Hoofdstuk 5: Lokale AI - Praktijk

Dit hoofdstuk transformeert je van een beginnende gebruiker transformeert tot een beheerder van je eigen, soevereine AI-ecosysteem. We laten de theoretische concepten achter ons en duiken met beide voeten in de praktijk. Stap voor stap, klik voor klik, bouwen we samen een volledig functionele, lokale AI-omgeving die volledig onder jouw controle staat.

In de komende pagina's leer je niet alleen hoe je grote taalmodellen (LLMs) op je eigen computer installeert, maar ook waarom bepaalde keuzes worden gemaakt. We ontrafelen de mysteries van quantization, verkennen de architectuur van het Model Context Protocol (MCP), en zetten een veilige en krachtige RAG-pijplijn op om je AI te laten praten met je eigen documenten. We gaan verder dan de basis en behandelen geavanceerde configuraties, troubleshooting voor veelvoorkomende problemen, en de cruciale beveiligingsaspecten die je moet kennen om met vertrouwen te kunnen werken.

Dit hoofdstuk is jouw routekaart naar digitale soevereiniteit in het tijdperk van AI. Na het voltooien van deze gids ben je niet langer een passieve consument van cloud-diensten, maar een actieve bouwer en beheerder van je eigen AI-assistent. Je zult de vrijheid ervaren van het werken zonder internetverbinding, zonder zorgen over privacy of oplopende abonnementskosten. Laten we de controle terugnemen en beginnen met bouwen.

5.1 Wat Heb Je Nodig? Een Complete Checklist

Voordat we beginnen met het bouwen van je lokale AI-omgeving, is het belangrijk om een duidelijk overzicht te hebben van alle benodigde componenten. Deze sectie fungeert als je pre-flight checklist: controleer elk punt voordat je aan de reis begint. Dit voorkomt frustratie en zorgt ervoor dat je installatie soepel verloopt.

