d!gital cluster uri

Urner MCP-Server

Abschlusspräsentation

Challange

Ziel

Schnittstelle für KI-Modelle schaffen, um auf **aktuelle Urner Informationen** zuzugreifen.

Wieso?

KI haben **veraltete** Infromationen, **halluzinieren** bei lokalen Fragen oft oder finden über die **Web Suche das Resultat nicht.**

Lösungsansatz

Nutzung des Model Context Protokoll (MCP)

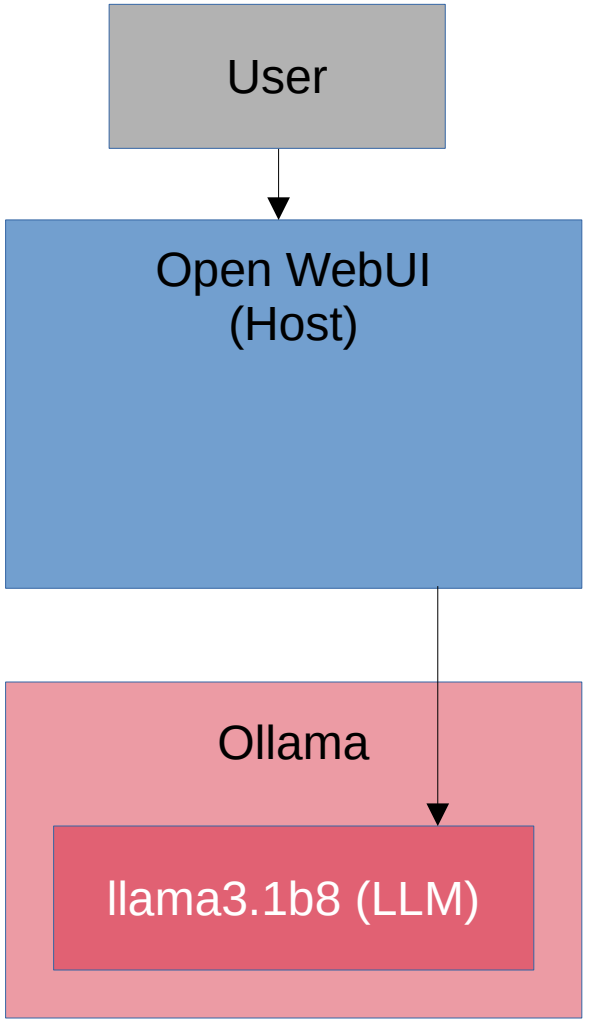
Architektur

Nutzung eines Chat Clients mit einem lokalen LLM, z.B.

Welche Events finden heute in Uri statt?

Probleme:

- Der LLM kennt das aktuelle Datum nicht
- Das LLM kennt keine aktuellen Events, da er mit alten Trainingsdaten von 2023 trainiert wurde



