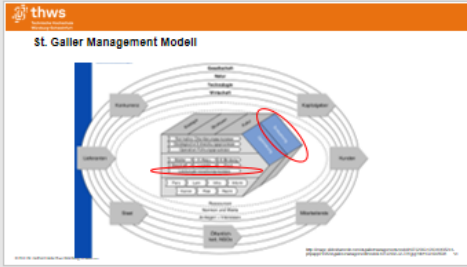




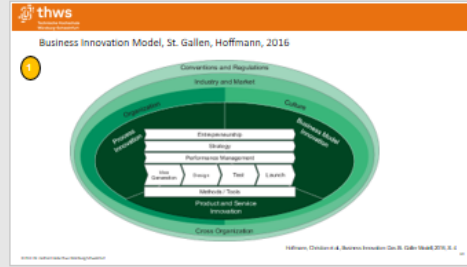
# Innovationsmanagement & KI in der Regionalentwicklung

# Innovation & Innovationsmanagement vielfältig und komplex

# Kleine Auswahl an Ansätzen und Konzepten für ein Innovationsmanagement



59



60



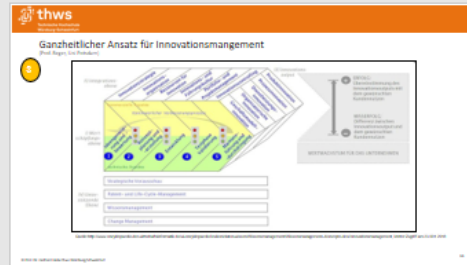
61



62



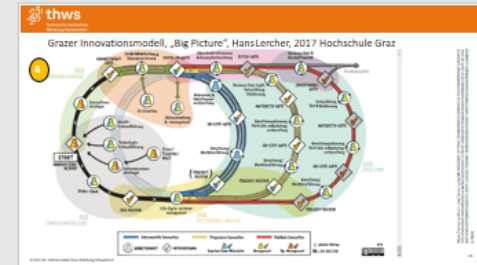
63



64



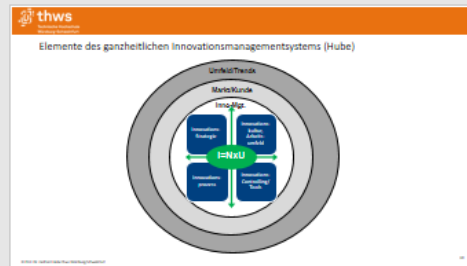
65



66



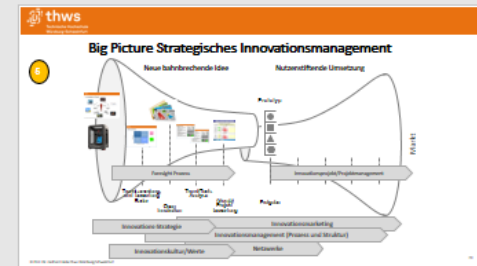
67



68



69



70



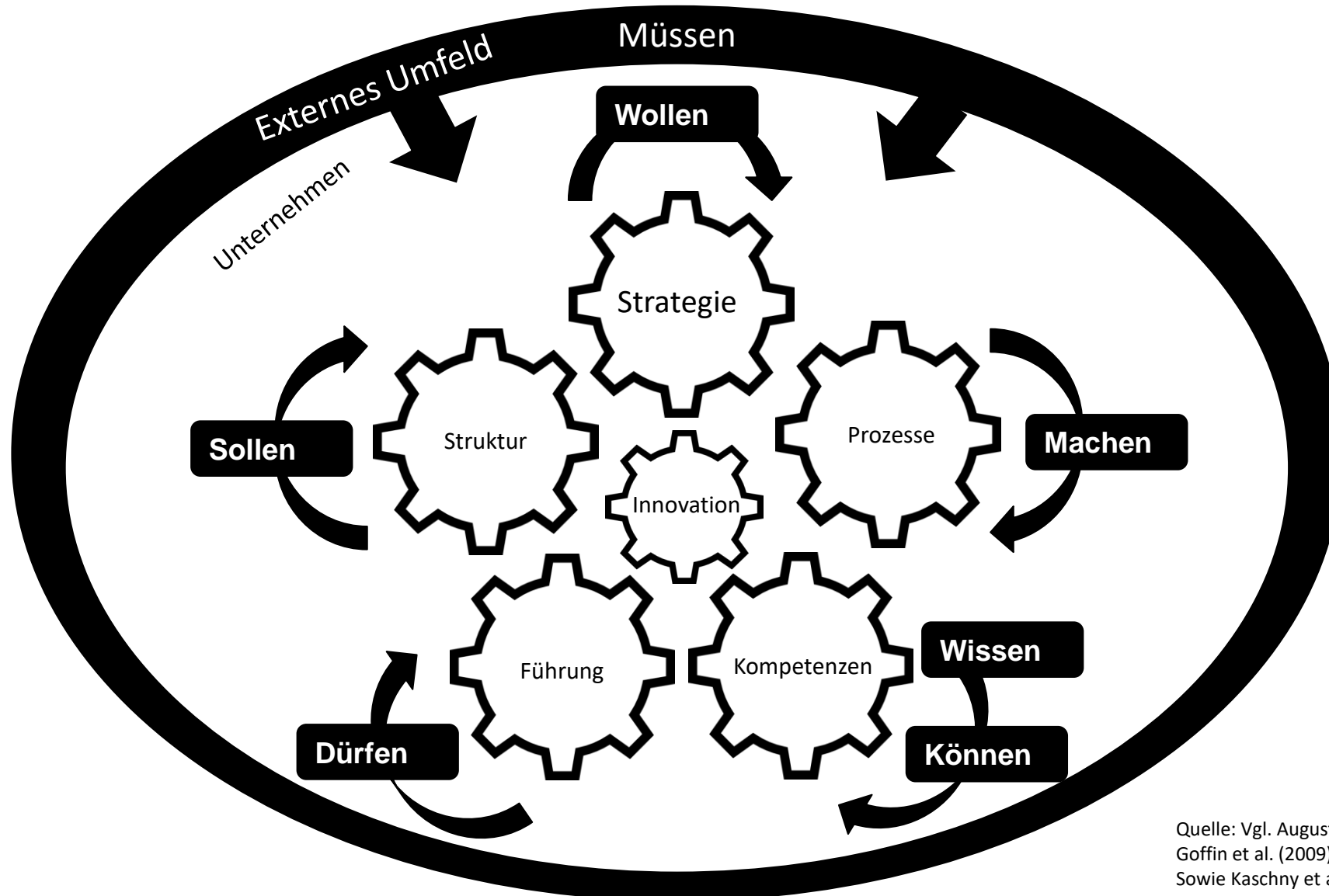
## Stark vereinfachter Ansatz für Innovationsmanagement

“Hire smart people and leave them alone”



Thomas Davenport

## Voraussetzungen für IM nach den Die Big-Five des Innovationskultur-Mixes



Quelle: Vgl. Augusten et al.(2017), S. 4;  
Goffin et al. (2009), S. 446ff.  
Sowie Kaschny et al. (2015), S.32

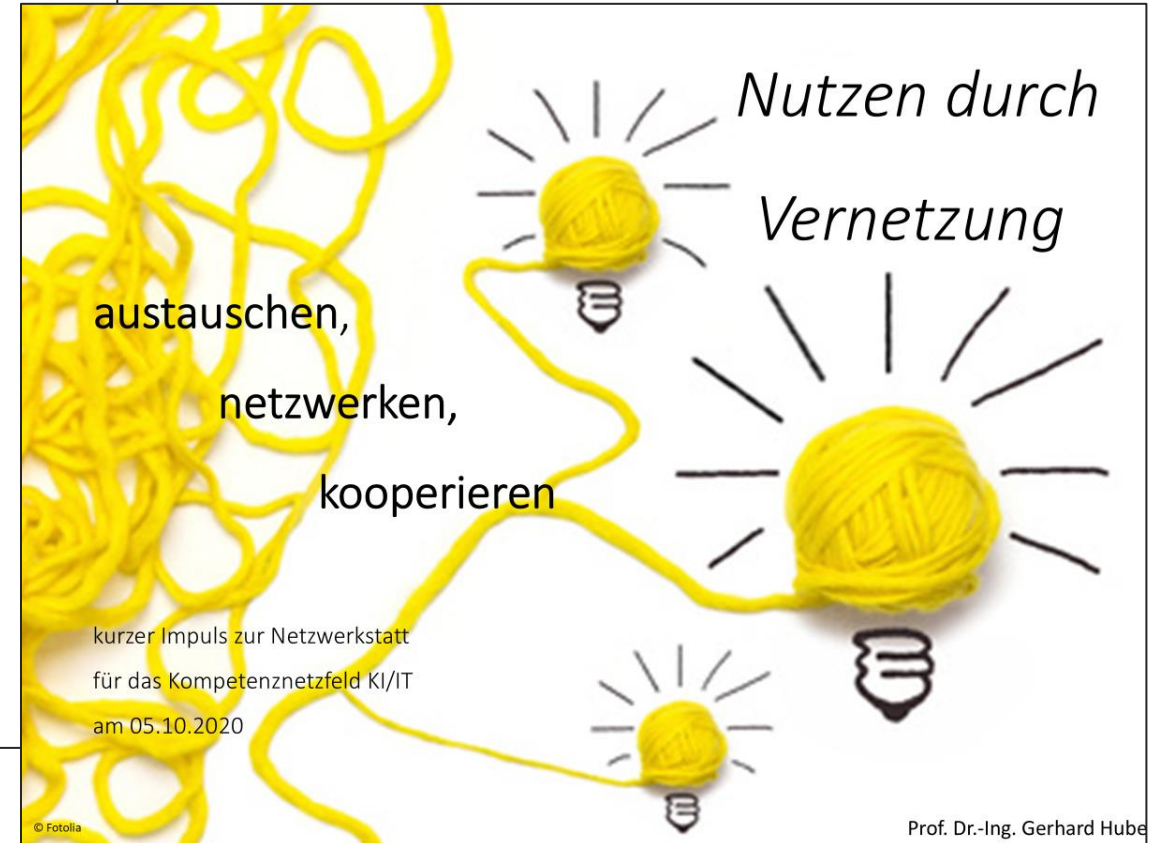
# Innovationsprojekt zu Innovation durch regionalen Netzwerke 2020

## Starkes Netzwerk für Mainfranken!



petersson@mainfranken.org  
weiss@mainfranken.org  
friehs@mainfranken.org

**Region Mainfranken GmbH**  
Regionalentwicklung  
Regionalmarketing  
**www.mainfranken.org**



kurzer Impuls zur Netzwerkstatt  
für das Kompetenznetzfeld KI/IT  
am 05.10.2020

Prof. Dr.-Ing. Gerhard Hube

# Optionen für Kooperationen zu Innovation und KI



Praxisprojekte



Masterarbeiten



I2m-Praxis-Integration  
Kooperationsmöglichkeiten



Beispielunternehmen  
in Lehrveranstaltung



Gastvorträge, Exkursionen



# Beispielunternehmen in Lehrveranstaltung



## Innovationsstrategie GLÖCKLE Bauunternehmung

Szenariobasierte Strategieentwicklung  
Prof. Dr. Gerhard Hube  
05.02.2024

Maria Alt  
Vanessa Sheila Milak  
Gerrik Noe  
Dominik Schmitt



### Szenariobasierte Strategieentwicklung für das Unternehmen Flyeralarm

Szenariobasierte Strategieentwicklung  
Master Integriertes Innovationsmanagement  
WS 2023/24, Prof. Dr. Gerhard Hube

Anna Merz, Lea Fröhlich, Viki Schmidt

05.02.2024

Linde Material Handling



## SZENARIOBASIERTE STRATEGIEENTWICKLUNG FÜR LINDE MATERIAL HANDLING

Christian Drotleff, Monica Genesi, Andrea Gräf,  
Patrick Klein, Luca Slangen

# Innovations- strategie



31.01.2022 –  
BENEDIKT BRAUN,  
HENRY FALLIS,  
JULIA HOFMANN,  
MARTINA SCHWARZ,  
HELENA SEUFFERT

# Beispielunternehmen in Lehrveranstaltung, Kreativitätsworkshops, Design Thinking



„Die Fähigkeit zur Innovation  
entscheidet über unser Schicksal“



Roman Herzog, 1997  
„Ruck“-Rede  
„Aufbruch ins 21. Jahrhundert“

# Innovationsmanagement und Zukunftstechnologien

Innovation ist kein Zufall und kann mit den richtigen Zutaten und der richtigen Strategie Zukunft sichern



**THWS – Business School**

Prof. Dr.-Ing. Gerhard Hube

[gerhard.hube@thws.de](mailto:gerhard.hube@thws.de)

# KI-gestützte Prozessanalyse und -verbesserung

Wie Künstliche Intelligenz hilft, Prozess schneller zu verstehen, zu analysieren und Verbesserungspotenziale zu finden



**THWS – Business School**

Prof. Dr.-Ing. Eva Klenk

[LinkedIn](#)

# Warum Prozessoptimierung?

Der „Happy Path“...



Antrag  
erhalten



Antrag  
fachlich  
prüfen



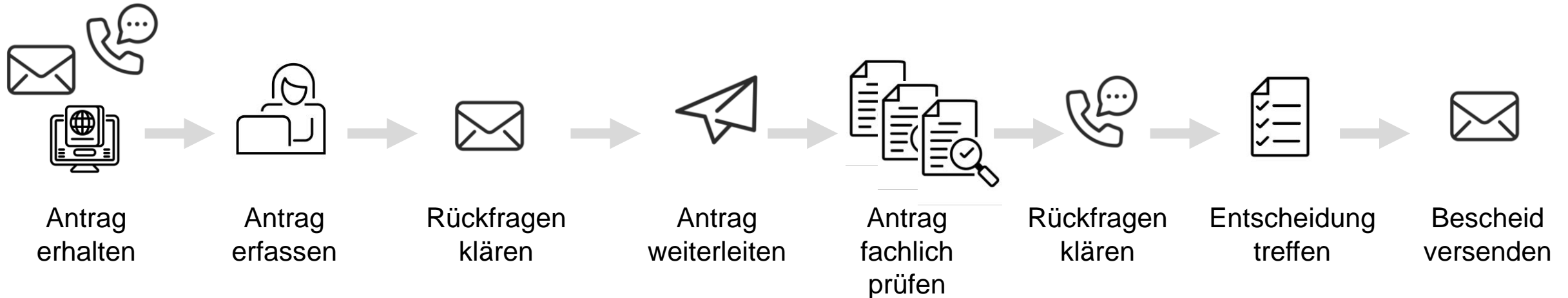
Entscheidung  
treffen



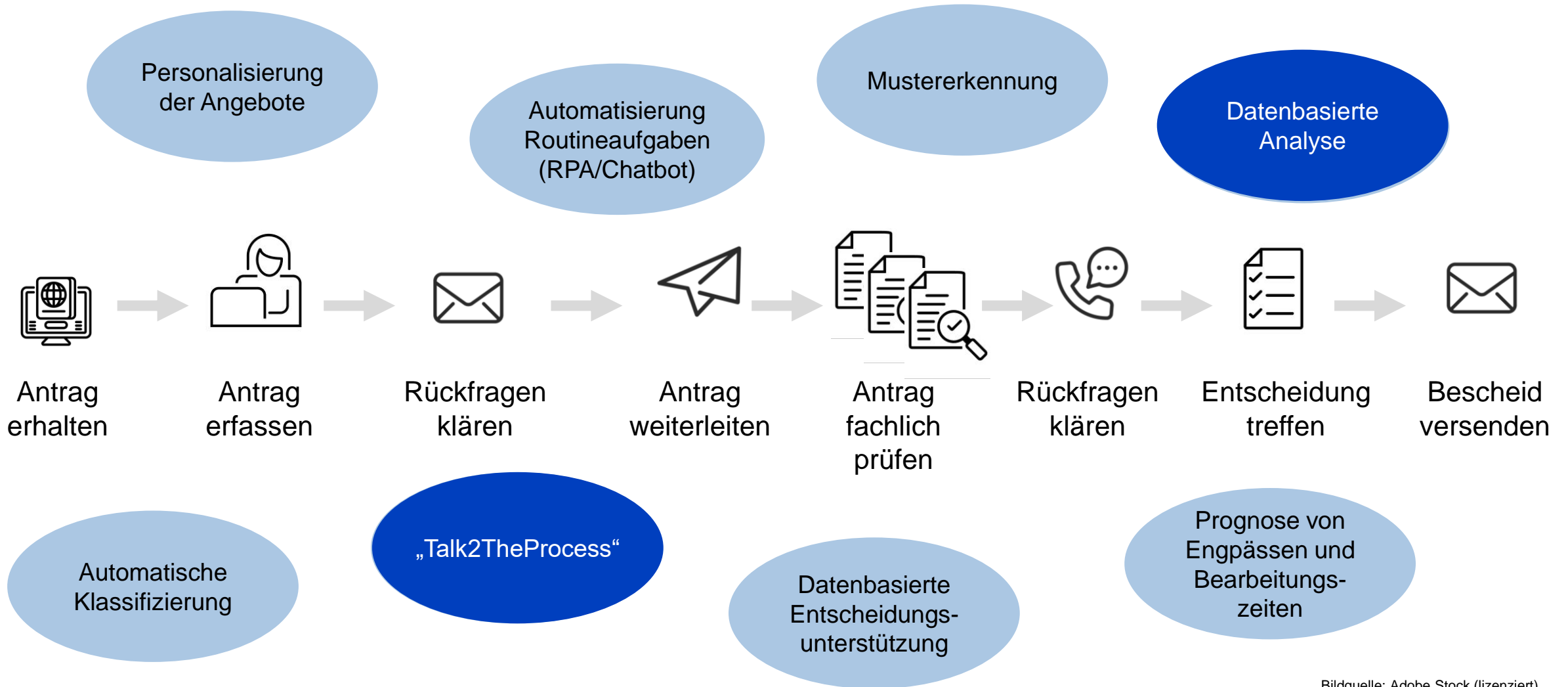
Bescheid  
versenden

# Warum Prozessoptimierung?

... oder so?



# Wie kann KI helfen?





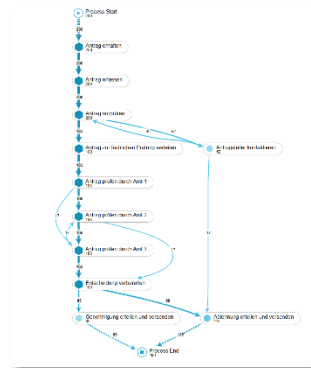
# Wie kann KI helfen?

