

IndustrialBrainMCP Pro

Installations- und Benutzerhandbuch für das native macOS-Speichersystem

1. Einleitung: Sinn, Zweck & Nutzen

IndustrialBrainMCP Pro ist ein kompromisslos robustes, für den industriellen und professionellen Dauereinsatz konzipiertes Gedächtnissystem für Large Language Models (LLMs). Es schließt die Lücke zwischen experimentellen Prototypen und ausfallsicherer Systemarchitektur.

- **Sinn:** Dauerhafte, fehlerresistente Speicherung von Kontext, Wissen und Interaktionshistorien von KI-Modellen ohne Abhängigkeit von Container-Virtualisierung (wie Docker).
- **Zweck:** Bereitstellung einer nativen System-Architektur unter macOS, die auf standardisierte Protokolle (Model Context Protocol, MCP) setzt und Transaktionssicherheit (ACID) garantiert.
- **Nutzen:** Absolute Datenintegrität durch SQLite im WAL-Modus, nahtlose Hintergrund-Synchronisation zu Neo4j und Markdown via `launchd`, sowie blitzschnelle Integration in Frontends wie LM Studio und Ollama.

2. Kernarchitektur & Technische Merkmale

- **ACID-konforme SQLite-Storage:** Transaktionssichere Speicherung aller Ereignisse im Write-Ahead Logging (WAL) Modus verhindert Datenverlust bei Abstürzen.
- **Nativer Launchd LaunchAgent:** Der Hintergrund-Archivar läuft als permanenter macOS-Dienst (`com.industrialbrain.archivist`) mit automatischem Neustart bei Abstürzen.
- **Pydantic & Instructor:** Strikte Datentypisierung und strukturierte LLM-Ausgaben garantieren fehlerfreie Datenstrukturen im Vektor- und Graphen-Speicher.
- **Dualer Sync (Neo4j & Markdown):** Automatisierte Spiegelung der Memory-Inhalte in eine Neo4j-Graphdatenbank und lesbare Markdown-Dateien im Dateisystem.

3. Installations- und Setup-Anleitung

Schritt 1: Verzeichnisstruktur initialisieren

Führe das Setup-Skript im Zielordner (z. B. `/Users/DEIN_BENUTZERNAME/IndustrialBrainMCP_pro`) aus:

```
chmod +x setup_mac.sh
./setup_mac.sh
```

Schritt 2: Launchd Hintergrunddienst einrichten

Erstelle die Datei `~/Library/LaunchAgents/com.industrialbrain.archivist.plist` mit folgendem Inhalt (passe `DEIN_BENUTZERNAME` an):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
```

```
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.industrialbrain.archivist</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/bin/python3</string>
    <string>/Users/DEIN_BENUTZERNAME/IndustrialBrainMCP_pro/background_archivist_pro.py</
string>
  </array>
  <key>WorkingDirectory</key>
  <string>/Users/DEIN_BENUTZERNAME/IndustrialBrainMCP_pro</string>
  <key>RunAtLoad</key>
  <true/>
  <key>KeepAlive</key>
  <true/>
  <key>StandardOutPath</key>
  <string>/Users/DEIN_BENUTZERNAME/IndustrialBrainMCP_pro/logs/launchd_stdout.log</string>
  <key>StandardErrorPath</key>
  <string>/Users/DEIN_BENUTZERNAME/IndustrialBrainMCP_pro/logs/launchd_stderr.log</string>
</dict>
</plist>
```

Aktiviere und starte den Dienst mit folgendem Befehl im Terminal:

```
launchctl load ~/Library/LaunchAgents/com.industrialbrain.archivist.plist
```

4. Integration in LM Studio (mcp.json)

Ergänze deine MCP-Konfiguration in LM Studio (unter macOS zu finden unter `~/Library/Preferences/com.linuxlabstudios.lmstudio.plist` oder direkt in den Einstellungen) um folgenden Server-Eintrag:

```
{
  "mcpServers": {
    "industrial-brain-pro": {
      "command": "/usr/bin/python3",
      "args": [
        "/Users/DEIN_BENUTZERNAME/IndustrialBrainMCP_pro/mcp_server_pro.py"
      ]
    }
  }
}
```

5. Befehls- und Statusübersicht (macOS)

Aktion	Befehl (macOS)	Beschreibung
Dienst starten	<code>launchctl load ~/Library/LaunchAgents/com.industrialbrain.archivist.plist</code>	Lädt und startet den Hintergrund-Archivar.
Dienst stoppen	<code>launchctl unload ~/Library/LaunchAgents/com.industrialbrain.archivist.plist</code>	Stoppt den laufenden Hintergrunddienst.

Aktion	Befehl (macOS)	Beschreibung
Status prüfen	<code>launchctl list grep com.industrialbrain.archivist</code>	Prüft, ob der Prozess läuft (zeigt PID und Return-Code).
Live-Logs anzeigen	<code>tail -f logs/archivist_pro.log</code>	Zeigt fortlaufend die Logausgaben des Archivars.
Fehler-Logs anzeigen	<code>tail -f logs/launchd_stderr.log</code>	Zeigt Systemfehler des <code>launchd</code> -Prozesses.

Hinweis zu Pfaden: Ersetze in allen Konfigurationen `DEIN_BENUTZERNAME` durch deinen tatsächlichen macOS-Benutzernamen (abfragbar per `whoami` im Terminal).