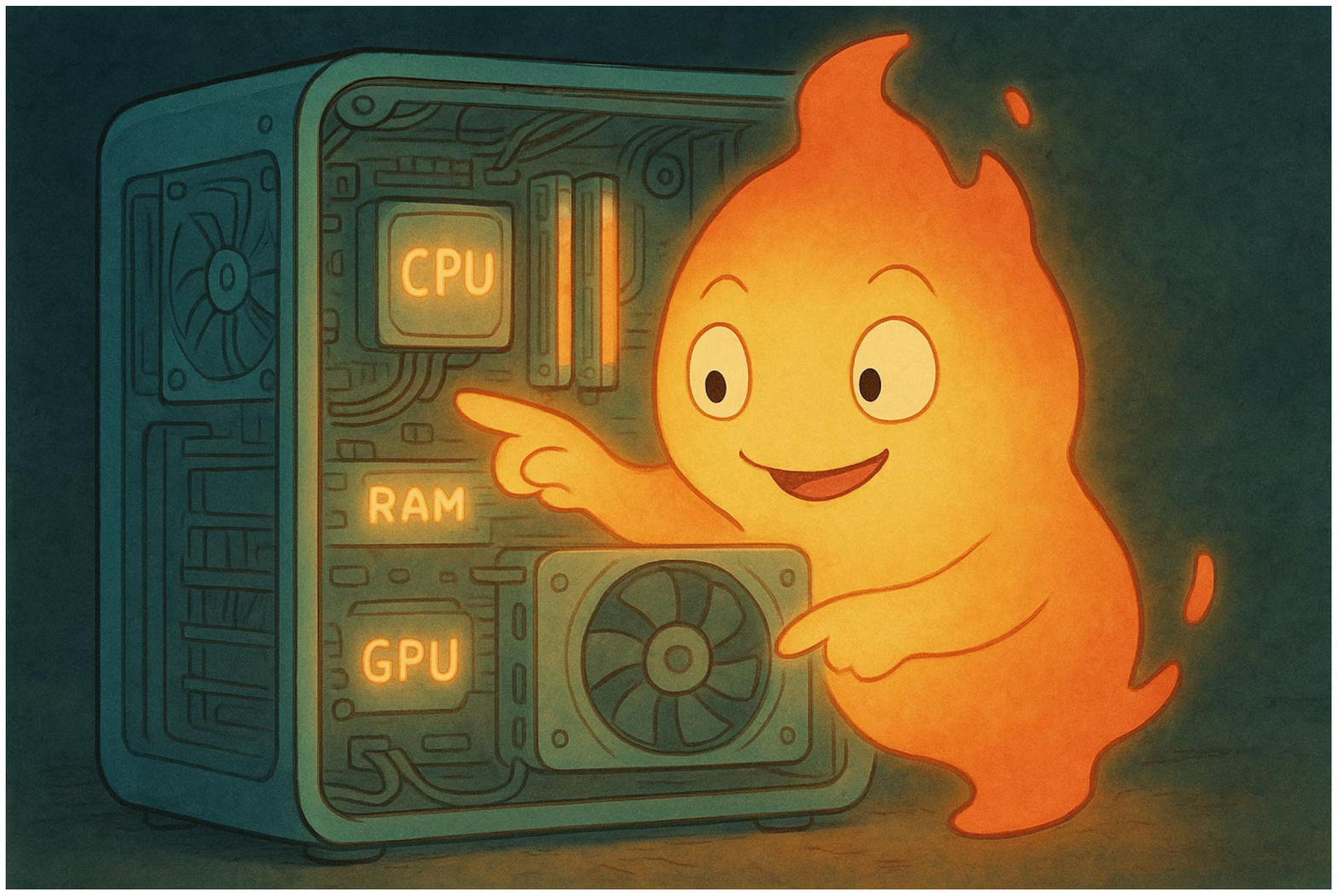
De Krachtbron: Hardware Vereisten

De kracht van je hardware bepaalt welke modellen je kunt draaien en hoe snel ze zullen reageren. Hier is een gedetailleerd overzicht:

Minimale Vereisten (Laptop Voor Kleine Taalmodellen)

Dit is het absolute minimum om te kunnen experimenteren met kleine AI-modellen (3B-7B parameters). Deze cursus is gebouwd voor de minimale vereisten:

- Processor (CPU): Een moderne multi-core processor (minimaal 4 cores)
  - Intel Core i5 (8e generatie of nieuwer) of AMD Ryzen 5 (2000-serie of nieuwer)
  - Moet de AVX2 instructieset ondersteunen (bijna alle CPU's vanaf 2015 hebben dit)
- Werkgeheugen (RAM): Minimaal 8GB, maar 16GB is sterk aanbevolen
  - 8GB is voldoende voor modellen tot 3B parameters met lage quantization
  - 16GB stelt je in staat om 7B modellen comfortabel te draaien
- Grafische Kaart (GPU): Optioneel, maar sterk aanbevolen voor snelheid
  - NVIDIA GPU met minimaal 4GB VRAM (bv. GTX 1650, RTX 3050)
  - AMD GPU met minimaal 4GB VRAM (bv. RX 5500 XT)
  - Apple Silicon Macs (M1/M2/M3/M4) hebben geïntegreerde GPU's die uitstekend werken
- Opslagruimte: Minimaal 50GB vrije schijfruimte
  - Modellen variëren van 2GB tot 50GB+ per model
  - Een SSD (Solid State Drive) is sterk aanbevolen voor snelle laadtijden



Aanbevolen Configuratie (Desktop Voor Middelgrote Taalmodellen)

Voor een comfortabele ervaring met 7B-13B parameter modellen:

- Processor (CPU): Intel Core i7 of AMD Ryzen 7 (recente generatie)
- Werkgeheugen (RAM): 32GB
- Grafische Kaart (GPU):
  - NVIDIA RTX 3060 (12GB VRAM) of hoger
  - AMD RX 6700 XT (12GB VRAM) of hoger
  - Apple Silicon M2 Pro/Max of M3 Pro/Max
- Opslagruimte: 100GB+ vrije SSD-ruimte

High-End Configuratie (Workstation / Server Voor Grote Taalmodellen)

Voor het draaien van 30B-70B parameter modellen:

- Processor (CPU): Intel Core i9 of AMD Ryzen 9
- Werkgeheugen (RAM): 64GB of meer
- Grafische Kaart (GPU):
  - NVIDIA RTX 4090 (24GB VRAM)
  - NVIDIA RTX A6000 (48GB VRAM)
  - Meerdere GPU's in SLI/NVLink configuratie
  - Apple Silicon M3 Max/Ultra met 64GB+ unified memory
- Opslagruimte: 500GB+ vrije NVMe SSD-ruimte

Platform-Specifieke Vereisten

Voor macOS Gebruikers:

- Alleen Apple Silicon: LM Studio ondersteunt geen Intel-based Macs
- Minimale OS Versie: macOS 13.4 (Ventura) of nieuwer
- Voor MLX Modellen: macOS 14.0 (Sonoma) of nieuwer vereist
- Unified Memory: Apple Silicon Macs gebruiken unified memory (gedeeld tussen CPU en GPU), dus meer RAM = betere GPU-prestaties

Voor Windows Gebruikers:

- Besturingssysteem: Windows 10 (64-bit) of Windows 11
- Architectuur: x64 (standaard) of ARM64 (voor Snapdragon X Elite laptops)
- NVIDIA GPU: Installeer de nieuwste NVIDIA GeForce Game Ready Drivers of Studio Drivers
- AMD GPU: Installeer de nieuwste AMD Adrenalin Drivers

Voor Linux Gebruikers:

- Distributie: Ubuntu 20.04 LTS of nieuwer (andere distributies kunnen werken maar zijn minder getest)
- Architectuur: Alleen x64 (aarch64/ARM64 wordt nog niet ondersteund)
- Display Server: X11 of Wayland
- NVIDIA GPU: Installeer de proprietary NVIDIA drivers (niet nouveau)

De Software: Je Digitale Werkplaats

Nu we de hardware hebben besproken, is het tijd voor het gereedschap zelf. Alle software die we in dit hoofdstuk gebruiken is volledig gratis en open-source. Dit is een kernprincipe van digitale soevereiniteit: de tools zijn van en voor iedereen.

