SLOGR

AI Agent Mission Control
Architecture, Design & Vision

Version 0.1.0 · March 2026 · Public Draft · MIT License

// TECHNICAL WHITEPAPER

github.com/slogrbot-stack/slogr

// TABLE OF CONTENTS

// 00

Abstract

Slogr is an open-source, self-hosted monitoring and command platform for AI agents. As AI agent deployments grow in complexity — spanning multiple frameworks, tasks, and tool calls — developers lack visibility into what their agents are doing in real-time.

Slogr solves this by providing a unified dashboard to monitor, debug, and communicate with any AI agent, regardless of the underlying framework. Built on Node.js with Socket.io for real-time communication and a Python SDK for agent integration, Slogr operates entirely on the user's own infrastructure.

```
// MISSION STATEMENT: Give every AI developer a cockpit — a real-time command center where they can see exactly what their agents are doing, intervene when needed, and ship with confidence.
```

// 01

The Problem

The rise of AI agent frameworks has made it dramatically easier to build autonomous AI systems. However, operating these agents in production remains a significant challenge.

1.1 OBSERVABILITY GAP — Most agent frameworks provide minimal built-in observability. Developers rely on print statements, log files, and ad-hoc debugging tools. When an agent fails, tracing the root cause is time-consuming and error-prone.

1.2 NO REAL-TIME INTERFACE — Existing monitoring solutions are largely post-hoc. There is no standard way to observe an agent's reasoning in real-time, send it commands mid-task, or adjust its behavior without restarting the process.

1.3 MULTI-FRAMEWORK FRAGMENTATION — Teams often use multiple agent frameworks. Each has its own logging format, callback system, and debugging interface. There is no unified layer across all of them.

1.4 COST BLINDNESS — AI agents consume API tokens at scale. Without per-agent cost tracking, teams have no visibility into which agents are burning resources and why.

// 02

The Solution

Slogr provides a real-time command center for AI agents. It consists of three components: a lightweight backend server, a Python SDK for agent integration, and a web dashboard for visualization and interaction.

SELF-HOSTED FIRST — Slogr is designed to run on your own server with your own API key. No SaaS dependencies, no data collection, no vendor lock-in. Every byte of data stays on your infrastructure.

// 03

Architecture

The system consists of three layers communicating via Socket.io and REST API. Python agents emit events to the Slogr server, which routes them to connected dashboard instances in real-time.

3.1 BACKEND LAYER — Node.js Express server with Socket.io handles agent registration, event routing, state management, and serves as a proxy for Anthropic API calls — ensuring API keys never appear in the browser.

3.2 AGENT SDK — Python SDK wraps any AI agent with minimal code changes. It emits structured events to the server via Socket.io, enabling real-time monitoring without modifying agent logic.

3.3 DASHBOARD — Single-page application built with vanilla JavaScript and HTML5 Canvas. Agents appear as starfighters, tasks as enemies. The dashboard has three panels: agent sidebar, game canvas, and agent chat.

// 04

Core Features

REAL-TIME MONITORING — Every event emitted by a connected agent appears on the dashboard within milliseconds. Task starts, tool calls, reasoning steps, errors — all visible as they happen.

DIRECT AGENT CHAT — Each agent has a dedicated chat channel. Dashboard users can send natural language commands and receive responses in real-time, powered by the Anthropic API via a backend proxy.

SHIELD HEALTH SYSTEM — Each agent has a shield health score (0-100). Shield decreases when tasks fail or errors occur, providing an instant visual indicator of agent reliability.

MULTI-AGENT FLEET — Multiple agents can connect simultaneously. The dashboard scales dynamically — new agents appear automatically in the sidebar and game canvas. No hard limit on

concurrent agents.

API KEY PROXY — All Anthropic API calls are routed through the backend server. The API key is stored in a server-side .env file and never exposed to the browser.

// 05

Framework Integrations

Slogr provides first-class integration with major agent frameworks through dedicated adapters.

LANGCHAIN — BaseCallbackHandler integration. Automatically tracks chain start/end, LLM calls, tool invocations, and errors with zero manual instrumentation.

CREWAI — Crew wrapper that tracks task delegation, agent handoffs, and final results across multi-agent crews.

AUTOGEN — Agent patcher that intercepts message exchange, function calls, and replies in AutoGen conversations.

CUSTOM — The base Python SDK supports any custom agent or framework with manual instrumentation via simple method calls.

// 06

Security Model

SELF-HOSTED ARCHITECTURE — All data stays on your own server. No telemetry is sent to Slogr or any third party. Agent events, task history, logs, and chat messages are private to your infrastructure.

API KEY MANAGEMENT — The Anthropic API key is stored in a server-side .env file. The frontend never has direct access. All AI API calls are proxied through the backend /api/chat endpoint.

AUTHENTICATION — In v0.1.0, authentication uses a simple API key (default: sk-slogr-demo). Production deployments should replace this with a strong random key in .env and place the server behind HTTPS.

// 07

Roadmap

```
v0.1.0 — Core monitoring, real-time chat, game visualization, Python SDK. [RELEASED]
```

```
v0.2.0 — SQLite persistence, full task history, per-agent cost tracking. [IN PROGRESS]
v0.3.0 — Authentication system, user accounts, team support.
v0.4.0 — LlamaIndex + Flowise integrations, plugin architecture.
v0.5.0 — Web3 wallet identity, on-chain agent activity proof.
v1.0.0 — Stable public API, optional cloud-hosted version, enterprise features.
```

// 08

Conclusion

As AI agents become more capable and more widely deployed, the need for robust observability tooling grows proportionally. Slogr represents a first step toward a universal command layer for AI agents — open, self-hosted, and framework-agnostic.

By combining real-time monitoring, direct communication, and a memorable game visualization, Slogr makes AI agent operations both productive and engaging. Better observability leads to better agents.

```
// JOIN THE REBELLION — Slogr is open source and community-driven. Star the repo, file issues, submit PRs. Help us build the cockpit every AI developer deserves.
```

github.com/slogrbot-stack/slogr · t.me/slogrAgent · x.com/slogrwork