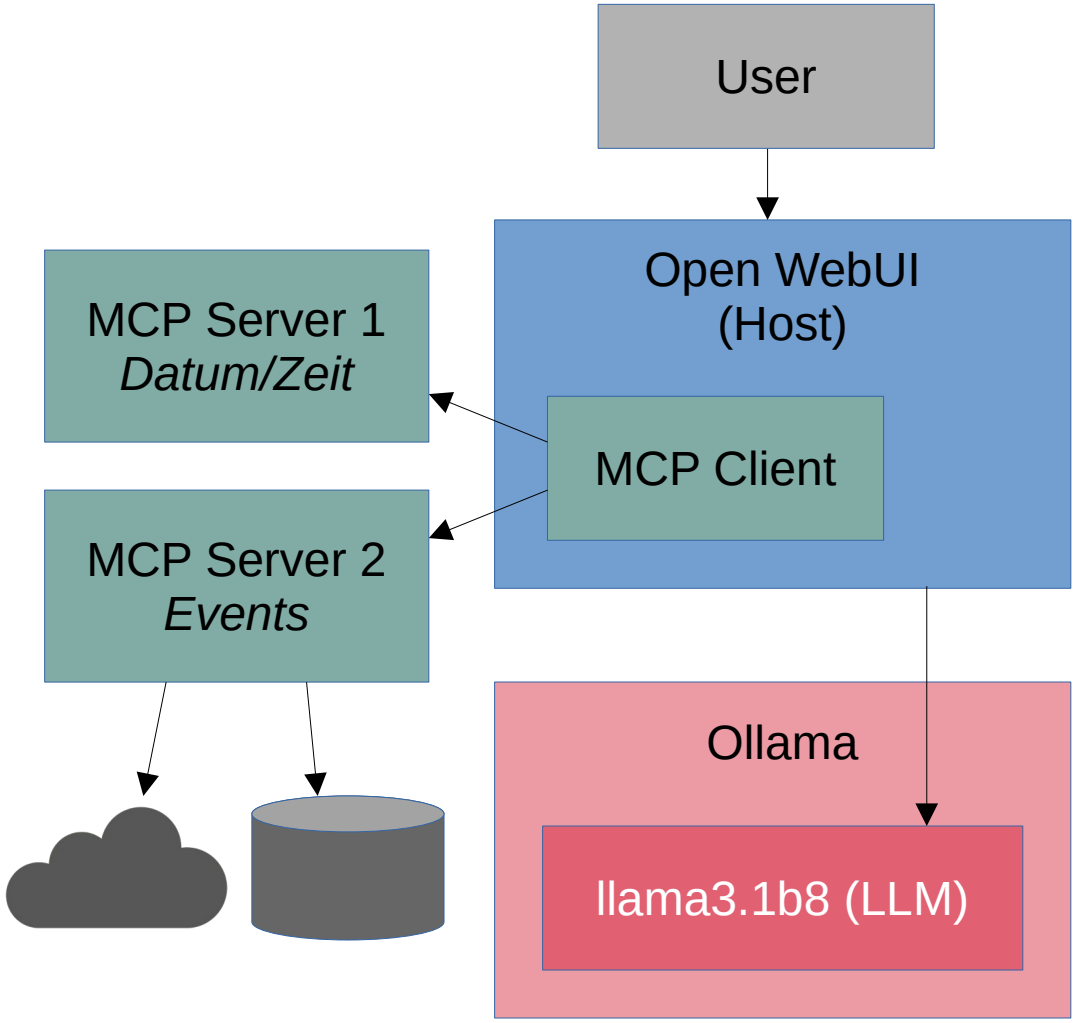
Architektur

Einbindung von MCP Servern in die Architektur.

Wichtig: Der Host muss einen MCP Client implementieren, z.B.

- Open WebUI (experimental)
- Claude Desktop

Daten könne “live” oder über einen Cache abgefragt werden.