Datenbasierte Analyse: *Process Mining*

Talk2TheProcess: *LLM und Bilderkennung*

CASE KEY	SORTING	EVENTTIME	ACTIVITY_EN	AUTOMATION	RESOURCE
1	1	2024-03-08 08:00:00	Antrag erhalten	WAHR	System
1	2	2024-03-08 08:00:00	Antrag erfassen	FALSCH	Sachbearbeiter A
1	3	2024-03-08 08:15:00	Antrag vorprüfen	FALSCH	Sachbearbeiter B
1	4	2024-03-08 09:31:00	Antragsteller kontaktieren	FALSCH	Sachbearbeiter B
1	5	2024-03-08 09:54:00	Antrag vorprüfen	FALSCH	Sachbearbeiter B
1	6	2024-03-27 11:11:00	Antrag zur fachlichen Prüfung verteilen	FALSCH	Sachbearbeiter B
1	7	2024-03-18 13:07:00	Antrag prüfen durch Amt 1	FALSCH	Sachbearbeiter C
1	8	2024-03-20 23:32:00	Antrag prüfen durch Amt 2	FALSCH	Sachbearbeiter D
1	9	2024-03-21 09:29:00	Antrag prüfen durch Amt 3	FALSCH	Sachbearbeiter E
1	10	2024-03-21 12:15:00	Entscheidung vorbereiten	FALSCH	Bereichsleiter
1	11	2024-03-21 15:03:00	Genehmigung erteilen und versenden	FALSCH	Sachbearbeiter B
2	1	2024-01-27 08:00:00	Antrag erhalten	WAHR	System
2	2	2024-01-27 08:13:00	Antrag erfassen	FALSCH	Sachbearbeiter A
2	3	2024-01-27 08:28:00	Antrag vorprüfen	FALSCH	Sachbearbeiter B
2	4	2024-01-27 08:30:00	Antrag zur fachlichen Prüfung verteilen	FALSCH	Sachbearbeiter B
2	5	2024-01-28 09:03:00	Antrag prüfen durch Amt 1	FALSCH	Sachbearbeiter C
2	6	2024-01-30 02:42:00	Antrag prüfen durch Amt 2	FALSCH	Sachbearbeiter D
2	7	2024-01-31 03:21:00	Antrag prüfen durch Amt 3	FALSCH	Sachbearbeiter E
2	8	2024-01-31 05:23:00	Entscheidung vorbereiten	FALSCH	Bereichsleiter
2	9	2024-01-31 07:25:00	Genehmigung erteilen und versenden	FALSCH	Sachbearbeiter B
3	1	2024-03-26 08:00:00	Antrag erhalten	WAHR	System
3	2	2024-03-26 08:18:00	Antrag erfassen	FALSCH	Sachbearbeiter A
3	3	2024-03-26 09:07:00	Antrag vorprüfen	FALSCH	Sachbearbeiter B
3	4	2024-03-26 09:09:00	Antrag zur fachlichen Prüfung verteilen	FALSCH	Sachbearbeiter B

Event Log



Prozessdokumentation

## Process Mining

*Process Discovery*: Welche Prozessabläufe gibt es?

*Process Conformance*: Wo weichen Prozessabläufe vom Standard ab?

*Enhancement*: Welche Auswirkungen haben Änderungen am Prozess oder an den Daten?

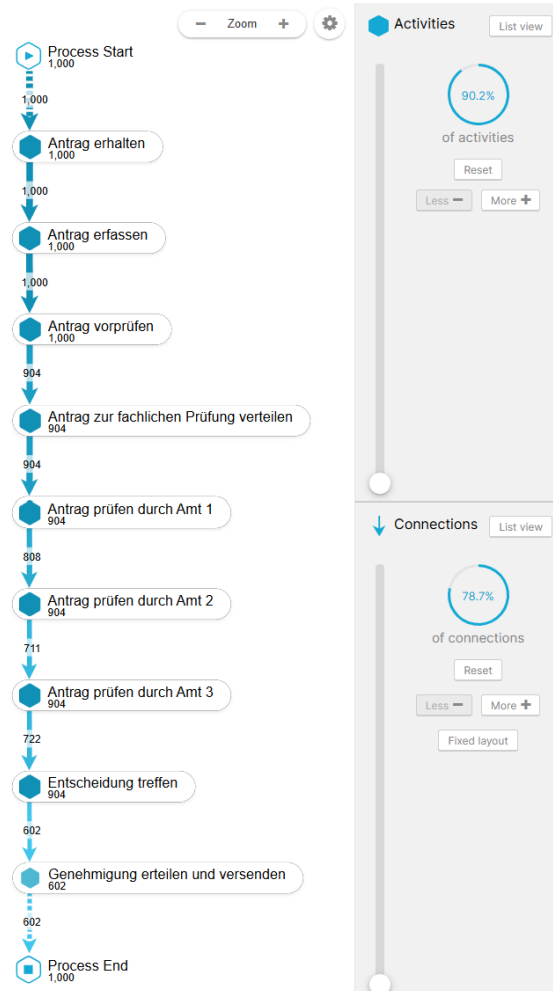
## LLM und Bilderkennung

Erkläre mir den Prozess!

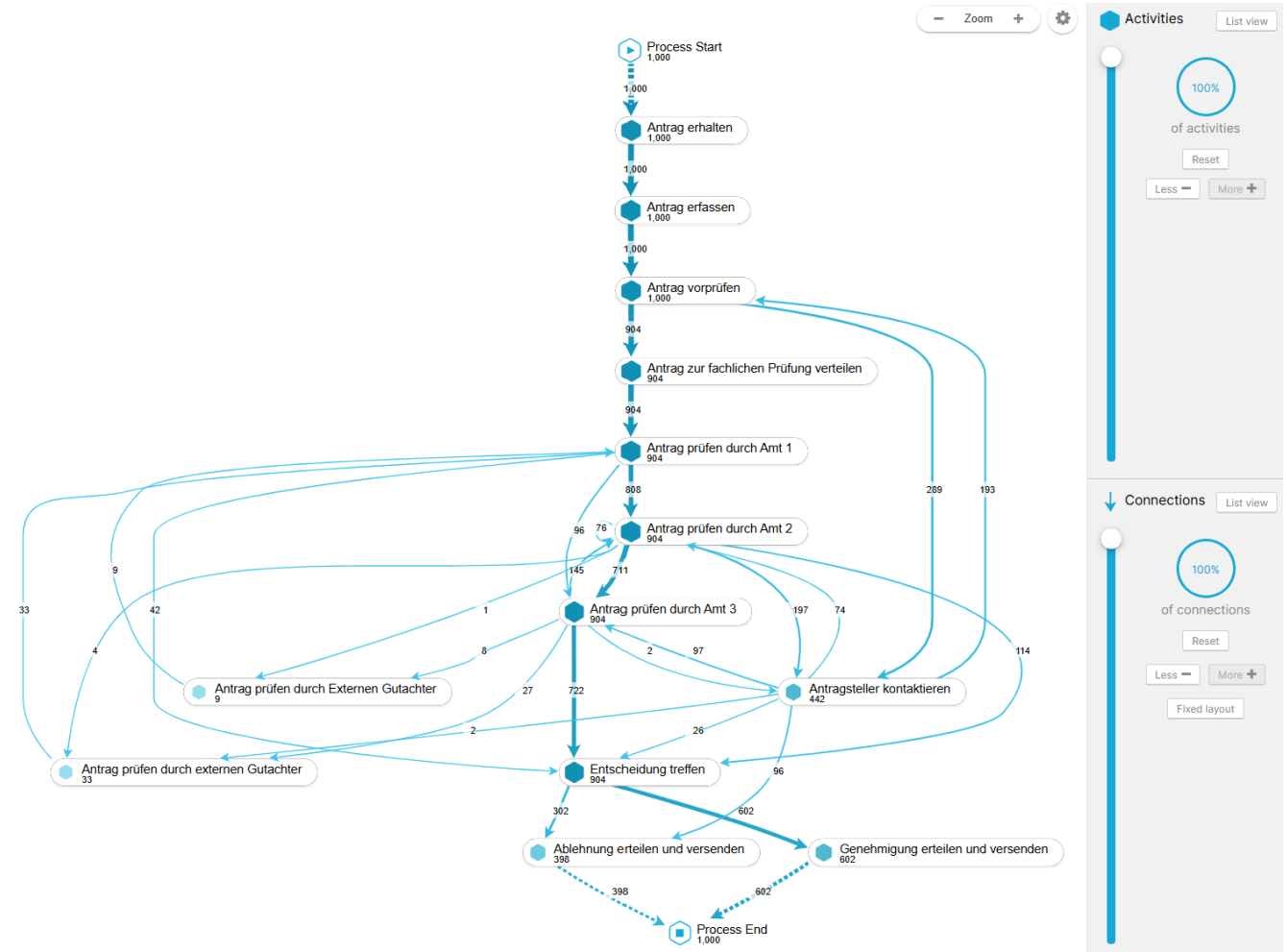
Wo steckt Verbesserungspotenzial?

# Wie kann KI helfen?

Process Discovery: Makro-Sicht

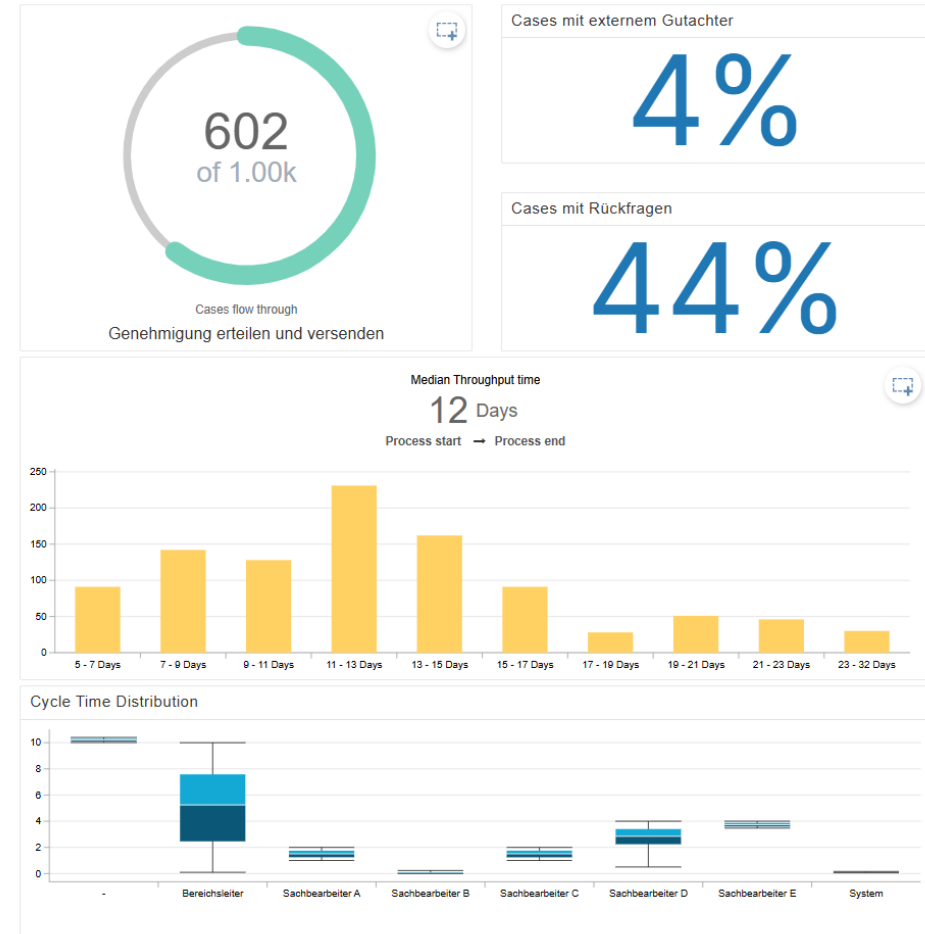
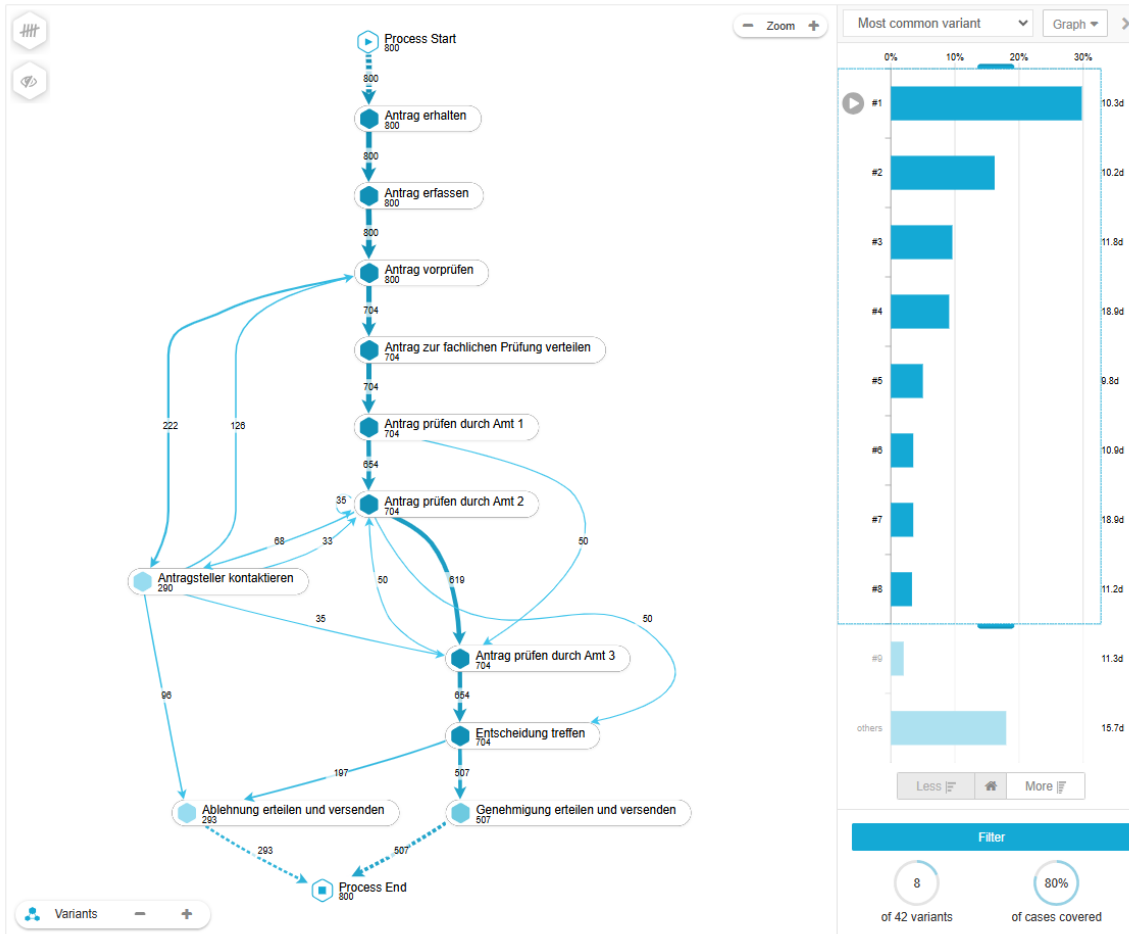


Process Discovery: Mikro-Sicht



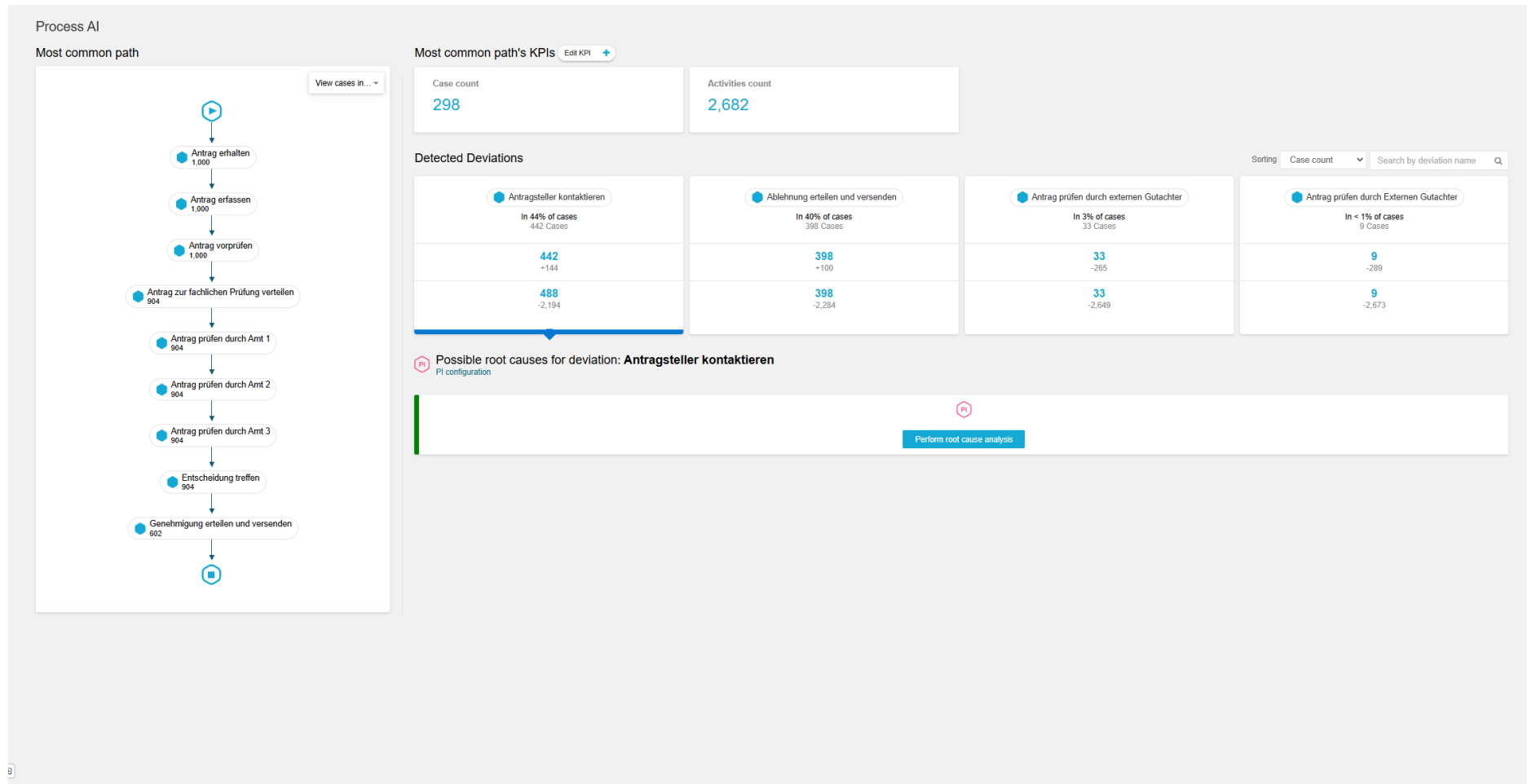
# Wie kann KI helfen?

## Process Enhancement



# Wie kann KI helfen?

## Process Enhancement



# KI-gestützte Dokumentenanalyse: Effiziente Informationsverarbeitung und -nutzung

Wie Künstliche Intelligenz hilft, Informationen in Dokumenten schneller zu strukturieren, zu verarbeiten und letztlich nutzbar zu machen.



