Sovereign Personal AI Assistant

A local assistant with the polish of a modern chat app, with encrypted phone access, scaling consumer → enterprise on one architecture

● Design & build plan – buildable at the consumer tier today, enterprise-scalable; not a measured deployment

Design & build plan. The consumer tier is buildable today with real, fitting models; the enterprise tier is a hardware step, not a rewrite. Numbers are model footprints and budgets, not measured results.

The idea

Cloud assistants are polished but put every prompt, response and history on someone else's server. Local assistants keep the data local but have been desktop-bound. This project closes the gap: a sovereign assistant with the comfort of a modern chat interface – chat history, projects, artifacts, tool use – reached from a phone, while the model and every conversation stay on owned hardware.

It is deliberately a **consumer product first** – not evaluation infrastructure (Projects A/B), not a professional operator workstation (C), not research (D). The design that works at home is the same one that scales to a team; the only variable is hardware.

Remote architecture (the differentiator)

The host runs the model behind an OpenAI-compatible endpoint; the phone is a thin client; the two are bridged by an end-to-end encrypted mesh (WireGuard over Tailscale) that opens no ports and exposes nothing to the public internet. Inference runs on the host, chat history stays on the devices, and the only thing that reaches the vendor's backend is the device-discovery list used to pair the machines.

Concretely this maps to LM Studio's **Locally** iPhone/iPad app + **LM Link** (shipped 2026), but the property is general: reach capable owned hardware, from anywhere, over an encrypted tunnel nobody else can read.

Current honest limits: first-party mobile client is iPhone/iPad, both ends run the same app, pairing is account-gated, and phone sessions default to a shorter context.

Honest hardware envelope

Sovereignty is only real if the model fits the card.

Consumer – 1× RTX 5090 (32 GB). Runs a model that fits with headroom:

Model	Footprint	Note
gpt-oss-20b	~13 GB (MXFP4)	fast daily driver
gemma4-31b	~19 GB	dense, strong reasoning
qwen3.6-35b-a3b	~24 GB	MoE, ~3B active

Enterprise – 2× 5090 (64 GB) or an 80 GB host. Adds the model the consumer card cannot hold:

Model	Footprint	Why not consumer
gpt-oss-120b	~60 GB (MXFP4)	116.8B total / 5.1B active; needs 64–80 GB. On 32 GB it would offload ~half the weights to system RAM and hit the bandwidth cliff.

The 120B is not a consumer claim – it is exactly what the enterprise tier adds when the hardware arrives. (MoE saves compute per token, not resident memory: all experts stay loaded.)

The stack

Because the server is OpenAI-compatible, the assistant is model-agnostic and tool-compatible:

- **Model backend** – Ollama / vLLM / LM Studio, any open-weight model.
- **OpenAI-compatible API** – `localhost:1234`, one endpoint.
- **UX layer** – chat history, projects, artifacts, tool-use.
- **Remote layer** – encrypted mesh, phone ↔ host, same endpoint.

Existing agentic tools (e.g. OpenCode) target the same endpoint and keep working, locally or remotely, with no reconfiguration.

Practical use case (STAR)

- **Situation.** A privacy-conscious professional – and later their small firm – wants a capable assistant with the comfort of a modern chat interface including phone access, but confidential material cannot go to a third-party cloud, and desktop-bound local setups are useless away from the desk.
- **Task.** Stand up a sovereign assistant that is genuinely usable day to day, reachable from a phone anywhere, and runs a model that actually fits the hardware – with a clean path to a team that doesn't require rebuilding it.
- **Action.** On the home RTX 5090, serve a 32 GB-fitting model (gpt-oss-20b or Qwen3.6-35B-A3B) through the OpenAI-compatible endpoint; wrap it in a polished chat UX; reach it from the phone over the encrypted mesh – no open ports, inference and history on owned hardware.
- **Intended result (practical impact).** A personal assistant the owner uses from their phone with nothing leaving their control – and a one-step enterprise path: add a second 5090 (or an 80 GB host) and the same stack serves a 120B-class model to the whole team over the same encrypted mesh, still with no public exposure. Consumer build today; enterprise by adding hardware, not by re-architecting.

Status

Buildable now at the consumer tier with real, fitting models; the enterprise tier is a hardware step, not a rewrite. The core property holds at every scale: **inference and history stay on owned hardware** – the whole reason to build locally.

Designed under CTC AI Operations, on the same local-inference discipline as the evaluation and workstation projects it sits beside.