LM Studio

Functie: De hoofdapplicatie voor het downloaden en draaien van lokale AI-modellen. Dit is ons commandocentrum.

Waarom het essentieel is: Het maakt het complexe proces van taalmodelbeheer toegankelijk voor iedereen.

Node.js

Functie: Een runtime-omgeving voor het uitvoeren van JavaScript-code.

Waarom het essentieel is: Vormt de ruggengraat voor de MCP-servers, die als brug fungeren tussen je AI en andere applicaties.

AnythingLLM Desktop

Functie: Een tool voor het opzetten van een RAG-systeem (chatten met documenten).

Waarom het essentieel is: Hiermee geven we onze AI een 'aktetas' met specifieke kennis, gebaseerd op onze eigen bestanden.

Het installeren van deze software vereist dat je administrator-rechten op je computer hebt. Dit is vergelijkbaar met het hebben van de sleutels van de werkplaats; je hebt toestemming nodig om nieuwe machines te installeren. Tijdens de installatie zal je computer je mogelijk om bevestiging vragen via een pop-up. Dit is een standaard veiligheidsmaatregel. Geef met een gerust hart toestemming, want we installeren alleen betrouwbare en wijdverbreide software.

Met je hardware-checklist voltooid en je software-gereedschapskist gevuld, ben je klaar voor de volgende stap. We gaan beginnen met het installeren van het hart van onze operatie: LM Studio.

5.2 Je Werkplaats Inrichten: LM Studio Installeren

Nu je gereedschapskist is gevuld, is het tijd om de werkbank te installeren: LM Studio. Zie dit programma als het hart van je lokale AI-operatie. Het is een prachtig ontworpen applicatie die de vaak intimiderende wereld van taalmodellen toegankelijk maakt. Het neemt de technische complexiteit weg en biedt een visuele, intuïtieve interface om modellen te ontdekken, te downloaden en mee te praten. Laten we samen deze cruciale eerste stap zetten.

De Deur Openen: Installatie van LM Studio

Het installeren van LM Studio is net zo eenvoudig als het installeren van elke andere applicatie. Het is een kwestie van een paar klikken.

01

Navigeer naar de Bron

Open je webbrowser en ga naar de officiële website: https://lmstudio.ai.

02

Download de Juiste Versie

De website herkent automatisch je besturingssysteem (Windows, macOS of Linux) en presenteert je de juiste downloadknop. Klik erop en het downloaden begint.

03

Installeer de Applicatie

Zodra de download voltooid is, voer je het installatiebestand uit. Op Windows is dit een .exe-bestand dat je door een simpele wizard leidt. Op macOS open je het .dmg-bestand en sleep je het LM Studio-icoon naar je Applications-map. Op Linux maak je het .AppImage-bestand uitvoerbaar en start je het met een dubbelklik.

En dat is het! Je hebt zojuist de deur geopend naar je eigen AI-werkplaats. Laten we nu naar binnen stappen en de ruimte verkennen.

Je Eerste Model Kiezen en Downloaden

Bij het openen van LM Studio word je begroet door een schone interface. Aan de linkerkant zie je een aantal iconen. De belangrijkste voor nu is het vergrootglas (🔍), het 'Discover'-tabblad. Dit is je poort naar Hugging Face, de grootste bibliotheek van open-source AI-modellen ter wereld. Laten we ons eerste model downloaden. We beginnen met een compact maar krachtig model: qwen3 4B 2507. Dit model is een uitstekende keuze voor beginners.

Zoek het Model

Klik op het 🔍-icoon en typ qwen3 4B 2507 in de zoekbalk.

Kies een Versie

Je krijgt een lijst met resultaten. Selecteer Qwen3 4B 2507. Aan de rechterkant verschijnen de details. Onder "Download Options" kan je via "Show all options" meerdere varianten vinden. Geen paniek, we ontcijferen dit samen.

De Kunst van het Comprimeren: Quantization Begrijpen

De verschillende bestanden die je ziet, representeren hetzelfde model, maar in verschillende groottes. Dit wordt bereikt door een proces genaamd quantization. Zie het als het comprimeren van een groot fotobestand. Het origineel is perfect, maar erg groot. Een gecomprimeerde versie is veel kleiner en daardoor sneller te laden, met slechts een heel klein, vaak onmerkbaar verlies in kwaliteit.

In de wereld van AI-modellen wordt de grootte en kwaliteit aangeduid met codes zoals `Q4_K_M` of `Q8_0`. Een hoger getal betekent een hogere kwaliteit en een groter bestand. Voor beginners is Q4_K_M de perfecte balans tussen kwaliteit en prestaties. Het is klein genoeg om soepel te draaien op de meeste moderne computers, maar krachtig genoeg voor indrukwekkende resultaten.

LM Studio maakt de keuze nog makkelijker. Naast elk bestand zie je een label dat aangeeft hoe goed het op jouw computer zal draaien:

- Groen: Perfect! Het model past volledig op je grafische kaart (GPU) voor maximale snelheid.
- Oranje: Mogelijk. Het model is wat groter en zal deels op je tragere werkgeheugen (RAM) moeten leunen.
- Rood: Waarschijnlijk te groot. Vermijd deze, tenzij je veel geduld hebt.

