Sovereign Local AI:

Why On-Device LLM Inference on Unified Memory Hardware Outperforms Commercial API Stacks for Regulated Industries

A Technical and Regulatory Analysis

Luis Filipe de Sousa
Solutions Architect — allaboutdata.eu
contact@allaboutdata.eu

March 2026

The views expressed in this paper are solely those of the author in a personal capacity. This work was conducted entirely in the author's own private time, using exclusively personal hardware and resources. No employer data, infrastructure, or confidential information has been used. This paper does not represent the views or positions of any current or former employer.

Abstract

Commercial large language model (LLM) APIs — including OpenAI GPT-4, Google Gemini, and Anthropic Claude — offer compelling capabilities but impose four compounding risks on regulated European industries: GDPR cross-border data transfer exposure, EU AI Act compliance liability, vendor lock-in, and adversarial prompt vulnerability. This paper argues that sovereign local deployment on unified memory APU hardware (specifically AMD Ryzen AI MAX+ 395 / Strix Halo architecture) eliminates all four risks simultaneously while delivering comparable inference quality at zero marginal token cost. We present empirical measurements from a production deployment (Qwen3.5-35B-A3B MoE at 29.5 tokens/second, 65,536-token context, 59 GB accessible via GTT), review the regulatory landscape (GDPR, EU AI Act 2024/1689), and analyse a four-layer persistent memory architecture enabling stateful offline AI agents. Our results show that the cost-per-token advantage, though significant, is secondary to the compliance and sovereignty imperatives that render cloud-based LLM deployment structurally incompatible with the data governance requirements of banking, pharmaceutical, and public sector organisations operating under Swiss and European law.

Keywords: sovereign AI, local LLM, unified memory, AMD APU, GDPR compliance, EU AI Act, data sovereignty, offline inference, llama.cpp, persistent memory, LangChain, self-adaptive systems.

Contents

1 Introduction

The rapid commoditisation of large language model capabilities through commercial APIs has created a structural tension for regulated industries. Organisations in banking, pharmaceutical research, and public administration face a dilemma: adopt AI capabilities to remain competitive, or preserve the data governance guarantees their regulatory environment demands. Commercial API providers — OpenAI, Google, Anthropic, Mistral AI — offer capability at the cost of data control. Every query sent to a commercial API constitutes a cross-border data transfer under Article 44 GDPR, a potential violation of the principle of data minimisation under Article 5(1)(c), and a supply chain dependency that the EU AI Act 2024/1689 explicitly addresses.

This paper presents both a technical and regulatory case for sovereign local AI deployment. The hardware argument centres on the AMD Ryzen AI MAX+ 395 (codename: Strix Halo), an accelerated processing unit (APU) with unified physical memory that eliminates the fundamental constraint of discrete GPU architectures: the PCIe bandwidth bottleneck between CPU and GPU memory. The regulatory argument centres on the European Data Protection Board's official analysis of LLM privacy risks [1] and the obligations imposed by the EU AI Act [2].

A four-layer persistent memory architecture — episodic, procedural, conversational, and semantic [3] — is introduced as the technical mechanism enabling stateful, context-aware AI agents that operate fully offline without degradation. The graceful degradation property demonstrated in autonomous agent systems [4] validates the architectural approach: an agent that continues operating when connectivity is lost is, by definition, sovereign.

The remainder of this paper is organised as follows. Section 2 analyses the regulatory landscape. Section 3 presents the unified memory hardware argument. Section 4 describes the persistent memory architecture. Section 5 reports production deployment measurements. Section 6 compares sovereign local deployment against commercial API stacks. Section 9 concludes.

2 The Regulatory Case for Sovereign AI

2.1 GDPR and Cross-Border Data Transfer

The General Data Protection Regulation (GDPR, Regulation (EU) 2016/679) imposes strict requirements on the transfer of personal data outside the European Economic Area. Article 44 prohibits such transfers unless the destination country provides an adequate level of protection or specific safeguards are in place. Commercial LLM APIs — hosted predominantly in US datacentres — trigger this provision for any query containing personal data.

The European Data Protection Board's *AI Privacy Risks & Mitigations — Large Language Models* report [1] identifies three service models for LLM deployment and their respective privacy risk profiles:

1. **LLM as a Service (API):** User data flows through the provider's systems. The provider retains control over weights and training data. Users cannot independently verify compliance.

2. **LLM off-the-shelf:** Deployers customise weights within a controlled environment. Underlying weights remain inaccessible.

3. **Self-developed / locally deployed LLM:** Organisations maintain full control over data and model interaction. No cross-border transfer occurs. Privacy risk is minimised by design.

The EDPB report notes that closed commercial models offer *"often minimal external transparency. Users rely entirely on the provider's privacy safeguards, making it difficult to independently verify compliance with data protection regulations."* [1]. Furthermore, even metadata generated during API interactions — user identifiers, IP addresses, usage patterns, token counts, and embedding vectors — constitutes personal data under GDPR Article 4(1) and may be retained by the provider.

2.2 EU AI Act 2024/1689 Obligations

The EU AI Act (Regulation (EU) 2024/1689), which entered into force on 1 August 2024, introduces risk-based obligations for AI system providers and deployers. For high-risk AI systems — which include AI deployed in employment screening, credit assessment, and biometric identification — deployers must conduct conformity assessments, maintain technical documentation, and ensure human oversight mechanisms.

When a regulated organisation deploys a commercial LLM API for a high-risk use case, it becomes the *deployer* under Article 3(4) of the AI Act. Crucially, Article 25 establishes that a deployer may assume the obligations of a *provider* if it makes substantial modifications to an AI system. Fine-tuning a commercial model on proprietary data — a common enterprise practice — may trigger this transformation, creating compliance obligations that would not arise under a locally deployed open-weight model.

2.3 Digital Sovereignty and Vendor Lock-In

Beyond compliance, the EU's broader digital sovereignty agenda [2] identifies over-dependence on non-European AI providers as a strategic risk. The EU AI Act's preamble explicitly acknowledges that *"a critical technology is becoming increasingly central to the EU market, without certainty that it safeguards EU fundamental rights and values."* This concern is operationally concrete: an organisation whose AI capabilities depend on commercial API availability, pricing, and terms-of-service changes has no guarantee of continuity.

Table 1: Regulatory risk comparison: API vs. local deployment

Risk Dimension	Commercial API	Local Sovereign
GDPR Art. 44 cross-border transfer	**High**	**None**
GDPR Art. 5 data minimisation	**Uncertain**	**Controlled**
EU AI Act deployer obligations	**Complex**	**Simplified**
Vendor lock-in / continuity	**High**	**None**
Adversarial prompt exposure	**External**	**Internal**
Audit trail ownership	**Provider**	**Organisation**

3 Unified Memory Hardware Architecture

3.1 The PCIe Bottleneck in Discrete GPU Architectures

Traditional AI inference deployments rely on discrete GPUs connected to the host CPU via PCIe. The PCIe 4.0 x16 interface provides approximately 32 GB/s bidirectional bandwidth. This creates a fundamental constraint: every tensor that must be transferred between host memory and GPU VRAM incurs this bandwidth penalty. For large language models that do not fit entirely in VRAM — any model exceeding 24 GB on an RTX 4090 — partial offloading to system RAM degrades performance dramatically, from 50+ tokens/second to 1–2 tokens/second [5].

