Technische Hochschule Augsburg

BRUNATA METRONA
Einfach. Mehr. Qualität.

KI-gestützte Generierung dynamischer Webseiteninhalte

Bachelorarbeit Informatik

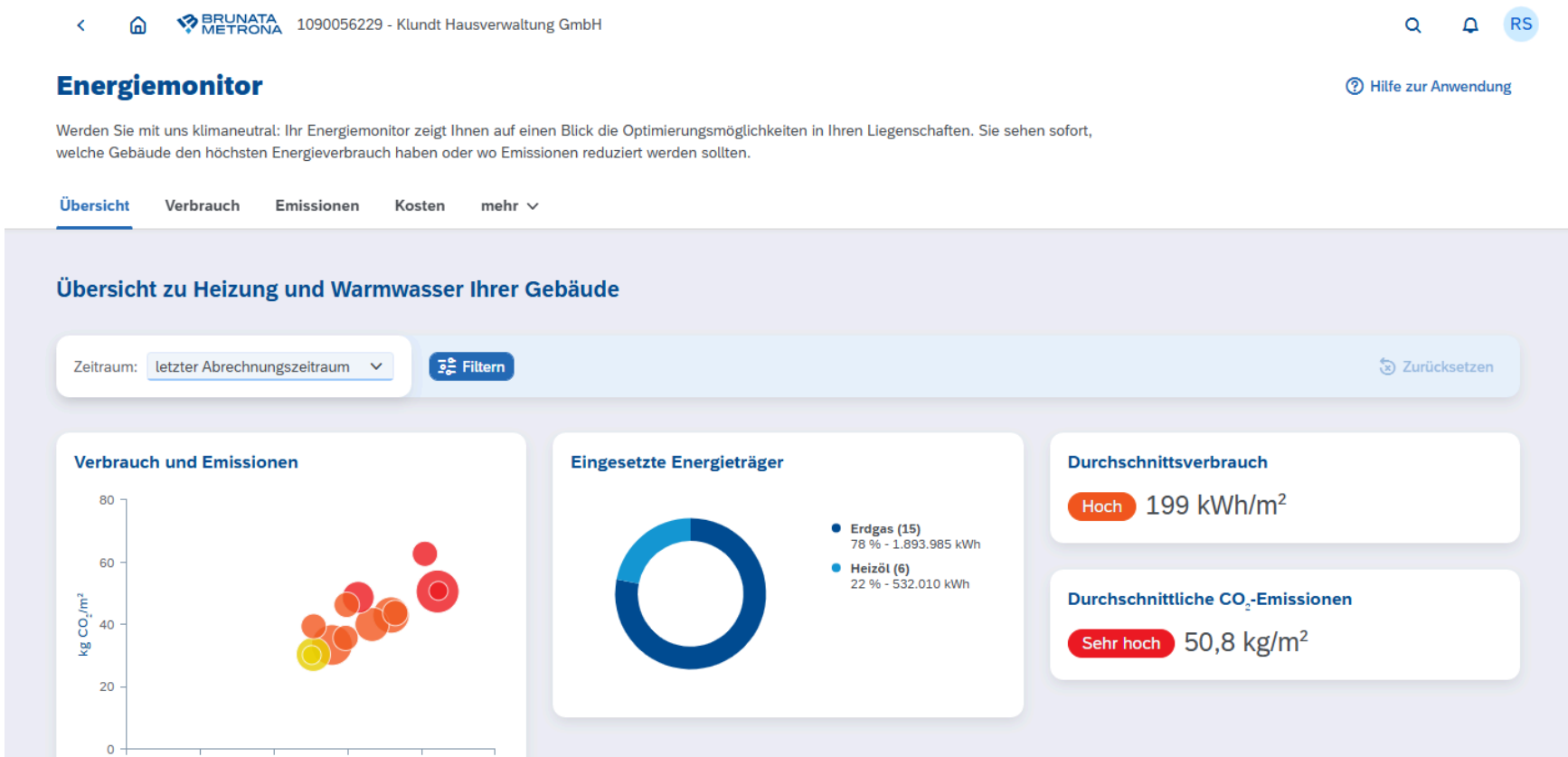
Rajbir Singh – 23.01.2025

Agenda

1. Motivation
2. Problem
3. Methodik
4. Umsetzung
5. Fazit

Motivation

Energiemonitor



Problem

Problem

- Unterschiedliche Anforderungen der Benutzer
- Komplexität des Zugangs zu digitalen Inhalten
- Mangel an flexiblen und personalisierbaren Benutzeroberflächen