Actie: Zoek in de lijst naar een Q4_K_M versie met een groen of oranje label en klik op de Download knop. Je ziet de voortgang onderaan het scherm.

Het Eerste Gesprek: Je AI Komt tot Leven

Zodra de download voltooid is, is het moment daar. We gaan praten met onze eigen AI.

1

Ga naar de Chatruimte

Klik op het tekstballon-icoon ( ) aan de linkerkant.

2

Laad het Model

Klik bovenaan op de knop Select a model to load en kies het Qwen-model dat je zojuist hebt gedownload.

3

Wacht even

Accepteer de standaard settings en klik op "Load Model". Het model wordt nu in het geheugen van je computer geladen. Dit kan een minuutje duren.

Zodra het model geladen is, verschijnt er een chatvenster. De werkplaats is operationeel. Typ je eerste vraag, bijvoorbeeld:

> Hallo! Wie ben jij en wat kun je doen?

Het antwoord dat verschijnt, wordt niet gegenereerd op een server ver weg, maar door de machine die voor je staat. Gefeliciteerd, je hebt zojuist je eigen, volledig lokale en soevereine AI tot leven gewekt!

Om de verschillen tussen CPU en GPU processing te ondervinden, kan je (indien mogelijk) spelen met de setting GPU Offload. Deze kan je vinden door naar het tabblad "My Models" te gaan. Daarna klik je op het radartje achter het gekozen model. Als je GPU correct gedetecteerd wordt, kan je een deel van het model, of het volledige model in de GPU laden. Zorg ervoor dat GPU Offload is aangevinkt en de slider op Max staat. Dit zorgt ervoor dat de kracht van je grafische kaart volledig wordt benut, wat resulteert in significant snellere antwoorden. Bij een minder performante GPU is partieel offloaden mogelijk, of zelfs werken zonder GPU, waarbij dan de value op 0 zal staan. Alle processing gebeurt dan op de CPU. Je kan het verschil duidelijk zien tijdens het chatten met een model, aan de hand van de snelheid waarmee de antwoorden verschijnen.

5.3 Je AI een Geheugen Geven: De Magie van RAG

Stel je een briljante, ervaren vakman voor. Hij beschikt over een enorme algemene kennis en kan de meest complexe problemen oplossen. Maar hij kent de specifieke blauwdrukken van jouw project niet, of de inhoud van jouw persoonlijke notitieboeken. Wat als je hem, precies op het moment dat je een vraag stelt, een perfect georganiseerde aktetas met exact de juiste documenten zou kunnen geven? Plotseling is hij niet alleen een algemene expert, maar een expert in jouw wereld. Dit is, in essentie, wat Retrieval Augmented Generation (RAG) doet. Het is een elegante en krachtige techniek die de algemene redeneerkracht van een taalmodel combineert met specifieke, externe informatie. RAG lost het meest fundamentele probleem van taalmodellen op: ze weten alleen wat ze tijdens hun training hebben geleerd. Ze hebben geen toegang tot actuele gebeurtenissen, jouw bedrijfsdata, of de inhoud van die ene PDF op je bureaublad.

> "RAG staat voor Retrieval Augmented Generation. En het betekent simpelweg dat je in staat zult zijn om alle informatie uit je lokale data op te vragen. We nemen dus documenten, importeren ze in AnythingLLM, dit zet ze om in een vectordatabase en alles wordt voor je gedaan."

Hoe Werkt RAG? Een Kijkje onder de Motorkap

Het proces klinkt misschien complex, maar het is te herleiden tot drie logische stappen, een dans tussen data voorbereiden, data opvragen en antwoorden genereren.

De Drie Stappen van RAG

1

Stap 1: Indexering (De Bibliotheek Inrichten)

Dit is de voorbereidende fase, die eenmalig plaatsvindt wanneer je jouw documenten toevoegt aan een RAG-systeem zoals AnythingLLM.

1. Versnipperen (Chunking): Je documenten – of het nu PDF's, Word-bestanden of notities zijn – worden opgedeeld in kleine, behapbare stukken tekst, die we 'chunks' noemen.
2. Betekenis Vatten (Embedding): Elke 'chunk' wordt vervolgens door een speciaal model gehaald dat de tekst omzet in een reeks getallen, een 'vector'. Zie deze vector als een coördinaat in een gigantische, onzichtbare bibliotheek van betekenis. Tekstfragmenten die over vergelijkbare onderwerpen gaan, krijgen coördinaten die dicht bij elkaar liggen.
3. Opslaan (Storage): Al deze vectoren worden opgeslagen in een gespecialiseerde, razendsnelle database, een 'vector database'.

2

Stap 2: Ophalen (De Juiste Pagina Vinden)

Dit gebeurt bliksemsnel, elke keer als jij een vraag stelt.