**THWS – Business School**

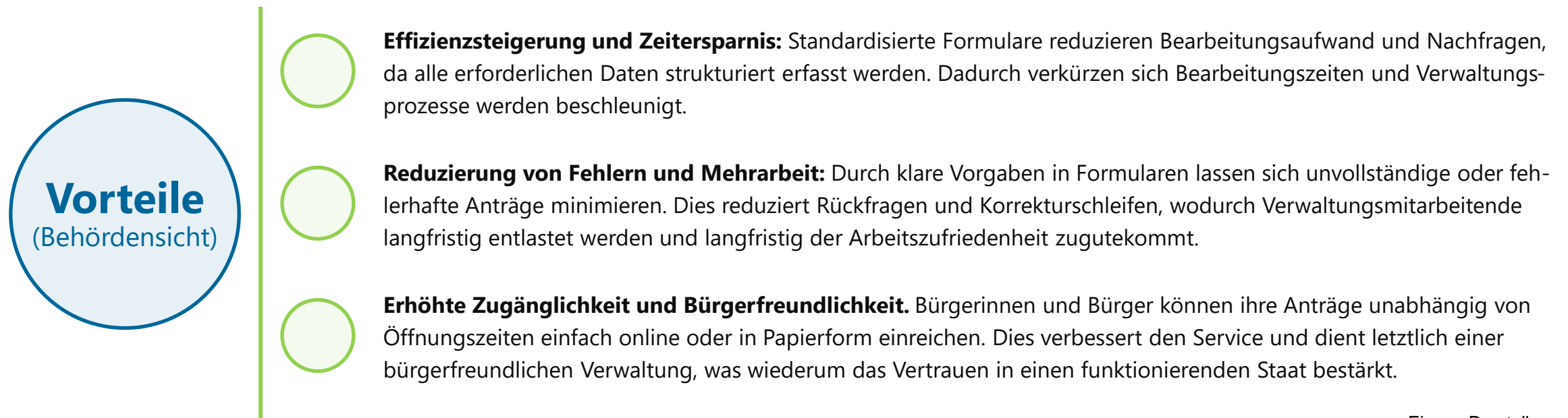
Prof. Dr. Robert Butscher

[LinkedIn](#)

# Use Case I:

## Retrieval Augmented Generation (RAG) zur Formulargestaltung

Formulare sind in der öffentlichen Verwaltung essenziell, um Anträge und Verwaltungsprozesse strukturiert und effizient zu bearbeiten. Sie standardisieren die Datenerfassung, minimieren Fehler und erleichtern die automatisierte Weiterverarbeitung. Dadurch ermöglichen sie eine rechtskonforme, transparente und nachvollziehbare Bearbeitung von Bürgeranliegen.



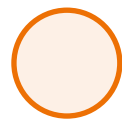
# Use Case I:

## Retrieval Augmented Generation (RAG) zur Formulargestaltung

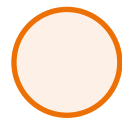
Formulare sind in der öffentlichen Verwaltung essenziell, um Anträge und Verwaltungsprozesse strukturiert und effizient zu bearbeiten. Sie standardisieren die Datenerfassung, minimieren Fehler und erleichtern die automatisierte Weiterverarbeitung. Dadurch ermöglichen sie eine rechtskonforme, transparente und nachvollziehbare Bearbeitung von Bürgeranliegen.



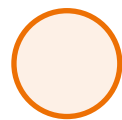
### Nachteile (Behördensicht)



**Digitale Barrieren für bestimmte Bevölkerungsgruppen.** Ältere Menschen oder Personen ohne digitale Kenntnisse können Schwierigkeiten bei der Nutzung von Online-Formularen haben. Dies kann zu einer ungleichen Zugänglichkeit und einer verstärkten Abhängigkeit von Unterstützung führen.



**Technische Komplexität und Wartungsaufwand.** Die Implementierung und Pflege einer sicheren, barrierefreien und stabilen Formularplattform erfordert kontinuierliche Updates und Anpassungen. Integrationen mit bestehenden Verwaltungssystemen sind oft aufwendig und fehleranfällig.

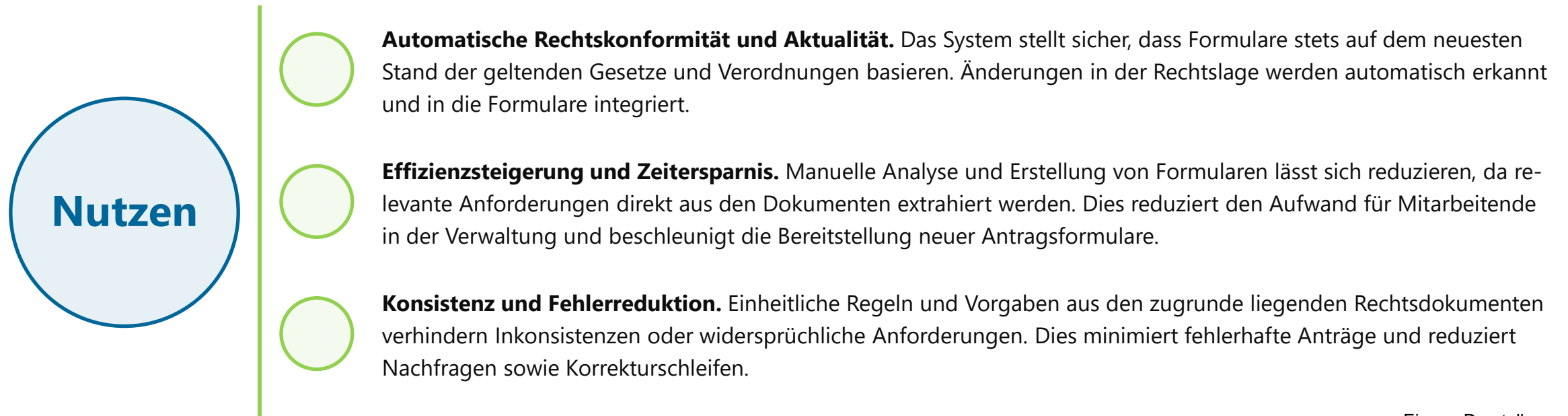


**Automatisierungshemmnisse durch unstrukturierte Eingaben.** Bürgerinnen und Bürger machen oft individuelle oder unvollständige Angaben in Freitextfeldern, was die automatische Weiterverarbeitung erschwert, v.a. handschriftlich ausgefüllte Formulare ziehen einen hohen manuellen Bearbeitungsaufwand nach sich.

# Use Case I:

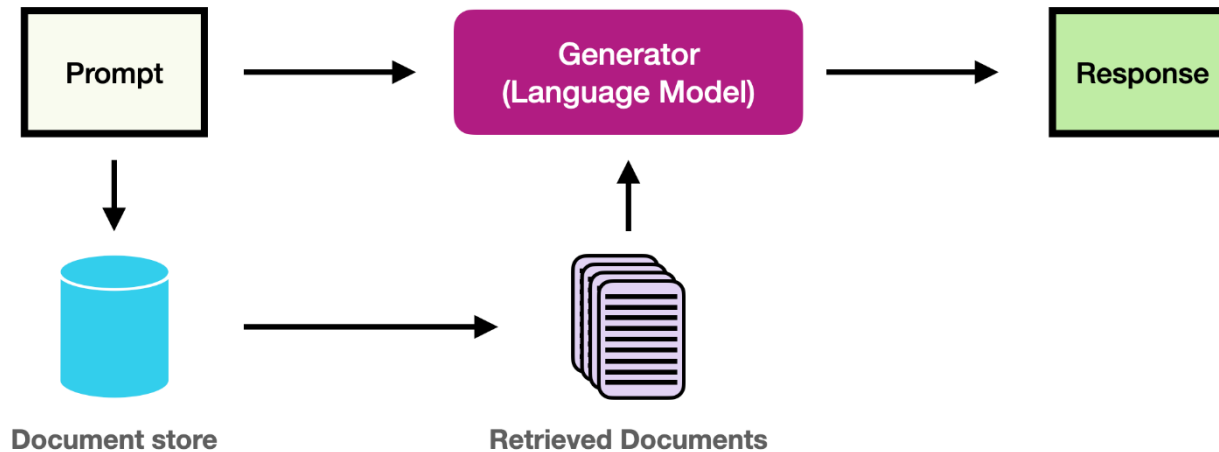
## Retrieval Augmented Generation (RAG) zur Formulargestaltung

Ein Retrieval-Augmented Generation (RAG)-System analysiert bestehende Gesetze und Verordnungen, um relevante Anforderungen für Antragsformulare automatisch zu extrahieren. Dabei werden strukturierte Formulare generiert, die Pflichtfelder, Nachweise und spezifische Bedingungen gemäß der Rechtsgrundlage enthalten. Behörden können so effizient aktualisierte und rechtskonforme Formulare bereitstellen.





# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung

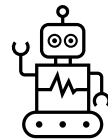


Der Ablauf bei Retrieval Augmented Generation (RAG) beginnt damit, dass das System per Prompt eine Benutzereingabe oder Frage erhält. Ein Suchmodul durchsucht anschließend externe Datenquellen (z.B. bereitgestellte Dokumente) nach relevanten Informationen zu dieser Anfrage.

Die abgerufenen Informationen werden mit der ursprünglichen Eingabe kombiniert, um eine erweiterte Kontextbasis zu schaffen. Basierend auf dieser kombinierten Information generiert ein Sprachmodell eine Antwort. Abschließend wird die generierte Antwort dem Benutzer präsentiert. RAG kombiniert somit das Wissen aus den externen Dokumenten mit seinem aus vorherigen Trainingsphasen gewonnenen Wissen. Diese Kombination ermöglicht es RAG, aktuelle und spezifische Informationen bereitzustellen, die über das vorhandene Wissen hinausgehen.

# Use Case I:

## Retrieval Augmented Generation (RAG) zur Formulargestaltung



Unterstützung durch KI



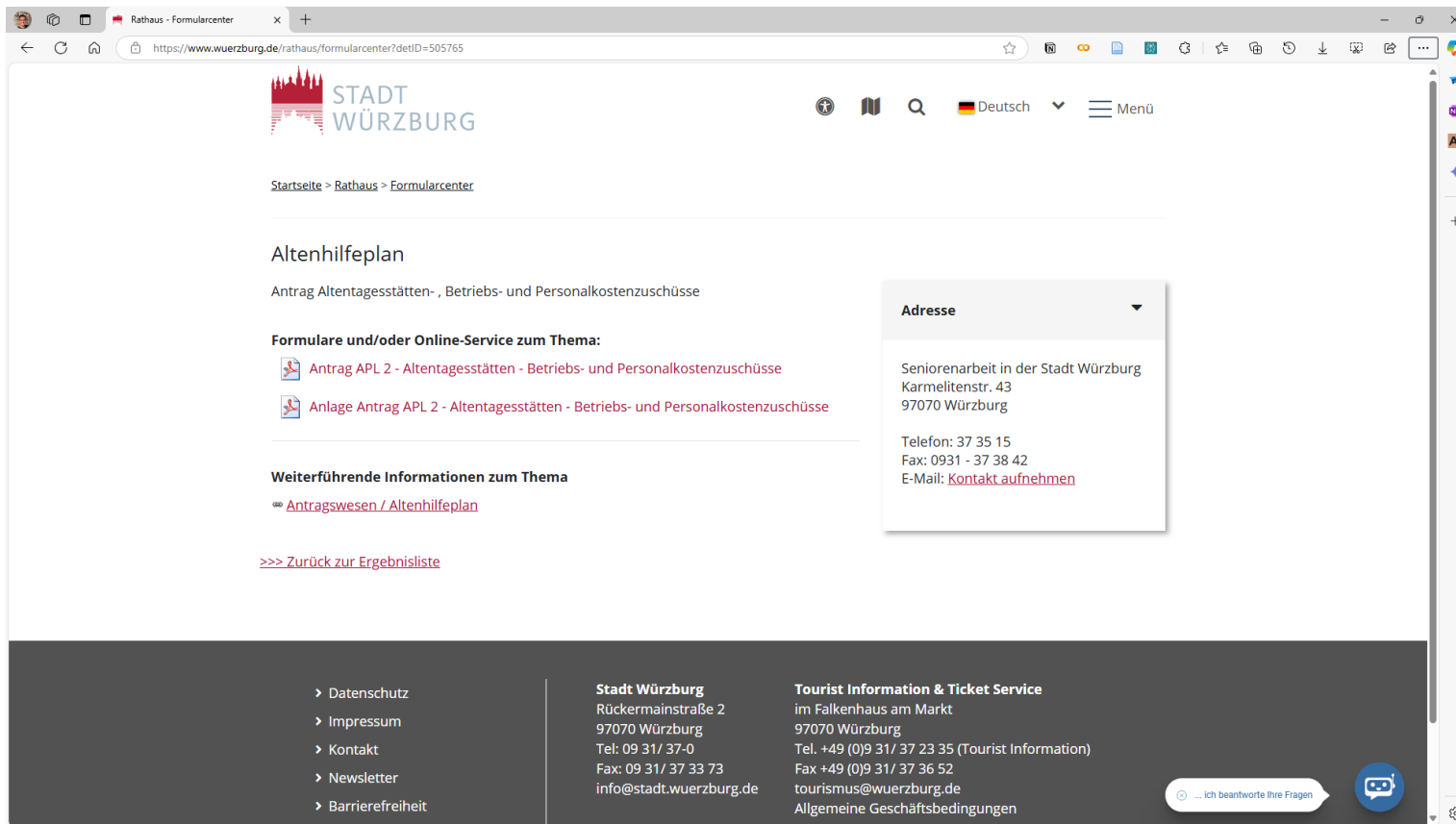
- Bestehende Formulare
- Gesetzestexte
- Rechtsverordnungen
- Gerichtsurteile
- Förderrichtlinien
- Taxonomien
- Ontologien

- Vorverarbeitung, etwa OCR
- Natural Language Processing
- Named Entity Recognition
- Optional Ontology Mapping

- Textanalyse ausschließlich bereitgestellter Dokumente
- Identifikation der Leistungsansprüche
- Identifikation der Voraussetzungen
- Vorschläge zur Formulierung in verständlicher Sprache
- Übersetzung der Formulare in verschiedene Sprachen
- Fachliche Validierung durch hinterlegte Ontologie/ Regeln
- Tests der Usability und Barrierefreiheit
- Formulare sind wesentlicher Teil der Customer Journey
- Sicherstellung der visuellen/ gestalterischen Konsistenz

- Programmcode, etwa JavaScript, SQL
- Mapping mit Taxonomien/ Ontologien
- Repräsentation in XML/ JSON/ RDF/ OWL usw.
- Automatisierte Analyse der Inhalte via Python/ R

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung

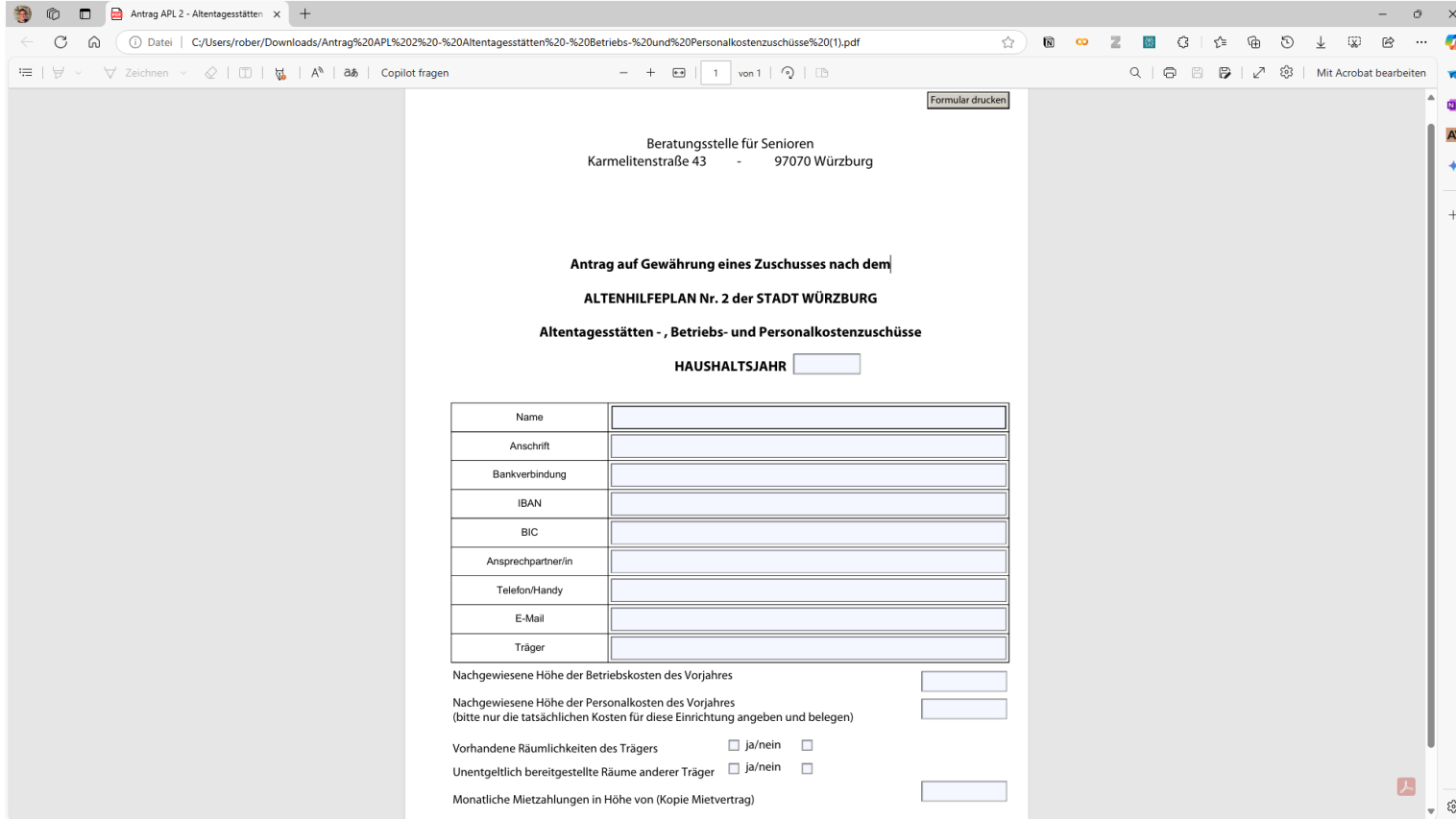


Die Stadt Würzburg bietet für ihre Bürgerinnen und Bürger ein umfangreiches Portal mit zahlreichen Antragsformularen.

Das CMS bietet die Möglichkeit, den Inhalt der Website in verschiedenen Sprachen anzugeben, was nicht das Image vor Würzburg als weltoffene Stadt fördert, sondern auch der gesellschaftlichen Teilhabe und Integration zugutekommt.

Die eigentlichen Inhalte der Formulare sowie die Formularfelder sind nicht übersetzt, zudem lassen sich die Formulare nicht direkt auf der Website online ausfüllen, was zu einem Medienbruch führt, Eingabefehler begünstigt oder gar manuellen Erfassungsaufwand in den Behörden nach sich zieht.

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung



The screenshot shows a PDF form for an application for a subsidy. The form is titled 'Antrag auf Gewährung eines Zuschusses nach dem ALTENHILFEPLAN Nr. 2 der STADT WÜRZBURG' and is for 'Altentagesstätten - , Betriebs- und Personalkostenzuschüsse'. It includes a header for 'Beratungsstelle für Senioren' at 'Karmelitenstraße 43 - 97070 Würzburg'. The form contains several input fields for personal and contact information, a table for financial data, and checkboxes for additional information.