Methodik

Integrationsansätze

Generierung von UI5-Code

Integrationsansätze

Generierung von HTML-Code

Integrationsansätze

Modifikation bestehender UI-Elemente

Integrationsansätze

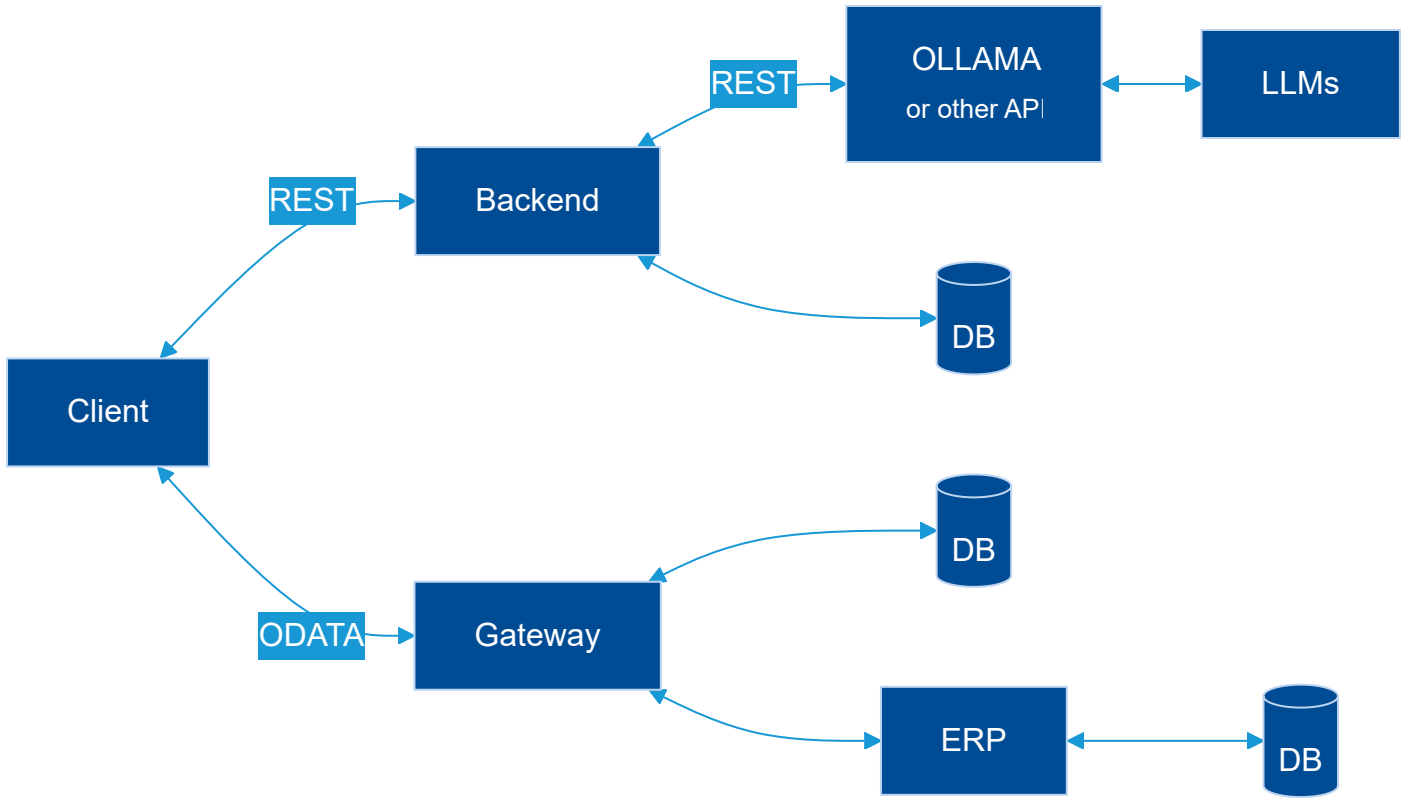
Auswahl vordefinierter Fragments und APIs

Integrationsansätze

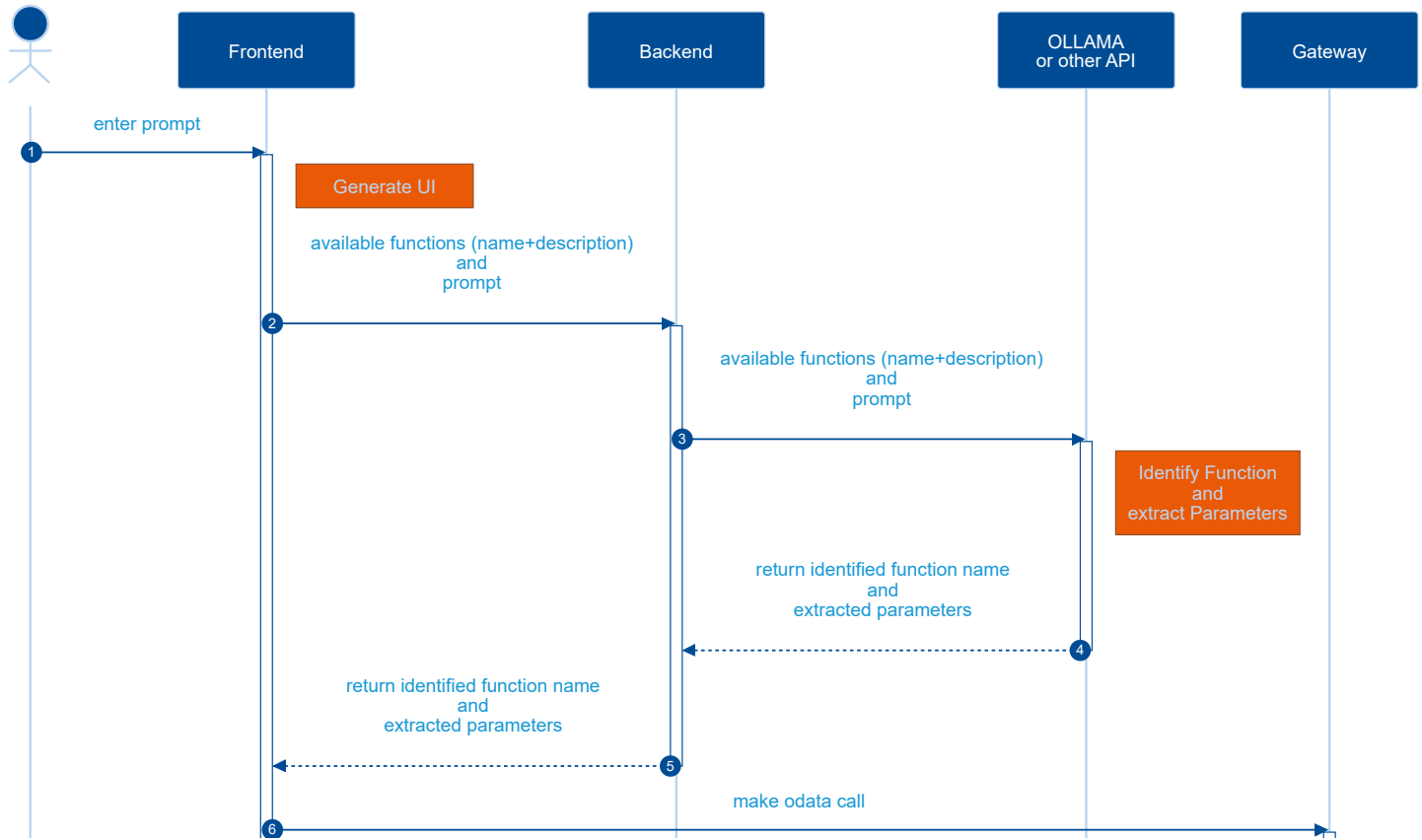
Dynamische Generierung der UI basierend auf API-Antworten