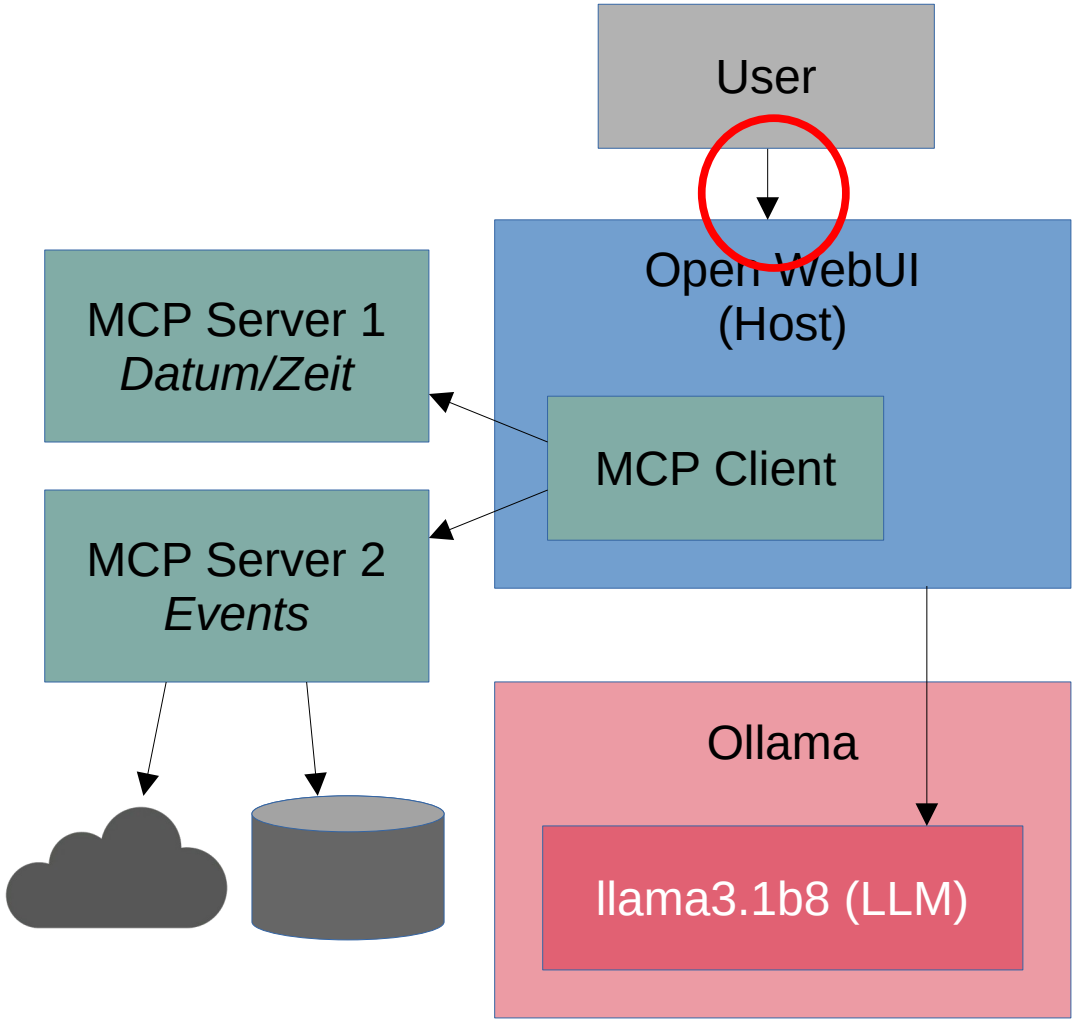


Ablauf

1)

User gibt einen Input im Chat:

Welche Events finden heute in Uri statt?



