Adelaide University

How to Build Your Own AI Research Assistant

Daniel Ebbert
Centre for Change and Complexity in Learning

12/10/2026

https://ebbert.nrw/blog

Who am I?

- Education researcher with a technical background
- Never satisfied with the limitations of technology
- Early adopter with a hacker mindset
- ChatGPT/Claude wasn’t good enough
- Adjusting AI setups/clients to my needs

Aim

Enable you to create your own AI assistant setup that fits your workflow.

Current State

- Which AI are you using?
- How are you accessing it?
- What purpose are you using it for?

Adelaide University

Introduction

- Any AI system is only as good as the information it has access to
  - garbage in, garbage out
- Generic chatbots (ChatGPT/Claude) answer based on training data, web search, and your input

Context Engineering

- Curating what an AI has access to
- What would be relevant context when using AI for research?

Literature

- Curated selection of work that you're interested in (and have read)
- Full-text PDFs
- Metadata (journal, co-authors, etc.)

-> use a reference manager

Notes (Personal Knowledge)

- Information beyond published literature, (e.g. meeting notes, papers, reviews, grant applications, etc.)
- Your point of view!

-> a note taking software might be useful

Machine Readability

- That a human can read something does not mean a machine can read it

Literature

- Depending on connection to reference manager
- Metadata might already be machine-readable
- PDFs will need to be converted to plain text

Notes

- Ideally use plain text files (.txt or .md)
- Word files will need to be converted
- Note-taking software on top of Markdown files make it easy

Self-Audit

- What is the current state of the context you want to provide an AI with?

Literature

- How are your sources organised?
- Are you using a reference manager?
- Is the metadata for your sources correct?
- Do you have the PDFs for all of your sources?

Notes

- Do you take digital notes?
- Are those notes machine readable?
- Are they sorted?
- Do they contain metadata?
- What might be useful to an AI that is not yet covered in your notes?

Make Progress Where You Are

Based on your self-audit, choose either your literature or your notes and work on making them useable as context for an AI.

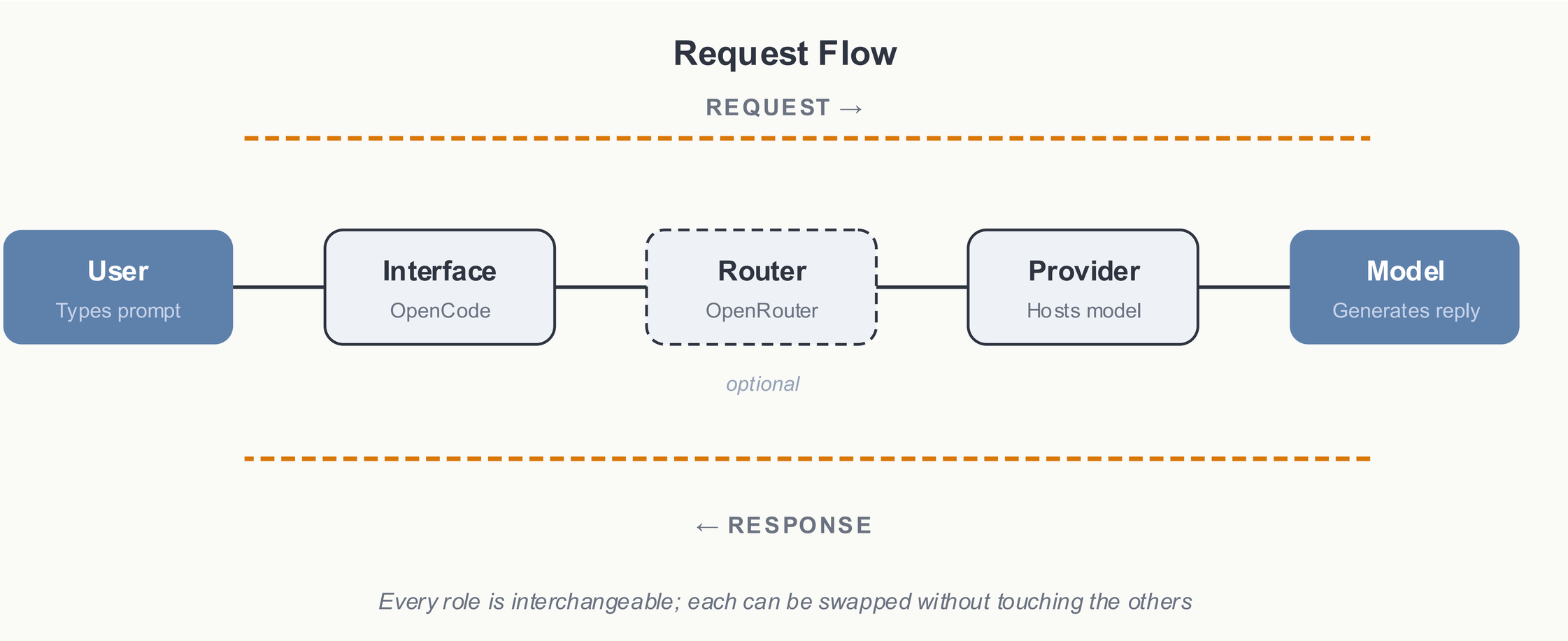
- No reference manager: install one and start structuring your sources
- Sources unsorted or metadata incomplete: sort and clean
- Notes not machine readable: start converting them to plain text or Markdown
- Already sorted: deepen metadata and improve organisation