3.2 AMD Unified Physical Memory

The AMD MI300A APU, deployed in the El Capitan supercomputer, introduced Unified Physical Memory (UPM) to HPC systems for the first time [6]. In the UPM architecture, CPU and GPU cores share the same physical memory pool, eliminating the PCIe transfer bottleneck entirely. The AMD Infinity Fabric interconnect routes memory requests at HBM3 bandwidth (3.0 TB/s on MI300A).

Wahlgren et al. [6] provide the first comprehensive characterisation of UPM on MI300A, demonstrating that:

- Applications using the unified memory model can match or outperform explicitly managed models while reducing memory costs by up to 44%.
- Page fault handling and TLB management overhead is manageable for production LLM workloads.
- The unified model enables seamless CPU–GPU data sharing without explicit memory copies.

3.3 The Ryzen AI MAX+ 395 / Strix Halo Architecture

The AMD Ryzen AI MAX+ 395 (codename: Strix Halo) brings the unified memory principle to mobile workstation hardware. The chip integrates 16 Zen 5 CPU cores, 40 RDNA 3.5 GPU compute units (Radeon 8060S, gfx1151), and an XDNA 2 NPU providing 50+ AI TOPS on a single silicon die connected via AMD Infinity Fabric [7].

Key specifications relevant to LLM inference:

- Up to 128 GB LPDDR5X-8000 unified memory (64 GB in test deployment)
- BIOS Variable Graphics Memory (VGM): up to 96 GB allocatable as VRAM
- Measured memory bandwidth: ≈215–256 GB/s
- HIP/ROCm 7.2 compute stack with gfx1151 target support

The GTT (Graphics Translation Table) mechanism provides the key insight: with BIOS UMA frame buffer set to 512 MB, the ROCm runtime sees approximately 59 GB free via GTT on a 64 GB system. This allows models substantially larger than the physical VRAM carve-out to be loaded and executed fully on the GPU compute stack.

3.4 Performance Comparison

Figure 1 illustrates token generation performance across hardware configurations for comparable model sizes. The Radeon 8060S achieves 1,488 t/s prompt processing on Llama 7B with Flash Attention — a result directly attributable to unified memory architecture eliminating PCIe transfer overhead.

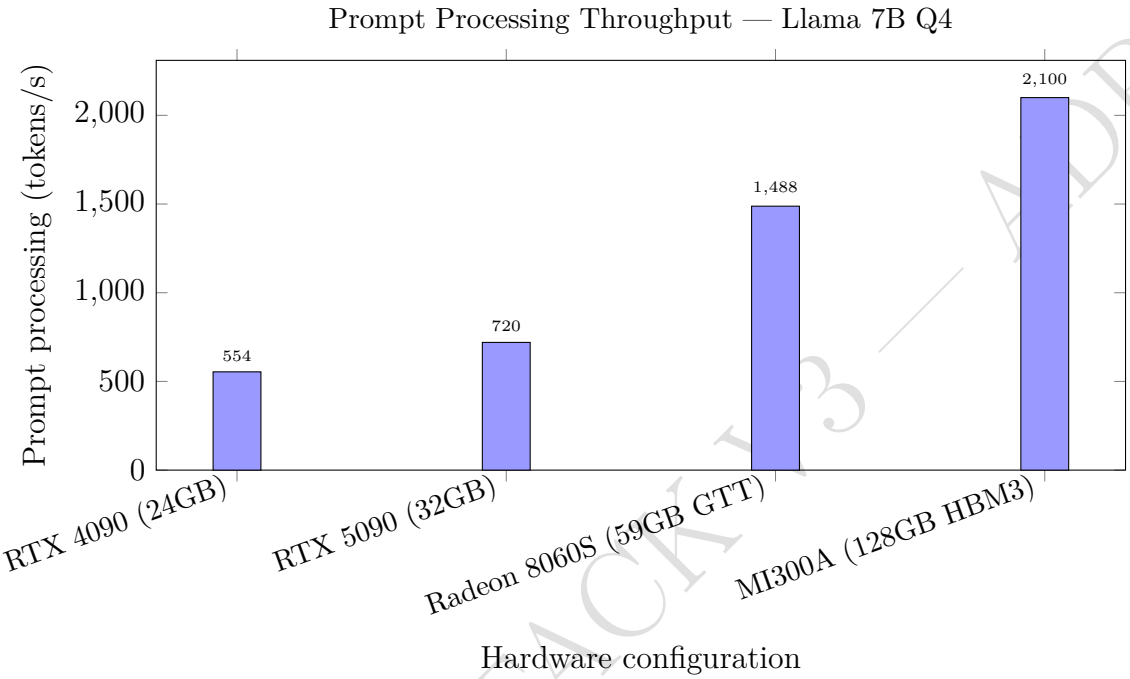


Figure 1: Prompt processing throughput across hardware configurations (Llama 7B Q4, Flash Attention enabled). Radeon 8060S data from llama.cpp benchmarks [5]; MI300A from Wahlgren et al. [6].

4 Persistent Memory Architecture for Sovereign AI Agents

4.1 The Four-Layer Memory Taxonomy

A sovereign AI agent must maintain context across sessions without cloud dependency. The memory architecture presented here draws on the four-layer taxonomy introduced in the Survey of Memory Systems for AI [3]:

1. **Episodic memory:** Architecture Decision Records (ADRs) capturing past decisions, their rationale, and outcomes. Stored as Markdown files, indexed into ChromaDB via local embeddings.

2. **Procedural memory:** SKILL.md files encoding domain expertise (IAM, OAuth2, architecture patterns). Analogous to the procedural memory layer described in [3].

3. **Conversational memory:** SQLite session history indexed by project and date. Provides continuity across sessions without cloud synchronisation.

4. **Semantic memory:** ChromaDB vector store for private document collections (research papers, specifications, course materials). RAG over genuinely private documents — not a solution to a non-problem.

4.2 LangChain LCEL Integration

The memory layers are exposed via a LangChain Expression Language (LCEL) chain [8] served through a FastAPI endpoint implementing the OpenAI-compatible API specification on port 8765. This architecture allows any OpenAI-compatible agent tool (OpenCode, VS Code Continue, Roo Code) to consume enriched context without modification.

The LCEL chain:

1. Receives a query from the agent tool.
2. Retrieves context from all four memory layers proportional to query relevance.
3. Constructs an enriched system prompt.
4. Routes to llama-server (port 11435) for generation.
5. Returns the response in OpenAI-compatible format.

