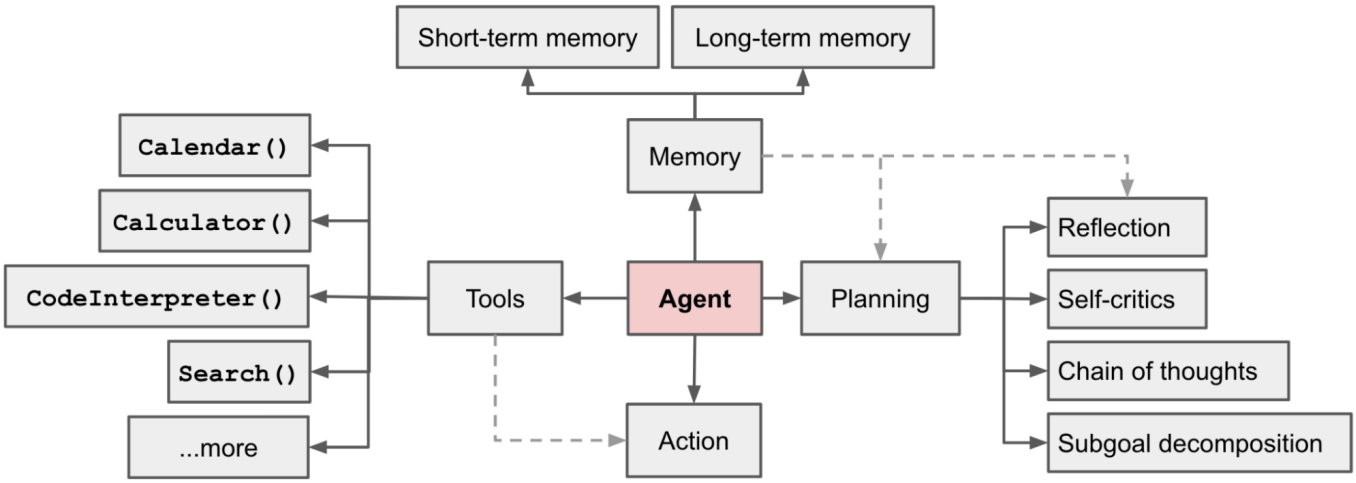
Masterarbeit "Privacy Enabled Agents" - Konzept

Problemstellung

Bei der Konzeption von KI-gestützten Systemen steigt die Nutzung des Pattern LLM-basierter Autonomer Agenten. Grundsätzlich werden hierbei große Sprachmodelle (Large Langage Models / LLM) eingesetzt, wie es auch kontrollierteren Umgebungen wie Chatbots oder Sprachassistenten der Fall ist. Die LLMs werden hierbei jedoch mit drei weiteren Komponenten[^1] verbunden:

- **Planung**: Der Agent plant und führt Aktionen aus, um seine Ziele zu erreichen.
- **Erinnerung**: Der Agent speichert Informationen über die Umgebung und seine Interaktionen.
- **Nutzung von Werkzeugen**: Der Agent nutzt Werkzeuge wie Funktionen oder externe Schnittstellen, um seine Ziele zu erreichen.



Durch die hohe Autonomie der Agenten und der zugrunde liegenden LLMs, die in der Regel aus der Cloud konsumiert werden, entstehen jedoch auch Risiken für die Privatsphäre der Nutzer. Der Aufruf der Werkzeuge selbst findet zwar in der Regel in der Umgebung des Betreibers statt, jedoch werden eventuelle Ein- und Ausgaben der Werkzeuge in der Regel an die LLMs weitergeleitet. Hierbei können sensible Daten wie personenbezogene Daten oder Geschäftsgeheimnisse betroffen sein. Dies ist besondere in Bereichen wie der medizinischen Versorgung, der Finanzbranche oder der öffentlichen Verwaltung problematisch.

