# City Service Agent

## Berlin prototype v0.1

Sven Djokic 31/01/2025

# Our Problem

The services of many cities are not provided in a user-friendly way to their citizens.

In many cities the services:

- take a lot of time to process
- are only available in one language
- their user experience and interfaces feel outdated
- cost a lot of taxpayers' money

# Improve your City's Services with Agentic AI

The CITY SERVICE AGENT enables you to provide an AI chatbot-like experience for your citizens

**< Citizen >**

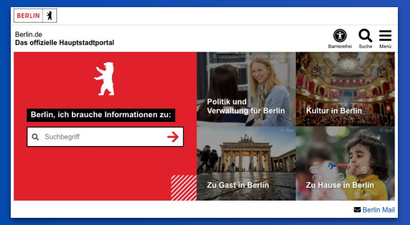
- Can ask questions about your city's services in a familiar chat interface

**< City Agent >**

- Has all data about your city's services stored in a vector database
- Understands the question, finds relevant information and summarizes it for the citizen
- Adds relevant links so citizen can check info on your official websites
- Is always available, speaks all languages and never gets tired!

# How the first Prototype berlin-v0.1 works

**berlin.de**




1060 web pages about the city's services

Stored as 3,792 records in a Pinecone vectorstore

Models used to create embeddings:
text-embedding-ada-002

**Citizens**

Ask questions in the chat

**App**

Chat interface deployed locally via Gradio

**City Agent**

Powered by gpt-4o-mini

Tool 1:
Retrieves relevant info from Pinecone

Tool 2:
Summarizes retrieved info and appends URLs

Big Brother LangSmith is always watching

# The Dataset for our Prototype

## 1st Scraping

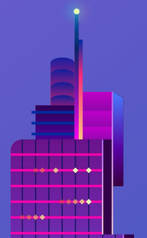
- Used the Scrapy library to build a spider
- Polite scraping to prevent getting blocked
- Spider crawled and scraped all 1060 pages of berlin.de
- Extracted text, title, source URL, sub-links
- Got rid off all html tags

## 2nd Chunking and Embedding

- Split up text of each page into chunks
- Appended the title of page to each chunk
- Added title, source URL, chunk #, text excerpt as Metadata
- Includes
  - Retry mechanism and debug prints
  - Sleep.time of 12 seconds to prevent hitting rate limits

## 3rd Upserting

- Created embeddings get upserted into Pinecone
- Includes
  - Retry mechanism and debug prints
  - Sleep.time of 12 seconds to prevent hitting rate limits



# Introducing our Agent

## The Agent

- gpt-4o-mini
- AgentType.OPENAI_FUNCTIONS
- Function to chat with agent is a simple While Loop

## The Memory

- ConversationBufferWindowMemory
- k=5 (number of past interactions retained)

## The Prompt

CUSTOM_PROMPT = """
You are an expert assistant specializing in providing detailed and comprehensive information about Berlin services.
Answer queries step by step, ensuring clarity and accuracy.

Respond to the user's question but do NOT generate references or links yourself. We (the system) will append any relevant source links automatically after your summary.

Important: The final answer must only include the links that are strictly relevant to the user's query. Omit any link that is NOT relevant to the user's question. Include links that ARE relevant.

Example:
User: "Wie melde ich mich in Berlin an?"

Assistant:
1. **Terminvereinbarung:** Vereinbaren Sie einen Termin beim örtlichen Bürgeramt in Berlin. Dies kann oft online oder telefonisch erfolgen.
2. **Erforderliche Dokumente:** Bringen Sie folgende Unterlagen mit:
  - Einen gültigen Personalausweis
  - Ein aktuelles biometrisches Passfoto
  - Ihren Mietvertrag oder Nachweis der Wohnadresse
3. **Gebühren:** Die Gebühren variieren je nach Stadtteil.
4. **Bearbeitungszeit:** Die Bearbeitung dauert in der Regel etwa eine Woche.
5. **Abschluss:** Nach der Bearbeitung erhalten Sie eine Meldebescheinigung.

Now, respond to the following question:

User: {input}

Assistant:
"""