Break
(10 minutes)

Anatomy of an AI Assistant

Term	Definition	Example
Developer	Whoever trained the model	Anthropic for Claude, OpenAI for ChatGPT
Provider	Whoever hosts the model and gives access to it	Anthropic's own API, AWS Bedrock, Microsoft Azure
Router (optional)	Routes your request to a provider based on your configuration	OpenRouter, Venice.ai
Interface	The UI you interact with	ChatGPT, Claude.ai, OpenCode

Request Flow



Closed-weight vs open-weight

- Closed-weight
  - Proprietary models that are hosted by the developer, or licensees
  - No (or very little) competition on price
  - Examples: ChatGPT, Claude
- Open-weight
  - Models that can be hosted on other infrastructure without paying a fee
  - Competition on price
  - Examples: DeepSeek, GLM

Which model to choose?

- Which model to use depends on the use case
- Things to consider when making a choice:
  - Input/Output modality (text, image, video, etc.)
  - Context window (max. length input)
  - Data storage / Data retention / Jurisdiction

Model decision flow

1. What access do you already have?
   - Subscription or institutional access.
2. What may happen to your data?
   - Retention / storage / jurisdiction.
3. Open-weight or closed-weight?
   - Might be a forced choice depending on data handling requirements.
4. Single provider or router?
   - Single provider might be cheaper, a router would be more flexible
5. What do you need to put in and get out?
   - Text, images, video, etc.
6. How much text at once?
   - Context window.
7. What does it cost?
   - Price per million tokens, input and output.

Evaluation Case

- Benchmarks are based on contrived tasks; rankings do not transfer to your use case
- Build your own test task instead: one you can judge
- Example: give a model a few pages of your writing and ask it to verify every claim against the cited sources
- Run the same task across your shortlist and compare

Tools

- Tools give AI models the ability to interact with other software
- Tool examples: web search, editing local files, fetching content from the web
- Tool categories:
  - Local tools as part of the interface
  - Server tools as part of the provider
  - **Model Context Protocol (MCP)**
  - **Agent Skills**
  - Various interface specific options, like hooks, plugins or so

Model Context Protocol (MCP)

- Standard for integrating external software
- All tools are fully loaded at startup
- Can be either:
  - Local connection to installed software on your device
  - Remote connection to server
    - Risky option as available tools can change without notice

Agent Skills

- Lightweight format for packaging knowledge and workflows (instruction manual)
- Skills are defined in markdown files (SKILL.md)
- Skills are flexible and can be used for a variety of tasks
  - Step by step instructions for tasks
  - Scripts that are repeatedly run
  - Templates for specified types of content
- Useful for telling an AI how to do a task

Interface (also called a Harness)

- Purpose-built interfaces: Claude Code, Claude Cowork, OpenCode, OpenWork, etc.
- Must be compatible with your chosen provider or router
- Must support the tools you want to connect: MCPs, skills
- Some interfaces include a browser the AI can use itself to navigate and read the web
- Some can also control the device itself
- Consider the intended purpose:
  - some are built for development (Claude Code, OpenCode) but work for a text based workflow,
  - others for general knowledge work such as documents or slides (Claude Cowork, OpenWork)
  - personal (semi-)autonomous agent (OpenClaw, Hermes, OpenHuman)

AI choices and tool mapping

- Following from the choices you've made about which model, provider, and interface to use determine how you're going to connect the tools you want your AI assistant to have
- Reference manager -> MCP
- Notes -> MCP or Skills
- Examples:
  - If you use Zotero you can use an MCP: https://github.com/54yyyu/zotero-mcp
  - If you use Obsidian you can use:
    - An MCP: https://3sztof.github.io/posts/obsidian-smart-connections-mcp/
    - Agent Skills: https://github.com/kepano/obsidian-skills

Build Plan

You now know all the pieces. Decide what to install and connect, and in what order.

- Base it on your self-audit, your stack choice, and your connection look-up results
- Order matters: knowledge base first (sources and notes), runtime second
- Prioritise: what is the first thing you will set up after this workshop?

If you have time left, start building.

Responsible Use Discussion

- Human oversight of AI generated content is essential
- Do not delegate what you could not do yourself
  - If you cannot assess the output, you should not use the AI for that task
- Consider the distribution of intellectual labour
  - The intellectual contribution should be yours
- Where you draw the line is your judgement call

What do you think?

Thank you!

daniel.ebbert@adelaide.edu.au

ebbert.nrw



Adelaide University