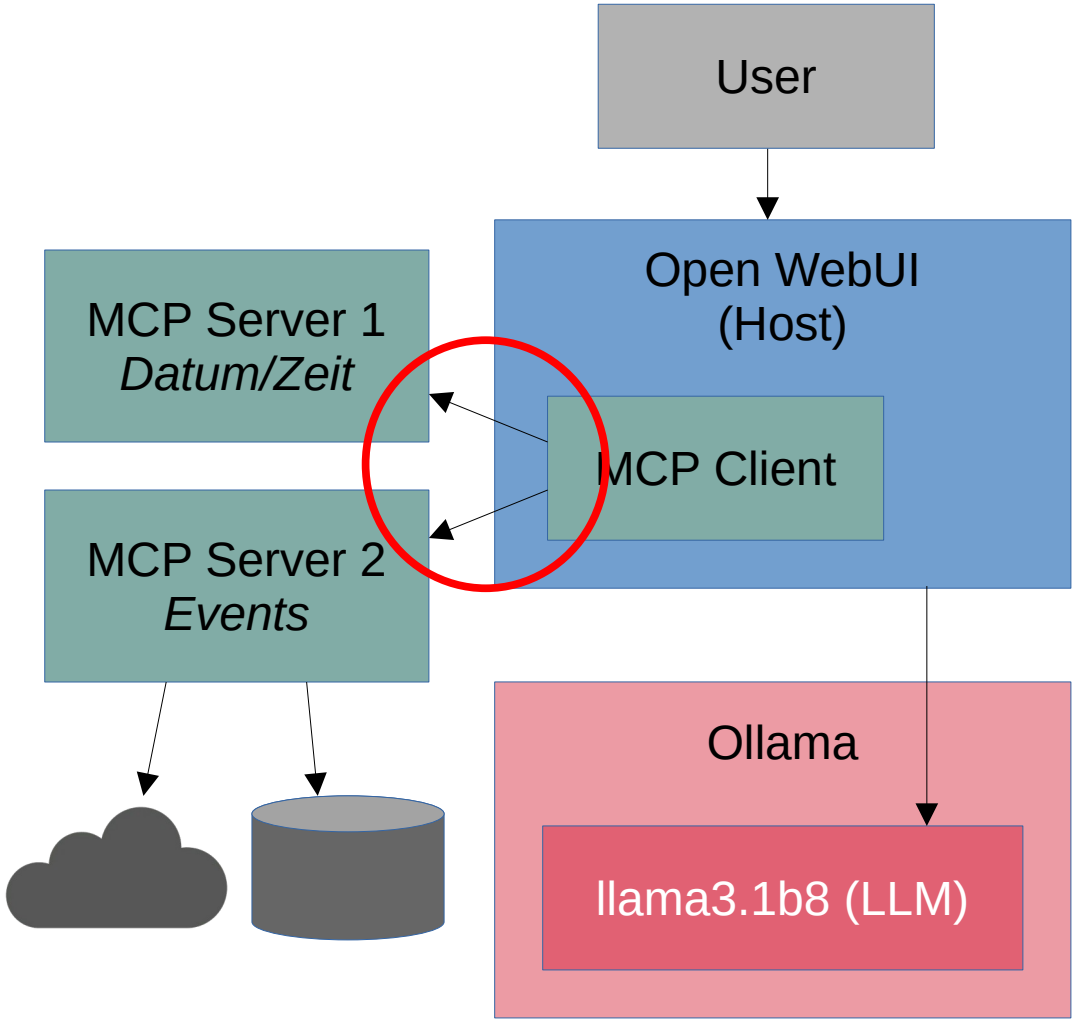
Ablauf

2)

Open WebUI stellt alle Tools zusammen:

- MCP Server 1: get_date, get_time
- MCP Server 2: get_events(date)

Die Funktionen besitzen eine Beschreibung in natürlicher Sprache.