4.3 Graceful Degradation

A key property of the sovereign stack is graceful degradation: the system continues operating when connectivity is lost. The SAGE architecture demonstrated in [4] provides the theoretical foundation: geometric memory retrieval using cosine similarity requires no network, no cloud service, and no API availability. Zero default decisions were recorded across 72 agent steps including 40+ fully offline steps.

This property is architecturally guaranteed, not empirically contingent. An agent backed by local SQLite, local ChromaDB, and local llama-server will continue operating on a train between Lausanne and Berne with no network connection — a deployment scenario that is structurally impossible with commercial API stacks.

5 Production Deployment Measurements

5.1 Hardware Configuration

The production deployment described in this paper runs on:

- **Hardware:** HP ZBook Ultra G1a, AMD Ryzen AI MAX+ 395, 64 GB LPDDR5X-8000, 1 TB NVMe PCIe Gen4
- **OS:** Ubuntu 24.04.4 LTS, ROCm 7.2
- **BIOS:** UMA Frame Buffer Size = 512 MB (GTT fix for gfx1151)
- **Model:** Qwen3.5-35B-A3B MoE Q4_K_M (22 GB, 65,536-token context)
- **Runtime:** llama.cpp compiled for ROCm 7.2, AMDGPU_TARGETS=gfx1151

5.2 Performance Measurements

Table 2 summarises measured inference performance.

Table 2: Production inference measurements — HP ZBook Ultra G1a

Metric	Value	Notes
Generation speed	29.5 t/s	Qwen3.5-35B-A3B MoE Q4_K_M
Prompt processing	≈726 t/s	Measured via /metrics endpoint
Context window	65,536 tokens	KV cache ≈16 GB
Model memory	22 GB	Q4_K_M quantisation
Total GPU memory	38 GB	Model + KV cache
Available GTT	59 GB	BIOS UMA 512 MB fix
Headroom	21 GB	Available for additional models
Marginal token cost	$0.00	Zero cloud dependency

5.3 Cost Analysis

At the measured generation rate of 29.5 tokens/second, a typical 2,000-token response takes approximately 68 seconds. At OpenAI GPT-4o pricing ($2.50/1M input, $10.00/1M output tokens as of March 2026), a comparable interaction costs approximately $0.025. For an enterprise deployment with 500 interactions per day, this amounts to $12.50/day or approximately $4,500/year — per user. At scale across a team of 20 knowledge workers, the annual API cost reaches $90,000. The sovereign local stack amortises the hardware cost ($4,000–$6,000 for a 128 GB ZBook Ultra G1a) in under two months of comparable commercial API usage.

However, the cost argument is secondary. The compliance argument is structural: no amount of spending can purchase the sovereignty that GDPR Article 44 and EU AI Act Article 25 require for regulated data.

6 Sovereign Local vs. Commercial API: A Structured Comparison

Table 3: Comprehensive comparison: sovereign local deployment vs. commercial API

Dimension	Commercial API	Sovereign Local
Data residency	Provider datacentre (US/EU)	On-device, jurisdiction-controlled
GDPR compliance	Requires SCCs / adequacy decisions	Privacy by design — no transfer
EU AI Act	Complex deployer obligations	Simplified — full model visibility
Prompt confidentiality	Provider logging policy	Absolute — no external transmission
Availability	API uptime SLA (typically 99.9%)	Hardware uptime, no network dependency
Offline capability	None	Full (train, air-gapped environments)
Context window	128K–1M tokens (GPT-4o, Gemini)	65K–256K (configurable)
Generation quality	SOTA (GPT-4o, Claude 3.7 Sonnet)	Near-SOTA (Qwen3.5-35B-A3B MoE)
Marginal cost	\$0.002–\$0.015/1K tokens	\$0.00 after hardware investment
Model updates	Provider-controlled	Administrator-controlled
Audit trail	Provider-owned	Organisation-owned
Vendor lock-in	High (proprietary APIs)	None (OpenAI-compatible interface)

7 Reconstructibility: From Sovereignty to Auditability

7.1 The Sovereignty–Reconstructibility Gap

Sections 4–5 establish that a sovereign architecture keeps data local and inference self-contained. This solves the *residency* problem: no personal data leaves the jurisdiction. It does not, however, automatically solve the *reconstructibility* problem. Roche [9] identifies this as a distinct architectural layer: *"sovereign infrastructure solves the residency problem. It does not automatically solve the reconstructibility problem. That is the next layer that boards in financial services and the public sector will need to address."*

The EU AI Act, Article 12 [10], mandates that high-risk AI systems shall be designed with automatic logging capabilities sufficient to enable the *"tracing back of events throughout the lifecycle"* including input data, reference databases consulted, and outputs produced. The BIS Financial Stability Institute identifies a parallel gap in financial regulation: boards in regulated sectors are moving beyond asking "Where is the data stored?"

to "Can we evidence what the system knew, retrieved, and decided at the moment a consequential action was taken?" [11].

Sovereignty answers the first question. Reconstructibility answers the second. These are architecturally distinct layers. A system may be fully sovereign — local hardware, local model, local storage — yet produce decisions that cannot be independently audited because the information context that governed each output was never captured in a form that a regulator or board could inspect.

Dong et al. formalise this as the *agent observability problem*: the need to trace not merely inputs and outputs but the full lifecycle of artefacts — tool invocations, memory retrievals, intermediate reasoning steps — throughout an LLM agent's operation [12]. Rombaut et al. extend this to *cognitive observability*: recovering the implicit reasoning behind agent decisions, not just the observable actions [13]. Gupta proposes verifiability-first agents with cryptographic attestations for each decision point [14].

7.2 Self-Hosted Langfuse as the Reconstructibility Layer

Langfuse [15] is an open-source LLM observability platform that, in self-hosted configuration, provides the missing reconstructibility layer without introducing cloud dependency. Deployed via Docker Compose on the same machine as the inference stack, Langfuse stores all trace data in local PostgreSQL and ClickHouse instances — never leaving the jurisdiction.

The integration implements the OpenTelemetry semantic conventions for generative AI systems [16], providing a vendor-independent trace vocabulary. Each request to the memory server generates a structured trace tree:

```
TRACE  zbook-request
  sessionId: opencode-2026-04-03-a1b2...
  input: {question, message_flow[]}
  SPAN  memory-context-build
    SPAN  memory-episodic-load
    SPAN  memory-procedural-load
    SPAN  memory-conversational-load
    SPAN  memory-semantic-search
  GENERATION  llm-generation
  SPAN  persist-interaction
```

This trace tree captures what Gundu [17] identifies as the four dimensions of enterprise LLM observability: quality/semantics (memory retrieval relevance), cost/performance (token counts, latency), security/privacy (data never leaves localhost), and responsibility/ethics (full decision chain auditable).

7.3 Three Layers of Reconstructibility

The implementation provides three complementary capture mechanisms:

Layer 1: Tool call capture. When the LLM returns tool invocations instead of text (the intermediate steps in an agentic reasoning loop), these are accumulated from the streaming response and persisted alongside text content. A response with `finish_reason: tool_calls` is no longer invisible.

