Your Private AI Coding Buddy

October 1, 2025

By Aaron Grothe
AlphaWall LLC

Introduction (Continued)

If you have questions/comments please feel free to ask them anytime. You don't have to hold them until the end of the talk.

This is just one example workflow. There are a lot of options.

Slides will be up on my website at https://grothe.us in the presentations section tonight or tomorrow

Why Private AI?

Why do Private AI?

- Free: You are the product. Your Queries/Code, etc. are being licensed to the provider
- Cost: If you're using an AI via API you're paying by number of API calls/tokens, etc. Can add up.
- Privacy: Are you willing to explain to work why you're sharing proprietary code with a search engine?

Why Private AI?

Some Limitations of Private AI?

- Power: How powerful is the machine you're using?
- Complexity: You're building a solution - multiple parts
- Learning Curve

Overview

For a Private AI Coding Buddy: You'll need the following pieces

- A Machine to run the LLM
- Large Language Model (LLM)
- LLM Engine
- IDE or User Interface

Machine to Run LLM

To run a Large Language Model you need a machine with some capability

My lab currently has 4 machines in it

- Macbook Air M1 - 8gb ram / Apple M1 silicon
- Acer Aspire 3 Laptop - 32gb ram / Ryzen 7500u/APU
- Beelink - 24gb ram / Ryzen 7500h/APU (mini-pc)
- HP z260 - 96gb ram / 4 x Nvidia k2200s (4gb each)

Machine to Run LLM - Tips

Apple Silicon machines are wonderful for running LLMs and dev

Notes

- Thermal throttling - You get about 10 - 15 minutes of full performance with a Macbook Air
- Cost - adding ram makes them kind of expensive
- Base specs M4 / 64gb ram makes a very nice machine

Machine to Run LLM - Tips

Acer Aspire 3 Laptop

Notes

- Thermal throttling - Fan goes pretty hard when running
- Cost - uses regular laptop memory, upgraded to 64gb for $50
- Lack of a discrete GPU is painful sometimes
- Might have to give memory hints to grub e.g. amdgpu.gttsize=12288

Machine to Run LLM - Tips

Beelink - Mini-PC

Notes

- Almost identical to Aspire Laptop
- Mini-desktop form factor better airflow
- Cost - uses regular laptop memory, can upgrade memory for similar prices to Acer Aspire 3
- Lack of a discrete GPU is painful sometimes

Machine to Run LLM - Tips

HP Z620

Notes

- Bunch of cheap Nvidia K2200s - 4 cards at about $25 / each
- Older Nvidia cards don't work for LMStudio
- Plenty of cooling, workstation

Machine to Run LLM - Tips

Next Machine

Contenders

- Build your own PC - with Intel ARC cards B580 - cheap 16gb
- Build your own PC - with Nvidia cards
- AMD Geekom A9 - 32gb Ram / AMD Ryzen AI ~$1,000
- Framework Workstation ~$2,000
- Windows Snapdragon Elite PC

Large Language Models

Huggingface.co - has thousands and thousands of available models

Some of the most popular models are

Llama3.2b - Meta's pretty loosely licensed model
Deepseek R1 - new kid on the block, has good reviews
Codellama - Customized version of Llama tuned for coding
Qwen2.5 - well regarded for coding purposes (main one I use)

Special models exist as well: sqlcoder, etc.

LLM Engines

Currently there are two very popular engines for running your LLMs locally

OpenLlama - https://ollama.com/
LM Studio - https://lmstudio.ai/

OpenLlama - is a bit older, has a well defined front end in openwebui
LM Studio - is newer, gives more options on model tuning. E.g. multiple versions of RocM, etc. requires newer cards

IDE or User Interface

You’ll need an interface to access the LLM

LM Studio:	has a chat interface
OpenWebUI:	provides a modern web interface
Ollama:	command line
Continue:	plugin for VS Codium, Command Line

Demo

Machine: Acer Aspire 3 / 16gb set aside to APU
LLM: Llama3.1 8b
LLM Engine: LM Studio
IDE: VS Codium

Let's do a simple game of life

Future Things

Two interesting future areas of development

- Command Line AI
- IDE Integration - Autocompletion, etc
- Small Language Models (SLM)

Command Line AI

Continue has command line access

- Currently in beta
- Is very much in beta

E.g. echo “generate git commit message for changes” | cn -p

Once it stabilizes could be a very useful tool.
Moves closer to being able to do more agentic things

This is going to be a great area of development

IDE Integration - Autocompletion, etc

Cline is a cool example of this plugging into your regular IDE.