Technologien

Architektur

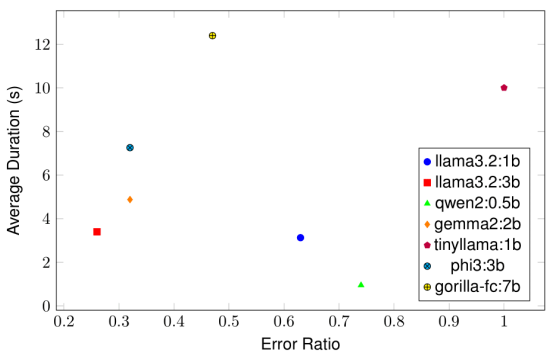


Datenfluss

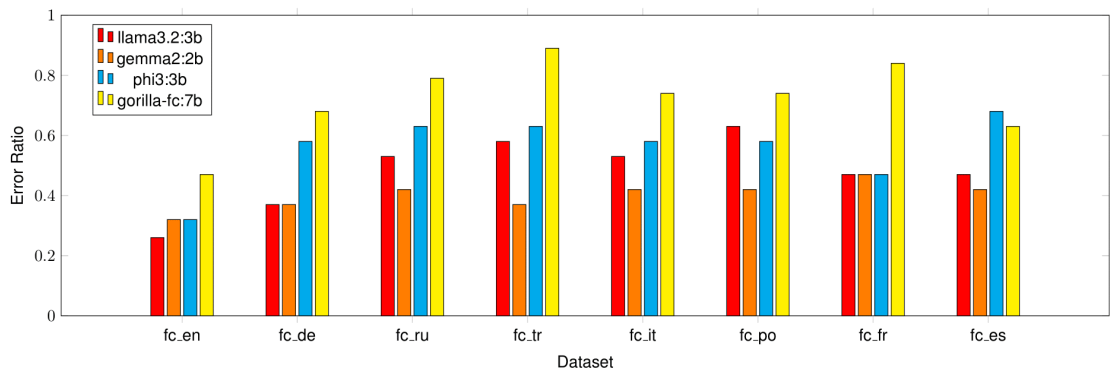

Frontend
Backend
OLLAMA
or other API
Gateway
enter prompt
1
Generate UI
available functions (name+description)
and
prompt
2
available functions (name+description)
and
prompt
3
Identify Function
and
extract Parameters
return identified function name
and
extracted parameters
4
return identified function name
and
extracted parameters
5
make odata call
6