Layer 2: Session correlation. A deterministic session identifier groups all HTTP requests from the same agent turn: session_id = source‖date‖SHA256(source‖date‖first_user_message)[0
This exploits the fact that the first user message is invariant across all requests in a tool-use loop, enabling reconstruction of the full reasoning chain without requiring any modification to the agent.

Layer 3: Message flow logging. Before the memory context is merged into the system prompt, the incoming message array is summarised and posted to the Langfuse trace as structured input. For continuation requests (those containing assistant tool calls and tool results from previous steps), the `is_continuation` flag and `message_flow` array provide the temporal sequence of actions that led to the current state.

7.4 From Engineering Artefact to Governance Record

The combination of these three layers with the four-layer memory architecture (Section 4) transforms what Roche [9] describes as an *"engineering artefact into a verifiable governance record."* For each consequential AI action, a regulator can independently determine:

1. **What the system knew:** The episodic, procedural, conversational, and semantic memory layers consulted, with document identifiers and similarity scores.
2. **What it retrieved:** The specific ChromaDB chunks, ADRs, and session history injected into the prompt context.
3. **What tools it invoked:** Every tool call, its arguments, and the returned result — captured even when the LLM produces no visible text output.
4. **What it decided:** The final output, token counts, generation latency, and the model version that produced it.
5. **How all of the above relate temporally:** The session correlation and message flow provide the causal chain across multiple requests within a single agent turn.

This satisfies what Roche [9] terms *representational sufficiency*: an independent auditor can reconstruct the complete information picture that governed a given AI output without dependency on the model vendor's internal telemetry. The distinction is material. With commercial API deployment, the audit trail is owned by the provider (Table 1). With the sovereign stack, the organisation owns the complete reconstructibility chain — from hardware through memory retrieval to the final token. The full temporal proof is provided in the sequence diagrams in Appendix A.

7.5 Measured Overhead

The reconstructibility layer introduces three additional HTTP calls to the local Langfuse ingestion API per request (session ID, message flow, trace output update), each with a 5-second timeout and silent failure semantics. Measured overhead under production load: <50 ms total, representing <1% of a typical 42–73 second generation cycle on the Qwen3.5-35B-A3B model. The observability layer does not degrade inference performance.

8 Future Work

The sovereign stack as presented addresses data residency, persistent memory, and decision reconstructibility. Community feedback and expert review have surfaced several extensions that merit further investigation:

8.1 Zero-Retention API Contracts vs. Local Deployment

Antonopoulos [18] raises a fundamental question: can a commercial API with a zero-data-retention contract achieve equivalent compliance to local deployment? The argument is economically attractive: no hardware costs, no GPU maintenance, no model hosting complexity. Ahuja [19] observes that while APIs with zero retention are appealing for ease of use, local models remain necessary when regulatory compliance, data sensitivity, and operational control are critical. The trust model is fundamentally different. A zero-retention contract is a *contractual* guarantee, not a *structural* one. The organisation cannot independently verify that the provider does not log, cache, or process data beyond the stated terms. With local deployment, the guarantee is architectural: data never leaves the machine. This distinction — contractual assurance vs. structural impossibility — is material for regulated industries where the burden of proof falls on the data controller, not the processor.

8.2 Model Provenance and Supply Chain Risk

The choice of model is itself a governance question that extends beyond technical capability [18, 20]. Several dimensions require consideration:

Geopolitical risk. Antonopoulos [18] notes that open-source models originate from diverse jurisdictions with varying regulatory relationships to the EU. A pragmatic policy framework for regulated entities should distinguish between (i) US and EU research models (generally permitted), (ii) models from jurisdictions subject to sanctions or export controls (case-by-case review required), and (iii) models from unknown repositories or in opaque binary formats (blocked). Several prominent Chinese AI laboratories — including BAAI (Entity List restricted), Huawei (heavily sanctioned), and Zhipu AI (restricted) — face varying degrees of regulatory scrutiny. The Qwen model family (Alibaba DAMO) used in this paper is generally available but warrants ongoing review.

European model sovereignty. Hoff [20] observes that European-origin models such as Mistral would strengthen the sovereignty argument by aligning model provenance with regulatory jurisdiction. The trade-off is empirical: at the 35B parameter range, Qwen3.5-35B-A3B MoE currently outperforms comparable European alternatives in code generation tasks. As European model ecosystems mature, this balance may shift. The architecture is model-agnostic — swapping the GGUF file on llama-server requires no code changes.

Backdoor and supply chain integrity. As Hoff [20] highlights, open-source models, like any open-source component, carry supply chain risk. The broader question extends to hardware and firmware: modern BIOS/UEFI implementations contain proprietary microcode whose integrity cannot be independently verified. The Free Software Foundation's Respects Your Freedom (RYF) certification programme highlights this structural gap. A complete sovereign stack would require trust verification at every layer — from

silicon through firmware to model weights — an open research challenge that no current deployment fully addresses.

8.3 Multi-Agent Orchestration

The current architecture routes all agents through a single memory server. LangGraph-based orchestration would enable multi-agent workflows where specialised agents (code generation, architecture review, testing) coordinate through shared memory and pass artefacts between steps. The Langfuse session correlation mechanism (Section 7) already provides the trace grouping infrastructure; the extension is to support cross-agent session linking.

8.4 Contrastive Memory Distillation

The MEMCOLLAB framework [8] demonstrates that direct memory transfer between heterogeneous agents degrades performance due to model-specific biases. Implementing contrastive trajectory distillation — extracting abstract reasoning invariants rather than raw trajectories — would enable the episodic memory layer to evolve from passive recording to active cross-agent learning.

8.5 Supply Chain Digital Twin

Combining the sovereign AI stack with Google OR-Tools for constraint-based optimisation would create a locally hosted supply chain simulation environment. The four-layer memory architecture would maintain state across planning horizons, while the reconstructibility layer would provide the audit trail required for supply chain compliance under EU regulation.

8.6 Scaling to 70B+ Models

The AMD Ryzen AI MAX+ 395 supports up to 128 GB LPDDR5X in dual-channel configuration. This headroom would enable 70B-parameter dense models or larger MoE architectures (e.g., Mixtral-8x22B) while maintaining the zero-cloud, offline-capable deployment model. The KV cache compression strategy (ADR-009, q4_0 quantisation) would scale proportionally, maintaining the 4:1 memory reduction ratio.

8.7 Formal Reconstructibility Certification

Gupta's verifiability-first framework [14] proposes cryptographic attestations for each agent decision point. Integrating lightweight hash chains into the Langfuse trace output — where each trace includes a cryptographic hash of the previous trace — would provide tamper-evidence guarantees suitable for financial services regulatory inspection.

9 Conclusion

This paper has presented a technical and regulatory case for sovereign local AI deployment as the structurally correct architecture for regulated European industries. The argument rests on five pillars:

1. **Regulatory compliance:** GDPR Article 44 and EU AI Act Article 25 create compliance obligations that commercial API deployment cannot satisfy for regulated data. The EDPB's official analysis [1] identifies the self-developed/locally deployed model as the privacy-optimal configuration.

