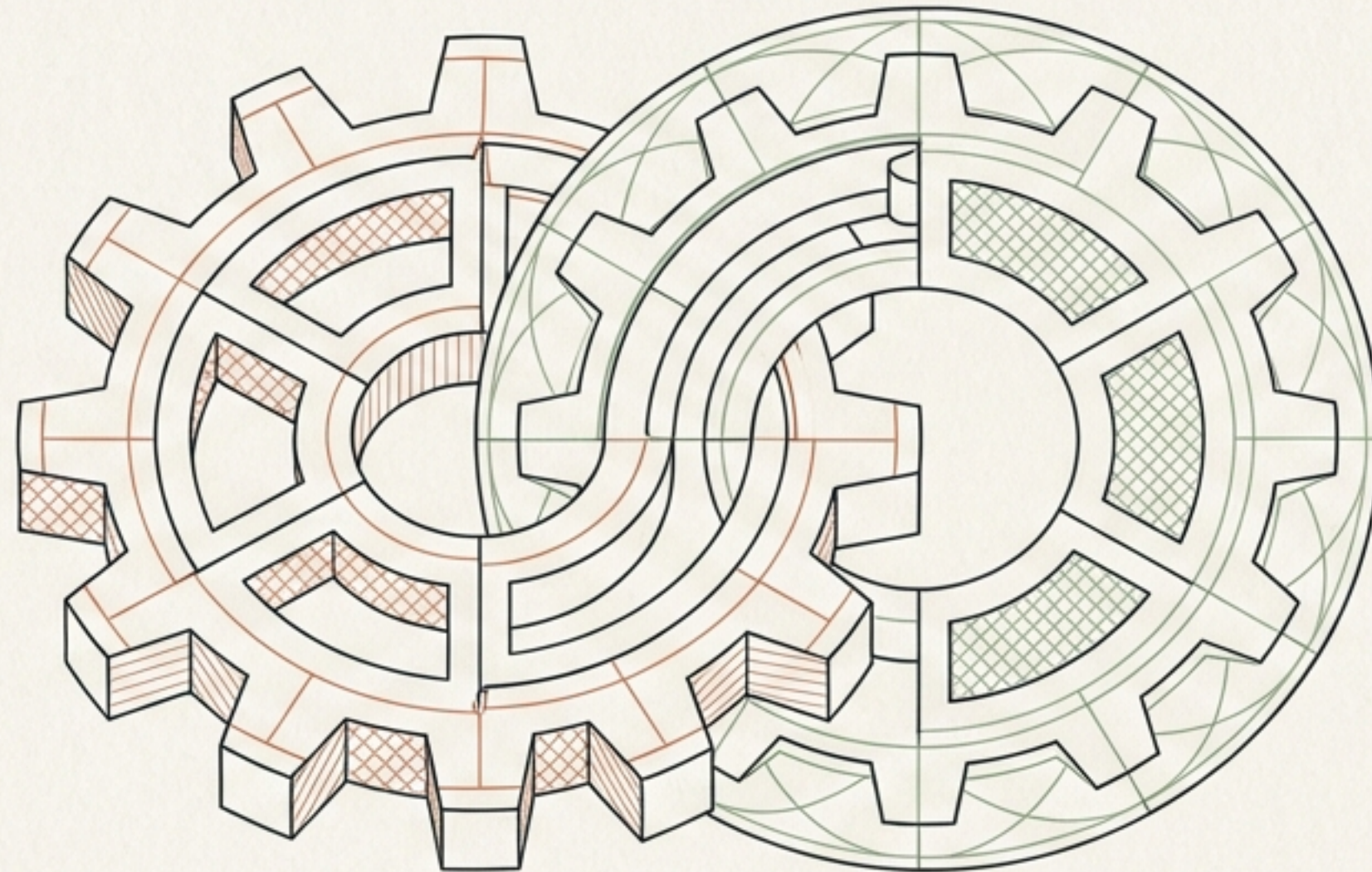


Ona & Ona Sphere

Your AI Squad. Your Mission. Safe for One—or Many.