Modelle

fc



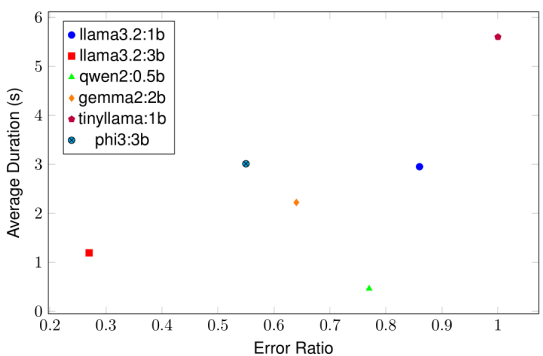
Englisch



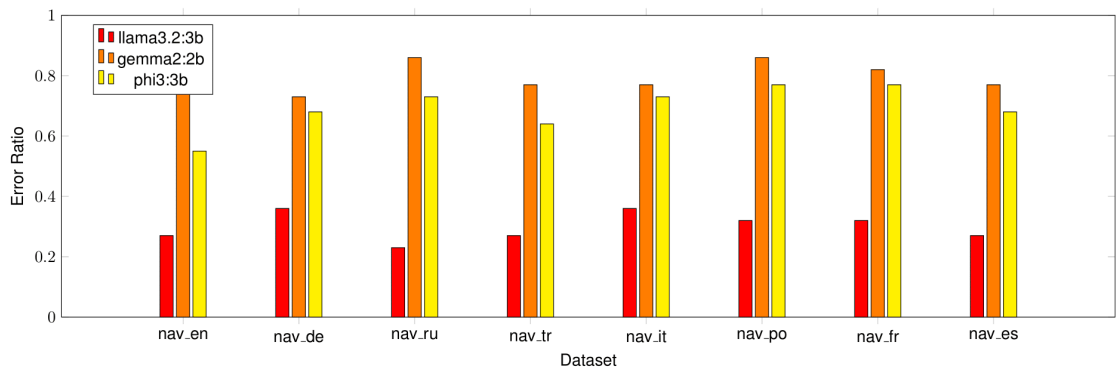
Deutsch

Modelle

nav



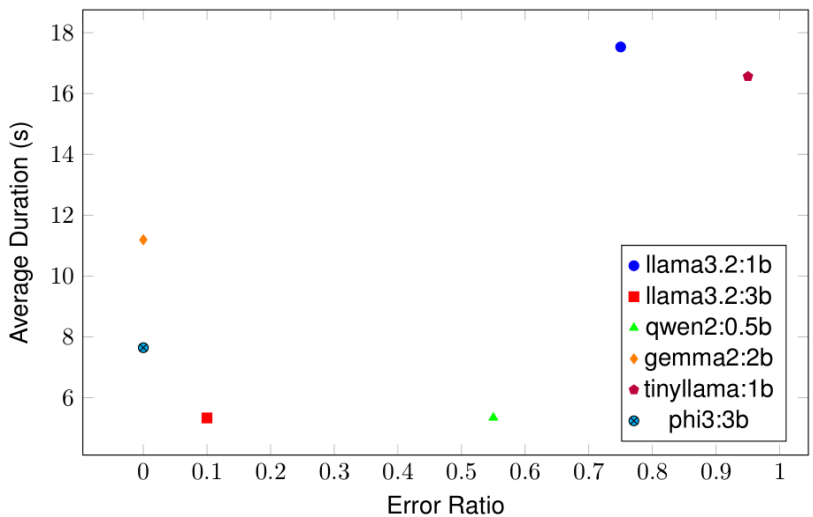
Englisch



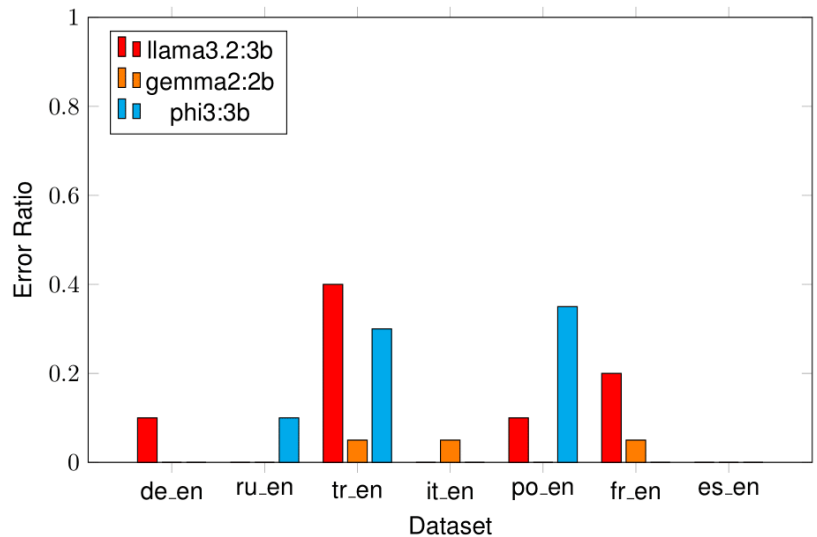
Deutsch

Modelle

translation



Englisch



Deutsch

Frontend Konzept

1_1

{"companyName": "TechCorp", "location": "NewYork", "isHiring": true, "departments": ["Engineering", "HR", "Sales", "Marketing"], "employees": [{"employeeId": 1001, "name": "AliceJohnson", "external": "true", "salary": 85000, "contactInfo": {"email": "alice.johnson@techcorp.com", "phone": "123-456-7890"}}, {"employeeId": 1002, "name": "BobSmith", "external": "false", "salary": 75000, "contactInfo": {"email": "bob.smith@techcorp.com", "phone": "987-654-3210"}}, {"employeeId": 1003, "name": "CarolWhite", "external": "false", "salary": 68000, "contactInfo": {"email": "carol.white@techcorp.com", "phone": "555-123-4567"}}], "__metadata": "someirrelevantdata", "companyId": "12345"}

Frontend Konzept

1_2

```
13      "employees":[
14         {
15            "employeeId":1001,
16            "name":"AliceJohnson",
17            "external":"true",
18            "salary":85000,
19            "contactInfo":{
20               "email":"alice.johnson@techcorp.com",
21               "phone":"123-456-7890"
22            }
23         },
```

