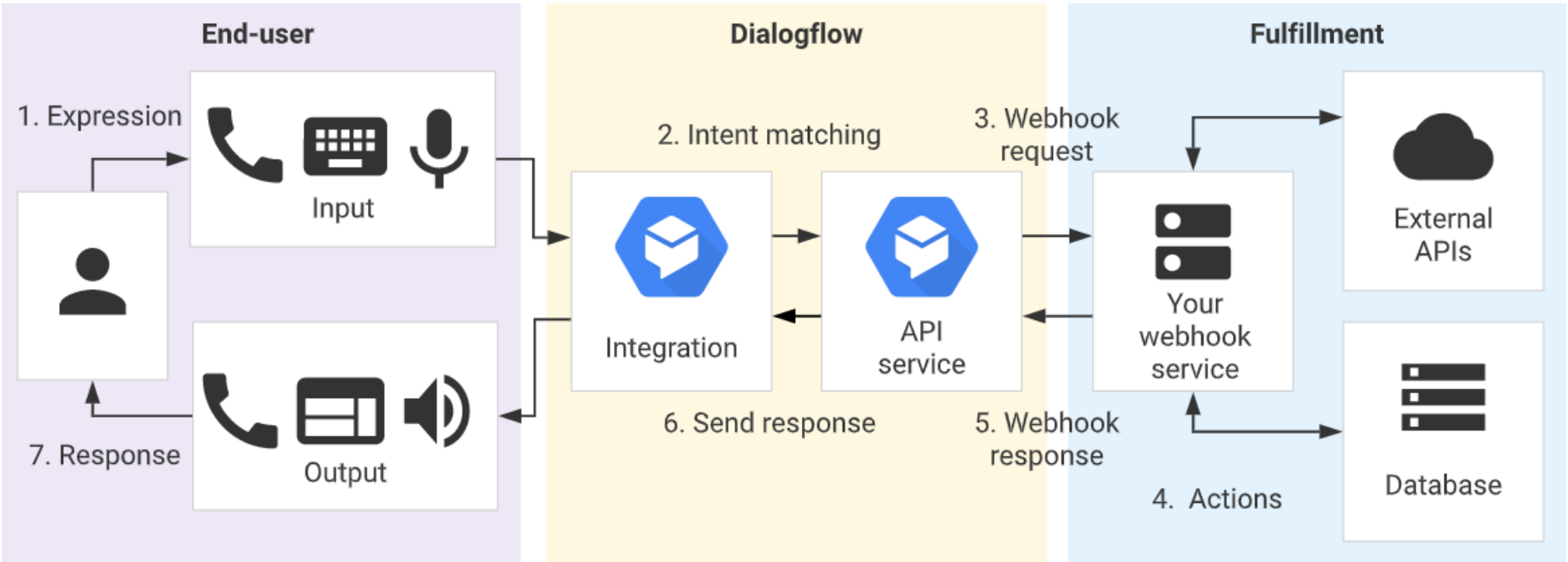
road bike
tune up
tomorrow
2017-11-09

System entities
@sys.time
@sys.date

Developer entities
@service-option
- Tune up
- Repair
- Tire change
- Upgrade

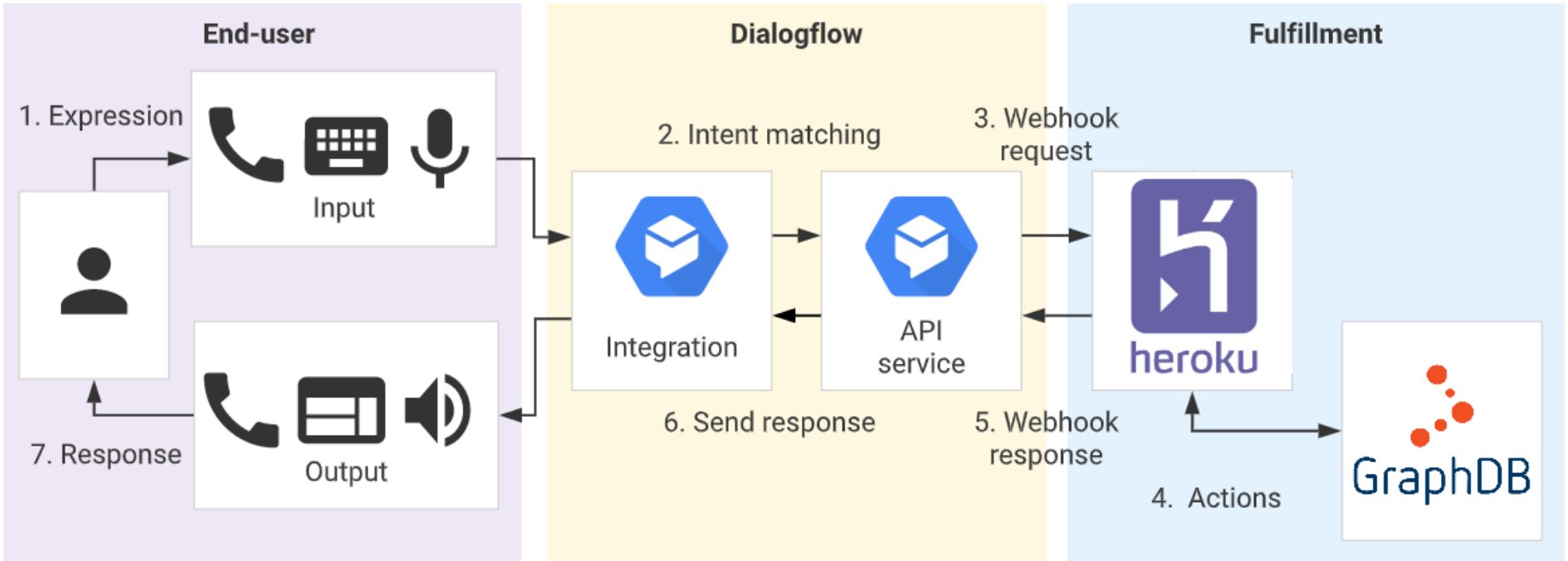
@bike-type
- Road bike
- Mountain bike
- Beach cruiser