# The Agent's Toolbox

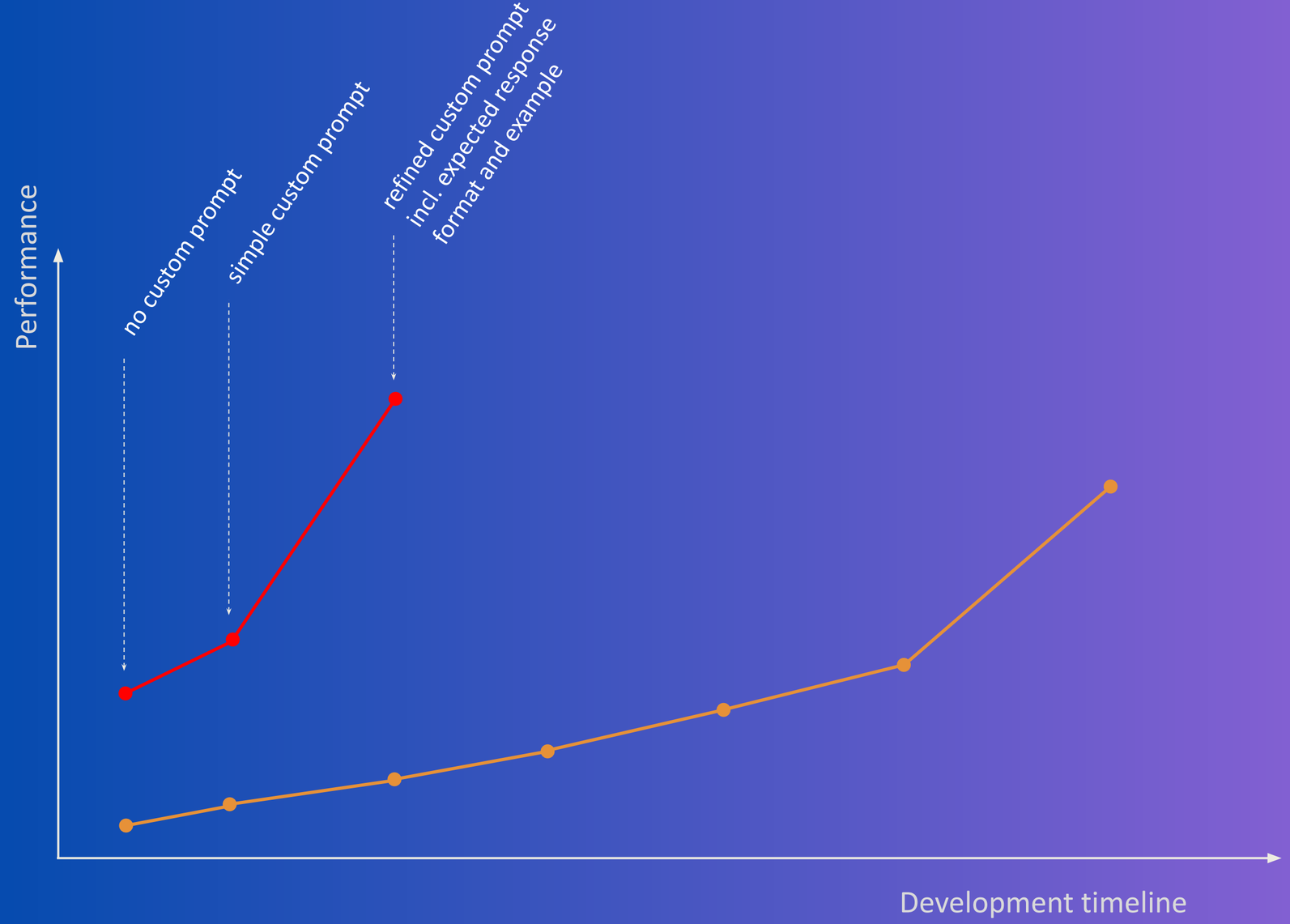
## Tool 1: Retriever

- Searches in Pinecone for embedded chunks that match the user's query
- Retrieves the matching chunks
- Provides first summary of the combined chunks

## Tool 2: Summary Generator

- Reviews the answer of the first tool and rewrites it
- Appends the source URLs of the pages of the retrieved chunks

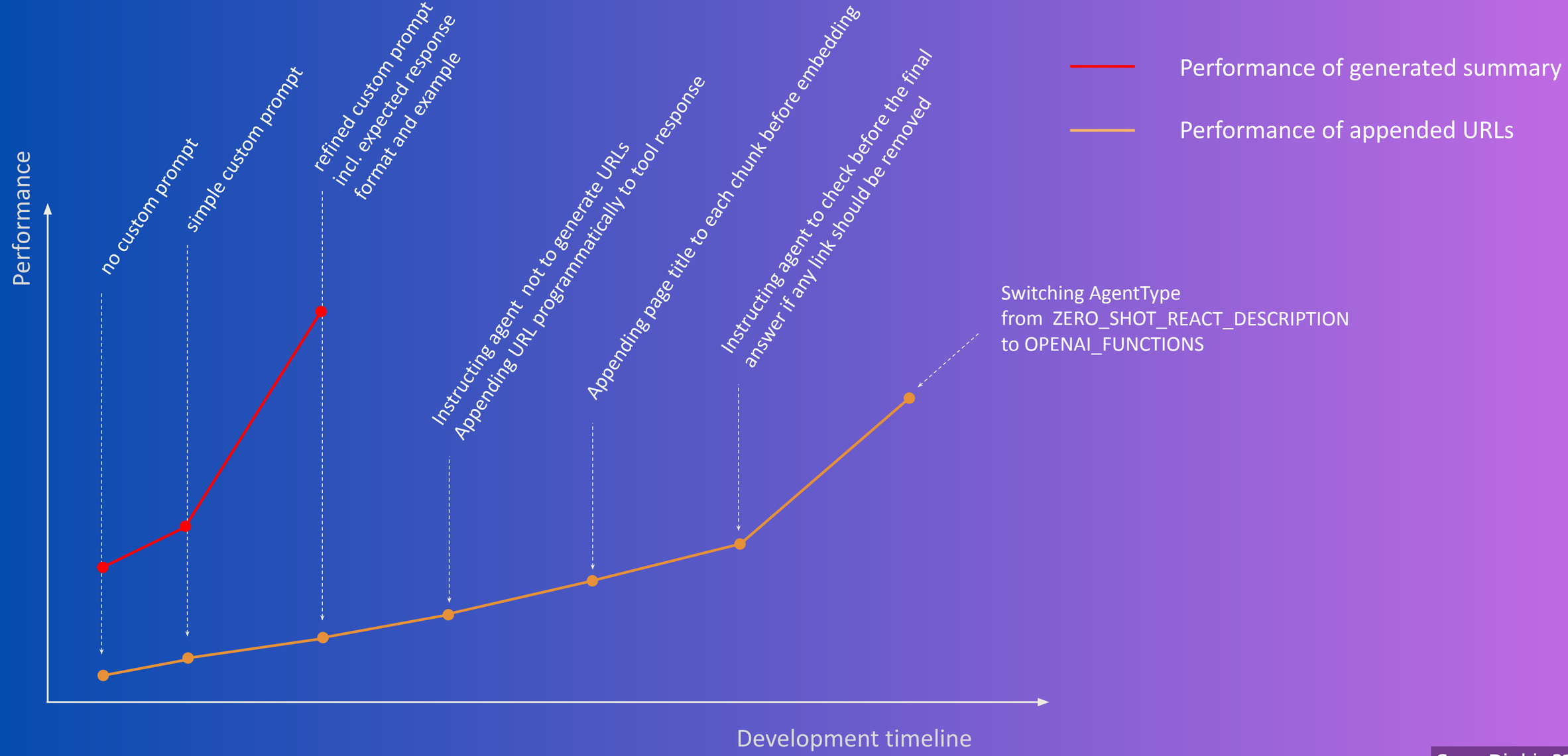
# Evaluating the Agent's Performance via manual testing