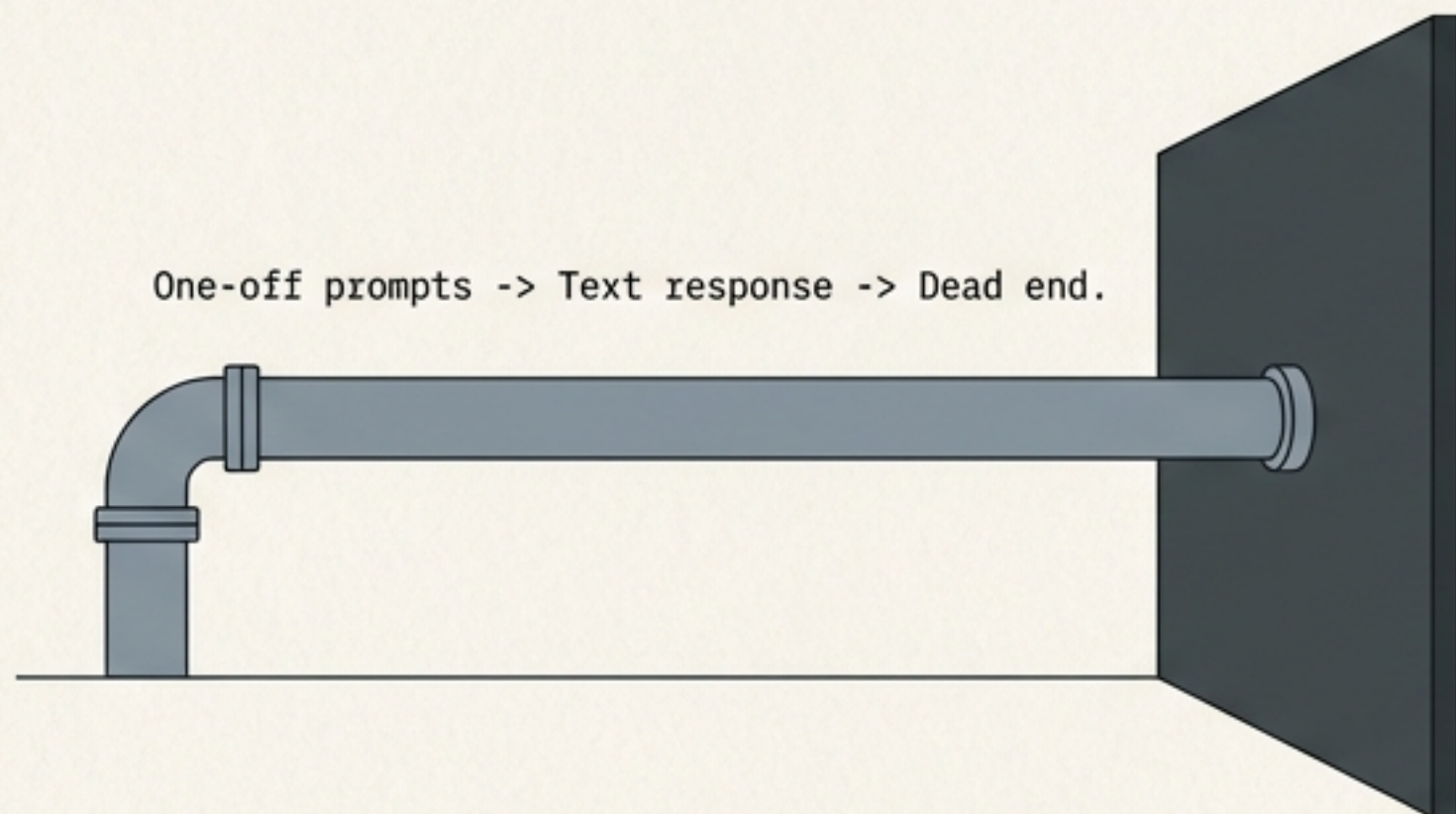
Status: System Operational
Architecture: Execution-Focused

Terminal_V.04

The Dead End of Chat-Only AI vs. The Mission Execution Paradigm

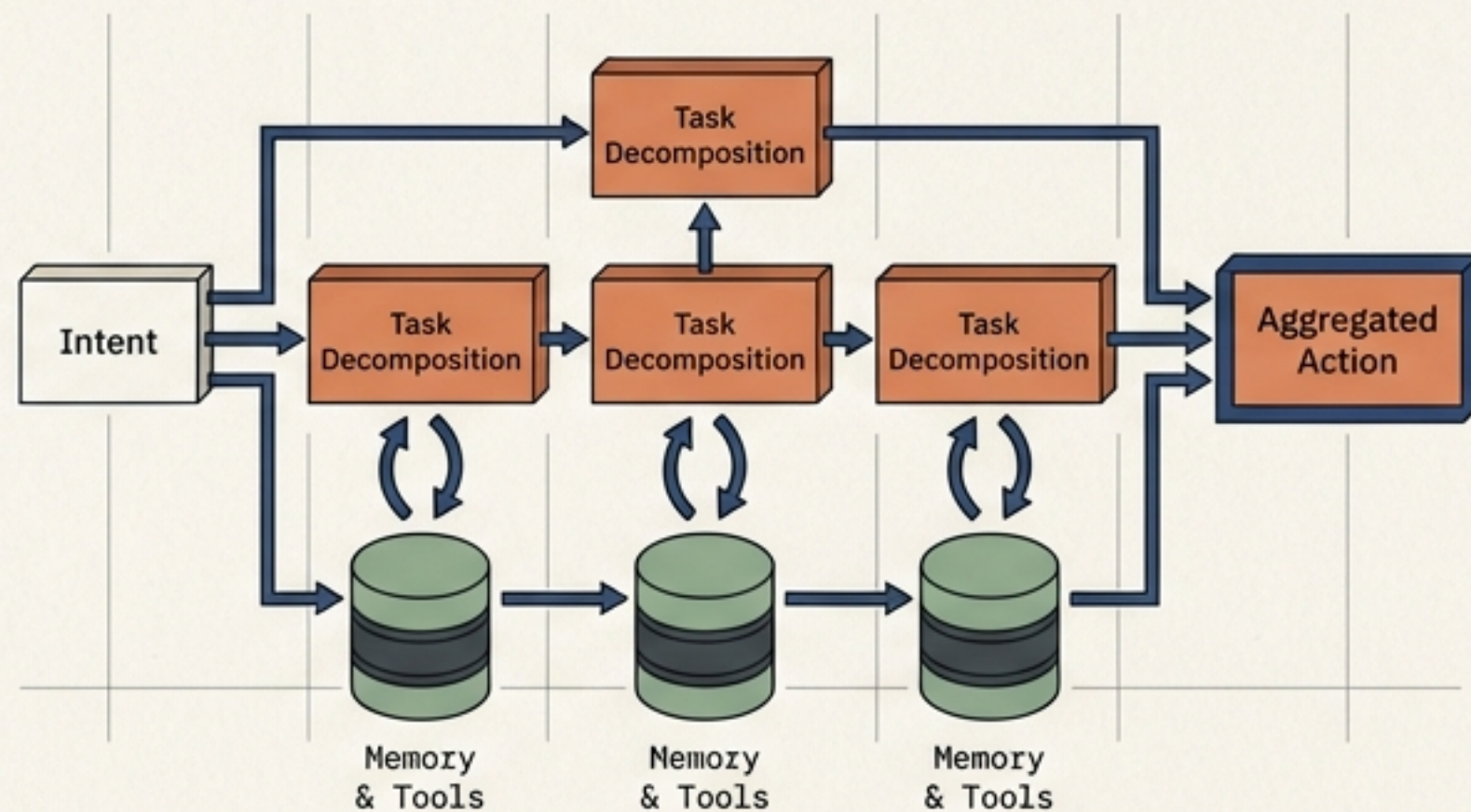
True productivity requires an operating system, not a conversation.