- Racing bike
- Fixed gear bike
- Cross bike

Architecture



Implementation

- Coding in JavaScript



Entities Extraction

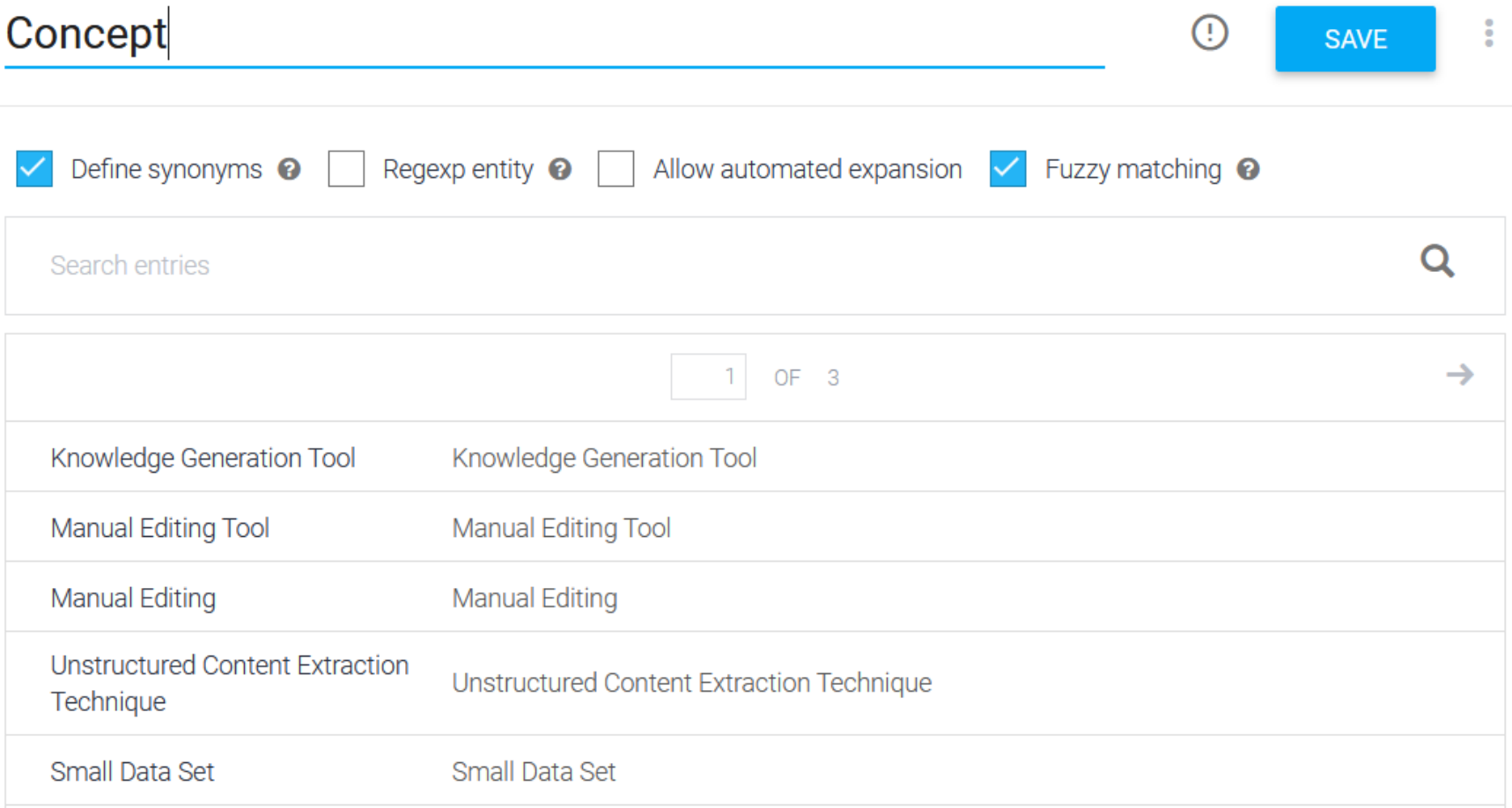
- An entity for each class in the knowledge graph is created

```
SELECT DISTINCT ?class
WHERE {
        ?s a ?class .
}
```