Lösungsansatz

Um die Privatsphäre der Nutzer zu schützen, soll ein System entwickelt werden, das die Übertragung von sensiblen Daten an die LLMs verhindert. Oft ist diese nämlich nicht notwendig, um die Ziele des Agenten zu erreichen. Dieser kann auch mit Pseudodaten oder Platzhaltern entscheiden, welches Werkzeug aufzurufen ist. Hierzu eignet sich eine Proxy-Schicht vor dem LLM, die alle Eingaben mithilfe eigener KI-Modelle prüft und gegebenfalls anonymisiert bzw. pseudonymisiert. Die Ersetzungen werden dann in einem Speicher abgelegt, um sie bei der Ausgabe wiederherzustellen. Welche Form der Ersetzung (Pseudonym, Platzhalter, ...) vorgenommen wird

Architektur

Die Architektur des Systems besteht aus drei Hauptkomponenten:

- **Agent**: Der Agent ist für die Planung und Ausführung von Aktionen verantwortlich. Er kommuniziert mit dem Proxy und den Werkzeugen.
- **Proxy**: Der Proxy ist für die Anonymisierung und Pseudonymisierung von Daten verantwortlich. Er kommuniziert mit dem Agenten und den Werkzeugen.
- **Werkzeuge**: Die Werkzeuge sind für die Ausführung von Aktionen verantwortlich. Sie kommunizieren mit dem Agenten und dem Proxy.

 alt text

Technologien

Für die Umsetzung des Systems sollen folgende Technologien verwendet werden:

- **Python**: Die Programmiersprache Python wird grundlegend für die Implementierung des Systems verwendet. Hierfür sprechen die hohe Verbreitung und die gute Unterstützung von KI-Bibliotheken.
- **LangChain / LangGraph**: LangChain bzw. LangGraph werden für die Orchestrierung der Komponenten verwendet. Dort sollen ggf. neue Komponenten für die Anonymisierung und Pseudonymisierung von Daten entwickelt werden.
- **FastAPI**: Das Web-Framework FastAPI wird für die Implementierung der REST-Schnittstellen verwendet.
- **Langfuse**: Langfuse wird zum Tracing und zur Evaluation der gesamten Architektur verwendet.
- **Podman**: Podman wird für die Containerisierung der Komponenten verwendet.
- **Compose**: Compose wird für das Deployment der Container verwendet.

Weitere vom Pattern abhängige Technologien und Komponenten wie z.B. Storage / Caches können im Laufe der Implementierung hinzugefügt werden.

Evaluation

Neben der grundsätzlichen Funktionalität des Systems soll vor allem die Qualität der Erkennung sowie die Kompatibilität mit diversen LLMs evaluiert werden. Die Qualität der Erkennung wird anhand von zu erstellenden Testdaten und Benchmarks evaluiert. Die Kompatibilität mit diversen LLMs wird anhand von zu erstellenden Testszenarien evaluiert.

Infrage kommende LLMs sind aktuelle Marktführer mit grundsätzlichem Support für Werkzeuge (Tool-Calls), die auch für Agenten grundsätzlich in Frage kommen bzw. genutzt werden. Dies sind zum aktuellen Zeitpunkt unter anderem:

- Anthropic Claude 3 Opus
- MistralAI Large-v2 / Nemo
- OpenAI GPT-4 / GPT-4o
- Cohere Command R+
- Google Gemini

Zeitplan

Der Zeitplan für die Implementierung des Systems sieht wie folgt aus:

- **September / Oktober 2024**: Konzeption und Planung

- **November / Dezember 2024**: Implementierung der Komponenten
- **Januar / Februar 2025**: Integration und Test
- **März / April 2025**: Evaluation und Dokumentation
- **Mai 2025**: Abschluss der Masterarbeit

Quellen

[^1] Lilian Weng: LLM Powered Autonomous Agents; 2023-06-23; Quelle: https://lilianweng.github.io/posts/2023-06-23-agent/