2. **Hardware viability:** AMD's unified memory architecture, characterised by Wahlgren et al. [6] and demonstrated in production, eliminates the PCIe bottleneck that historically made local LLM deployment impractical for large models. A 35B-parameter MoE model runs at 29.5 tokens/second on a 1.7 kg laptop.

3. **Persistent memory:** The four-layer memory architecture — drawing on the taxonomy in [3] and the MEMCOLLAB framework [8] — provides stateful, context-aware AI agents without cloud dependency.

4. **Graceful degradation:** The geometric memory approach demonstrated in [4] validates the architectural principle: a system that continues operating offline is, by definition, sovereign.

5. **Reconstructibility:** Self-hosted Langfuse [15] with OpenTelemetry-native tracing [16] provides the audit trail mandated by EU AI Act Article 12 [10] — transforming the sovereign stack from an engineering artefact into a verifiable governance record that a regulator can independently inspect.

The cost-per-token advantage of local deployment ($0.00 vs. $0.025 per interaction) is real but secondary. The compliance imperative is structural. For organisations operating under Swiss data protection law (revDSG) and EU GDPR — including banks, pharmaceutical companies, and public sector entities — sovereign local AI is not a preference. It is a legal requirement.

The sovereignty–reconstructibility distinction, first articulated in this context by Roche [9] and developed in Section 7, is an underappreciated architectural concern. Most current "sovereign AI" proposals address data residency without addressing decision auditability. As Mügge [21] observes, the question of *for whom* AI sovereignty operates is as consequential as the technical implementation. Reconstructibility ensures that sovereignty serves the regulated entity and its stakeholders — not merely the engineering team.

Future work will address multi-agent orchestration via LangGraph, supply chain digital twin construction using Google OR-Tools, and scaling to the 128 GB RAM configuration to enable 70B+ parameter models.

References

[1] I. Barberá, "AI Privacy Risks & Mitigations – Large Language Models (LLMs)," tech. rep., European Data Protection Board (EDPB), Support Pool of Experts Programme, March 2025. Document submitted February 2025, updated March 2025. https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf.

[2] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council – Artificial Intelligence Act." Official Journal of the European Union, August 2024. https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai.

[3] Multiple Authors, "Survey of Memory Systems for AI Agents," *SSRN*, 2025. Survey_of_Memory_SSRN.pdf.

[4] I. Likov, "Graceful Degradation in Autonomous Agents: SAGE Memory-Augmented Drone Navigation Without Language Model Dependency," tech. rep., Independent, March 2026. sage_drone_paper_final.pdf. Code: https://github.com/Ivelin2022/sage-drone.

[5] lhl, "Strix Halo (Ryzen AI MAX+ 395) LLM Benchmark Results." Level1Techs Forums / GitHub, July 2025. https://github.com/lhl/strix-halo-testing.

[6] J. Wahlgren *et al.*, "Dissecting CPU-GPU Unified Physical Memory on AMD MI300A APUs," *arXiv preprint*, vol. arXiv:2508.12743, August 2025.

[7] S. Tandon, L. Grinberg, G.-T. Bercea, C. Bertolli, M. Olesen, S. Bna, and N. Malaya, "Porting HPC Applications to AMD Instinct™ MI300A Using Unified Memory and OpenMP," in *ISC High Performance 2024 Research Paper Proceedings (39th International Conference)*, pp. 1–9, Prometeus GmbH, 2024.

[8] Multiple Authors, "MEMCOLLAB: Contrastive Memory Distillation Between Language Models," *arXiv preprint*, vol. arXiv:2603.23234, March 2026.

[9] I. Roche, "Personal communication: Sovereignty, reconstructibility, and the governance gap in local AI architectures," April 2026. Dr. Ivan Roche, Non-Executive Director, AI & Technology Governance. Series of LinkedIn discussions (March–April 2026) identifying reconstructibility as a distinct architectural layer above sovereignty, motivating the Langfuse integration and the three-layer trace capture design presented in Section 8.

[10] European Parliament and Council of the European Union, "EU AI Act, Article 12: Record-Keeping." Official Journal of the European Union, 2024. Mandates automatic logging for high-risk AI systems. https://artificialintelligenceact.eu/article/12/.

[11] Bank for International Settlements, Financial Stability Institute, "Managing Explanations: How Regulators Can Address AI Explainability," Tech. Rep. Occasional Paper No. 24, BIS FSI, September 2025.

[12] L. Dong, Q. Lu, and L. Zhu, "AgentOps: Enabling Observability of LLM Agents," *arXiv preprint*, vol. arXiv:2411.05285, November 2024.

[13] B. Rombaut *et al.*, "Watson: A Cognitive Observability Framework for the Reasoning of LLM-Powered Agents," in *Proc. ASE 2025*, 2025. arXiv:2411.03455.

[14] A. Gupta, "Verifiability-First Agents: Provable Observability and Lightweight Audit Agents for Controlling Autonomous LLM Systems," *arXiv preprint*, vol. arXiv:2512.17259, December 2025.

[15] Langfuse GmbH, "Langfuse: Open Source LLM Engineering Platform." GitHub, 2023. Self-hostable. OpenTelemetry-native since v4. https://github.com/langfuse/langfuse.

[16] OpenTelemetry Authors, "Semantic Conventions for Generative AI Systems." OpenTelemetry Specification, 2024. Status: Development. Defines `gen_ai.*` attributes. https://opentelemetry.io/docs/specs/semconv/gen-ai/.

[17] V. Gundu, "Methodological Foundations of AI Observability for Enterprise LLM Applications," *International Journal of Engineering and Computer Science*, vol. 14, no. 10, 2025.

[18] V. Antonopoulos, "Personal communication: Zero-retention API contracts and model provenance policy for regulated industries," April 2026. Deputy Head of Technology (IT Operations, Infrastructure, Security & Risk). LinkedIn discussion on sovereign AI deployment, April 2026.

[19] A. Ahuja, "Personal communication: Local models for enterprise-grade compliance and control," April 2026. Lead Technical Program Manager, Cloud and AI-led Transformation in Financial Services. LinkedIn discussion on sovereign AI deployment, April 2026.

[20] A. Hoff, "Personal communication: Model provenance and European sovereignty in local AI deployments," April 2026. LinkedIn discussion on sovereign AI deployment, April 2026.

[21] D. Mügge, "EU AI sovereignty: for whom, to what end, and to whose benefit?," *Journal of European Public Policy*, vol. 31, no. 8, pp. 2200–2225, 2024.

A Sequence Diagrams: Temporal Proof of Reconstructibility

The following sequence diagrams provide the temporal evidence that complements the structural C4 architecture. Together they demonstrate that every information state governing an AI decision is captured in a form that can be independently reconstructed.

A.1 Query Flow: Agent to Inference Engine via Memory Layer

Figure 2 shows the complete request lifecycle from an agent (OpenCode) through the memory server to the inference engine and back, with every Langfuse trace point annotated.