The Prompt Trap



Standard AI waits for your next instruction.

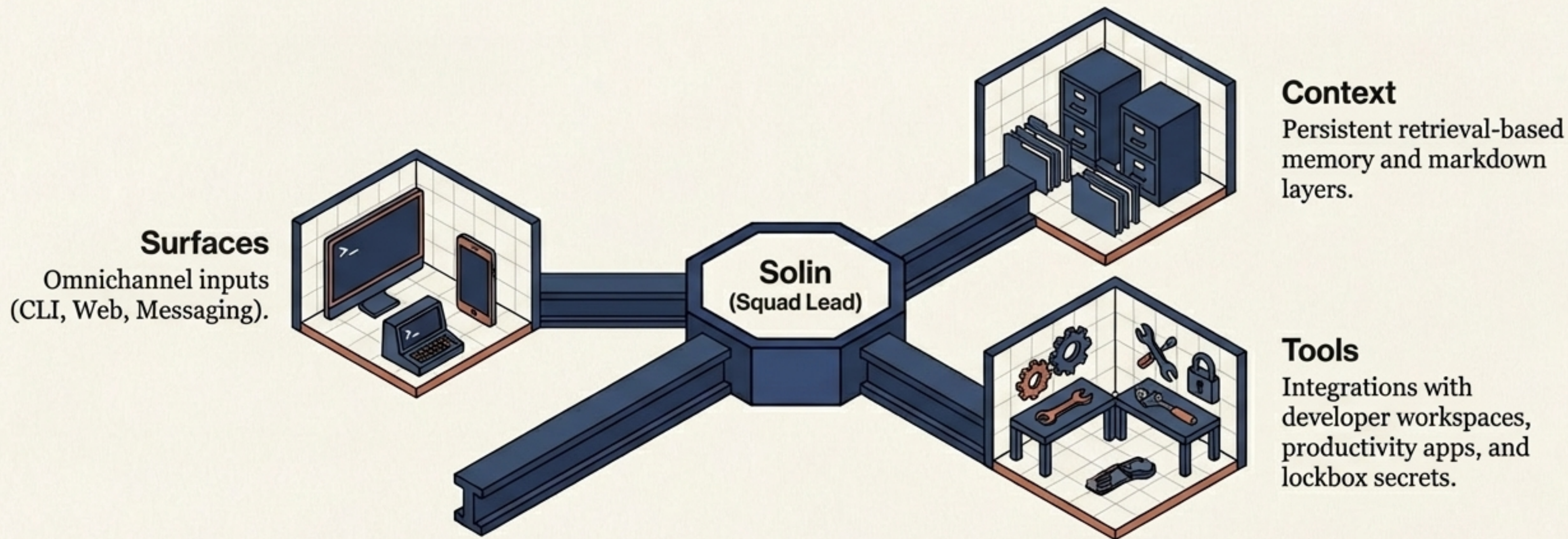
The Ona Mission Pipeline



Ona takes a mission, breaks it into tasks, coordinates specialists, and delivers a completed outcome.

Layer 01: The Ona Execution Engine

Your private, local-first intelligence substrate.



Surfaces

Omnichannel inputs (CLI, Web, Messaging).

Solin
(Squad Lead)

Context

Persistent retrieval-based memory and markdown layers.

Tools

Integrations with developer workspaces, productivity apps, and lockbox secrets.