1. Vraag Vertalen: Jouw vraag wordt door hetzelfde model gehaald en ook omgezet in een vector-coördinaat.
2. Semantisch Zoeken: Het systeem zoekt nu in de vector database niet naar trefwoorden, maar naar de 'chunks' waarvan de coördinaten het dichtst bij de coördinaat van jouw vraag liggen. Het zoekt naar betekenis, niet naar letterlijke tekst.

3

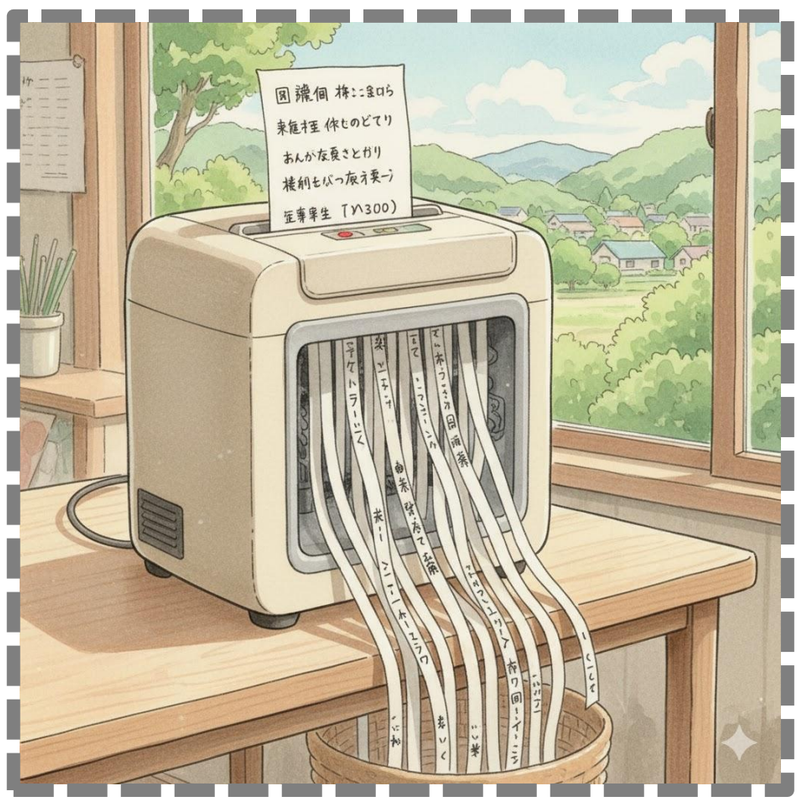
Stap 3: Verrijkte Generatie (Het Antwoord Formuleren)

Nu komt het taalmodel, dat we in LM Studio hebben geladen, in het spel.

1. Context Bieden: Je oorspronkelijke vraag wordt gecombineerd met de 4 (instelbaar) meest relevante 'chunks' die in de vorige stap zijn gevonden. Dit vormt een nieuwe, verrijkte prompt.
2. Genereren met Grond: Deze verrijkte prompt wordt naar het taalmodel gestuurd. Het model heeft nu niet alleen jouw vraag, maar ook de relevante context uit jouw documenten. Het hoeft niet meer te gokken of te 'hallucineren', maar kan een antwoord formuleren dat feitelijk onderbouwd is door de informatie die jij hebt aangeleverd.

Het resultaat is een AI die niet alleen slim is, maar ook wijs over jouw specifieke onderwerpen. Het is de perfecte samenwerking tussen algemene intelligentie en domeinspecifieke kennis.

Architectuur van een RAG-Systeem



RAG - SYSTEEM

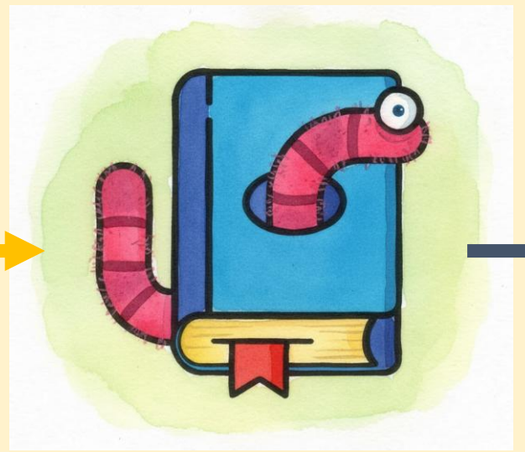
1. Initiële Vraag komt binnen over chat

2. Vraag Vertalen

Slimme Opzoeker

3. Semantisch zoeken

4. Context bieden



5. Vraag wordt naar LLM doogestuurd met context

6. Antwoord generatie

7. LLM stuurt het gegronde (geïnformeerde) antwoord naar de gebruiker

5.4 Praktijk: Je Kennisbank Bouwen met AnythingLLM

Nu we de elegante theorie van RAG begrijpen, is het tijd om onze handen vuil te maken en deze kennis om te zetten in een tastbaar resultaat. In deze sectie bouwen we een volledig lokaal en privé RAG-systeem met AnythingLLM. Dit is een prachtig stuk gereedschap: een gebruiksvriendelijke, open-source applicatie die speciaal is ontworpen om een brug te slaan tussen jouw documenten en een taalmodel. We gaan het koppelen aan de lokale server die we zojuist in LM Studio hebben leren opzetten.