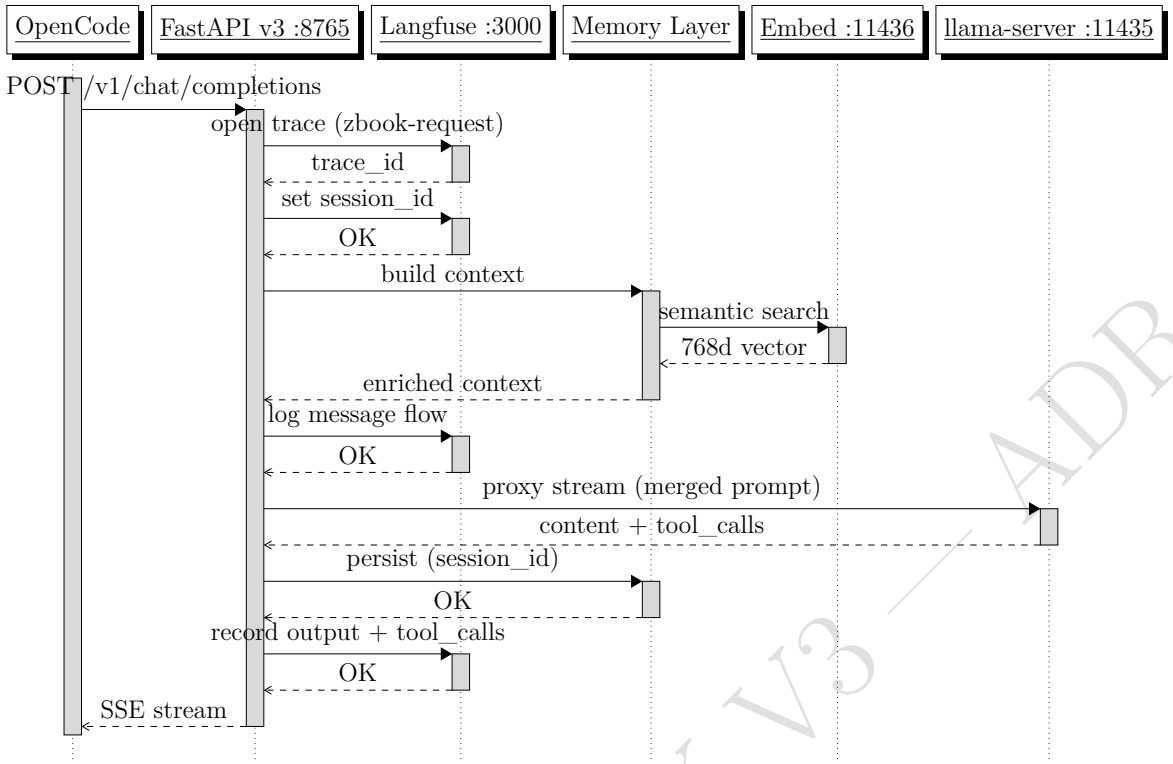


Figure 2: End-to-end observability flow. The memory server opens a Langfuse trace, sets the session identifier, builds context from four memory layers (episodic, procedural, conversational, semantic — the semantic layer calls the dedicated embedding server on port 11436 per ADR-013), logs the incoming message flow, proxies to llama-server, and records the output including tool calls and finish reason. Each step is individually traced and queryable.

A.2 Session Correlation: Multi-Request Tool-Use Loop

Figure 3 shows how multiple sequential requests from the same agent turn are correlated via the deterministic session identifier. This is the temporal proof that the full reasoning chain — across tool invocations and intermediate steps — can be reconstructed.

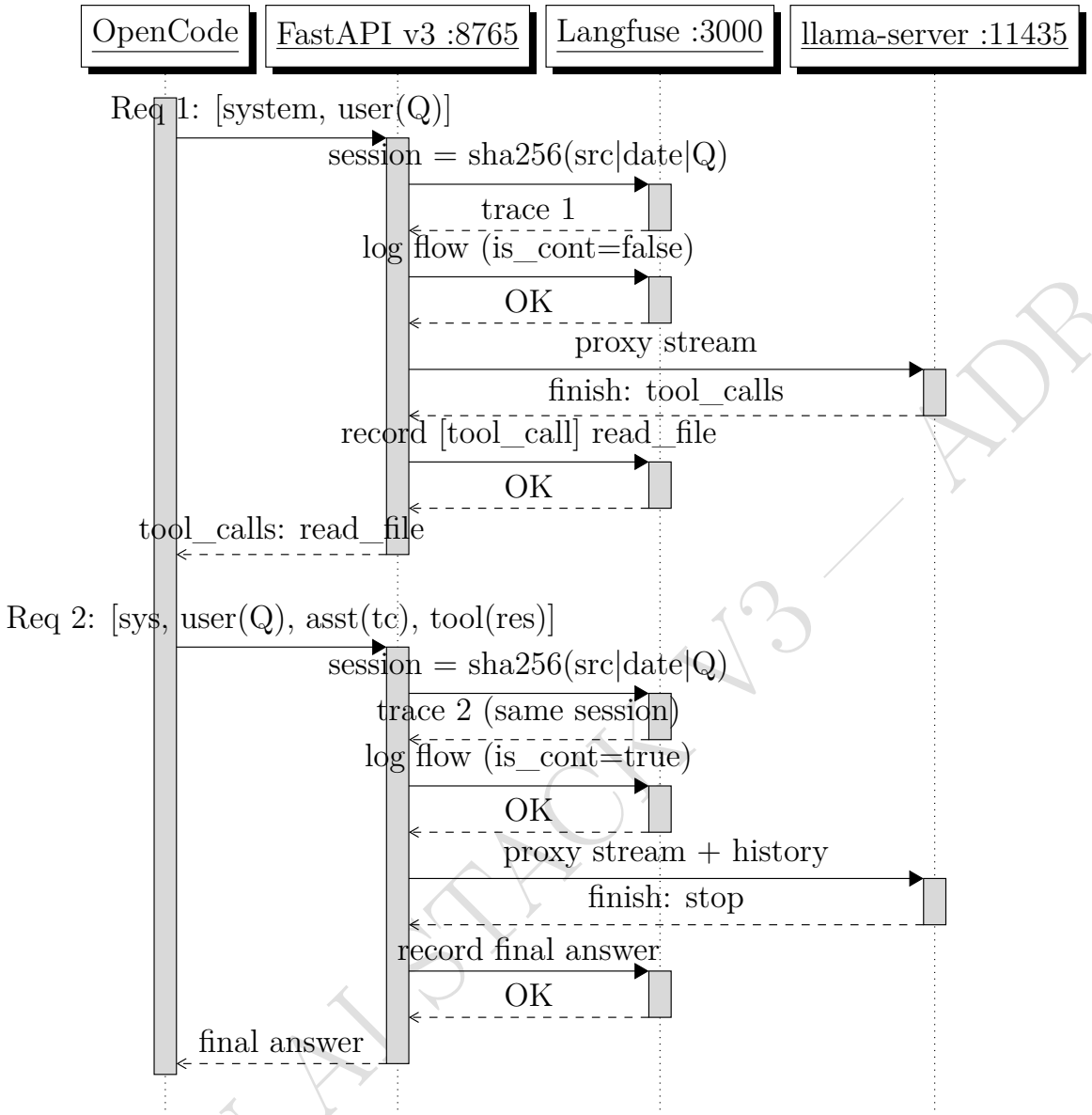


Figure 3: Multi-request tool-use loop with session correlation. Request 1 triggers a tool call (`finish_reason: tool_calls`); the agent executes the tool locally, then sends Request 2 with the tool result appended. Both requests produce the same `session_id` because the first user message (Q) is invariant across the loop. In Langfuse, both traces appear under one session, enabling full chain reconstruction. The `is_continuation` flag distinguishes fresh questions from mid-loop requests.