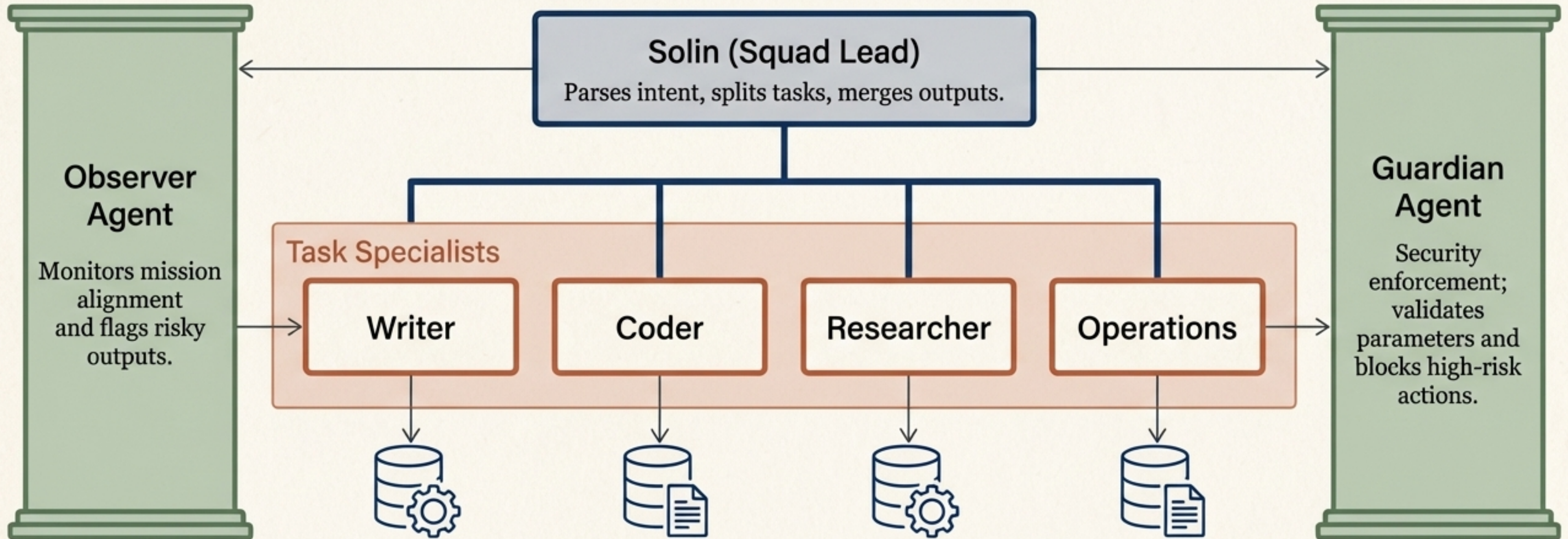
Ona is your private multi-agent execution layer. It manages personal context and private files without exposing them.

Solin is your personal lead agent within Ona. You speak to Solin; Solin coordinates the system.

Built for local-first privacy.
You own the runtime.

The Multi-Agent Squad Organisation

Solin never works alone. Missions are delegated to specialised, role-bound workers and overseers.



Specialisation

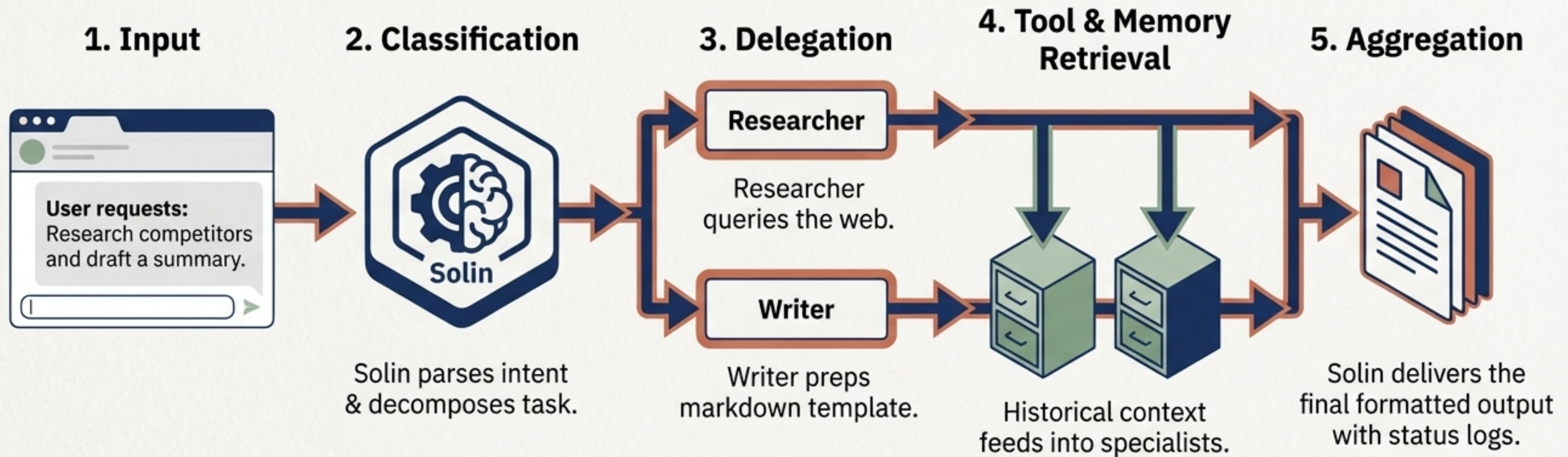
Complex work requires focused system prompts. Solin delegates to personas tailored for deep research or coding.

Oversight

Guardian and Observer agents do not execute tasks; they act as parallel safety rails to prevent hallucination and misuse.

The Mission Pipeline in Action

How an ambiguous request becomes a structured, multi-step execution.



Missions differ from chat because they are **persistent, stateful jobs**. Ona supports quick replies for fast latency, but automatically **scales** to **deep, multi-step orchestration** for complex jobs.