Stap 1: De Brug Opzetten in LM Studio

Voordat AnythingLLM met ons model kan praten, moeten we in LM Studio een 'lokale server' starten. Zie dit als het openen van een communicatielijn. De server fungeert als een brug, die ons model toegankelijk maakt voor andere applicaties via een gestandaardiseerde taal (een OpenAI-compatibele API).

1. Open LM Studio en ga naar het Developer tabblad in de linkerkolom.
2. Selecteer een Model: Kies bovenaan het model dat je wilt gebruiken. Een model als qwen/qwen3-4b-2507 is een uitstekende keuze voor RAG-taken, dankzij hun grote 'contextvenster' en vermogen om instructies goed te volgen. Let hier op de "context length" setting. Dat bepaalt de grootte van het werkgeheugen van de LLM (maar bepaalt dus ook extra resources gebruik op je systeem).
3. Load Model: Klik op de grote knop Load Model. Nu kan je in de rechterkolom op "Info" klikken en alzo de details van je server terugvinden. Zet de slider op “running” zodat het model beschikbaar is voor Andere applicaties. Ook is het interessant om eens om "Settings" te klikken, vlak naast de slider waar staat "Status: Running".

Stap 2: AnythingLLM Installeren en Configureren

AnythingLLM is onze tweede werkplaats, speciaal voor het beheren van kennis. De installatie is eenvoudig.

1. Download de Software: Ga naar de officiële website https://anythingllm.com/ en download de Desktop-versie voor jouw besturingssysteem.
2. Installeer de Applicatie: Volg de standaard installatieprocedure, vergelijkbaar met hoe we LM Studio hebben geïnstalleerd.

Bij de eerste keer opstarten, word je begroet door een vriendelijke setup-wizard. Dit is de belangrijkste stap, waar we de verbinding met onze LM Studio-brug leggen.

1. LLM Provider: Kies LM Studio uit de lijst.
2. Model Configuratie: De standaardinstellingen zijn hier perfect. De Model Base Path (http://localhost:1234/v1) is het adres van onze brug. Max Tokens gaat over de context length. Je kan deze beide ongewijzigd laten.

   Mocht je niet voldoende kracht op je PC hebben om een lokale LLM te draaien, is dit het moment om een API key van bijvoorbeeld een publiek model (zoals OpenAI) te gebruiken.

3. Embedding Engine & Vector Database: Laat deze op de standaardwaarden (AnythingLLM Embedder en LanceDB). Dit zijn de ingebouwde, efficiënte systemen voor het indexeren en opslaan van je documenten.

Klik door de laatste stappen en de configuratie is voltooid. De twee werkplaatsen zijn nu met elkaar verbonden.

Je Eerste Kennisbank in Actie

Stap 3: Je Eerste Kennisbank Creëren

Een 'workspace' in AnythingLLM is een geïsoleerde omgeving voor een specifiek project. Laten we er een maken.

1. Creëer een Workspace

Klik op New Workspace en geef het een duidelijke naam, bijvoorbeeld "Syntra".

2. Upload Documenten

Klik op “Embed a document". Sleep simpelweg de bestanden die je wilt gebruiken (PDF's, Word-documenten, tekstbestanden) naar het upload-gebied.
Selecteer het bestand en klik “Move to workspace”.
Klik op “Save and Embed”.
AnythingLLM begint onmiddellijk met het 'embedding' proces: het versnipperen, analyseren en indexeren van de inhoud in zijn vector database. Dit kan even duren, afhankelijk van de hoeveelheid informatie.

3. Chatten met je Eigen Kennis

Zodra de documenten zijn verwerkt, is het magische moment aangebroken. Je kunt nu een gesprek voeren met je eigen bestanden.
Klik in de linkerkolom “Syntra” open.
Stel een specifieke vraag die alleen beantwoord kan worden met de kennis uit jouw documenten. Dit is de ultieme test.

Het antwoord dat je krijgt, is niet zomaar een antwoord; het is een synthese, onderbouwd met bronvermeldingen. Je ziet precies welke fragmenten uit welke documenten zijn gebruikt om het antwoord te formuleren. Dit is de essentie van betrouwbare, transparante en gepersonaliseerde AI.

Je hebt nu een krachtig, volledig lokaal en privé RAG-systeem gebouwd. Elke verzameling documenten, hoe complex ook, is nu een interactieve kennisbank waarmee je in natuurlijke taal kunt praten.

5.5 Je AI Handen Geven: Het Model Context Protocol (MCP)

We hebben onze AI een brein gegeven (LM Studio), een geheugen (AnythingLLM), en nu is het tijd om het handen te geven. Wat als je AI niet alleen kon praten, maar ook kon doen? Dit is precies waar het Model Context Protocol (MCP) voor is ontworpen. MCP is een ingenieuze open-source standaard, ontwikkeld door Anthropic, die fungeert als een universele vertaler tussen een taalmodel en externe tools of systemen. Het stelt de AI in staat om met de wereld buiten zijn chatvenster te interageren. De tool stuurt een beschrijving van zijn mogelijkheden naar het model. Het model analyseert deze informatie en stuurt een commando terug, zoals "zoek informatie op in de CRM" of "stuur een mail naar een emailadres". Dit creëert een krachtige, interactieve lus waarbij de AI een actieve deelnemer wordt in je digitale wereld, in plaats van een passieve gesprekspartner.