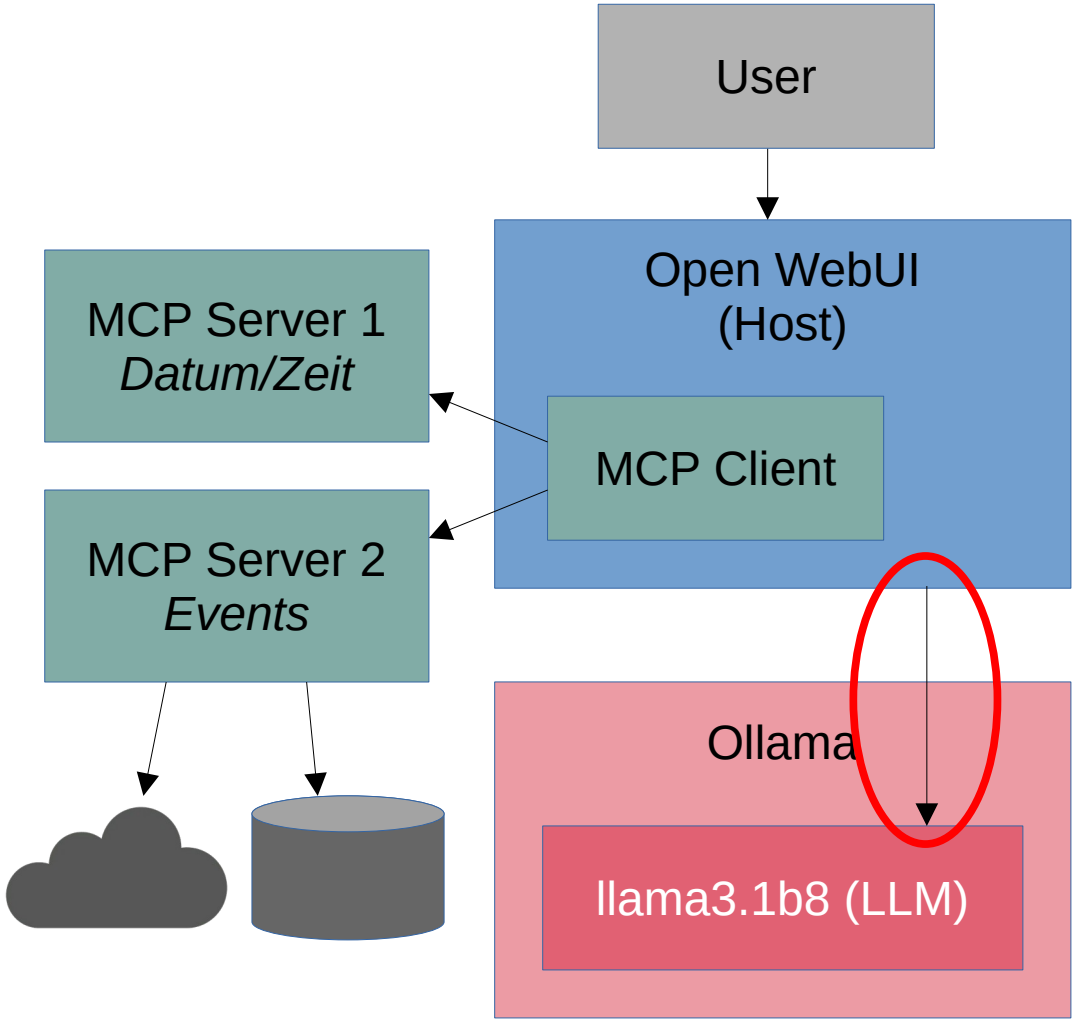


Ablauf

3)

Open WebUI sendet den **Prompt** des Users und die **Liste der verfügbaren Tools** ans LLM.

Das LLM meldet zurück, dass das Tool **get_date** genutzt wird.

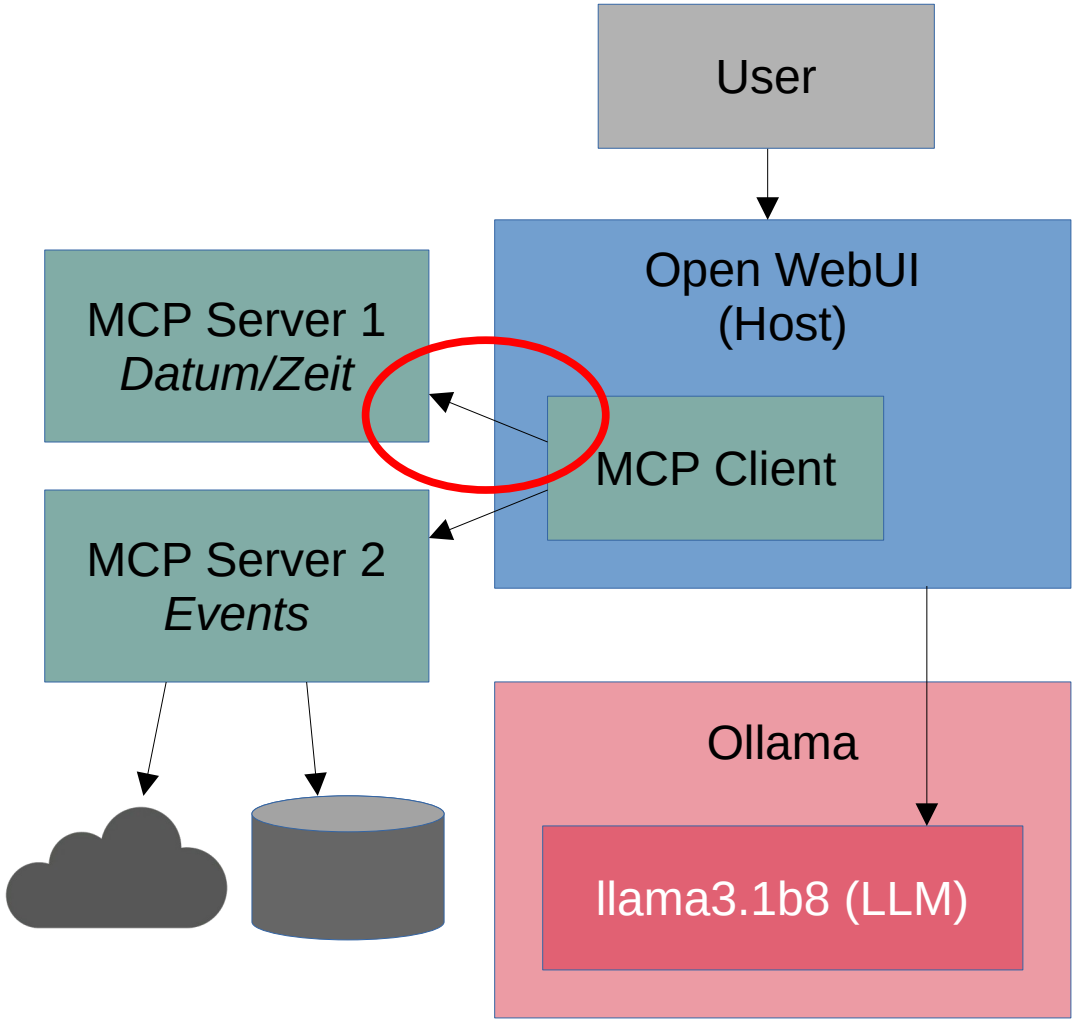


Ablauf

4)