Name	
Anschrift	
Bankverbindung	
IBAN	
BIC	
Ansprechpartner/in	
Telefon/Handy	
E-Mail	
Träger	

Nachgewiesene Höhe der Betriebskosten des Vorjahres

Nachgewiesene Höhe der Personalkosten des Vorjahres  
(bitte nur die tatsächlichen Kosten für diese Einrichtung angeben und belegen)

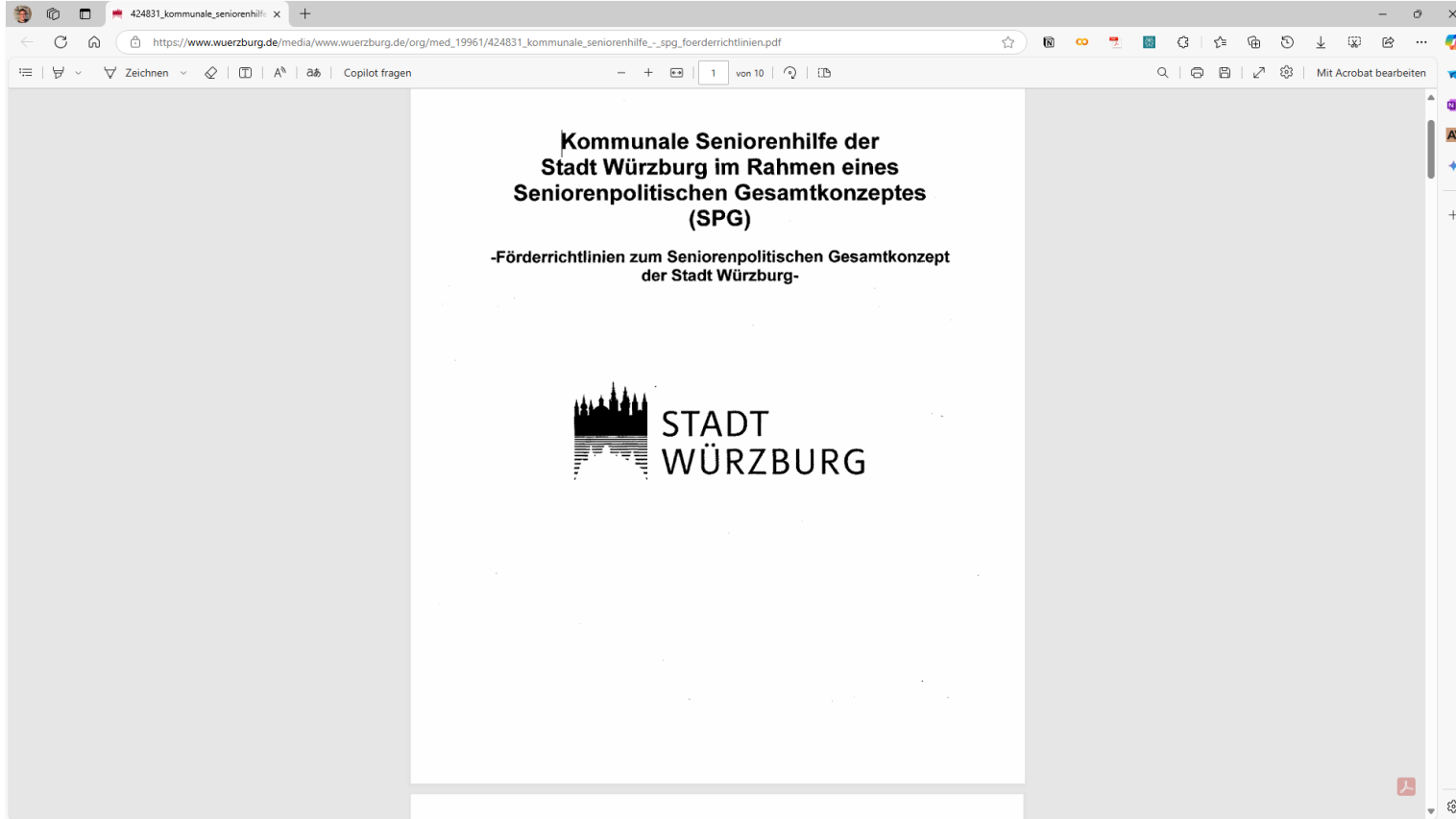
Vorhandene Räumlichkeiten des Trägers  ja/nein

Unentgeltlich bereitgestellte Räume anderer Träger  ja/nein

Monatliche Mietzahlungen in Höhe von (Kopie Mietvertrag)

Der Screenshot zeigt ein typisches Formular. Es soll hier nur der Anschauung dienen und als Ausgangspunkt für die maschinelle, KI-gestützte Weiterverarbeitung fungieren. Das Formular ist Input für das RAG-System.

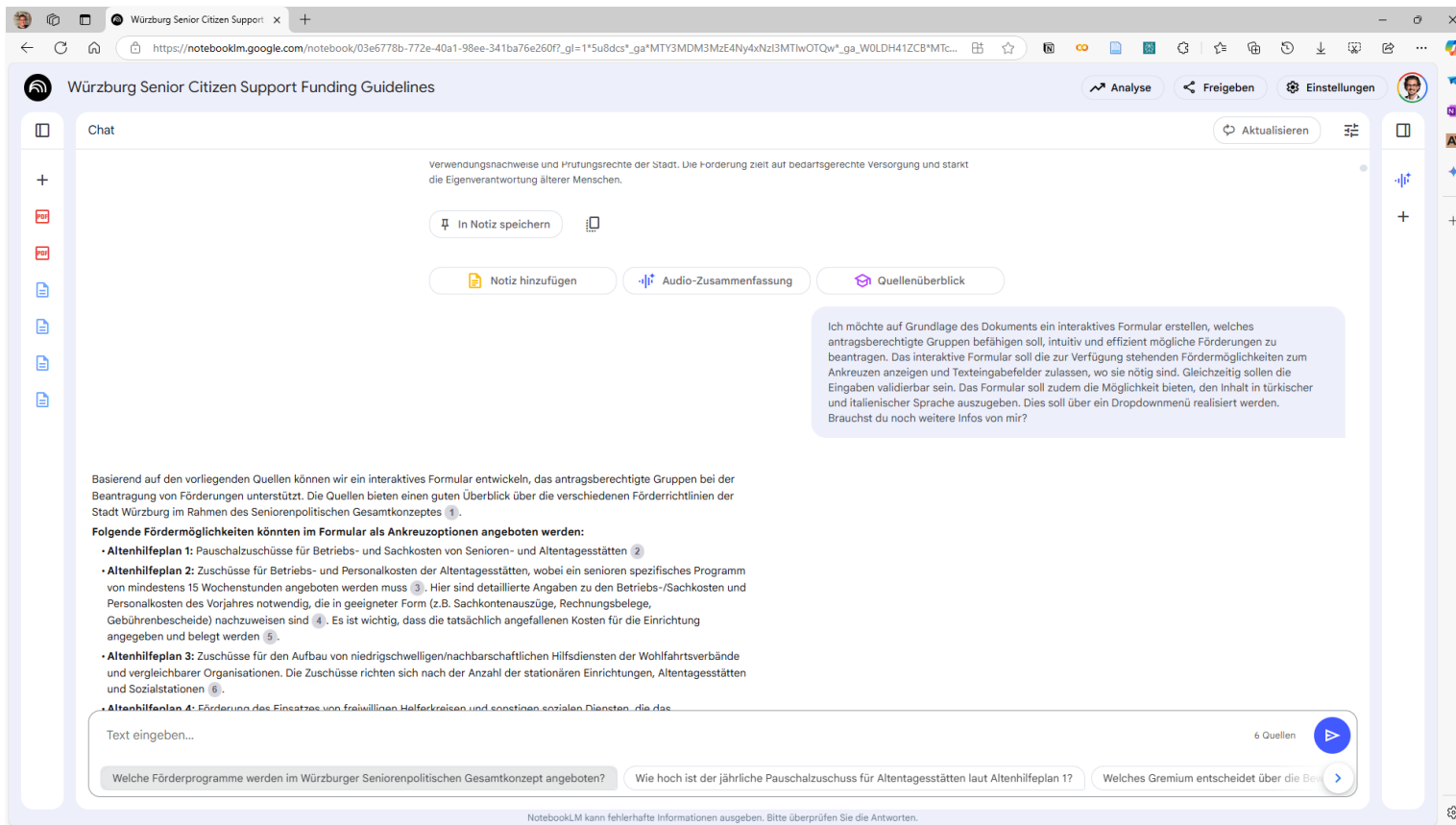
# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung



Den von der Stadt Würzburg bereitgestellten Formularen liegt eine umfangreiche Förderrichtlinie zugrunde (10 Seiten PDF), die als Grundlage in ein RAG-fähiges System eingespielt wird. Das RAG-System macht alleinig inhaltliche Vorschläge, welche Fragen für ein Formular, über welches sich ein Förderzuschuss beantragen lässt, relevant sind.

Darüber hinaus lassen sich über ein RAG-System auch inhaltlich passende Fragen automatisch generieren. Diese böten sich als FAQ für Bürgerinnen und Bürger an oder könnten Grundlage für ein Hilfesystem sein. Auch für das Information Retrieval (etwa die Suche nach dem sachlich korrekten Formular) bieten Fragestellungen eine Hilfestellung.

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung



The screenshot shows the Google LM Notebook interface. The browser tab is titled 'Würzburg Senior Citizen Support'. The URL is [https://notebooklm.google.com/notebook/03e6778b-772e-40a1-98ee-341ba76e260f?\\_gl=1\\*5u8dcs\\*\\_ga\\*MTY3MDM3MzE4Ny4xNzI3MTwOTQw\\*\\_ga\\_W0LDH41ZCB\\*MTc...](https://notebooklm.google.com/notebook/03e6778b-772e-40a1-98ee-341ba76e260f?_gl=1*5u8dcs*_ga*MTY3MDM3MzE4Ny4xNzI3MTwOTQw*_ga_W0LDH41ZCB*MTc...). The document title is 'Würzburg Senior Citizen Support Funding Guidelines'. The main text in the chat area reads: 'Verwendungsnachweise und Pflichtenrechte der Stadt. Die Forderung zielt auf bedarfsgerechte Versorgung und stärkt die Eigenverantwortung älterer Menschen.' Below this, there are buttons for 'In Notiz speichern', 'Notiz hinzufügen', 'Audio-Zusammenfassung', and 'Quellenüberblick'. A blue chat bubble contains the prompt: 'Ich möchte auf Grundlage des Dokuments ein interaktives Formular erstellen, welches antragsberechtigte Gruppen befähigen soll, intuitiv und effizient mögliche Förderungen zu beantragen. Das interaktive Formular soll die zur Verfügung stehenden Fördermöglichkeiten zum Ankreuzen anzeigen und Texteingabefelder zulassen, wo sie nötig sind. Gleichzeitig sollen die Eingaben validierbar sein. Das Formular soll zudem die Möglichkeit bieten, den Inhalt in türkischer und italienischer Sprache auszugeben. Dies soll über ein Dropdownmenü realisiert werden. Brauchst du noch weitere Infos von mir?'. Below the chat bubble, the system's response is visible, starting with 'Basierend auf den vorliegenden Quellen können wir ein interaktives Formular entwickeln, das antragsberechtigte Gruppen bei der Beantragung von Förderungen unterstützt. Die Quellen bieten einen guten Überblick über die verschiedenen Förderrichtlinien der Stadt Würzburg im Rahmen des Seniorenpolitischen Gesamtkonzeptes 1.' and listing four 'Altenhilfspläne'. At the bottom, there is a text input field with the placeholder 'Text eingeben...' and a '6 Quellen' indicator. Below the input field, there are three suggested prompts: 'Welche Förderprogramme werden im Würzburger Seniorenpolitischen Gesamtkonzept angeboten?', 'Wie hoch ist der jährliche Pauschalzuschuss für Altagestagesstätten laut Altenhilfsplan 1?', and 'Welches Gremium entscheidet über die Bew...'. The footer of the interface states 'NotebookLM kann fehlerhafte Informationen ausgeben. Bitte überprüfen Sie die Antworten.'

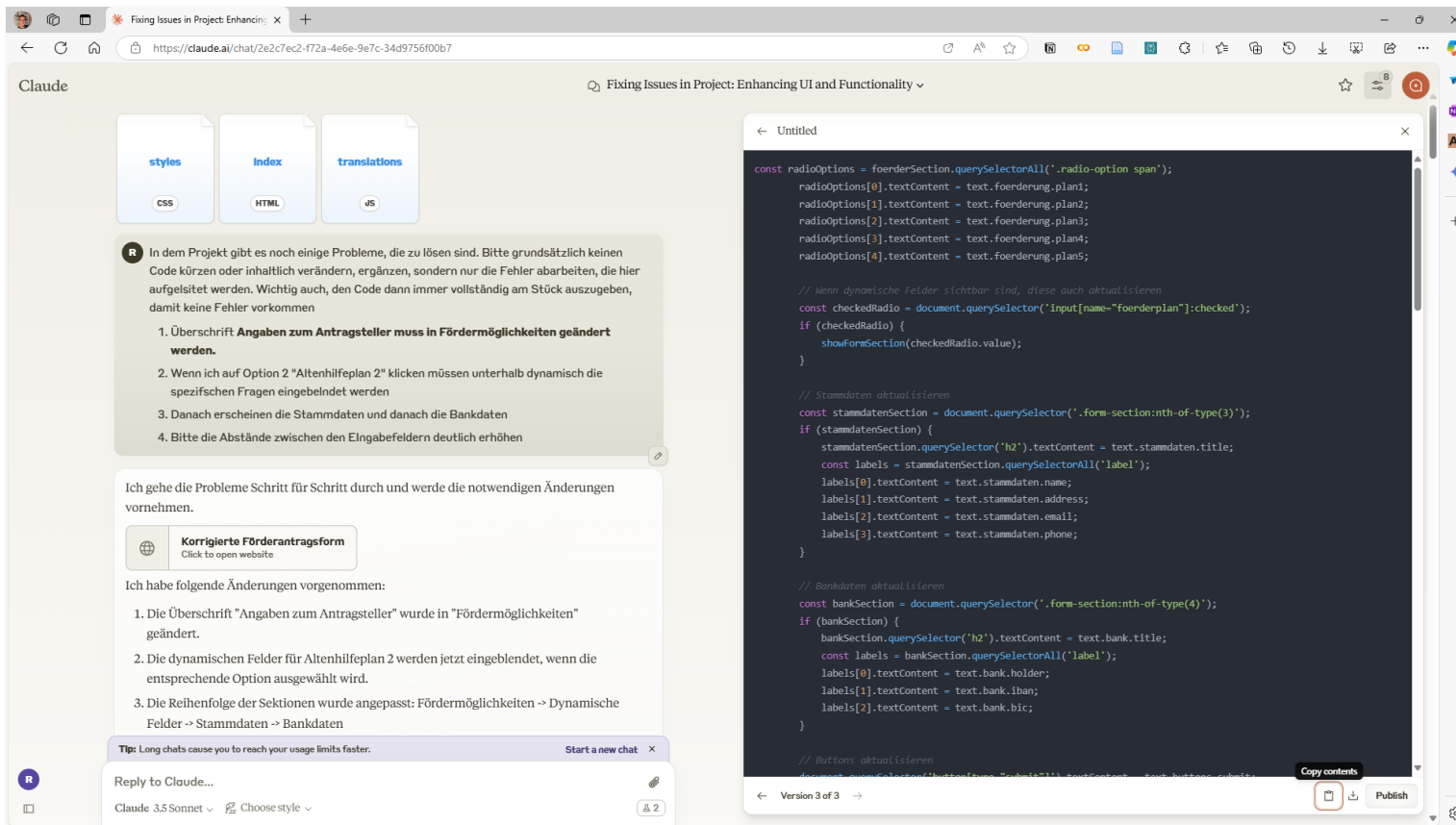
Als RAG-System wird im Folgenden Google LM Notebook verwendet (keine Werbung).

Es werden zum Start in das Cloud-basierte System nur zwei Dokumente hochgeladen:

- Antragsformular APL 2
- Förderrichtlinie

Auf ein entsprechendes Prompt hin erstellt das System einen Vorschlag, welche Aspekte aus der Förderrichtlinie für ein Antragsformular als relevant erscheinen. Zudem bietet das System eine Reihe vorgedachter Fragen an, die sich allesamt aus den beiden Dokumenten speisen. Es ist zudem möglich, auch eigene Fragen zu erstellen und so mit den vorhandenen Dokumenten zu chatten.

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung

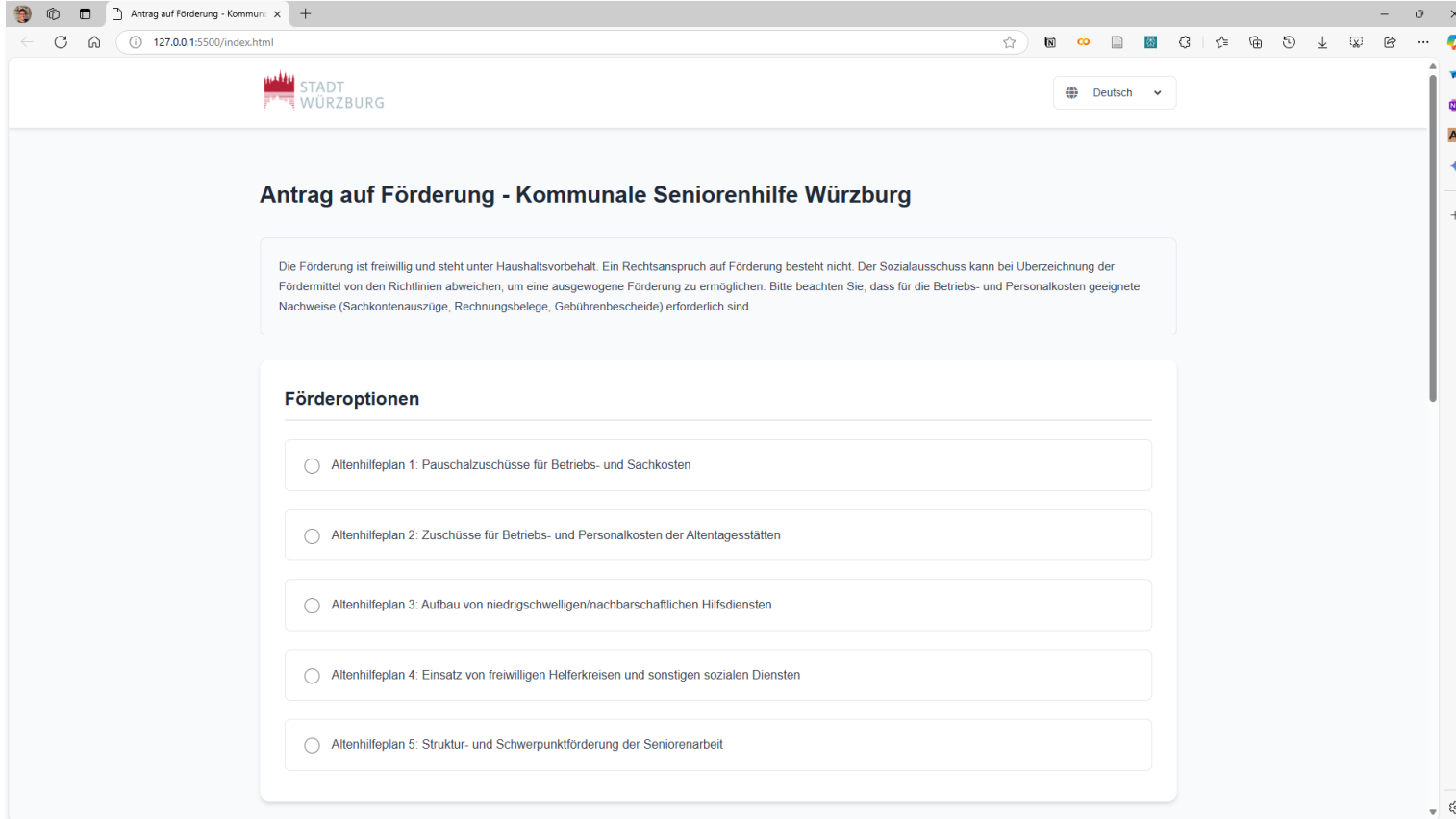


The screenshot shows the Claude.ai interface. On the left, there are tabs for 'styles', 'index', and 'translations'. Below them are buttons for 'CSS', 'HTML', and 'JS'. A chat window contains a request in German: 'In dem Projekt gibt es noch einige Probleme, die zu lösen sind. Bitte grundsätzlich keinen Code kürzen oder inhaltlich verändern, ergänzen, sondern nur die Fehler abarbeiten, die hier aufgelistet werden. Wichtig auch, den Code dann immer vollständig am Stück auszugeben, damit keine Fehler vorkommen'. It lists four issues: 1. Change the title of the application form to 'Fördermöglichkeiten geändert werden'. 2. Dynamically show specific questions when option 2 is selected. 3. Show personal data and bank data. 4. Increase spacing between input fields. Below the chat is a 'Korrigierte Förderantragsform' button and a list of changes made. On the right, a code editor shows JavaScript code for form validation, including functions for showing sections, updating personal data, bank data, and buttons.