Frontend Konzept

2

getCompany

companyName	TechCorp
location	New York
isHiring	true
departments	[object],[object],...
employees	[object],[object],...
__metadata	some irrelevant data
companyId	12345

employees

employeeId	name	external	salary	contactInfo
1001	Alice Johnson	true	85000	[object]
1002	Bob Smith	false	75000	[object]
1003	Carol White	false	68000	[object]

Frontend Konzept

3

getCompany

companyName	TechCorp
location	New York
isHiring	true
departments	▸
employees	▸
__metadata	some irrelevant data
companyId	12345

employees

employeeId	name	external	salary	contactInfo
1001	Alice Johnson	true	85000	›
1002	Bob Smith	false	75000	›
1003	Carol White	false	68000	›

Frontend Konzept

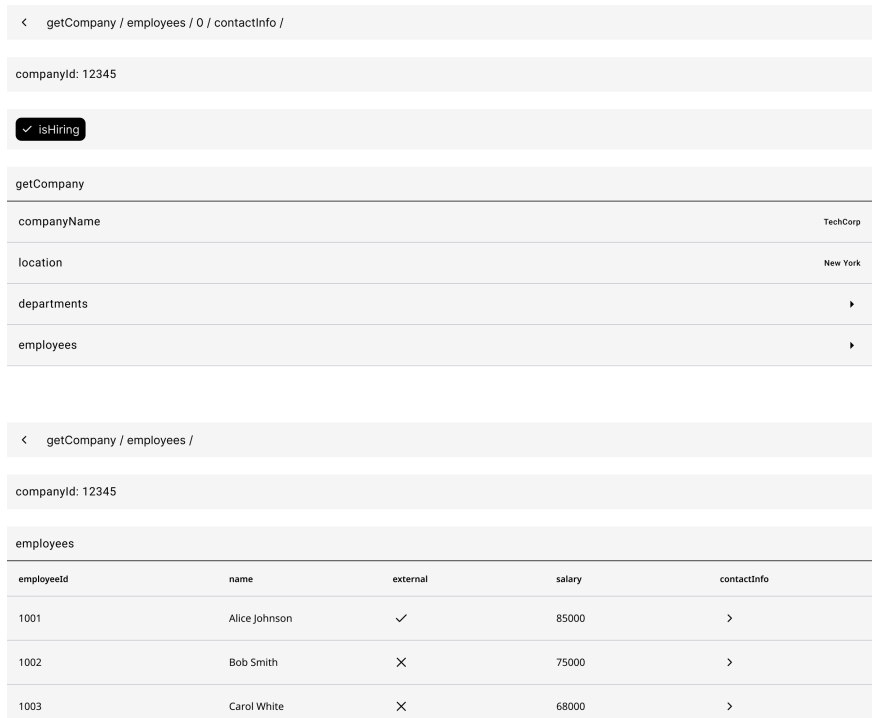
4

< getCompany / employees / 0 / contactInfo /

getCompany	
email	alice.johnson@techcorp.com
phone	123-456-7890

Frontend Konzept

5

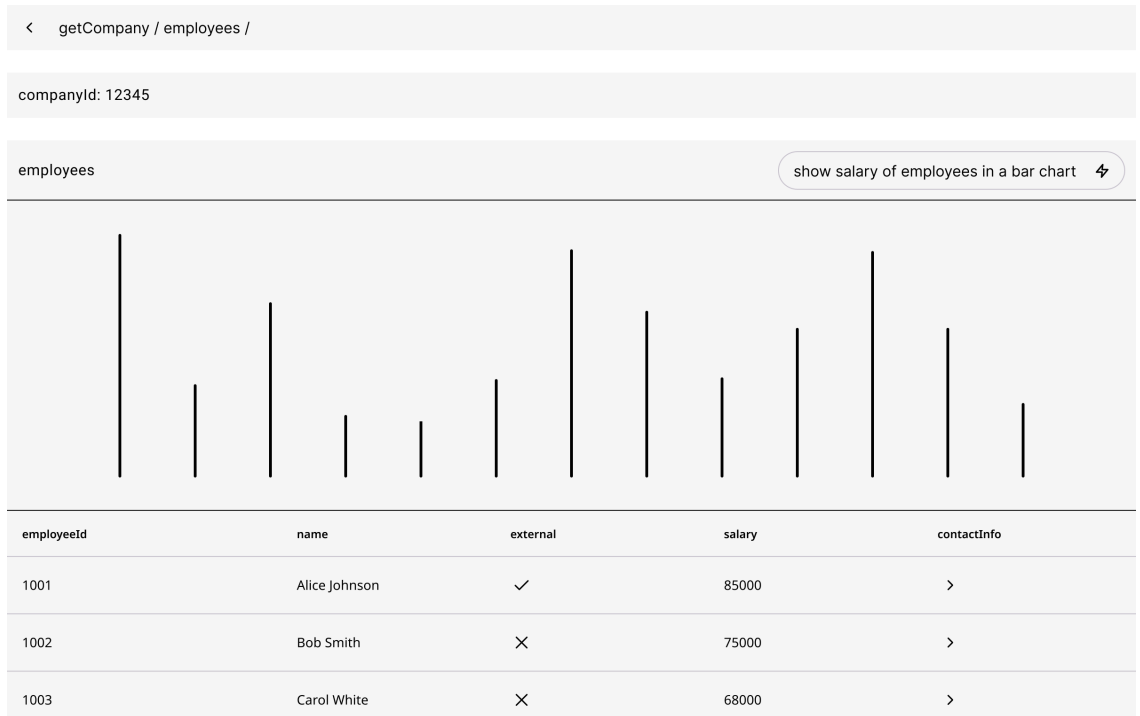


Frontend Konzept

6

Frontend Konzept

7



Umsetzung

Backend

- SAP Backend
- NGINX Backend
- SQLite Datenbank
- Kommuniziert mit OLLAMA (oder anderen APIs)
- Speichert User und deren Verlauf

Frontend

- MVC
- ...