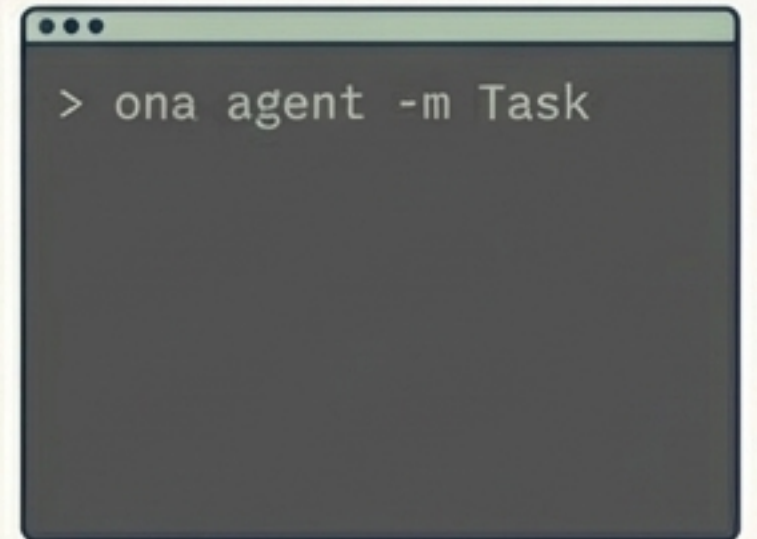
Omnichannel Control Surfaces

One mission state, accessible across all your daily workflows.



Web/Desktop UI

Mission Control dashboard, Jobs list, Approvals queue. Start a deep research mission on your desktop.



CLI (Terminal)

Monitor progress via CLI while coding.

Ona
Mission
State



IDE Integration (/ide)

Code editor interface with developer-agent terminal bridge.

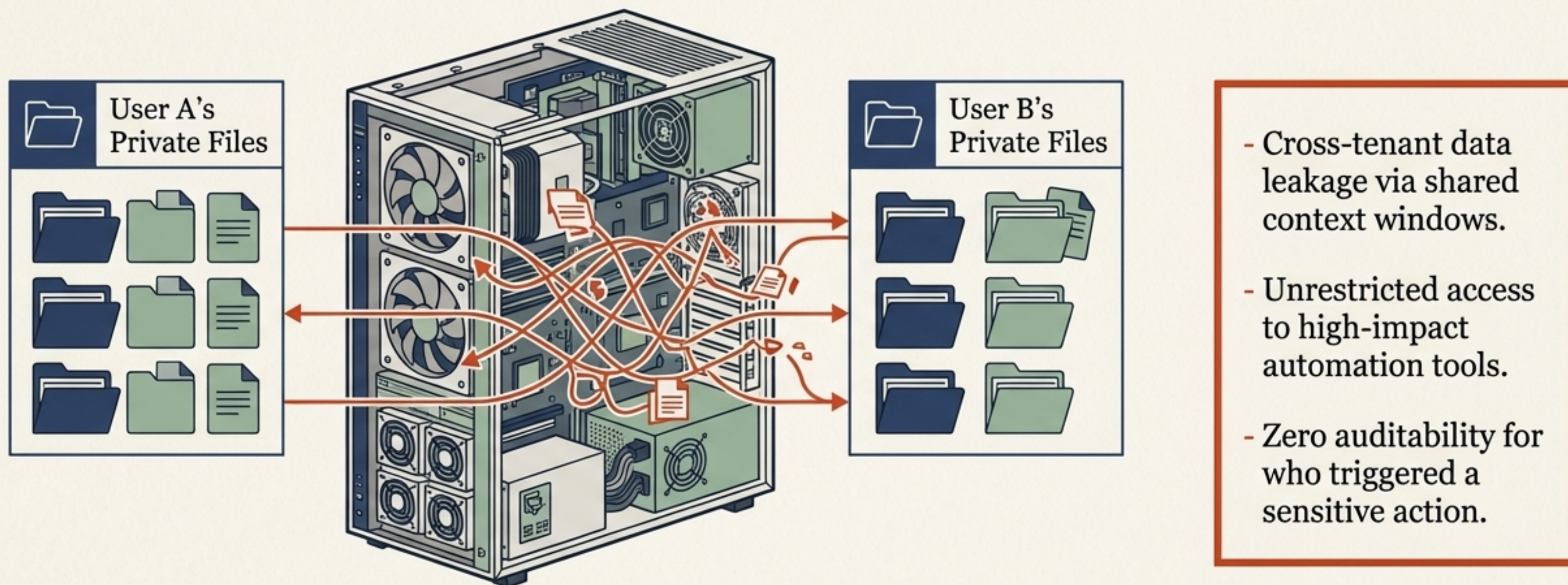
Messaging Messaging Channels

Telegram, WhatsApp, and Discord. Approve a sensitive external email via a quick reply on your phone.