Der Input aus Google LM Notebook dient als Grundlage für eine zunächst statische Erfassungsmaske in HTML. Ziel ist am Ende ein interaktives Formular, daher entstehen im Laufe der Arbeitssitzung noch weitere Dateien für CSS- und JavaScript-Skripte.

Für die Entwicklung wird Claude.ai (keine Werbung) und Deep Seek (keine Werbung) zur Qualitätssicherung genutzt.

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung



The screenshot shows a web browser window displaying a form for 'Antrag auf Förderung - Kommunale Seniorenhilfe Würzburg'. The form includes a header with the city logo and a language dropdown set to 'Deutsch'. Below the title, there is a text box explaining that the funding is voluntary and subject to household reservation. The main section, 'Förderoptionen', contains five radio button options for different types of support: 1. Pauschalzuschüsse für Betriebs- und Sachkosten, 2. Zuschüsse für Betriebs- und Personalkosten der Alltagsstätten, 3. Aufbau von niedrigschwelligen/nachbarschaftlichen Hilfsdiensten, 4. Einsatz von freiwilligen Helferkreisen und sonstigen sozialen Diensten, and 5. Struktur- und Schwerpunktförderung der Seniorenarbeit.

Für die nebenstehende Implementierung als interaktives Web-Formular wurden bewusst mehrere KI-Systeme genutzt: Claude.ai (Lead Developer) und Deep Seek (Test Engineer).

Der Inhalt des Formulars stammt vollständig aus zwei Dokumenten und wurde per RAG von Google LM Notebook abgeleitet.



# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung

Antrag auf Förderung - Kommun - X

127.0.0.1:5500/index.html

Altagestättenplan 2. Zuschüsse für Betriebs- und Personalkosten der Altagestätten

Altagestättenplan 3. Aufbau von niedrigschwelligen/nachbarschaftlichen Hilfsdiensten

Altagestättenplan 4. Einsatz von freiwilligen Helferkreisen und sonstigen sozialen Diensten

Altagestättenplan 5. Struktur- und Schwerpunktförderung der Seniorenarbeit

**Angaben zu Altagestätten - Betriebs- und Personalkosten**

Betriebskosten des Vorjahres (€):

Personalkosten des Vorjahres (€):

**Räumlichkeiten**

Vorhandene Räumlichkeiten des Trägers:

Unentgeltlich bereitgestellte Räume anderer Träger:

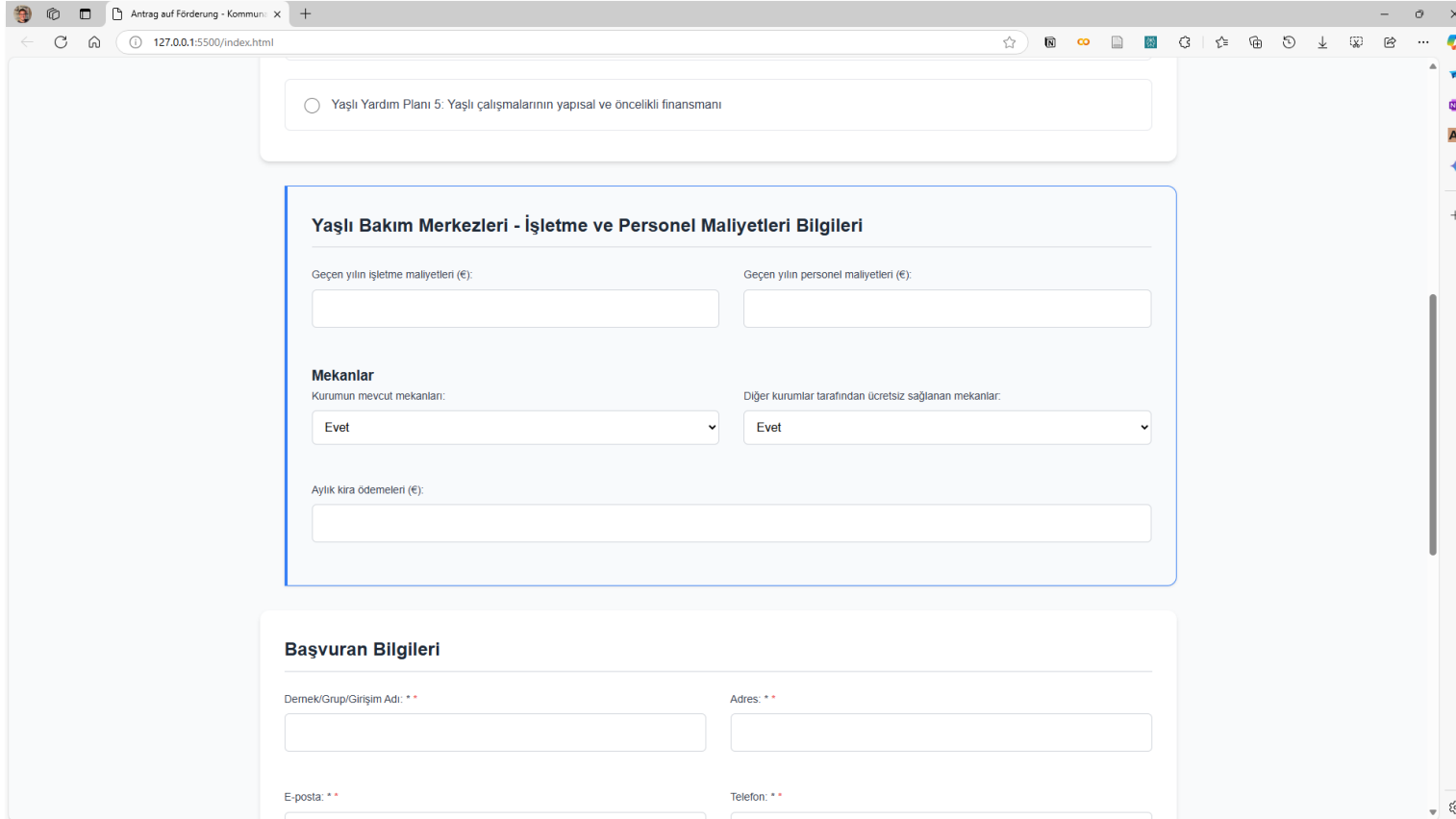
Monatliche Mietzahlungen in Höhe von (€):

Das Formular ist dynamisch und interaktiv, je nach gewählter Option werden unterschiedliche Erfassungsfelder eingeblendet.

Das Erscheinungsbild sollte modern und zeitlos sein, technisch wäre es kein Problem, einen Styleguide für ein einheitliches und konsistentes Design zugrunde zu legen und die KI die Einhaltung überwachen zu lassen.

Die Codierung hat in diesem Fall Claude.ai im Zusammenspiel mit Google LM Notebook und Google Gemini übernommen.

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung



Antrag auf Förderung - Kommun... X

127.0.0.1:5500/index.html

Yaşlı Yardım Planı 5: Yaşlı çalışmalarının yapısal ve öncelikli finansmanı

**Yaşlı Bakım Merkezleri - İşletme ve Personel Maliyetleri Bilgileri**

Geçen yılın işletme maliyetleri (€):

Geçen yılın personel maliyetleri (€):

**Mekanlar**

Kurumun mevcut mekanları:

Diğer kurumlar tarafından ücretsiz sağlanan mekanlar:

Aylık kira ödemeleri (€):

**Başvuran Bilgileri**

Dernek/Grup/Girişim Adı: \*\*

Adres: \*\*

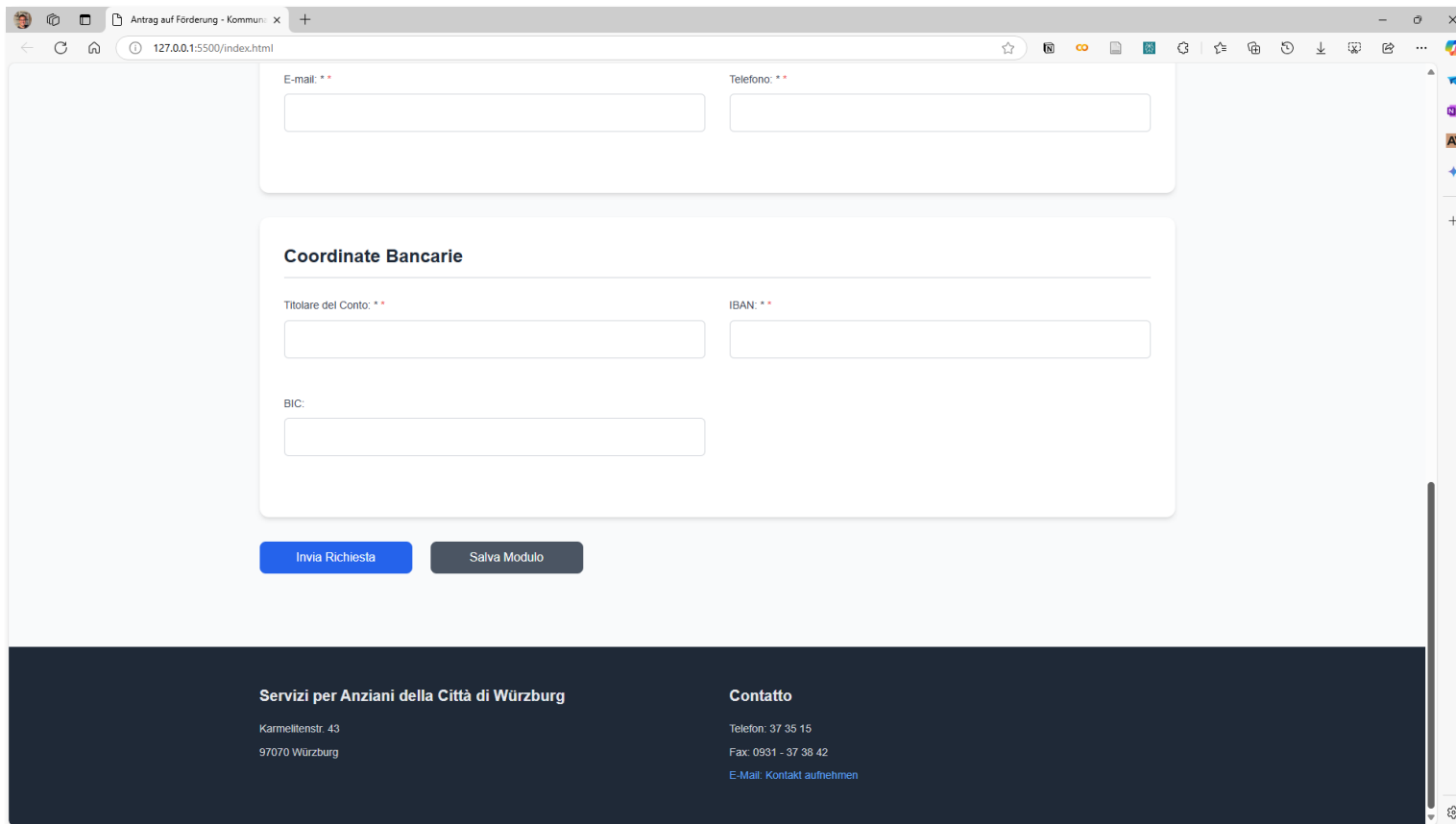
E-posta: \*\*

Telefon: \*\*

Das interaktive Formular wurde mithilfe von KI zu Illustrationszwecken in verschiedene Sprachen übersetzt – hier auf Türkisch.

Ein Formular ist auch ein Kommunikationsmittel zwischen Behörden und Bürger/ -in. Sprache spielt hierbei eine Schlüsselrolle. Hier liegt großes Potenzial durch KI Barrieren abzubauen, auch im Hinblick auf leicht verständliche Sprache oder Accessibility.

# Use Case I: Retrieval Augmented Generation (RAG) zur Formulargestaltung



The screenshot shows a web browser window with the URL 127.0.0.1:5500/index.html. The page displays a form titled 'Coordinate Bancarie' in Italian. The form has two rows of input fields. The first row contains 'E-mail: \*\*' and 'Telefono: \*\*'. The second row contains 'Titolare del Conto: \*\*' and 'IBAN: \*\*'. Below these is a 'BIC:' field. At the bottom of the form are two buttons: 'Invia Richiesta' (blue) and 'Salva Modulo' (grey). The footer of the page is dark blue and contains the following text:

**Servizi per Anziani della Città di Würzburg**  
Karmelitenstr. 43  
97070 Würzburg

**Contatto**  
Telefon: 37 35 15  
Fax: 0931 - 37 38 42  
E-Mail: [Kontakt aufnehmen](#)

Das interaktive Formular wurde mithilfe von KI zu Illustrationszwecken in verschiedene Sprachen übersetzt – hier auf Italienisch.

Ein Formular ist auch ein Kommunikationsmittel zwischen Behörden und Bürger/ -in. Sprache spielt hierbei eine Schlüsselrolle. Hier liegt großes Potenzial durch KI Barrieren abzubauen, auch im Hinblick auf leicht verständliche Sprache oder Accessibility.

# Use Case I – Fazit: Retrieval Augmented Generation (RAG) zur Formulargestaltung

Mit den KI-Systemen Google LM Notebook, Claude.ai und Deep Seek wurde prototypisch ein interaktives Formular auf Basis von HTML, CSS und JavaScript implementiert. Als Grundlage dienten nur die entsprechende Förderrichtlinie sowie ein vorhandenes Formular.



## Vorteile



**Sehr leistungsstarke textanalytische Fähigkeiten durch LLM.** RAG nutzt für die Ausgabe ein LLM, welches imstande ist, Texte wie ein Mensch zu formulieren oder Texte in verschiedenen Sprachen ausgeben kann.



**Überführung statischer PDF-Dokumente in interaktive Webanwendungen.** Der vom RAG-System generierte Text bildet die Grundlage für interaktive Formulare, die sich zur Datenerfassung in bestehende IT-Systeme sehr gut eignen.



**Code-Generierung durch LLMs.** LLMs bieten die Möglichkeit, qualitativ hochwertigen Code zu schreiben. Dies macht es möglich, interaktive Webanwendungen durch die Fachabteilung zu entwickeln und Entwicklungskosten einzusparen.



**Ausbau der Digitalisierung.** Isolierte Formulare sind Medienbrüche und erschweren Prozessautomatisierung. Mittels KI lassen sich Inhalte aus Dokumenten in Formate überführen, die sich gut zum maschinellen Datenaustausch eignen.

# Use Case I – Fazit :

## Retrieval Augmented Generation (RAG) zur Formulargestaltung

Mit den KI-Systemen Google LM Notebook, Claude.ai und Deep Seek wurde prototypisch ein interaktives Formular auf Basis von HTML, CSS und JavaScript implementiert. Als Grundlage dienten nur die entsprechende Förderrichtlinie sowie ein vorhandenes Formular.



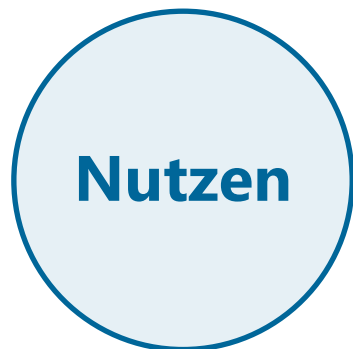
### Herausforderungen

-  **Eventuelle Abhängigkeit zu Big Tech.** Die derzeit leistungsstärksten LLMs sind derzeit noch proprietär. In der Folge stellt sich die Frage zum fairen Zugang zu solchen Technologien und eventuellen Abhängigkeiten.
-  **KI als Allheilmittel.** Die vergleichsweise rasche Umsetzbarkeit begünstigt die (problematische) Vermutung, dass sich jedes IT-Problem irgendwie mit KI lösen ließe. Dies weckt falsche Erwartungen und allokiert zu starke finanzielle Mittel.
-  **Einhaltung von Datenschutz und EU AI Act.** Aus Datenschutz und Privacy-Gründen sind "Quick Wins" mit Cloud-basierten LLMs nicht ohne weiteres möglich. Eine on-Prem-Umsetzung zieht Rüst- und Wartungsaufwand nach sich.
-  **"Wildwuchs" an Lösungen.** Anwendungen lassen sich mit KI vergleichsweise schnell programmieren, sie können aber qualitative Einschränkungen haben oder sich schlecht in die bestehende IT-Landschaft integrieren lassen.

# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide

Das Anwendungsbeispiel zeigt, wie sich mithilfe von NLP-Methoden ("Natural Language Processing") aus einem unstrukturierten Textdokument (hier Förderrichtlinie) die relevanten Konzepte und Beziehungen extrahieren und in eine formale Ontologie überführen lassen. Diese dient als Grundlage, anhand der extrahierten Regeln und Zusammenhänge Antragsformulare automatisiert zu bearbeiten und zu entscheiden, ob eine beantragte Förderung gewährt und wenn ja, in welcher Höhe bewilligt werden kann.



**Effizienzsteigerung:** Eine Ontologie automatisiert abzuleiten und als Prüfraster anzuwenden, trägt dazu bei, Anträge effizienter und schneller bearbeiten zu können. In der Folge lassen sich manuelle, zeitintensive Prozesse reduzieren. Eine Ontologie lässt sich zudem nutzen, Prozesse nach einem "Standard Procedure" abarbeiten zu können.



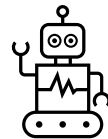
**Konsistente Entscheidungsfindung:** Eine formale Ontologie prüft alle Förderkriterien einheitlich und nachvollziehbar. Dies führt zu konsistenten Entscheidungen, die frei von subjektiven Einflüssen Dritter sind und somit auch Entscheidungen nachvollziehbarer (und damit weniger willkürlich) macht.