Code Snippets

function calling

```
1   const fn = await this._getFn(prompt);
2   const data = await this[fn.name](fn.parameters);
3   const path = await this._getPath(data, prompt);
```

Code Snippets

CustomNavJSON

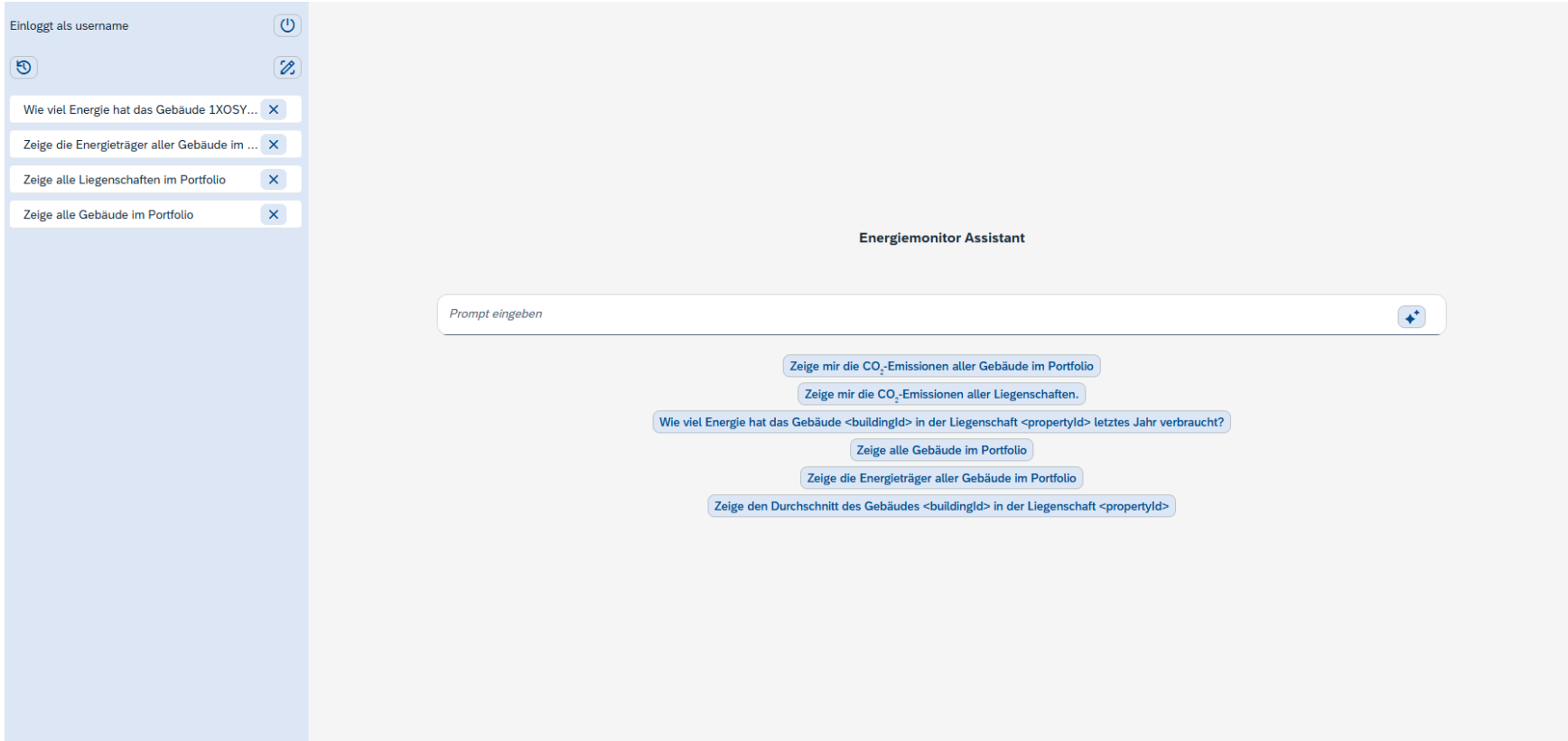
```
<custom:CustomNavJSON
    id="idCustomNavJSON"
    title="{currentData>/title}"
    params="{currentData>/params}"
    data="{currentData>/data}"
    path="{currentData>/path}"
>
    <custom:noData>
        <core:Fragment
            fragmentName="energiemonitor.fragment.NoData"
            type="XML"
        />
    </custom:noData>
</custom:CustomNavJSON>
```