The Shared Server Dilemma

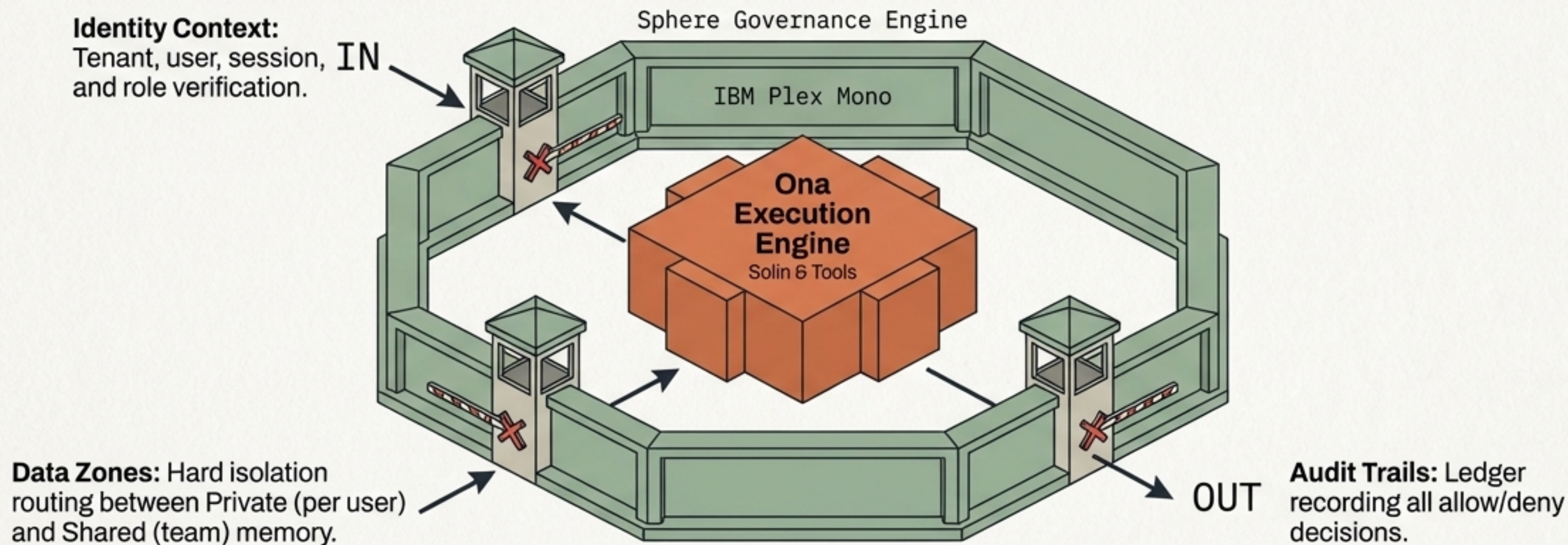
Multi-user AI introduces critical risks: shared intelligence cannot mean shared private data.



As soon as a family, team, or company shares a single AI runtime, local-first privacy breaks down. You need strict boundaries. You need governance.

Layer 02: Ona Sphere

The definitive trust boundary and control plane for shared-server deployments.




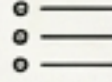
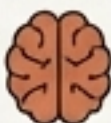

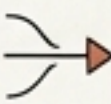

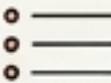

– **Sphere evaluates shared-server policy boundaries** before execution is permitted.

– Operations without valid identity scope are blocked by default.

– **Runs standalone:** If Sphere is down, Ona's private local workflows continue uninterrupted.

System Separation: Execution vs. Governance

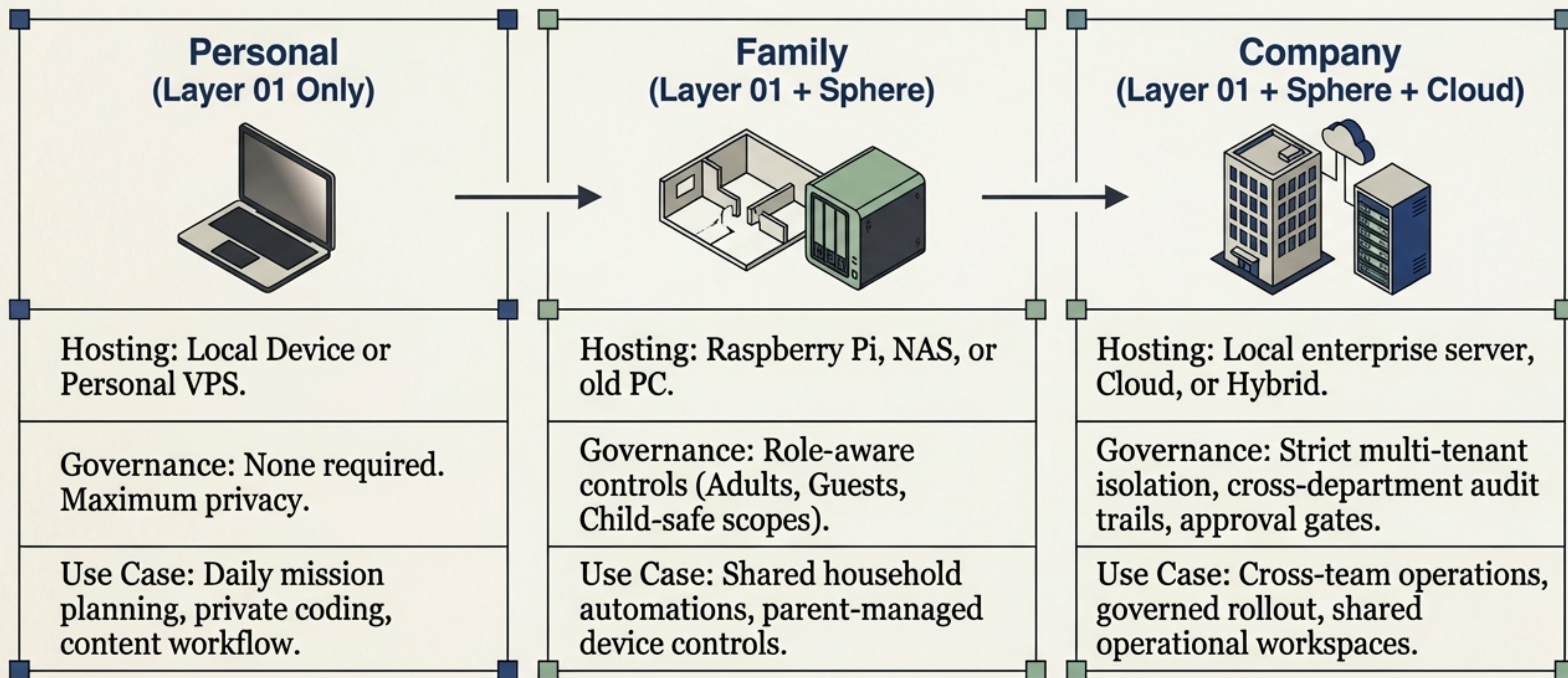
Ona gets the mission done. Sphere ensures it is done safely for many.