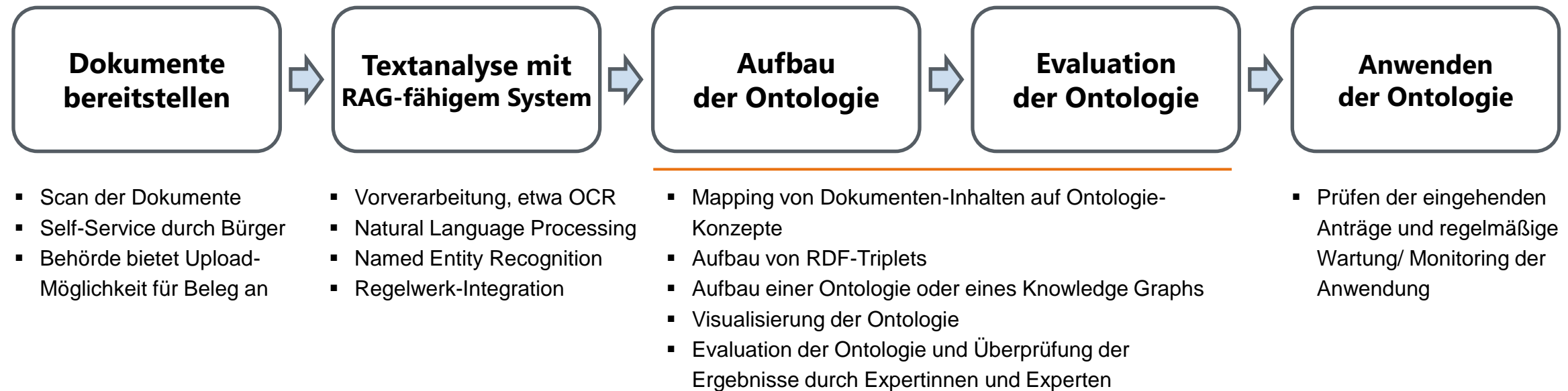
**Flexibilität und Anpassungsfähigkeit:** Die Integration von NLP ermöglicht die dynamische Extraktion relevanter Informationen aus sich ändernden Förderrichtlinien. Dadurch lässt sich das System leicht an neue Anforderungen und Regelungen anpassen oder auch komplexe Regelwerke zuverlässig umsetzen.

# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide

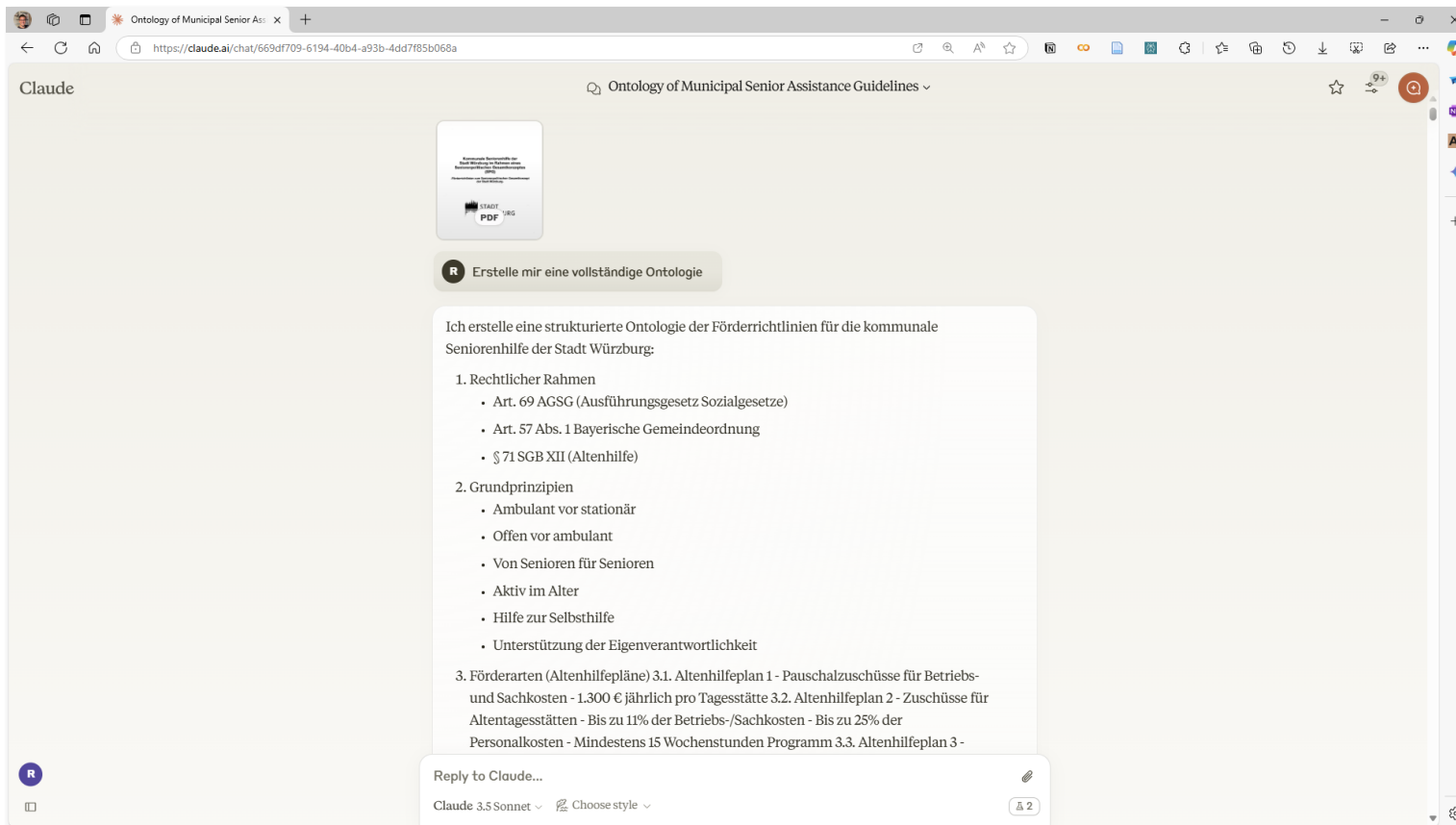


Unterstützung durch KI



# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide



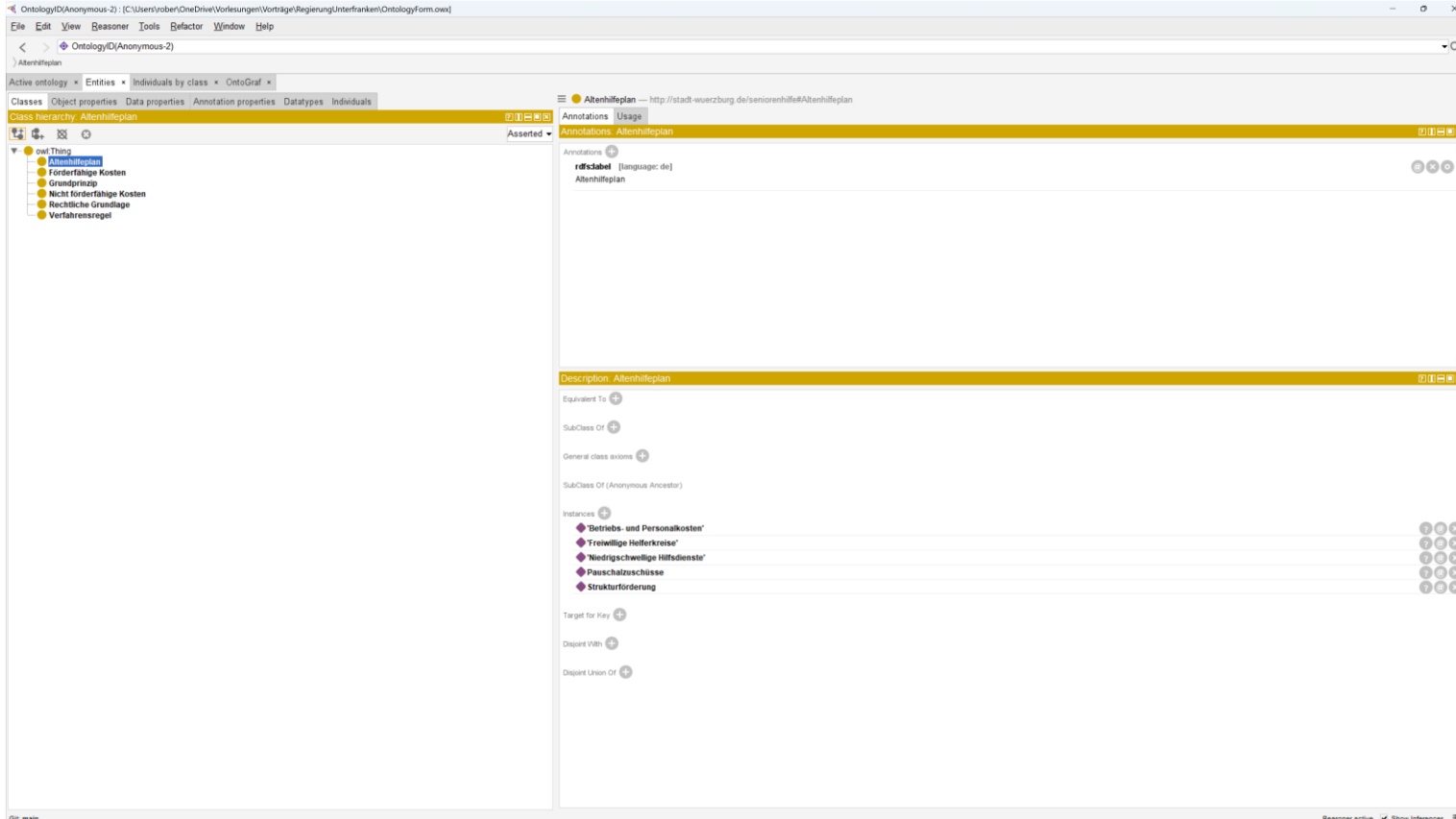
Die Ontologie setzt auf den Förderrichtlinien auf, sie wurde schrittweise durch Prompting von KI-Tools wie Claude.ai (Screenshot), Deep Seek und ChatGPT aufgebaut. Nach einer initialen Domain-Analyse wurden Kernkonzepte und deren Beziehungen identifiziert.

Formal wurde sie in die RDF-Notation überführt und besteht aus sog. RDF-Triplets. Die formale Repräsentation erfolgt in RDF/OWL mit definierten Namespaces, Properties und Axiomen. Die Ontologie umfasst Klassen, deren hierarchische Beziehungen, Attribute sowie Kardinalitäten.



# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide



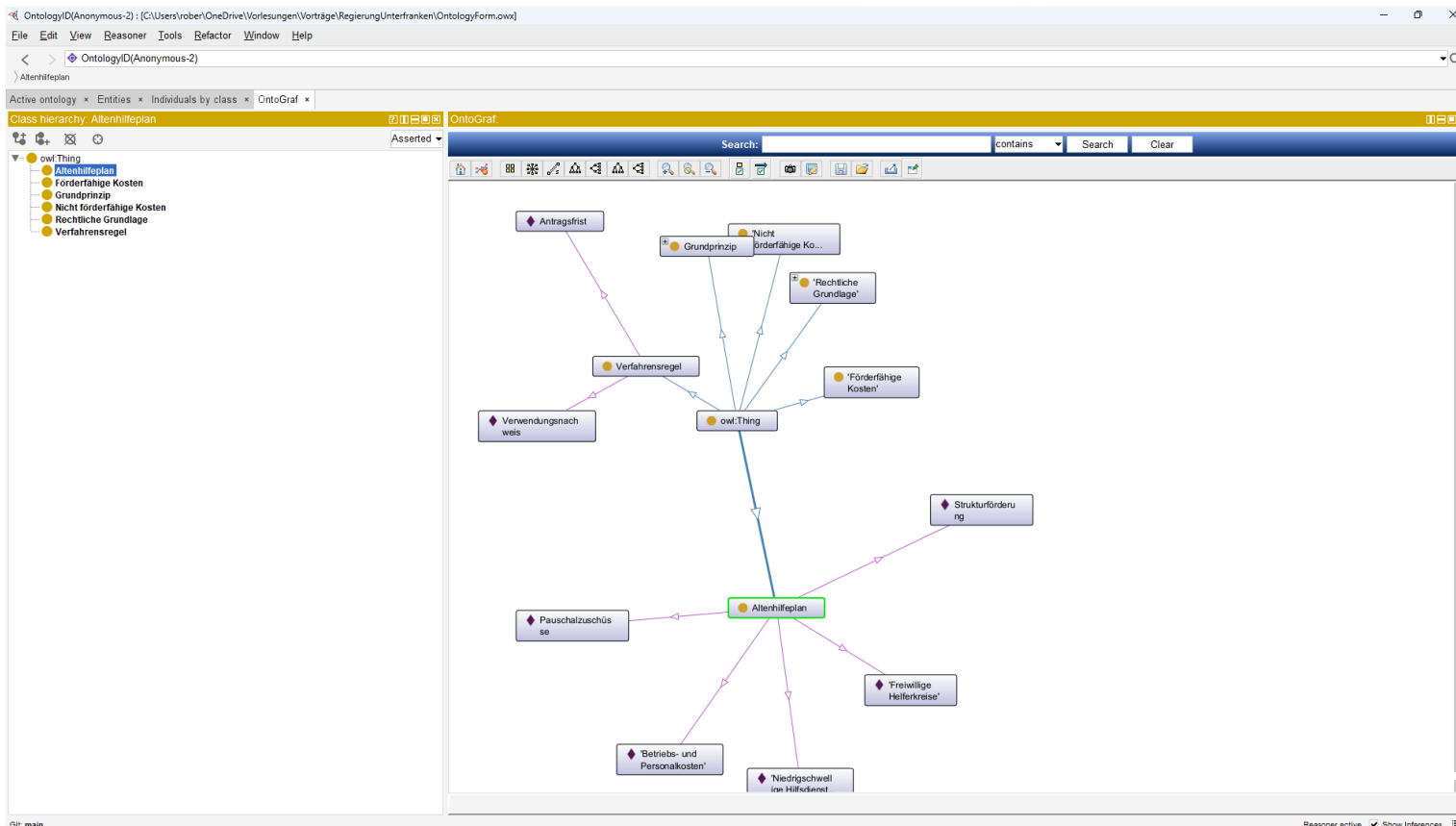
Eine Ontologie ist eine formale, maschinenlesbare Repräsentation von Wissen in einem spezifischen Domänenbereich, die aus Klassen, Eigenschaften, Beziehungen und logischen Axiomen besteht und in Sprachen wie RDF/OWL kodiert wird.

### Vorteile:

- Explizite Wissensmodellierung ermöglicht maschinelles Schlussfolgern (Inferenz)
- Standardisierte Semantik erlaubt Interoperabilität zwischen Systemen
- Graphstruktur ermöglicht komplexe Abfragen und Validierungen auf Basis der modellierten Beziehungen mit SPARQL (Abfragesprache für RDF-Graphen)

# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide



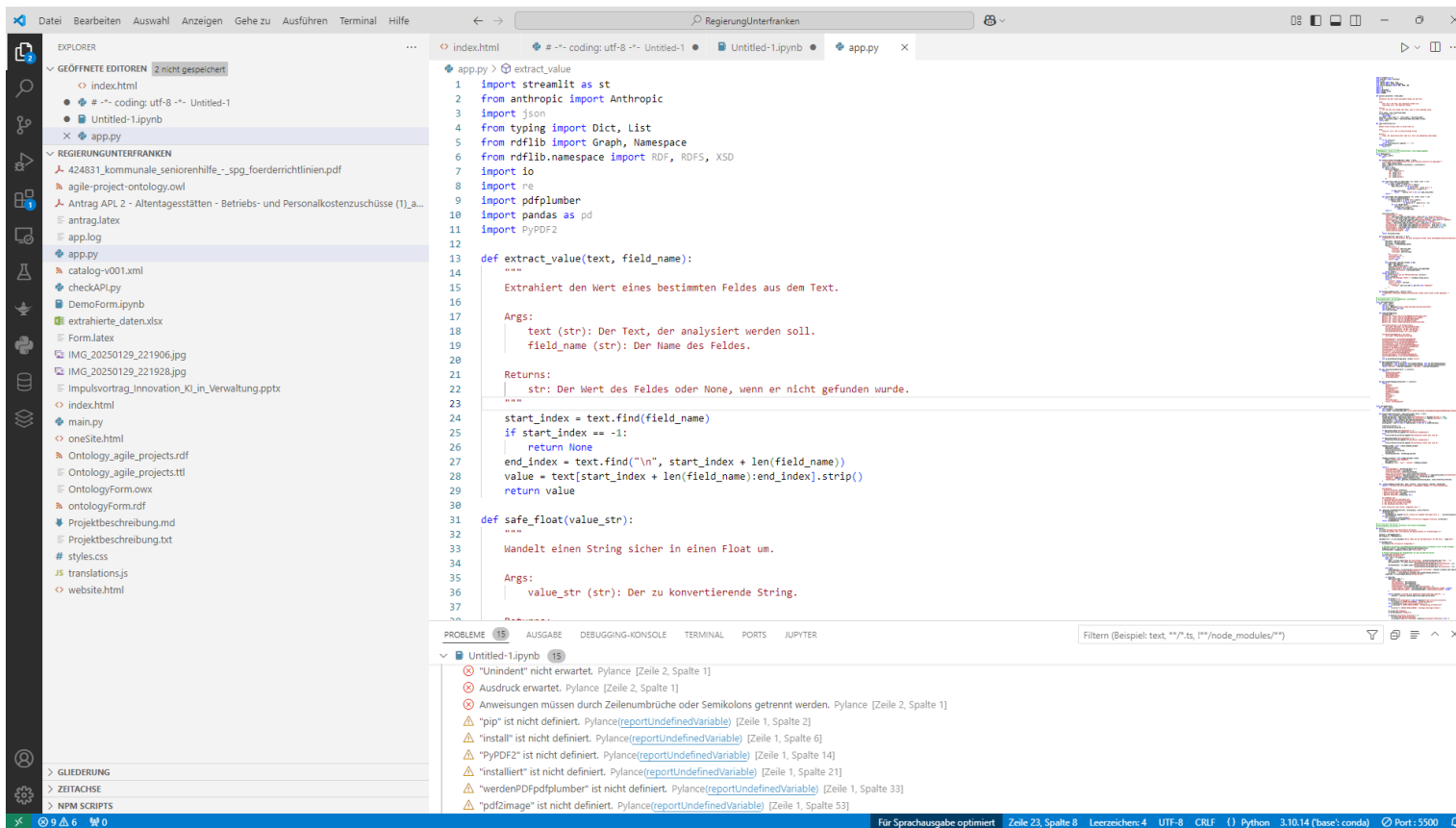
Der Screenshot zeigt eine im Tool Protégé modellierte Ontologie für einen Altemhilfeplan, dargestellt als gerichteter Graph mit Klassen und deren hierarchischen sowie semantischen Beziehungen.

Die Hauptklassen (owlThing als Root) umfassen Konzepte wie "Rechtliche Grundlage", "Förderliche Kosten" und "Verfahrensregel", die durch verschiedene Relationstypen (dargestellt durch unterschiedlich farbige Pfeile) miteinander verbunden sind. Die Baumstruktur auf der linken Seite zeigt die taxonomische Hierarchie der Ontologie, während der Graph in der Mitte die komplexeren semantischen Beziehungen visualisiert. Die Ontologie dient für das POC das Regelwerk.



# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide



Die vorab aufgestellte Ontologie definiert die semantische Basis für die NLP-gestützte Formularanalyse (Python). Die Formularfelder werden dabei über ein definiertes Mapping-Schema mit den entsprechenden Ontologie-Konzepten verknüpft.

Die API von Claude.ai wird genutzt, um basierend auf diesen semantischen Annotationen die Formularinhalte zu analysieren. Die in der Ontologie formalisierten Regeln, Eigenschaften und Beziehungen ermöglichen dann eine automatisierte, wissensbasierte Validierung der eingehenden Anträge.

# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide

Beratungsstelle für Senioren  
Karmelitenstraße 43 - 97070 Würzburg

**Antrag auf Gewährung eines Zuschusses nach dem  
ALTENHILFEPLAN Nr. 2 der STADT WÜRZBURG**  
Altentagesstätten -, Betriebs- und Personalkostenzuschüsse

HAUSHALTSJAHR 2024

Name	Helke Musterfrau
Anschrift	Musterweg 23
Bankverbindung	DE894576
IBAN	4784543
BIC	
Ansprechpartner/in	
Telefon/Handy	
E-Mail	
Träger	Senioritas

Nachgewiesene Höhe der Betriebskosten des Vorjahres 1502

Nachgewiesene Höhe der Personalkosten des Vorjahres (bitte nur die tatsächlichen Kosten für diese Einrichtung angeben und belegen) 1402

Vorhandene Räumlichkeiten des Trägers  ja/nein

Unentgeltlich bereitgestellte Räume anderer Träger  ja/nein

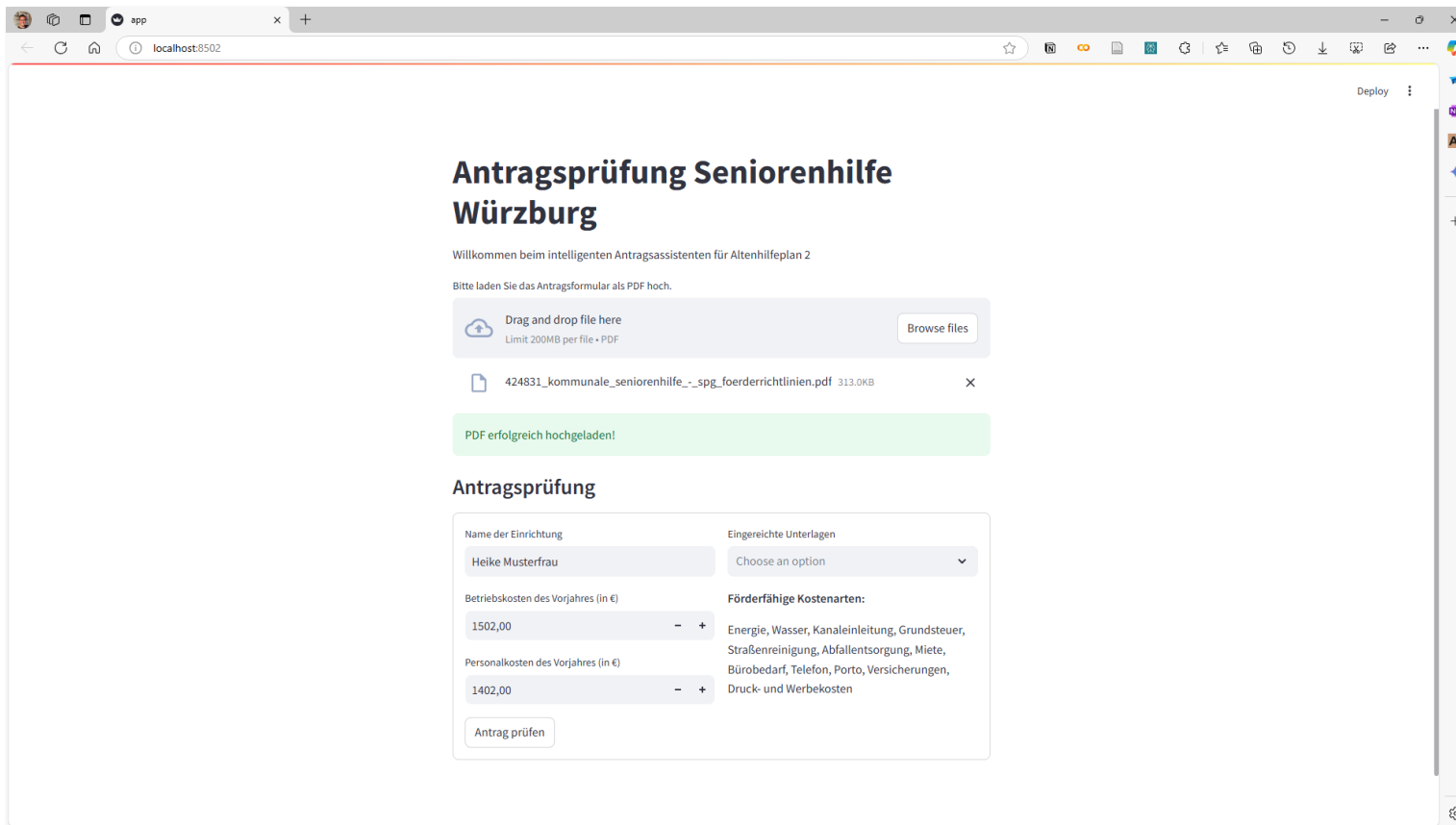
Monatliche Mietzahlungen in Höhe von (Kopie Mietvertrag) 985

Der Screenshot zeigt das mit Fakedaten ausgefüllte Antragsformular, welches als Blanko von dem Internetauftritt der Stadt Würzburg heruntergeladen werden kann.

Das Formular kann am Computer oder handschriftlich ausgefüllt worden sein. Die im Dokument eingetragenen Daten lassen sich mit Python maschinell z.B. in ein Webformular oder in eine Datenbank usw. einlesen. Hat ein Bürger oder Bürgerin die Daten handschriftlich eingetragen, ist das Dokument z.B. per Scanner zu digitalisieren und dann per OCR-Prozess die bildliche Textinformation in Maschinentext zu überführen.

# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide



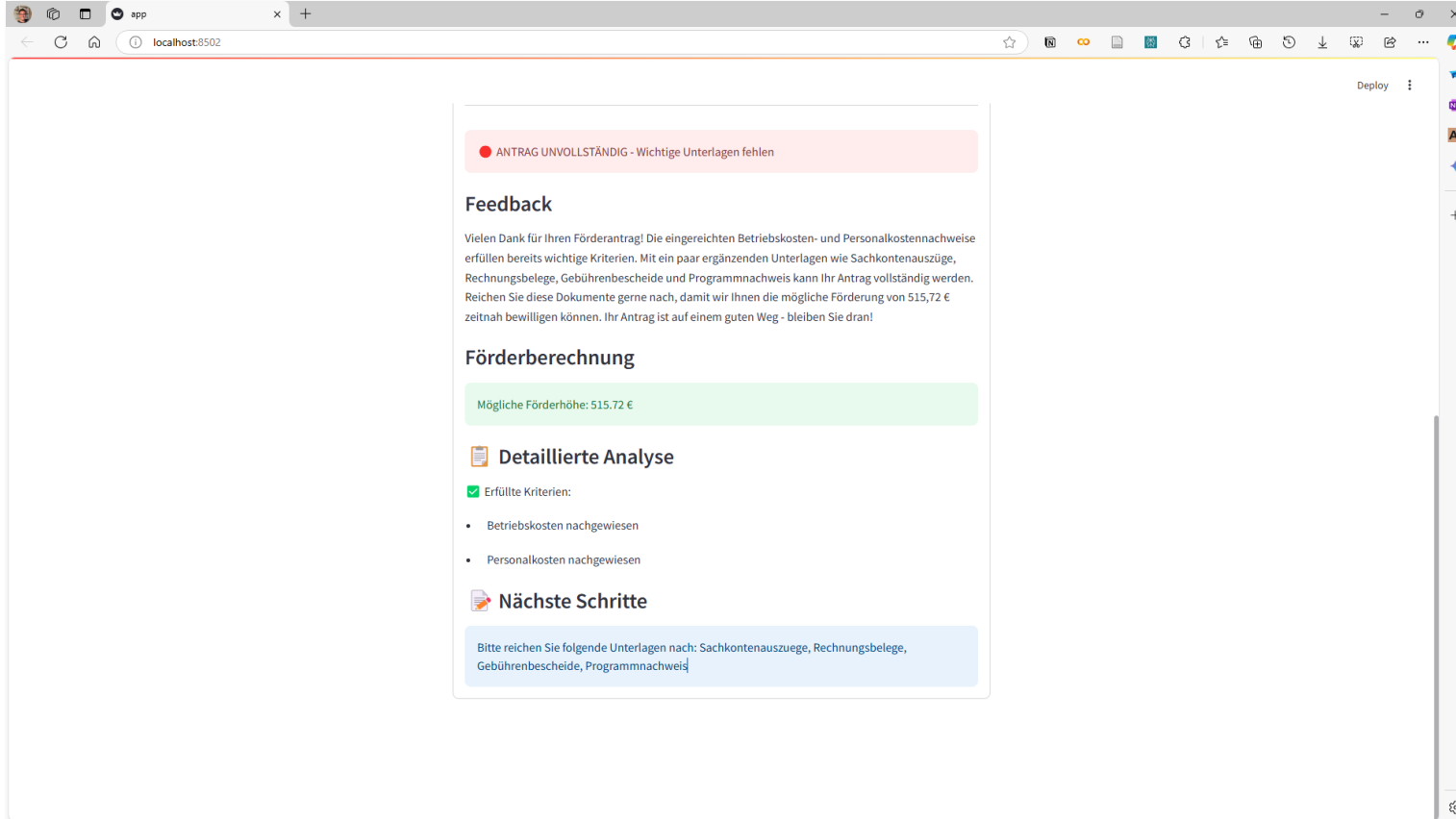
The screenshot shows a web browser window at localhost:8502 displaying a form titled "Antragsprüfung Seniorenhilfe Würzburg". The form is for an "intelligent application assistant for the elderly care plan 2". It features a file upload section with a "Drag and drop file here" area, a "Browse files" button, and a list of uploaded files. A green message indicates "PDF erfolgreich hochgeladen!". Below this is the "Antragsprüfung" section with input fields for "Name der Einrichtung" (Heike Musterfrau), "Eingereichte Unterlagen" (Choose an option), "Betriebskosten des Vorjahres (in €)" (1502,00), and "Personalkosten des Vorjahres (in €)" (1402,00). A "Förderfähige Kostenarten" section lists categories like "Energie, Wasser, Kanalleitung, Grundsteuer, Straßenreinigung, Abfallentsorgung, Miete, Bürobedarf, Telefon, Porto, Versicherungen, Druck- und Werbekosten". A final "Antrag prüfen" button is at the bottom.

Der Screenshot zeigt den im Rahmen des POC erstellten Prototyp: Die Bürgerin lädt ihren Antrag samt Belege auf die Portalseiten der Behörde. Die Felder aus dem Formular werden automatisch eingelesen, alternativ ließen sich auch die Werte direkt in das Webformular schreiben.

Der Klick auf den Button "Antrag prüfen" startet den Überprüfungsprozess.

# Use Case II:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide



In diesem Fall wurde der Antrag abgelehnt, da wichtige Belge beim Upload nicht enthalten waren.

Ein LLM formuliert eine wertschätzende Begründung der Ablehnung, da das soziale Engagement der Antragstellerin gewürdigt werden soll. Es werden Lösungswege aufgezeigt, wie sich das Problem beseitigen ließe. Zudem wird auf die potenzielle Förderhöhe hingewiesen, die der Antragstellerin zustände, wenn die Unterlagen noch eingereicht werden. Alle Reaktionen des Systems sind dynamisch und setzen entweder auf der formalen Ontologie auf oder stammen als Antwort von einem LLM.

# Use Case II – Fazit:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide

Mit den KI-Systemen Google LM Notebook, Claude.ai und Deep Seek im Zusammenspiel mit Python und der API von Claude.ai wurde ein ontologiebasierter und stark automatisierbarer Prüfungsprozess für eingehende Antragsformulare vorgestellt.



### Vorteile



**Sehr leistungsstarke textanalytische Fähigkeiten durch LLM.** RAG nutzt für die Ausgabe ein LLM, welches imstande ist, Texte wie ein Mensch zu formulieren oder Texte in verschiedenen Sprachen ausgeben kann.



**Überführung statischer PDF-Dokumente in semantisch angereicherte Dokumente.** Die mittlerweile durch KI vergleichsweise einfach zu erstellende Ontologie bildet die Grundlage für eine regel-/ inferenzbasierte Automatisierung.



**Semantisch annotierte Formulareingabefelder.** Durch die semantische Annotation werden Daten maschinell interpretierbar und damit auch autonom verarbeitbar. Dies eine der zentralen Aspekte der Vision von Berners-Lee zum Web 3.0.



**Ausbau der Digitalisierung.** Isolierte Formulare sind Medienbrüche und erschweren Prozessautomatisierung. Mittels KI lassen sich Inhalte aus Dokumenten in Formate überführen, die sich gut zum maschinellen Datenaustausch eignen.



# Use Case II – Fazit:

## Ontologien | Automatisierbare Antragsbearbeitung und zeitnahe Bescheide

Mit den KI-Systemen Google LM Notebook, Claude.ai und Deep Seek im Zusammenspiel mit Python und der API von Claude.ai wurde ein ontologiebasierter und stark automatisierbarer Prüfungsprozess für eingehende Antragsformulare vorgestellt.



### Herausforderungen

- **Eventuelle Abhängigkeit zu Big Tech.** Die derzeit leistungsstärksten LLMs sind derzeit noch proprietär. In der Folge stellt sich die Frage zum fairen Zugang zu solchen Technologien und eventuellen Abhängigkeiten.
- **KI als Allheilmittel.** Die vergleichsweise rasche Umsetzbarkeit begünstigt die (problematische) Vermutung, dass sich jedes IT-Problem irgendwie mit KI lösen ließe. Dies weckt falsche Erwartungen und allokiert zu starke finanzielle Mittel.
- **Einhaltung von Datenschutz und EU AI Act.** Aus Datenschutz und Privacy-Gründen sind "Quick Wins" mit Cloud-basierten LLMs nicht ohne weiteres möglich. Eine on-Prem-Umsetzung zieht Rüst- und Wartungsaufwand nach sich.
- **"Wildwuchs" an Lösungen.** Anwendungen lassen sich mit KI vergleichsweise schnell programmieren, sie können aber qualitative Einschränkungen haben oder sich schlecht in die bestehende IT-Landschaft integrieren lassen.

# Use Case III: (Handschriftlich ausgefüllte) Formulare und deren Handling

Das Anwendungsbeispiel zeigt, wie sich mithilfe der API von Claude.ai und der Programmiersprache Python ein interaktives System zur Handschrifterkennung aufbauen lässt. Hierbei dient ein Formular als Grundlage für ein per generativer KI erstelltes Datenmodell und liefert Impulse für ein verbessertes Daten- und Prozessmanagement.



**Effizienzsteigerung und Zeitersparnis:** Handschriftlich ausgefüllte Formulare müssen nicht manuell in IT-Systeme übertragen werden, was nicht nur den Prozess beschleunigt, sondern auch den Menschen von Routineaufgaben befreit. Wichtig aber, OCR funktioniert trotz großer Fortschritte auch nicht verlässlich mit 100 % Genauigkeit.



**Kopplung der Erkennung mit Datenmanagement:** Die Texterkennung der handschriftlichen Formulareingaben wirft auch die Frage auf, wie und in welcher Technologie die Daten zu speichern sind. Formulare können auch komplexe Datenstrukturen repräsentieren und so böte es sich an, die erkannten Daten in einer NoSQL-Datenbank zu speichern.



**Erhöhte Zugänglichkeit und Bürgerfreundlichkeit.** Auch wenn papierne Formulare aus einer IT-Sicht nicht ideal bleiben, stellen sie für Bürgerinnen und Bürger keine allzu hohe Einstiegshürde dar. Zudem lassen sich auch Endgeräte für den Prozess nutzen, etwa wenn der Antrag per Smartphone fotografiert und dann hochgeladen wird.

# Use Case III: (Handschriftlich ausgefüllte) Formulare und deren Handling

Das Anwendungsbeispiel zeigt, wie sich mithilfe der API von Claude.ai und der Programmiersprache Python ein interaktives System zur Handschrifterkennung aufbauen lässt. Hierbei dient ein Formular als Grundlage für ein per generativer KI erstelltes Datenmodell und liefert Impulse für ein verbessertes Daten- und Prozessmanagement.

Personliche Daten - Fiktives Formular

Section 1: Persönliche Informationen

Vollständiger Name: Heike Musterfrau

Geburtsdatum: 04.02.1983

Telefon: 09104/7350

E-Mail: H.Musterfrau@mail.com

Section 2: Adresse

Straße & Hausnummer: Münzstraße 12

Stadt: Musterort

PLZ: 93045

Land: Deutschland

Section 3: Kontoinformation (fiktiv)

Bankname: Demobank

Währung: EUR

IBAN/Kontonummer: DE 89 3704 0044 0532 0130 00

Routing-Nummer (z. B. BIC):

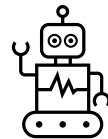
Hinweis: Dieses Formular ist rein fiktiv und dient nur zu Übungszwecken. Verwenden Sie keine echten persönlichen oder finanziellen Daten.

Die Ausgangslage bilden zwei (fiktive und per KI erzeugte Formulare), die ausgedruckt und handschriftlich ausgefüllt wurden.

