

BLUEPRINT

The Self-Hosted AI Stack

Private inference, agent memory and assistants on your own
Kubernetes. Data sovereignty by architecture, not by policy - the
stack we run ourselves.

Why self-host AI at all

Managed AI APIs are excellent - until the data question arrives. Enquiries containing customer details, internal documents feeding an assistant, agent memory accumulating business knowledge: each leaves your perimeter when the provider is external. Self-hosting does not reject outside AI; it draws a line where sensitive context never crosses.

The reference architecture we operate

- Kubernetes on Talos Linux - immutable, minimal attack surface, managed through GitOps so the cluster state is reviewable in Git
- Model serving for private inference - LLMs and embeddings run inside the cluster for tasks that must not leave it
- Cognee for agent memory - a knowledge-graph layer that gives agents persistent, inspectable context instead of opaque prompt stuffing
- Typebot for guided assistants - conversational intake and internal helpers, self-hosted with restricted allowed origins
- MinIO for object storage - documents, artifacts and model-adjacent data under the same sovereignty rules
- Keycloak for identity - assistants and tools authenticate against the same IdP as the team

This is not a diagram we sell. It is the stack this company's own assistants, memory and pipelines run on today.

Sizing honestly

A CPU-only cluster handles embeddings, retrieval, orchestration and small models well. GPU nodes matter when you want fast private inference on larger models - they add cost and scheduling complexity, so the honest answer is: start with the assistants and memory layer, measure the real demand, then decide whether dedicated inference hardware pays for itself. For many organisations the pattern is hybrid - public APIs for low-sensitivity tasks, private inference where context is sensitive.

What operating it actually means

- ☐ Versioned deployments through GitOps - every model, assistant and memory service is a reviewed change, not a console click
- ☐ Backup and restore procedures that cover agent memory and object data, with documented restore tests
- ☐ Monitoring that treats AI services like any other workload - availability, latency, resource use
- ☐ Access governed through the company IdP - assistants are not a shadow identity system
- ☐ Update paths tested on a copy before production - a broken memory layer is an outage, not an experiment

This document is part of the UHL resources collection. Direct download and related work:
<https://uhl.cloud/en/resources/>