[ONA]	[ONA SPHERE]
<ul style="list-style-type: none">Function: The Brain & Workflow. 	<ul style="list-style-type: none">Function: The Rules & Boundaries. 
<ul style="list-style-type: none">Core Entities: Solin, Specialist Agents, Tools, Memory. 	<ul style="list-style-type: none">Core Entities: Gateway Agents, Lockbox, Guardian, Audit Logs. 
<ul style="list-style-type: none">Primary Goal: Task decomposition and outcome generation. 	<ul style="list-style-type: none">Primary Goal: Identity verification, policy enforcement, role gating. 
<ul style="list-style-type: none">Scope: Private, personal layer. Runs on its own. 	<ul style="list-style-type: none">Scope: Multi-user, shared-server infrastructure. 

Start with Ona for your personal workflows. Enable Sphere when you invite others to the server—without needing to rebuild your flow.

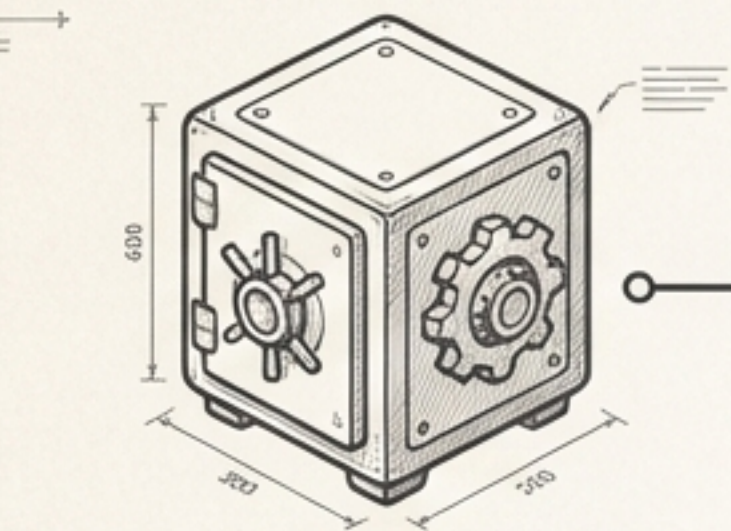
The Scaling Architecture

From private desktop intelligence to company-wide AI infrastructure.



Model Hosting & Sovereign Routing

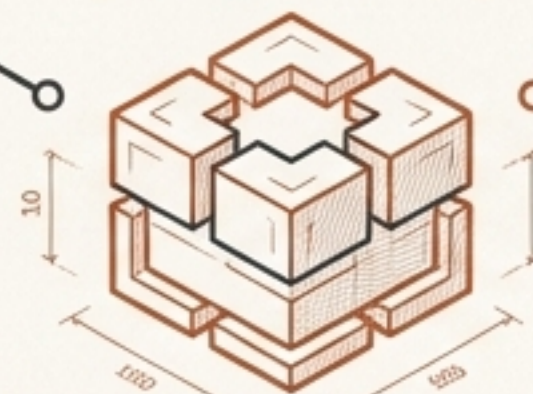
Complete control over privacy, cost, and reasoning depth.



Self-Hosted Only
(llama.cpp, Ollama, vLLM)

- Benefit: Maximum local privacy, zero per-token cost.

[Ona Core]



Hybrid (The Recommended Standard)

- Local routing for core tasks, deliberate escalation gate to cloud models for heavy lifting.



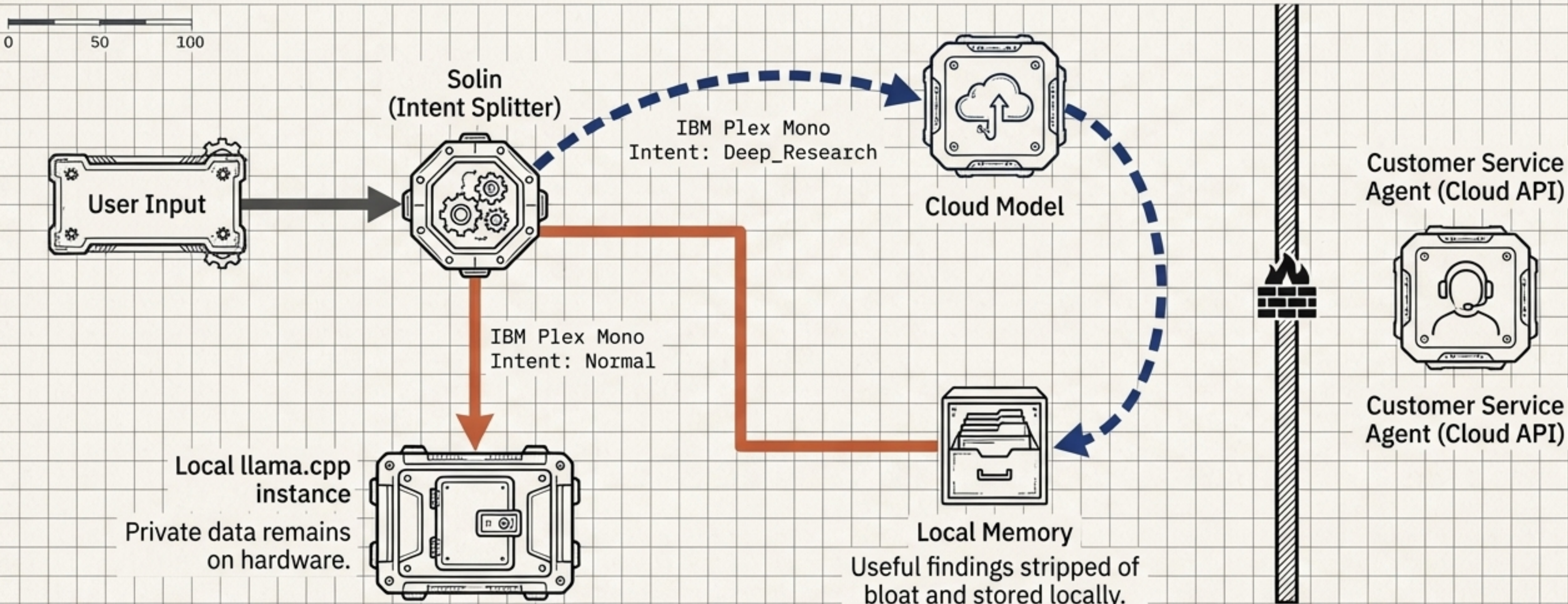
Cloud API
(OpenAI, Anthropic, Gemini)