Adds capabilities to IDE for things like AI autocomplete and so on. Works pretty well if you have a good AI machine. Not so much on mine.

Small Language Models (SLM)

Small Langauge Models are custom models that are trained on specific data. Think of it as a trained version of Retrieval Augmented Generation.

Think of making a model of all of your Git models, documentation, wiki entries and so on.

The SLM can be chained first so it is queried first then refer to a LLM if you don't find a match

Minimal Configuration

I want to try this out and get an idea of how/if this will work for me.

You only need 2 items to give this a spin

Llamafile - combined LLM and engine
Continue - IDE integration if desired

Minimal Configuration (part 1)

Llamfile - https://github.com/Mozilla-Ocho/llamafile

This combines an LLM with a runtime engine. Is also multi-platform so same executable runs on Linux/Windows/Mac OS/BSDs

Download an LLM from their github repo. Recommend

LLaMA 3.1 8B Instruct

```
# chmod +x ./llamafile
# ./llamafile -m llama-3.1-8b-instruct-q4_0.gguf --server
--nobrowser
```

Minimal Configuration (part 2)

Basic config file for continue plugin

```
~/.continue/config.yaml
{
 "models": [
   {
     "title": "Llamafile Mistral-7B",
     "provider": "openai",
     "model": "mistral-7b",  // This can be any string; it's for display
     "apiBase": "http://localhost:8081/v1",
     "apiKey": "EMPTY"  // Or "sk-no-key-required"; llamafile doesn't need a real key
   }
```

Minimal Configuration (Notes)

Couple of things to know

- You might need to change the apiBase from 8080 to 8081
- Performance of Llamafile may not be optimzied
- If you have issues try and use the web version of llamafile at http://localhost:8080

Retrieval Augmentation Generation (RAG)

Retrieval Augmentation Generation (RAG)

Easiest way to get your code, documents, spec guides, etc into your AI system. Open WebUI makes it very easy to do this

Fine Tuning/Retraining, etc. are outside scope for this talk.

Open Source?

Can I do a fully Open Source/Free version of a Private AI coding buddy?

Ollama is open source
There are open source LLMs on huggingface.co
Codium is a version of Microsoft Code with telemetry removed
Continue is open source, but has telemetry you can turn off

Things I Wish I knew Earlier

Couple of things that might give you a hand

- You will get different results from asking the same question
- Minimum Viable Product (MVP) is a good way to get started
- If one model doesn't work for you, try another Llama-3.2:8b is pretty good
- If you have an amd apu keep in mind amdgpu.ggtsize it can be a game changer on Linux
- Alex Ziskund's Youtube Chanel is a big help - https://www.youtube.com/@AZisk

Things I Wish I knew Earlier

Couple of things that might give you a hand

- AI Omaha - Has a meetup, very good group about general purpose AI
- Mac or Nvidia cards are the easy setup for Private AI
- OpenWebUI is amazing - very easy to install with docker, makes RAG pretty easy, interface is similar to ChatGPT's and others
- If OpenWebUI gives you grief clear your localhost cookies in cache, bug in recent version of OpenWebUI
- Hallucinations are real in AI, check everything

Summary

You can start with Private AI for free

This is a quick talk. Matt Payne did a talk about Private AI Coding Assistants in January for the Omaha Java User Group - https://www.grothe.us/presentations/ojug-202501-aicoding.pdf

Thanks for coming :-)

Links

Models for LLMs

- Hugging Face - hold thousands of ready to use models - Huggingface.co

LLM Engines

- OpenLlama - https://ollama.com/
- LM Studio - https://lmstudio.ai/

Links

LM Studio: Web interface, provides API access, Chat, ability to tun models

- LM Studio: https://lmstudio.ai/

OpenWebUI: provides a modern web interface to OpenLlama

- OpenWebUI: https://openwebui.com/

Links

Ollama: Command Line, LLM Manager

- Ollama: https://ollama.com/

Continue: Plugin for VS Code/Codium, and Intellij, others

- Continue: https://www.continue.dev/

Links

LLamfile - LLM in one file

- Github page: https://github.com/Mozilla-Ocho/llamafile

Youtube Pages

- Alex Ziskind: https://www.youtube.com/@AZisk
- Dave's Garage: https://www.youtube.com/@DavesGarage