Via MCP Client wird das aktuelle Datum abfragt:

get_date

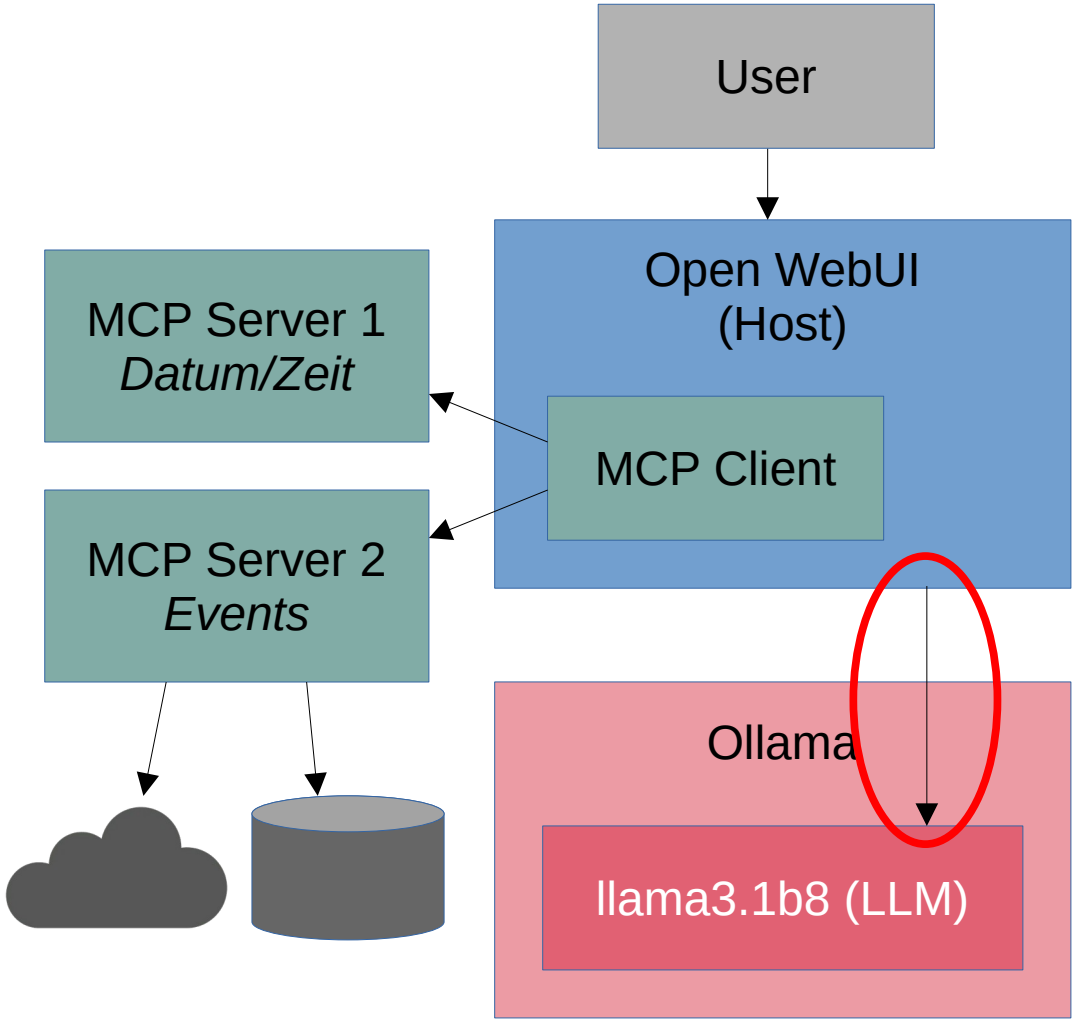


Ablauf

5)

Open WebUI sendet das aktuelle Datum ans LLM.

Das LLM meldet zurück, dass das Tool **get_events** mit dem Parameter **date=2026-03-28** verwendet werden soll.