De Architectuur van MCP: Een Client-Server Model

De kern van MCP is een klassieke client-server architectuur die ontworpen is om lokaal en veilig te functioneren:

- MCP Host (De AI-Applicatie): Dit is de AI-omgeving waar het taalmodel draait. Voorbeelden zijn in ons geval LM Studio, maar andere toepassingen zoals Claude Desktop, of Cursor kunnen natuurlijk ook. De host coördineert de communicatie.
- MCP Client: Voor elke tool of applicatie waarmee de host wil praten, creëert het een MCP Client. Deze client is verantwoordelijk voor de één-op-één communicatie met een specifieke MCP Server.
- MCP Server: Dit is een klein programma dat naast een specifieke applicatie draait (bv. je webbrowser, je agenda, je terminal). Het fungeert als een tolk:
    - Het observeert de status van de applicatie (bv. welke webpagina is open, welke bestanden staan in een map) en rapporteert dit als "context" aan de AI.
    - Het accepteert commando's van de AI (bv. "klik op die knop", "open dit bestand") en vertaalt deze naar concrete acties in de applicatie.

De Primitives van MCP: De Bouwstenen van Interactie

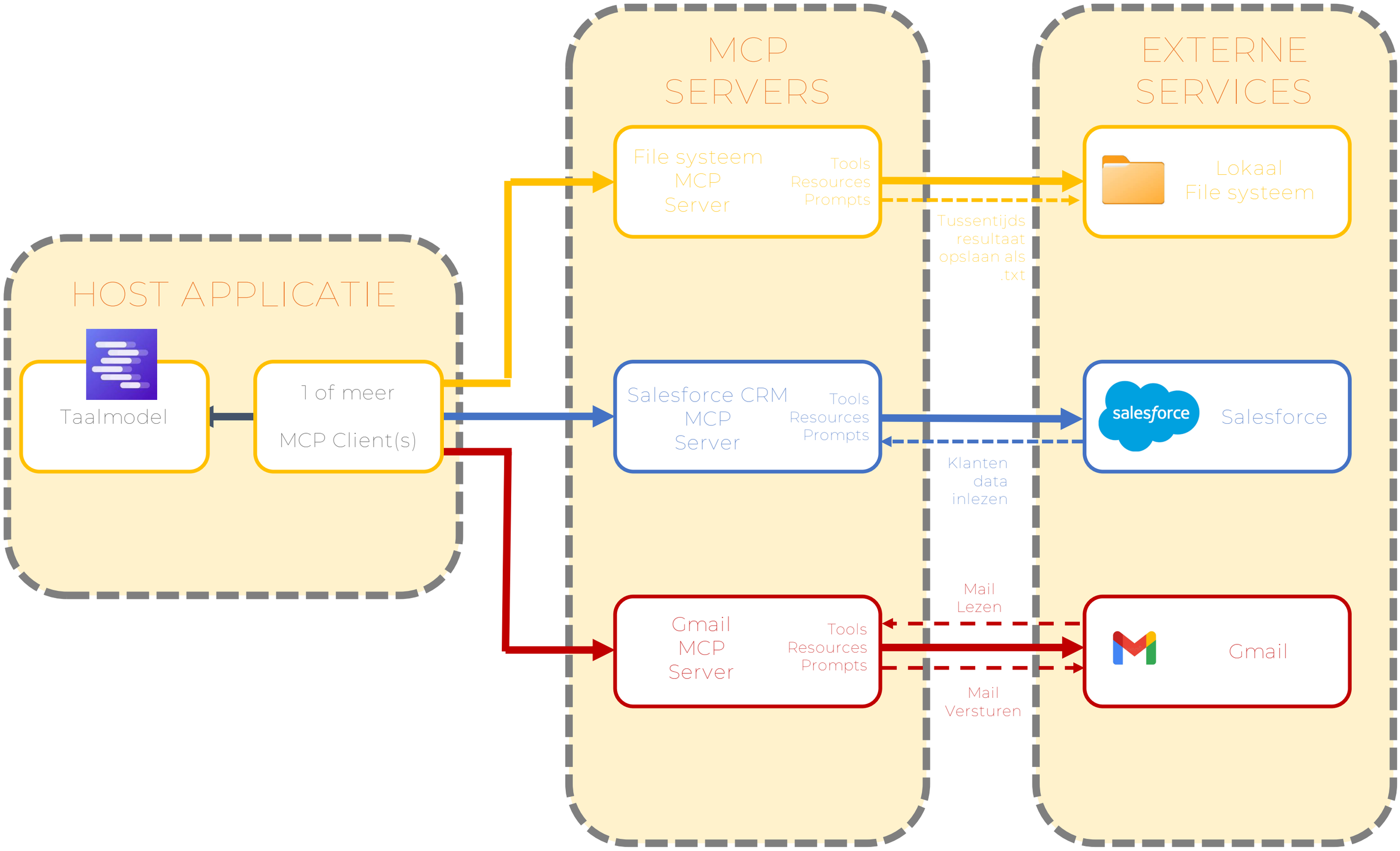
MCP definieert een aantal standaard "primitives" of bouwstenen die servers kunnen aanbieden aan de AI. De belangrijkste drie zijn:

- Tools: Dit zijn uitvoerbare functies die de AI kan aanroepen. Denk aan browser.navigate(url), filesystem.readFile(path), of calendar.createEvent(...). De AI kan een lijst van beschikbare tools opvragen (tools/list) en ze vervolgens aanroepen (tools/call).
- Resources: Dit zijn databronnen die contextuele informatie leveren. Voorbeelden zijn de inhoud van een webpagina, de structuur van een database, of de lijst van bestanden in een map. De AI kan ontdekken welke resources beschikbaar zijn (resources/list) en hun inhoud opvragen (resources/get).
- Prompts: Dit zijn herbruikbare templates die de AI helpen om beter te interageren. Een server kan bijvoorbeeld een specifieke system prompt aanbieden die het model instrueert hoe het de beschikbare tools het beste kan gebruiken.

Door deze gestandaardiseerde bouwstenen kan elke AI die MCP spreekt, in principe met elke applicatie praten die een MCP-server heeft, zonder dat ze elkaar van tevoren hoeven te kennen.

Om het principe duidelijk te maken, gaan we een zeer simpel MCP voorbeeld opzetten in de volgende paragrafen, maar denk eraan, er zijn veel meer mogelijkheden! Bijvoorbeeld: Salesforce integratie, integreren met office applicaties, databases, ...

Architectuur van een Model Context Protocol-Systeem



Bestanden Beheren: Filesystem MCP in de Praktijk

Een gemakkelijle en directe en praktische toepassing van MCP is het verbinden van je AI met het bestandssysteem van je computer. Hiermee geven we onze lokale AI de mogelijkheid om simpele bestanden (txt, csv) te lezen, te schrijven en te beheren. Het opzetten vereist een paar technische stappen, maar door ze rustig te volgen, geven we onze AI de mogelijkheid om tastbare resultaten te creëren.

1

Node.JS Installeren

- Ga naar de Website: Open je browser en navigeer naar https://nodejs.org/.
- Download Node.js.
- Installeer Node.js:
  - a. Op Windows: Voer de .msi-installer uit. Volg de wizard en accepteer de standaardinstellingen. Vink de optie aan om "Automatically install the necessary tools" als deze wordt aangeboden.
    b. Op macOS: Open het .pkg-bestand en volg de installatiewizard.
- Verifieer de Installatie: Open een terminal (Command Prompt op Windows, Terminal op macOS/Linux) en typ node --version. Als je een versienummer (bv. v20.11.0 of nieuwer) ziet, is de installatie geslaagd.
- Rebooten aub!

2

De MCP Server Configureren in LM Studio

Nu gaan we LM Studio vertellen over deze nieuwe 'tool' die onze AI kan gebruiken.

- Open LM Studio en open een chat.
- Klik op "Program" en dan "Install", gevolgd door "Edit MCP.json".
- Kopieer de volgende tekst integraal en overschrijf de oorsponkelijke tekst - de opgegeven folder mag je zelf kiezen maar hij moet al bestaan!:

```
{
 "mcpServers": {
  "filesystem": {
   "command": "npx",
   "args": [
    "@modelcontextprotocol/server-filesystem",
    "C:/TestSyntra"
   ]
  }
 }
}
```

7

De Verbinding Testen met een Praktisch Voorbeeld

Nu komt het spannende moment. We gaan onze AI vragen om een bestand aan te maken en er inhoud in te schrijven.

1. Ga naar het Chat-tabblad (💬) in LM Studio.
2. Laad het Model: Zorg ervoor dat je het qwen/qwen3-4b-2507 model hebt geladen. Dit is een compact maar capabel model dat uitstekend geschikt is voor tool-gebruik.
3. Activeer de MCP Tool: In het rechterpaneel, onder Program, zorg je ervoor dat de mcp/filesystem is aangevinkt. De mogelijke acties worden automatisch gedetecteerd en kunnen granulair gekozen worden.
4. Geef de Opdracht: Typ de volgende vraag in het chatvenster: *Maak een tekstbestand aan met de naam "Syntra.txt" en schrijf daarin een lang, informatievf artikel over wat kunstmatige intelligentie is. Houd het begrijpelijk voor beginners. Minstens 2000 woorden.*
5. Observeer de Magie: Het model zal nu de opdracht analyseren, de MCP Filesystem tool gebruiken om het bestand aan te maken, en een artikel schrijven. Je ziet in het chatvenster dat het model de tool aanroept. Zodra het klaar is, kun je naar je AI_Workspace map navigeren en het bestand Syntra.txt openen. Je zult zien dat de AI daadwerkelijk een artikel heeft geschreven en opgeslagen!