- Benefit: Low maintenance, massive reasoning capacity.

Ona is backend-agnostic. Mix and match Docker-hosted local services with cloud endpoints. Keep sensitive operational context local by default.

Deep Research Routing & Dual-Bot Separation

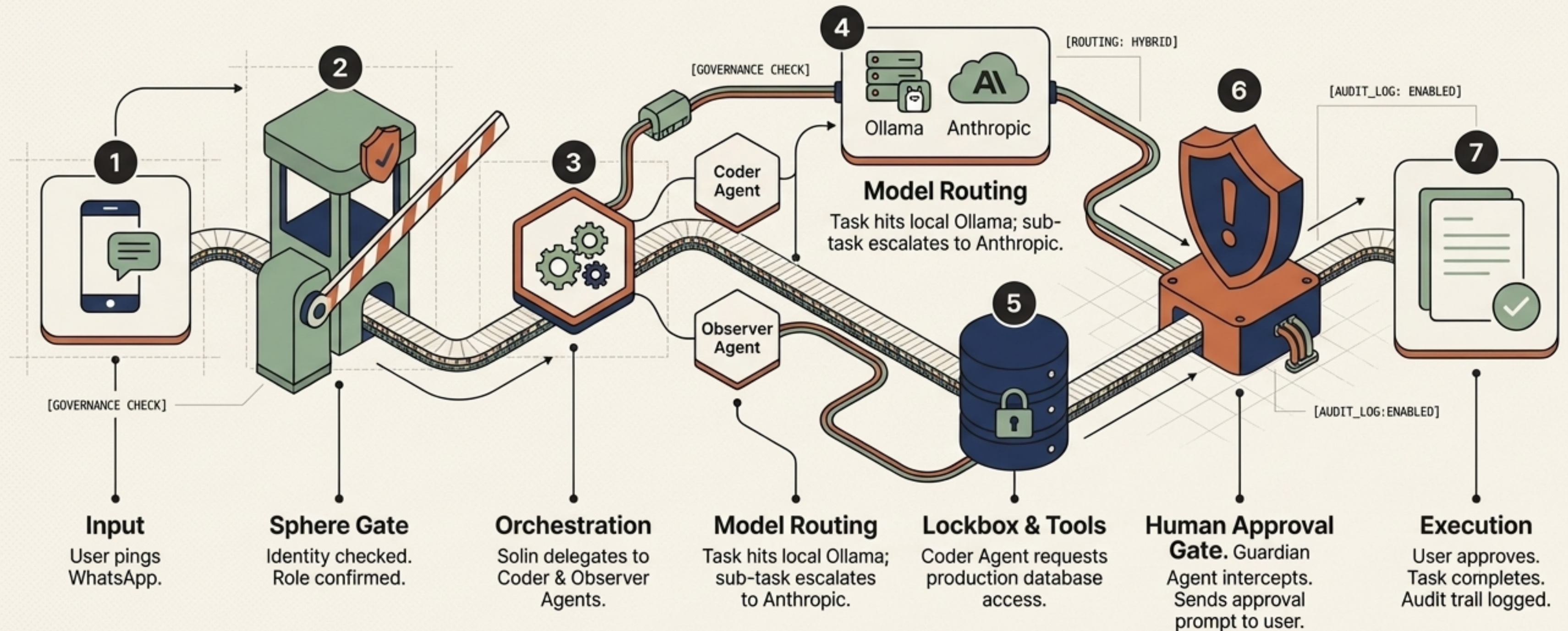
Isolating core workflows from intensive tasks and customer-facing surfaces.



Rule: Dual Bot Separation. Solin and Customer Service Assistants must run on distinct LLM services to guarantee reliability and security isolation.

The Universal Intelligence Substrate

The complete end-to-end lifecycle of a governed mission. in Georgia.



Initialise Your System

Scale from personal workflows to secure shared infrastructure today.

```
> ona start --sphere
```

Documentation & Architecture

Access full setup guides, API parity matrices, and UI capabilities at [/REPOSITORY](#) and [/SPHERE_DOCS](#).

Community Hub

View deployment patterns and contribute at [onapublic](#).

Operations Contact

For enterprise scaling and support:

Discord · WhyNot Productions
Zerwiz (Josef Lindbom) — josef.lindbom@gmail.com