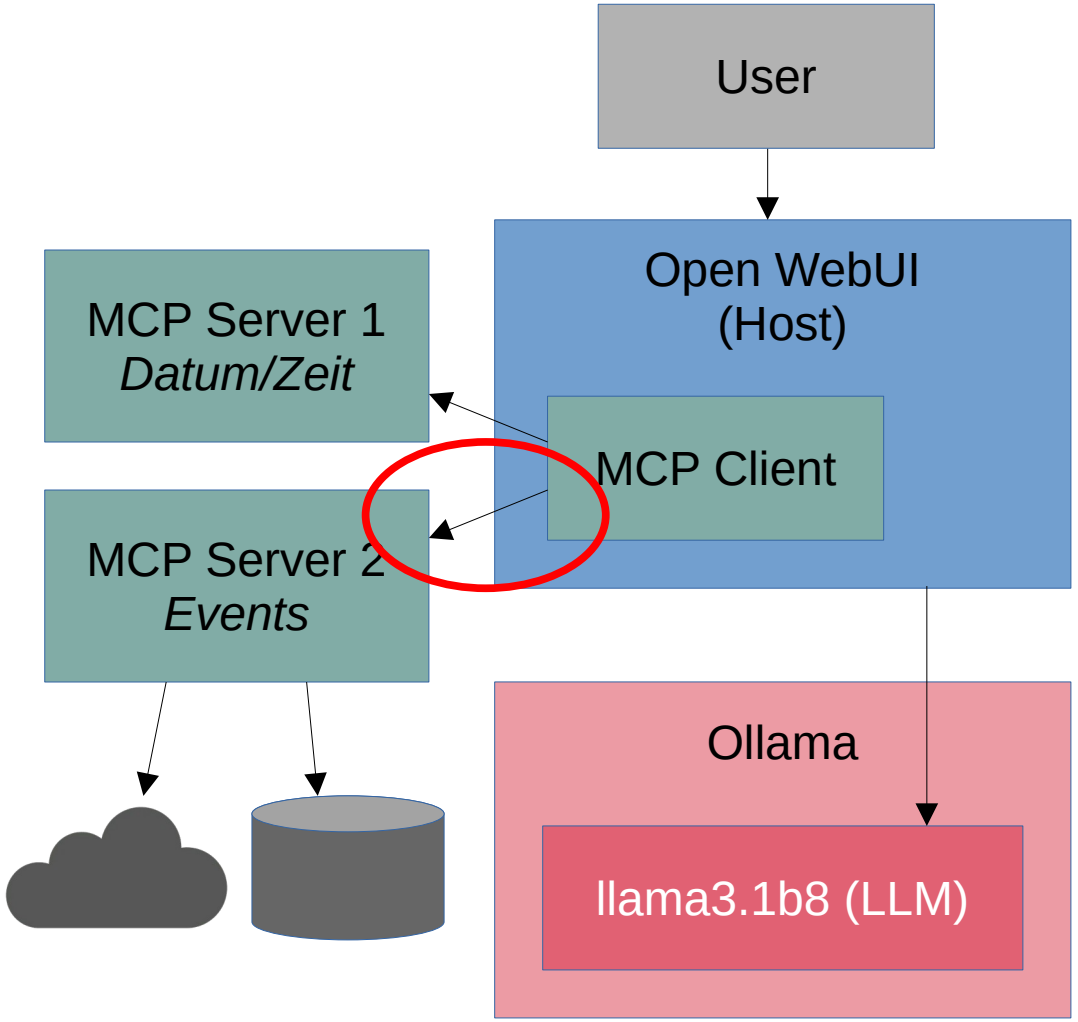


Ablauf

6)

Via MCP Client werden die Events von heute abgefragt:

get_events(2026-03-28)