Code Snippets

chart

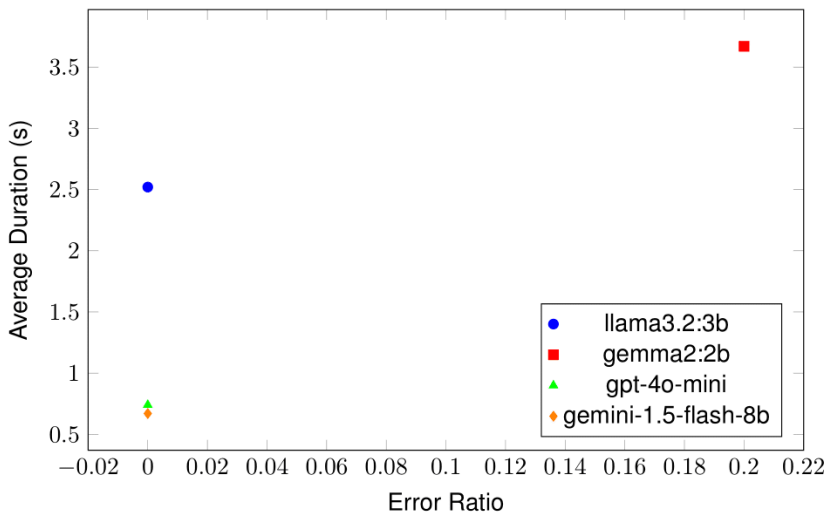
```
16
17  this._createChartConfig(chartResp, extractedData);
18
19  this._setOptions(chartResp);
20
21  this.destroyChart();
22
23  this._chart = new Chart(
24      document.getElementById("chartCanvas").getContext("2d"),
25      this._chartConfig
26  );
```

Demo

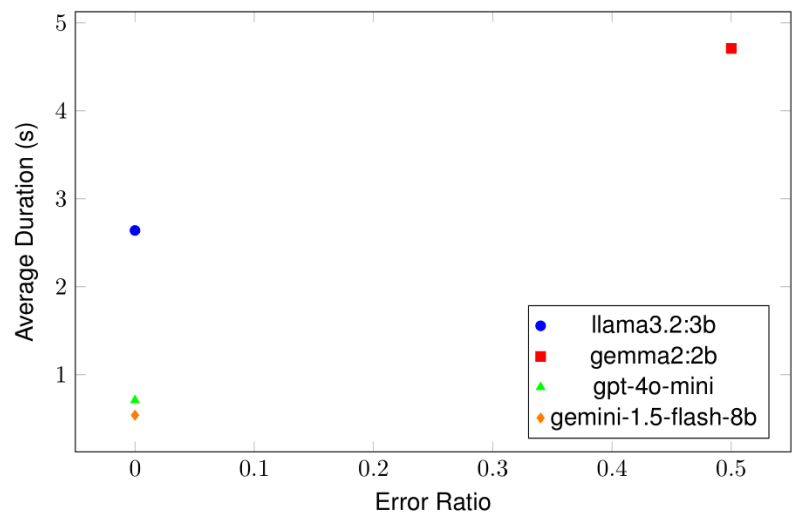

Einloggt als username
Wie viel Energie hat das Gebäude 1XOSY...
Zeige die Energieträger aller Gebäude im ...
Zeige alle Liegenschaften im Portfolio
Zeige alle Gebäude im Portfolio
Energiemonitor Assistant
Prompt eingeben
Zeige mir die CO2-Emissionen aller Gebäude im Portfolio
Zeige mir die CO2-Emissionen aller Liegenschaften.
Wie viel Energie hat das Gebäude <buildingId> in der Liegenschaft <propertyId> letztes Jahr verbraucht?
Zeige alle Gebäude im Portfolio
Zeige die Energieträger aller Gebäude im Portfolio
Zeige den Durchschnitt des Gebäudes <buildingId> in der Liegenschaft <propertyId>

