LoKI User Guide

Benutzerhandbuch / User Manual

Version: 1.0.0 and 1.1.0
Projekt: LoKI – Local AI Assistant
Autor: Knut Schneider
Datum: 08.05.2026

Inhaltsverzeichnis / Table of Contents

Deutsch

1. LoKI

LoKI ist ein Desktop-Assistent zum lokalen Ausführen von Large Language Models (LLMs) In der Version 1.1.0 sind auch Multimodale LLM (Image→Text) integriert. Das Programm wurde in C++ mit dem Qt-Framework entwickelt und nutzt die llama.cpp Inferenzbibliothek. LoKI lädt abhängig von der Hardware automatisch das passende Backend (CPU, CUDA, Vulkan oder OpenCL). Dadurch kann das Programm sowohl auf reinen CPU-Systemen als auch auf GPU-beschleunigten Systemen betrieben werden. Je nach dem was das Programm beim ersten Start vorfindet werden eventuell System-Treiber über ein install-Skript nach installiert.

Das AppImage enthält alle notwendigen Laufzeitkomponenten wie Python, Node-Module und verschiedene Runtime-Bibliotheken. Dadurch kann LoKI weitgehend unabhängig von der Linux-Distribution betrieben werden.

Ordnerstruktur

Beim ersten Start mit --install wird ein Arbeitsordner erstellt:

LoKI/

├─ Channels (Telegram, WhatsApp, weitere werden folgen)
├─ Data (Chat-Kontext Datenbank)
├─ Docs (Dokumentation)
├─ Model (installierte LLM Modelle)
├─ ModelUseFiles (Dateizugriff für LLM’s)
├─ Plugins (MCP Tools)
└─ Voice (TTS Stimmen)

Data: Dort befindet sich die sqlite3 (ist auch im AppImage)Datenbank für den Chat-Kontext. Der Chat-Kontext ist auf der Hauptseite von LoKI auf der linken Seite und führt alle geführten Chats auf. gespeicherte Chats können wieder aktiviert und fortgesetzt werden oder per rechten Mausklick gelöscht werden. Ist man in einem Chat kann man mit dem Button „+ Neuer Chat“ in einen neuen Chat wechseln.

Docs: Hier liegen die Dokumentation.

Model: Hier werden die LLM’s gespeichert die per Download oder Drag&Drop erstellt werden. Beim Download kann man im Download Bereich von LoKI aus einer Liste verfügbarer LLM’s (Hugging Face) auswählen. Es können reine .GGUF Modelle aber

auch HF Modelle geladen werden, wobei bei den HF Modellen nach dem Download das Modell automatisch nach GGUF gewandelt wird, da nur GGUF Modelle geladen werden können. Auf der Downloadseite ist es auch möglich ein GGUF Modell zu Requantisieren. Das bringt mehr Speed geht aber zu lasten der Genauigkeit des Modells. Weiteres in Quantisierung-Doc.

ModelUseFiles: Das ist der Ordner den das geladen Model benutzen kann um Files anzulegen und zu manipulieren. Im MCP-Modus „Local“ sind ein paar Tools die dieses ermöglichen. Der Pfad zum Ordner ist in den Settings frei wählbar und kann auch wo anders hin gelegt werden. **Achtung**: Überlegen sie genau worauf sie eine LLM zugreifen lassen!

Plugins: Dieser Ordner enthält die Ordner „local“, „stdio“ und „http“ in denen sich die Beispiel-Tools befinden. Dazu siehe „Tools-Doc“.

Voice: Hier sind ein paar Stimmen für die akustische Ausgabe. Es sind deutsche und englische Stimmen vorhanden. In diesem Zusammenhang vielen Dank an Thorsten-Voice für seine Stimme mit frei wählbaren Emotionen die man auch über die Settings einstellen kann. (Hier gilt mein Dank an Thorsten-Voice)

Es können auch andere Stimmen dort gespeichert werden. Es ist nur zu beachten das die Stimme mit ihrer .json Konfigurations-Datei gespeichert wird.

Nach dem ersten Start von LoKI mit **–install** kann man dann das Programm auch direkt aus „Anwendungen“ oder durch Doppelklick starten. Es ist keine Konsole mehr notwendig.

2. Editor

Der Editor ist ein integrierter Code- und Texteditor. Er ermöglicht das Bearbeiten von Plugin-Scripts, Konfigurationsdateien und Dokumentationen. Syntax-Highlighting erleichtert dabei die Arbeit mit JSON, Python oder JavaScript.

Der Editor kann Dokumente direkt an das aktuell laufende Modell, über die 3 unteren Buttons, übergeben, um z. B. Code zu erklären, zu überprüfen oder automatisch Dokumentation generieren zu lassen.

Tipp: wenn man in den Settings den ModelUseFiles-Pfad auf den Ordner Plugins/lokal, Plugins/stdio oder Plugins/http setzt kann das LLM über die File-Tools direkt auf die Tools zugreifen, lesen, mit dem richtigen Prompt erweitern und so weiter.