A.3 Langfuse Trace Tree Structure

Table 4 documents the complete span hierarchy recorded per request. Each span is individually queryable in the self-hosted Langfuse instance.

Table 4: Langfuse trace tree per request to `/v1/chat/completions`

Level	Span Name	Type	Source Function
0	zbook-request	Trace	_build_memory_context()
1	memory-episodic-load	Span	EpisodicLoader.load()
1	memory-procedural-load	Span	ProceduralLoader.load()
1	memory-conversational-load	Span	SessionMemory._load_from_sqlite()
1	memory-semantic-search	Span	ZBookVectorStore.search()
1	llm-generation	Generation	_proxy_stream() / httpx POST
1	persist-interaction	Span	_persist()

Trace metadata: sessionId, input (structured message flow), output (answer + tool call summaries), finish_reason, has_tool_calls, n_messages, is_continuation, prompt_tokens, completion_tokens.

B C4 Architecture Diagrams

The following C4 diagrams — generated from the Structurizr DSL workspace — provide the structural proof that complements the temporal evidence in Appendix A. Together they demonstrate that the sovereign stack is architecturally designed for both operational sovereignty and decision reconstructibility.

B.1 C4 Level 1: System Context

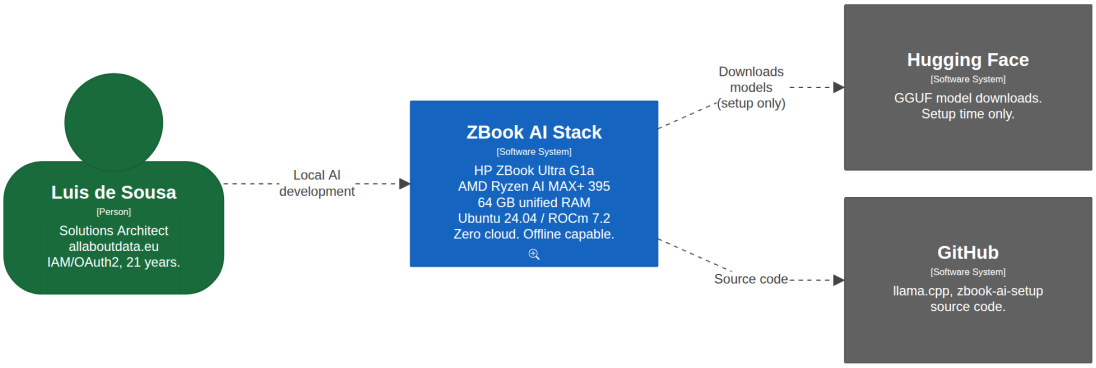


Figure 4: C4 Level 1 — System Context. The ZBook AI Stack operates as a self-contained system with no runtime cloud dependencies. External systems (Hugging Face, GitHub) are consulted only during initial setup. All inference, memory, and observability execute locally on the HP ZBook Ultra G1a with AMD Ryzen AI MAX+ 395.

B.2 C4 Level 2: Container Diagram

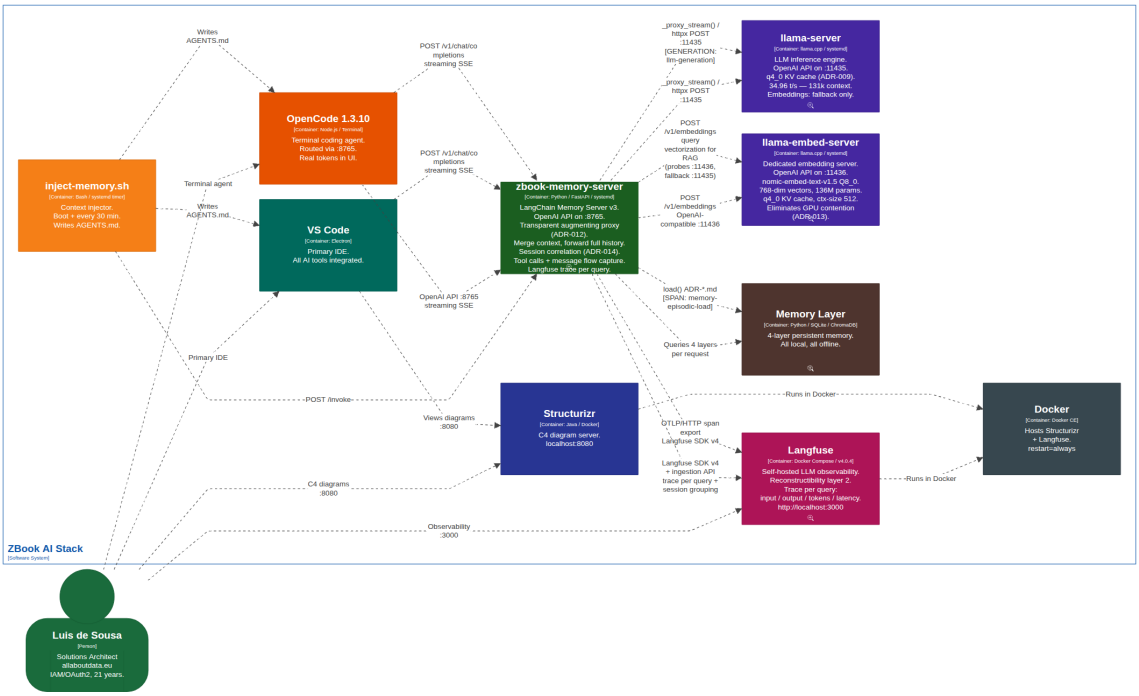


Figure 5: C4 Level 2 — Containers. The memory server (:8765) sits as a mandatory intermediary between all agents (OpenCode, VS Code extensions) and the inference engine (:11435). The dedicated embedding server (:11436) eliminates GPU contention (ADR-013). Langfuse provides self-hosted observability. All arrows represent local communication — no network egress.