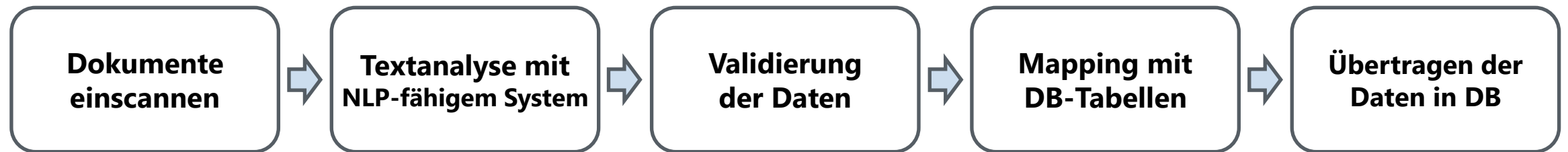
Für die Behörde ergeben sich damit etliche Herausforderungen:

- Digitalisierung des Papierformulars
- Archivierung des Belegs
- Maschinelle Verarbeitung

# Use Case III: (Handschriftlich ausgefüllte) Formulare und deren Handling



Unterstützung durch KI



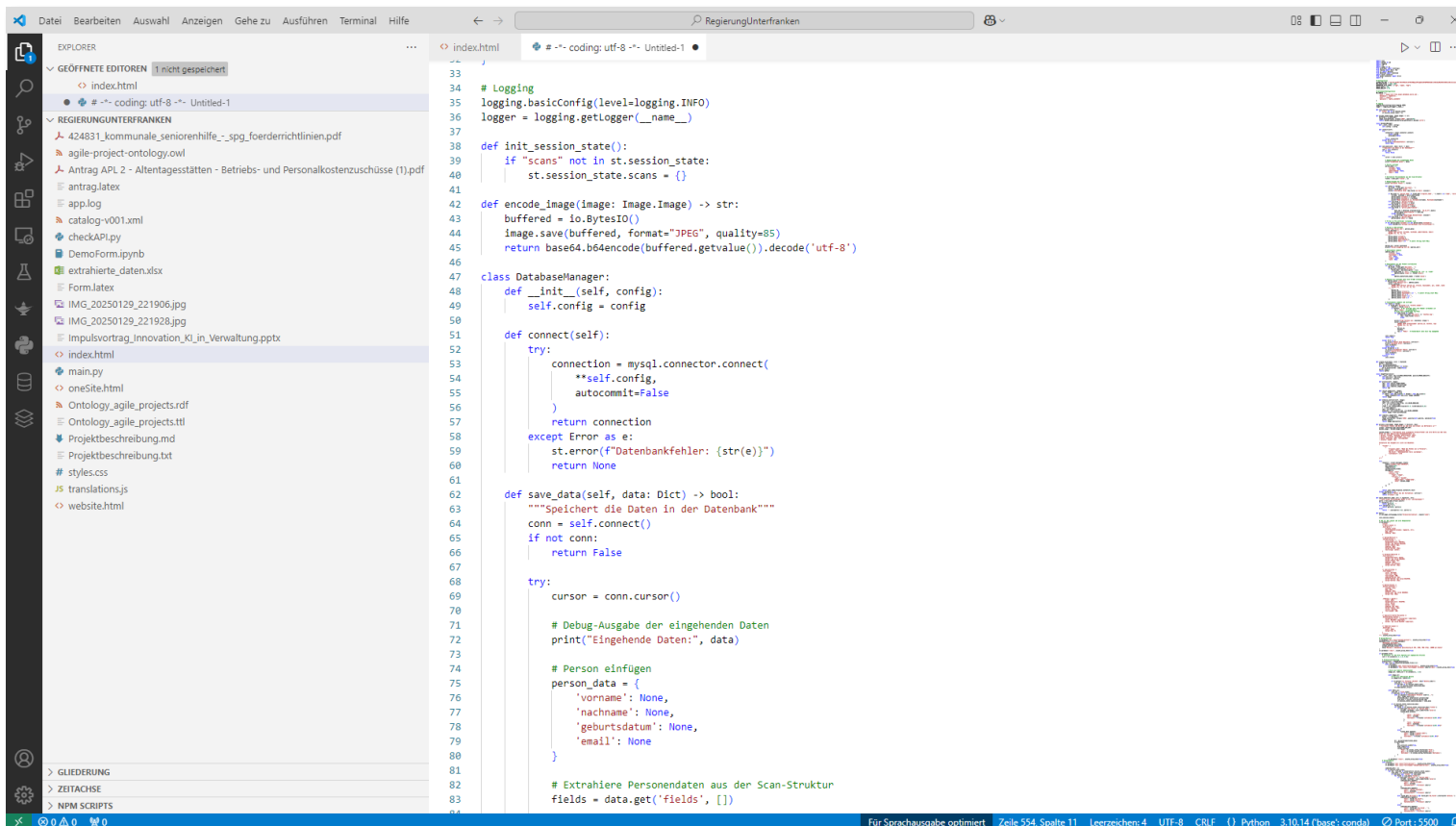
- Scan der Dokumente
- Self-Service durch Bürger
- Behörde bietet Upload-Möglichkeit für Beleg an

- Vorverarbeitung durch OCR
- Semantische Analyse
- Erkannte Entitätstypen bilden die Relationen (beim relationalen DB-Modell)
- Generierung eines DB-Modells und einer DB durch ein LLM

- Validierung der erkannten Daten
- Automatische Korrektur nicht valider Daten
- Mapping der per OCR in Maschinentext überführten Daten mit den korrespondierenden Attributen in den jeweiligen Datenbank-Tabellen (bei relationalem DB-Modell)

- Speichern der Daten in einer Datenbank-Tabelle oder in einer Collection (z.B. MongoDB)

# Use Case III: (Handschriftlich ausgefüllte) Formulare und deren Handling



```
33
34 # Logging
35 logging.basicConfig(level=logging.INFO)
36 logger = logging.getLogger(__name__)
37
38 def init_session_state():
39     if "scans" not in st.session_state:
40         st.session_state.scans = []
41
42 def encode_image(image: Image.Image) -> str:
43     buffered = io.BytesIO()
44     image.save(buffered, format="JPEG", quality=85)
45     return base64.b64encode(buffered.getvalue()).decode('utf-8')
46
47 class DatabaseManager:
48     def __init__(self, config):
49         self.config = config
50
51     def connect(self):
52         try:
53             connection = mysql.connector.connect(
54                 **self.config,
55                 autocommit=False
56             )
57             return connection
58         except Error as e:
59             st.error(f"Datenbankfehler: {str(e)}")
60             return None
61
62     def save_data(self, data: Dict) -> bool:
63         """Speichert die Daten in der Datenbank"""
64         conn = self.connect()
65         if not conn:
66             return False
67
68         try:
69             cursor = conn.cursor()
70
71             # Debug-Ausgabe der eingehenden Daten
72             print("Eingehende Daten:", data)
73
74             # Person einfügen
75             person_data = {
76                 'vorname': None,
77                 'nachname': None,
78                 'geburtsdatum': None,
79                 'email': None
80             }
81
82             # Extrahiere Personendaten aus der Scan-Struktur
83             fields = data.get('fields', [])
```

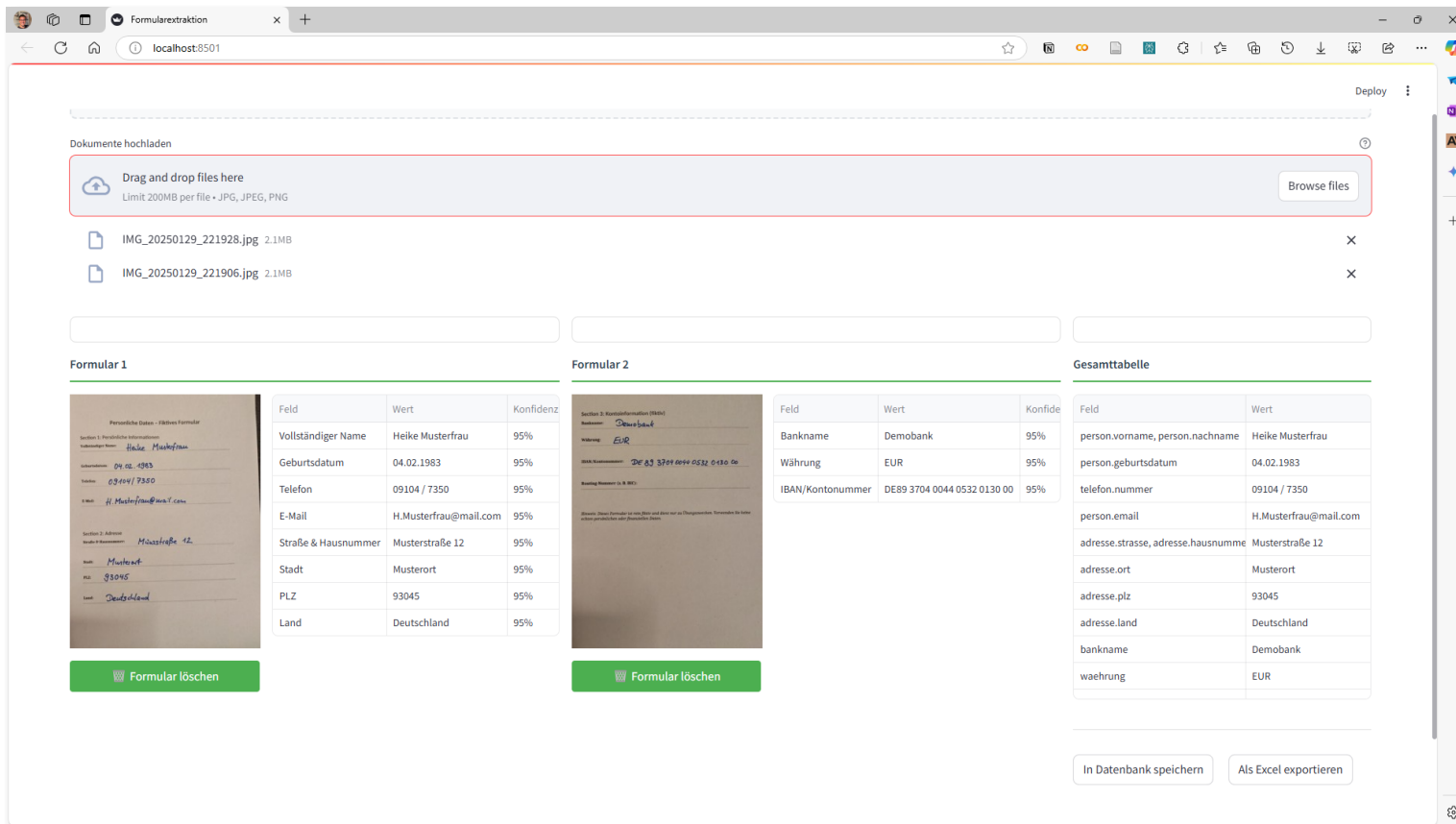
Ausschnitt aus der Python-Datei zur Aufbereitung des Dokuments, NLP-Analyse und Identifikation der relevanten Entitätstypen aus dem Formular.

Ein solches Formularfeld ist das Attribut "Vollständiger Name" eines Entitätstyps wie "Person". Zur Erfüllung der 1. Normalform (1NF) wird es in die Attribute "Vorname" und "Nachname" aufgespalten, im Projekt mithilfe regulärer Ausdrücke.



# Use Case III:

## (Handschriftlich ausgefüllte) Formulare und deren Handling



Dokumente hochladen  
 Drag and drop files here  
 Limit: 200MB per file • JPG, JPEG, PNG  
 Browse files

IMG\_20250129\_221928.jpg 2.1MB  
 IMG\_20250129\_221906.jpg 2.1MB

**Formular 1**

Feld	Wert	Konfidenz
Vollständiger Name	Heike Musterfrau	95%
Geburtsdatum	04.02.1983	95%
Telefon	09104 / 7350	95%
E-Mail	H.Musterfrau@mail.com	95%
Straße & Hausnummer	Musterstraße 12	95%
Stadt	Musterort	95%
PLZ	93045	95%
Land	Deutschland	95%

Formular löschen

**Formular 2**

Feld	Wert	Konfide
Bankname	Demobank	95%
Währung	EUR	95%
IBAN/Kontonummer	DE89 3704 0044 0532 0130 00	95%

Formular löschen

**Gesamttabelle**

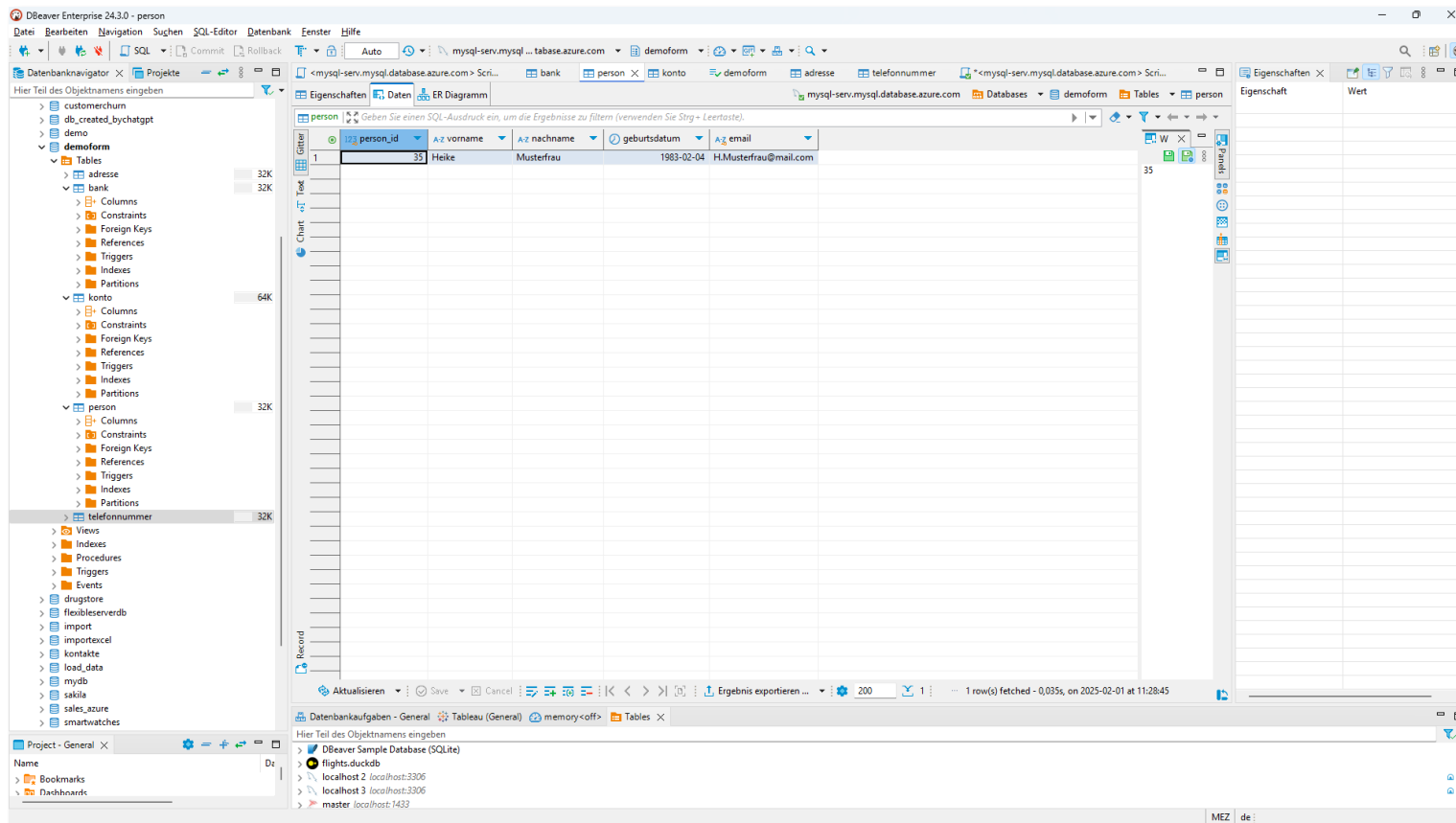
Feld	Wert
person.vorname, person.nachname	Heike Musterfrau
person.geburtsdatum	04.02.1983
telefon.nummer	09104 / 7350
person.email	H.Musterfrau@mail.com
adresse.strasse, adresse.hausnumme	Musterstraße 12
adresse.ort	Musterort
adresse.plz	93045
adresse.land	Deutschland
bankname	Demobank
waehrung	EUR

In Datenbank speichern    Als Excel exportieren

Der Screenshot zeigt die verarbeiteten Formulare und die zu speichernden Daten.

Die dritte Spalte ("Gesamttabelle") zeigt die Daten, wie sie in die Datenbank gespeichert werden. In diesem Szenario dient MySQL als Datenbankserver, der auf Microsoft Azure gehostet wird. Die Webanwendung schreibt die Daten nach einer Validierung (derzeit nicht implementiert) in die jeweiligen Datenbank-Tabellen. In diesem Use Case wird eine relationale Datenbank verwendet, eine flexiblere Alternative wäre jedoch, die erkannten Datenfeld-Wert-Paare als Key-Value in JSON-Notation zu speichern, z.B. in einer dokumentenorientierten NoSQL-DB wie MongoDB.

# Use Case III: (Handschriftlich ausgefüllte) Formulare und deren Handling



Der Screenshot (DBeaver) zeigt den erfolgreich gespeicherten Datensatz 😊.

In diesem POC wurde ein relationales DBMS (MySQL Server) genutzt. Flexiblere Möglichkeiten böten NoSQL-Datenbanksysteme und semistrukturierte Datenformate wie etwa XML oder JSON, da sie weniger restriktiv im Schema sind und sich leichter an verändernde Datenanforderungen anpassen lassen.

Diese Systeme eignen sich insbesondere dann, wenn sich die Datenstrukturen im Laufe der Zeit ändern oder wenn unstrukturierte bzw. heterogene Daten zu verarbeiten sind.



# Use Case III – Fazit :

## (Handschriftlich ausgefüllte) Formulare und deren Handling

Es wurde mit SQL, Python und der API von Claude.ai ein Prototyp erstellt, der Daten aus einem unstrukturierten Textdokument ausliest und in eine strukturierte (= tabellierte) Datenbank (als eine von mehreren) Speicheroptionen überführt.



### Vorteile



**Sehr leistungsstarke textanalytische Fähigkeiten durch LLM.** RAG nutzt für die Ausgabe ein LLM, welches imstande ist, Texte wie ein Mensch zu formulieren oder Texte in verschiedenen Sprachen ausgeben kann.



**Überführung von Inhalten aus unstrukturierten Texten in ein Datenbanksystem.** Texte besitzen zwar eine inhärente linguistische Struktur und Semantik, diese wurden durch NLP in eine maschinell verarbeitbare Form überführt.



**Effiziente und reibungslosere Datenverarbeitung.** Die in den Formularen enthaltenen Daten werden in ein (hier relationales) Datenbanksystem überführt, was ein effizientes, sicheres und konsistentes Datenmanagement begünstigt.



**Ausbau der Digitalisierung.** Isolierte Formulare sind Medienbrüche und erschweren Prozessautomatisierung. Mittels KI lassen sich Inhalte aus Dokumenten in Formate überführen, die sich gut zum maschinellen Datenaustausch eignen.

# Use Case III – Fazit :

## (Handschriftlich ausgefüllte) Formulare und deren Handling

Es wurde mit SQL, Python und der API von Claude.ai ein Prototyp erstellt, der Daten aus einem unstrukturierten Textdokument ausliest und in eine strukturierte (= tabellierte) Datenbank (als eine von mehreren) Speicheroptionen überführt.



### Herausforderungen

-  **Eventuelle Abhängigkeit zu Big Tech.** Die derzeit leistungsstärksten LLMs sind derzeit noch proprietär. In der Folge stellt sich die Frage zum fairen Zugang zu solchen Technologien und eventuellen Abhängigkeiten.
-  **KI als Allheilmittel.** Die vergleichsweise rasche Umsetzbarkeit begünstigt die (problematische) Vermutung, dass sich jedes IT-Problem irgendwie mit KI lösen ließe. Dies weckt falsche Erwartungen und allokiert zu starke finanzielle Mittel.
-  **Einhaltung von Datenschutz und EU AI Act.** Aus Datenschutz und Privacy-Gründen sind "Quick Wins" mit Cloud-basierten LLMs nicht ohne weiteres möglich. Eine on-Prem-Umsetzung zieht Rüst- und Wartungsaufwand nach sich.
-  **"Wildwuchs" an Lösungen.** Anwendungen lassen sich mit KI vergleichsweise schnell programmieren, sie können aber qualitative Einschränkungen haben oder sich schlecht in die bestehende IT-Landschaft integrieren lassen.