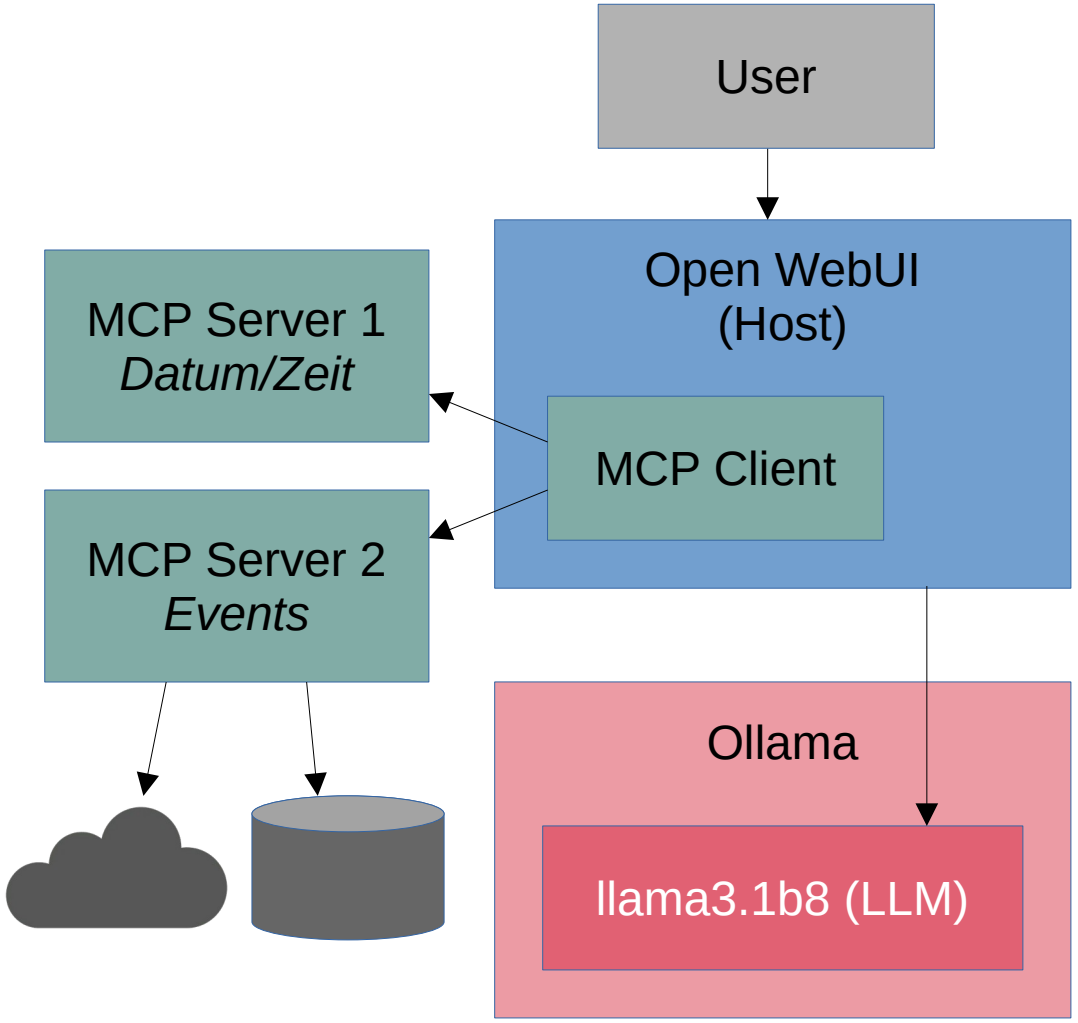
Ablauf

7)

Open WebUI sendet die Liste der Events ans LLM.

Das LLM meldet die Ausgabe den den Benutzer:

Heute am 28.03.2026 finden in Uri folgende Events statt:
- Data Hackdays Uri 2026
...



Demo

d!gital cluster uri

Ziele erreicht?

Basis-Architektur mit Host, LLM, MCP-Server

- Nachvollzogen und reproduzierbar aufgebaut
- Dockerfiles und Compose im Repo verfügbar

MCP Server

- Server 1 (Python): Veranstaltungen + Kinoprogramm, Erweiterbarkeit gegeben
- Server 2 (Rust): PoC mit aktuellem Datum und Zeit

Datenquellen

- Web Scraping für Kinoprogram implementiert
- MCP Serven nutzen gecachte JSON Dateien

Challanges

- Mehrere Betriebssysteme (Windows, Mac OS & Linux) und Architekturen (x86/64, ARM64)
- Zusammenarbeit der Komponenten verstehen und Komponenten integrieren
- Tool Design
  - Saubere Parametrisierung fürs LLM (z.B. Datumsformat)
  - Tools Spezifizierung: Generalisierung vs. Spezialisierung
- Integration unterschiedlicher Datenquellen und -formate

Outlook

- Data-Loading zu definieren und aufzubauen
  Wie können Dritte Daten auf die Plattform pushen?
- MCP Tool Design für die Daten
- Definition Hosting und Betrieb
- User Dokumentation und Veröffentlichung

Fragen?

d!gital cluster uri