B.3 C4 Level 3: Memory Server Components

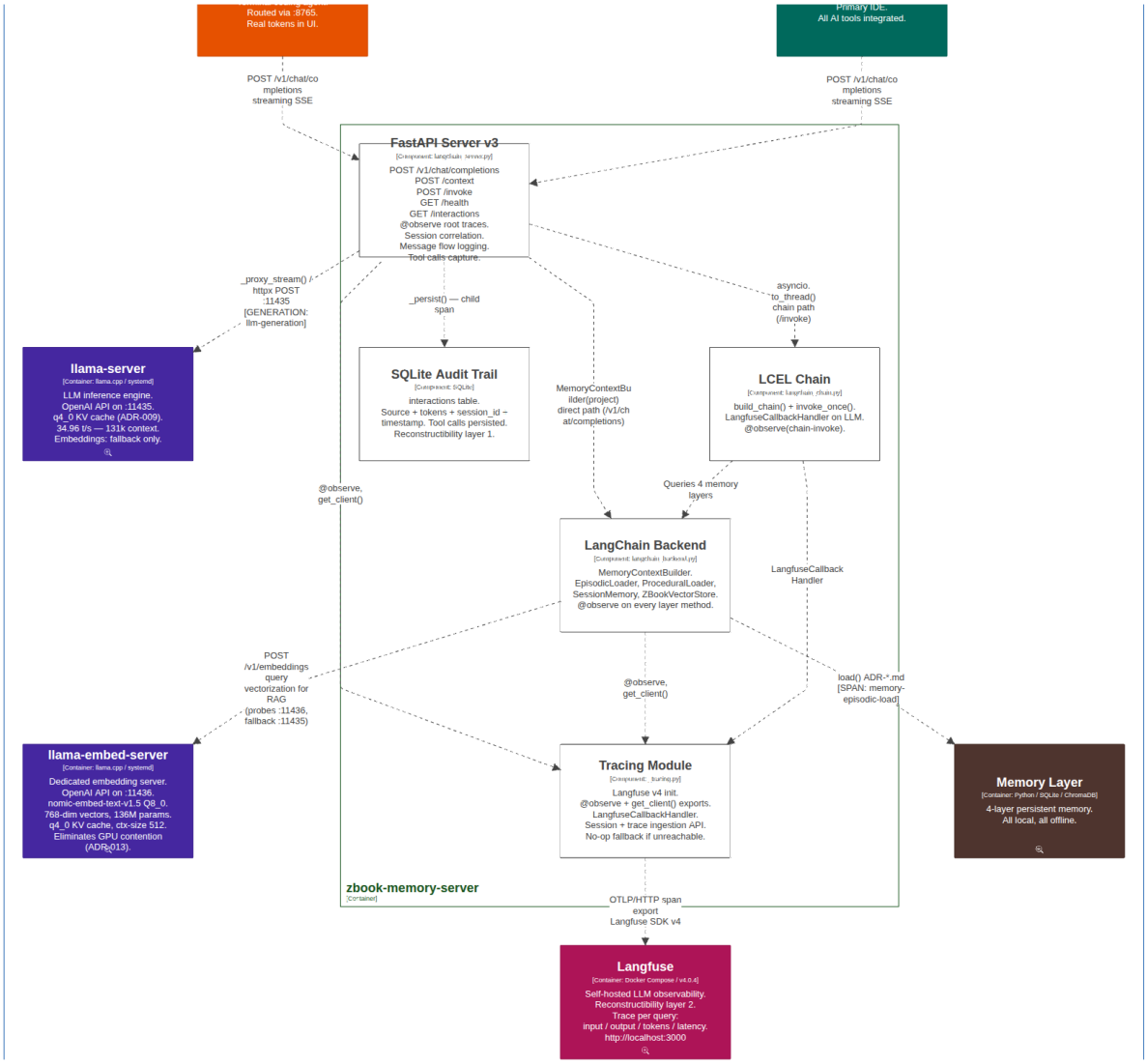
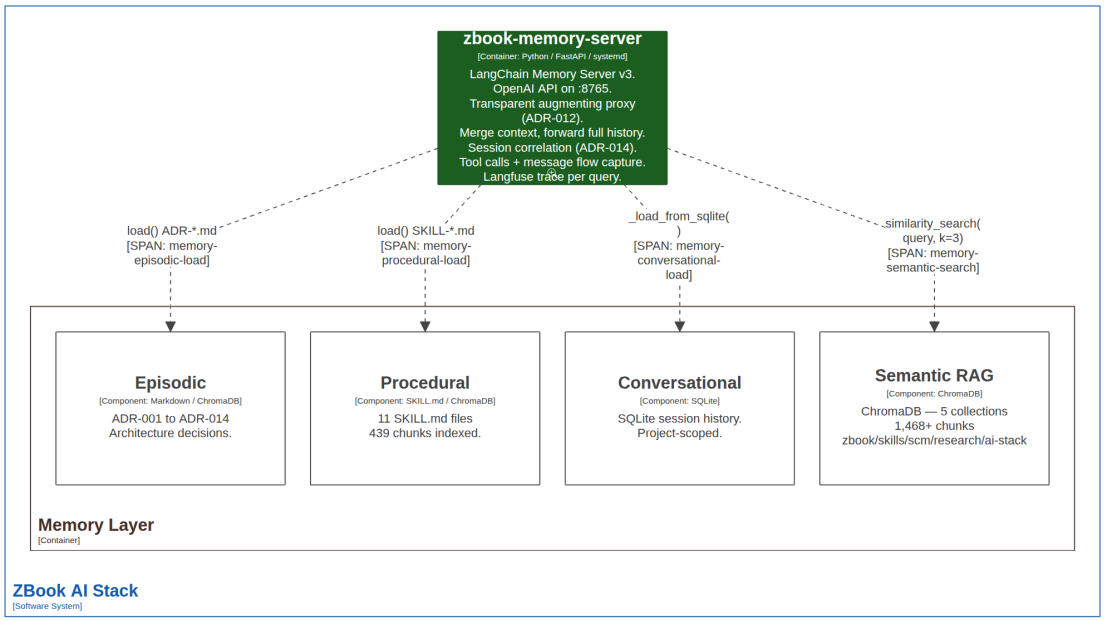


Figure 6: C4 Level 3 — Memory Server v3 component decomposition. The FastAPI Server v3 (langchain_server.py) orchestrates the request lifecycle: the LangChain Backend queries four memory layers, the LCEL Chain handles the /invoke path, the Tracing Module exports spans to Langfuse, and the SQLite Audit Trail persists every interaction with session identifiers for reconstructibility (ADR-014).

B.4 C4 Level 3: Memory Layer Components



ZBook AI Stack — C4 Level 3: Memory Layer
vendredi 3 avril 2026 à 11:19 heure d'été d'Europe centrale

Figure 7: C4 Level 3 — Memory Layer decomposition. Four complementary storage mechanisms: Episodic (ADR-001 to ADR-014 in Markdown, indexed into ChromaDB), Procedural (11 SKILL.md files, 439 chunks), Conversational (SQLite session history, project-scoped), and Semantic RAG (ChromaDB with 5 collections and 1,468+ chunks across private documents).