Benchmark

function



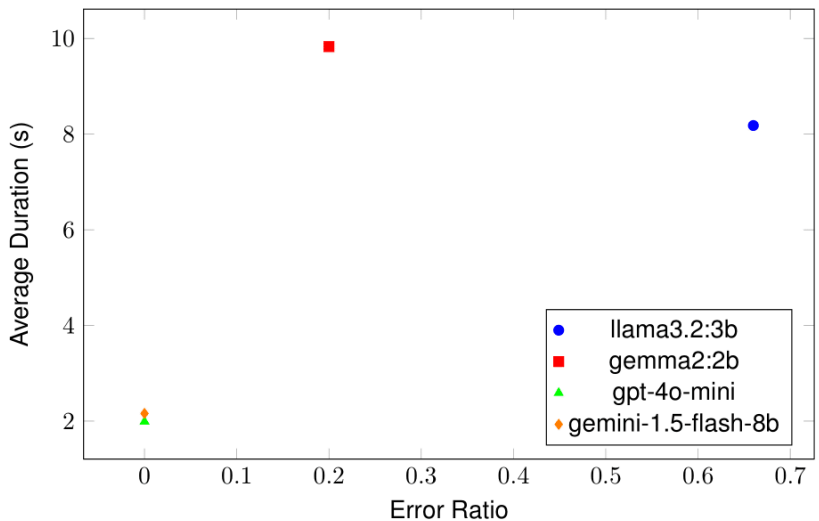
Englisch



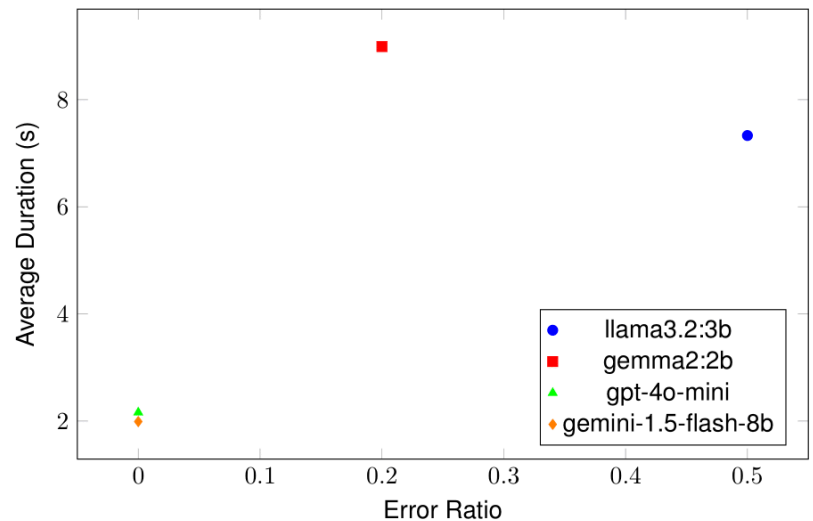
Deutsch

Benchmark

path



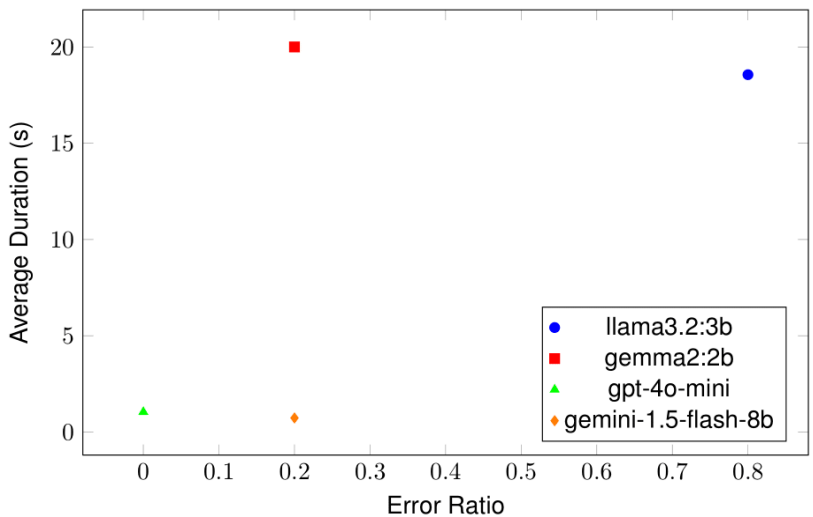
Englisch



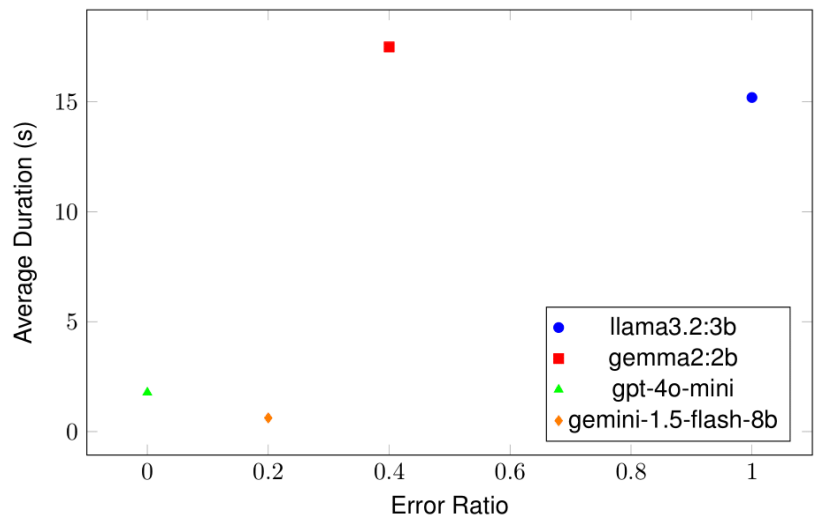
Deutsch

Benchmark

chart



Englisch



Deutsch

Fazit

Fazit

- Integration von LLMs in Webanwendungen bietet erheblichen Mehrwert
- Nur das nötigste von LLMs generieren lassen, so wenig wie möglich
- LLMs machen Fehler
- Fehler müssen mit Programmierlogik erkannt und behoben werden
- Leistung der LLMs hängt von eingesetzter Hardware ab

Optimierungsmöglichkeiten

- JSON Responses anpassen - Design kann im Backend festgelegt werden
- Vorschaufunktion bei verschachtelten Objekten
- Keys der JSON Responses als Keys für i18n-Texte verwenden
- Möglichkeit UI-Elemente auf Startseite zu fixieren
- Dem User die Möglichkeit geben eigene LLMs einzubinden (Datenschutz)
- KI-APIs vor Missbrauch schützen (Rate Limiting)

Ausblick

- Tabellen und Listen um zusätzlich KI-Features erweitern
- Konzept für POST-, UPDATE- und DELETE-Operationen
- Fine-Tuning & Prompt-Tuning
- Parallele Anfragen