Hinweis: Die in den Ordner vorhandenen Plugins/Tools sind sehr einfach gehalten und sollen nur die Funktionsweise bei LoKI zeigen. Zum Beispiel ist im Local-Ordner das Tool Calc nur sehr einfach. Es eignet sich aber hervorragend für eine Erweiterung durch ein brauchbares LLM wenn man den obigen Tipp folgt.

3. Download

Der Download-Bereich ermöglicht das direkte Laden von Modellen aus HuggingFace. Sowohl GGUF-Modelle als auch vollständige HuggingFace-Repositories können geladen werden.

HF-Modelle werden nach dem Download automatisch in das GGUF-Format konvertiert, da LoKI ausschließlich GGUF-Modelle für die Inferenz nutzt. Beim HF Download wird im Model-Ordner ein Ordner mit dem Namen des Model’s erstellt der alle Dateien des HF Downloads bündelt. Dieser Ordner wird nach der Konvertierung wieder gelöscht (wegen Platzbedarf) und es steht dann das fertige Model im GGUF Format zur Verfügung.

Hinweis: Es sind nicht alle LLM’s die angezeigt werden aud ladbar. Einige LLM’s aus der Liste sind “gated” und damit gesperrt. Der Download wird dann abgebrochen. In Der Suchleiste können sie nach LLM’s ihrer Wahl suche.

Mit LoKI getestete Modelle sind: DeepSeek-R1-Distill-Llama-8B.gguf
Qwen3-8B-Q4_K_M.gguf (LoKI quantisiert)
Llama3.1-8b-f16 und Q4_K_M
gpt-oss-20b_f16 (gepackt- mehr dazu in der Quantisierungs-Doc)

Multimodal:

Gemma-3-12b
Qwen3.5-9b
Llava-1.6-mistral-7b(GGUF Download da es nicht konvertiert werden kann wegen anderer Model Struktur.)

4. Quantisierung

Unten im Download-Bereich ist der Bereich für die Quantisierung. Quantisierung reduziert die Speichergröße eines Modells, indem Gewichte mit geringerer numerischer Präzision gespeichert werden. Während ein Modell im F16-Format 16-Bit Floating-Point-Gewichte verwendet, nutzen Quantisierungsformate wie Q8, Q5 oder Q4 komprimierte Darstellungen.

Typische Vorteile der Quantisierung:
• geringerer RAM / VRAM Bedarf
• schnellere Inferenz
• kleinere Modell-Dateien

Typische Quantisierungsstufen:
F16 → höchste Genauigkeit, größter Speicherbedarf
Q8 → sehr gute Qualität bei moderater Speicherreduktion
Q5 → guter Kompromiss zwischen Geschwindigkeit und Qualität
Q4_K_M → sehr effizient für Consumer-Hardware

Einige moderne Modelle sind bereits intern optimiert (z. B. durch Tensor Packing oder Shared Tensors). In solchen Fällen kann eine erneute Quantisierung die Dateigröße sogar erhöhen da das packing zerstört wird. Ein Beispiel-Kandidat ist gpt-oss-20b_f16 das dadurch nur eine Größe von 12GB hat statt 30GB (geschätzt).

Wählen sie hier das LLM und die Quantisierungs-Stufe aus und starten sie die Quantisierung. Danach steht das LLM zum testen bereit.

5. Settings

Auf der Setting Seite können verschiedene Parameter von LoKI eingestellt werden. LoKI selbst kann auf **deutsch/englisch** oder **hell/dunkel** erscheinen. Dies ist zur Laufzeit einstellbar.

Speaker-Settings: Der Pfad zum Voice Verzeichnis welches die Sprachmodelle beinhaltet.
Emotions ID gilt nur für das Voice Model „Thomas“.

Download-Settings: HuggingFace Token: Man sollte sich auf der HF Web-Seite einen solchen Token generieren lassen. Dadurch gehen etwas mehr Downloads. Der Token wird verschlüsselt gespeichert.
Hier kann man auch auswählen ob nach dem Download und eventueller Konvertierung(HF Download) gleich eine Quantisierung erfolgen soll und

die Quantisierungs-Stufe die in diesem Fall benutzt werden soll.

Die Settings erlauben die Anpassung wichtiger Inferenz-Parameter des LLM.

n_ctx – Kontextgröße
Bestimmt wie viele Tokens gleichzeitig im Kontextfenster gehalten werden.
Größere Werte ermöglichen längere Gespräche, benötigen aber mehr RAM/VRAM.

n_batch – Batchgröße
Steuert wie viele Tokens gleichzeitig verarbeitet werden.
Höhere Werte erhöhen die Geschwindigkeit, benötigen aber mehr Speicher.

n_gpu_layers – GPU Layer
Bestimmt wie viele Transformer-Layer auf der GPU berechnet werden.
0 = CPU only
-1 = so viele Layer wie möglich auf der GPU

Temperatur
Kontrolliert die Kreativität der Antworten.
Niedrige Werte → präzise Antworten
Hohe Werte → kreativere Antworten

top_p und top_k
Diese Parameter steuern das Sampling der Token-Wahrscheinlichkeiten und beeinflussen die Vielfalt der generierten Antworten.