Performance of generated summary

Performance of appended URLs

# Evaluating the Agent's Performance via manual testing and LangSmith

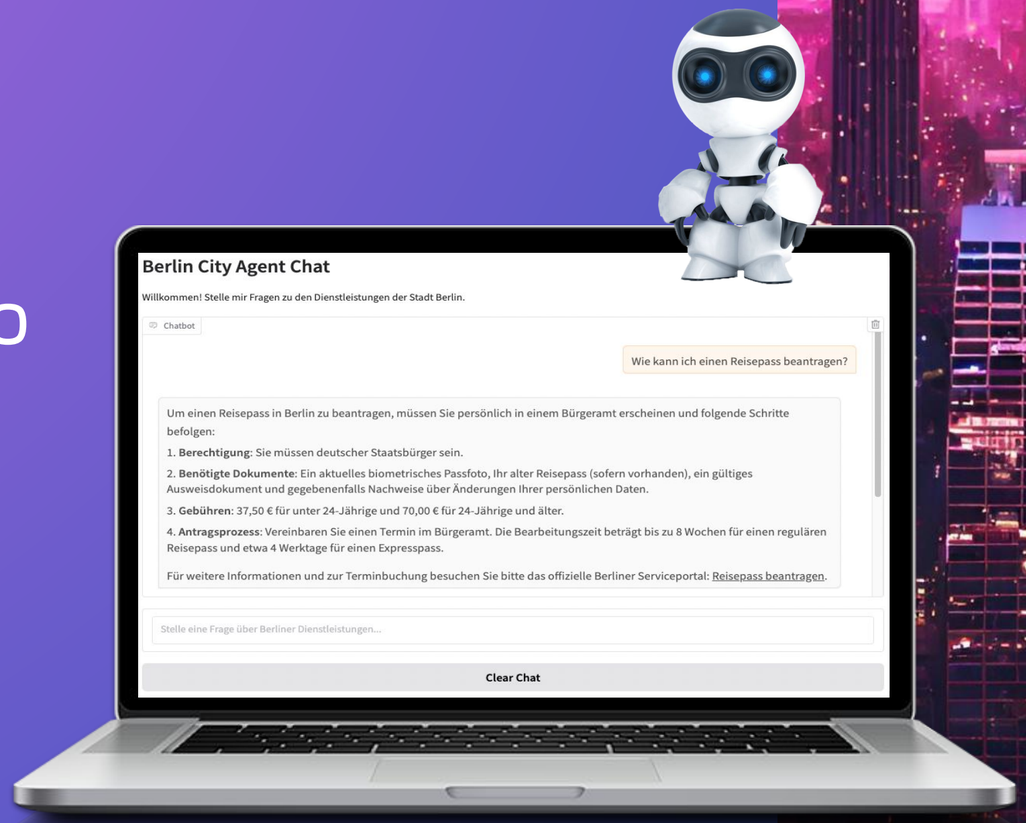


# Conclusions

- Not only the selected model but also the selected agent type make a big impact on the generated responses
- Refactoring code is a thing
- Speed of generating responses is an issue
- Keeping unrelated (metadata) out of the final response can be tricky

# The Future of City Services is Agentic

Now let's see the demo



# Thank you!

Any questions?