- Then schema.org's property *name* is used to fill the entities with values

```
PREFIX sc: <http://schema.org/>
SELECT ?name
WHERE {
        ?resource  a [entityTypeName] .
        ?resource sc:name ?name .
}
```

Entities Extraction cont'd



What Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description ?purpose where {
    {
        ?Concept schema:name ?name.
        OPTIONAL { ?Concept schema:description ?description . }
        OPTIONAL { ?Concept kgbs:purpose ?purpose . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
    union
    {
        ?Concept schema:alternateName ?name.
        OPTIONAL { ?Concept schema:description ?description . }
        OPTIONAL { ?Concept kgbs:purpose ?purpose . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
}
```

Difference Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description where {
    {
        ?Concept schema:name ?name
        OPTIONAL {?Concept kgbs:differsFrom ?relatesTo.}
        OPTIONAL {?relatesTo schema:description ?description.}
        filter (LCASE(?name) = LCASE("${first_parameter}") || LCASE(?name) = LCASE("${second_parameter}"))
    }
    UNION
    {
        ?Concept schema:alternateName ?name
        OPTIONAL {?Concept kgbs:differsFrom ?relatesTo.}
        OPTIONAL {?relatesTo schema:description ?description.}
        filter (LCASE(?name) = LCASE("${first_parameter}") || LCASE(?name) = LCASE("${second_parameter}"))
    }
}
```

List Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

select ?description where {
    {
        ?Concept schema:name ?name
        OPTIONAL {?Concept skos:narrower ?specialization.}
        OPTIONAL {?specialization schema:name ?description.}
        filter (LCASE(?name) = LCASE("${parameter}")) .
    }
    union
    {
        ?Concept schema:alternateName ?name
        OPTIONAL {?Concept skos:narrower ?specialization.}
        OPTIONAL {?specialization schema:name ?description.}
        filter (LCASE(?name) = LCASE("${parameter}")) .
    }
}
```

Example Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
select ?description where {
    {
        ?Concept schema:name ?name.
        optional { ?Concept skos:example ?example . }
        optional {?example schema:description ?description . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
    UNION
    {
        ?Concept schema:alternateName ?name.
        optional { ?Concept skos:example ?example . }
        optional {?example schema:description ?description . }
        filter (LCASE(?name) = LCASE("${parameter}"))
    }
}
```

Step Type Question

```
PREFIX schema: <http://schema.org/>
PREFIX kgbs: <http://knowledgegraphbook.ai/schema/>

select ?description where {
    {
        ?Concept schema:name ?name .
        ?Concept schema:step: ?Object .
        OPTIONAL { ?Object schema:text ?description . }
        filter contains (LCASE(?name), LCASE("${parameter}")) .
    }
    UNION
    {
        ?Concept schema:alternateName ?name .
        ?Concept schema:step: ?Object .
        OPTIONAL { ?Object schema:text ?description . }
        filter contains (LCASE(?name), LCASE("${parameter}")) .
    }
}
```

Problems

- Changing namespaces of GraphDB
- No JavaScript library for GraphDB with authentication

Live Demo

Thank you for your attention!