MCP aktivieren: Aktiviert den MCP-Modus und blendet auf der Haupseite das MCP Fenster ein.

MCP-Modus: Auswählbar ist stdio, local oder http. Http dann über den MCP-Endpunkt.

Hinweis: Die Einstellung für die KI sollten nicht bei laufender LLM geändert werden. Das hat dann keine Auswirkungen auf den laufenden Thread. Die Parameter werden beim laden der LLM benutzt.

Hinweis: Die KI Settings werden beim laden der LLM von LoKI dynamisch überwacht und eventuell, bei nicht passen, herabgesetzt. Dies wird nach dem laden des LLM dann angezeigt. Diese Herabstufung ist etwas konservativ(sicher) und man kann noch etwas nach oben gehen.

6. Tools

LoKI unterstützt MCP-Tools (Model Context Protocol). Diese Tools können lokale Programme, Web-APIs oder Automatisierungen in den Kontext des LLM integrieren.

Hier sind die Tools der einzelnen MCP Modi. Diese sind beispielhaft und streckenweise nicht ausgebaut. Dieses ist mit Absicht, damit man zu mindest die funktionsweise der Tools sieht und selber den Ausbau oder die Entwicklung neuer Tools angehen kann. Dazu mehr in der Editor-Doc.

Alle Tools verwenden das jsonrpc 2.0 Protokoll.

Im Ordner Plugins befinden sie die Ordner:

local: Hier befinden sich die Tools für den lokalen MCP-Modus. Diese sind beispielhaft und in Python geschrieben. Die Python-Tools benutzen auch das Python312-Enviroment aus dem AppImage so das keine System-Abhängigkeiten vorhanden sind. Hier sind auch Tools die auf den Ordner ModelUseFiles zugreifen und von der LLM genutzt werden kann.

stdio: Hier sind Tools die mehr System Zugriffe erlauben. Darunter sind auch Tools die den im AppImage integrierten Browser(ms-playwright) nutzen. Auch hier wird das integrierte Node-modules genutzt. Zu Beachten ist das alle Tools in diesem Ordner einen mcp-tool-loader benutzen der das scannen der Tools übernimmt. Bei eigenen Tools bitte an den schon vorhanden Tools orientieren um die Funktion zu gewährleisten.

http: Hier liegt ein beispielhafter funktionierender Server. Dieser Server wird automatisch von LoKI in diesem Modus gestartet.

Der jeweilige Modus wird über die Settings-Seite eingestellt. Die Vorwahl muss vor dem Start der LLM ausgewählt werden. Währen die LLM aktiv ist, ist keine Änderung des MCP-Modus möglich da die Plugins/Tools beim Start der LLM, je nach Modi, gescannt werden und der LLM dann im Kontext bekannt gegeben werden. Die jeweils gefundenen Plugins/Tools werden auf der Hauptseite auf der rechten Seite im Fenster „Geladene Plugins“ angezeigt.

7. System Monitor

Der System Monitor zeigt CPU-, RAM- und GPU-Auslastung in Echtzeit. Besonders wichtig ist die VRAM-Überwachung bei GPU-Inference.

Diese Seite kann man mit **F9** abdocken und auf einen anderen Monitor legen so das man während eines Chats das Verhalten des Systems bewerten kann.Mit **F9** ist er auch wieder Andockbar oder einfach über „schließen“.

Empfehlung:
Nach dem Laden eines Modells sollten mindestens 512 MB bis 1 GB VRAM Reserve vorhanden sein.

Erweiterungen ab Version 1.1.x

Online-Provider

LoKI unterstützt jetzt zusätzlich Online-Provider wie OpenAI GPT-5 und Claude Sonnet. Online-Modelle erscheinen direkt in der Modellliste neben lokalen GGUF-Modellen. Die API-Keys werden verschlüsselt gespeichert. Alle bestehenden MCP-Tools können auch mit Online-Providern genutzt werden.

Live-Streaming

Antworten von OpenAI und Claude werden tokenweise ausgegeben. Dadurch erscheinen Antworten deutlich schneller im Chatfenster.

Tool-Freigaben

LoKI zeigt alle erkannten MCP-Tools in einer modernen Kachel-Ansicht an. Im Online-Modus kann für jedes Tool individuell festgelegt werden, ob ein Modell dieses Tool verwenden darf.

Channels

LoKI unterstützt externe Kommunikations-Channels wie WhatsApp und Telegram. Chats aus diesen Channels erscheinen automatisch in der Memory-Liste.

WhatsApp-Modi

Der WhatsApp-Connector unterstützt sowohl den Betrieb als zusätzliches Gerät (Linked Device) als auch einen eigenständigen WhatsApp-Account mit eigener SIM-Karte. Zusätzlich steht eine Funktion „Neu koppeln“ zur Verfügung.

Agent-Modus

LoKI kann komplexe Aufgaben autonom in mehreren Schritten ausführen. Das Modell entscheidet selbstständig, welche Tools verwendet werden müssen.