



AgentLand Observability

Air Traffic Control for AI Coding Agents

```
-> connecting to agent_network...
[INFO] Agent_001 active
[INFO] Agent_001 active
-> monitoring resource usage...
-> trace: 78e23a11-aacc-bo2e-b33e-4076-b81c7098a3fc
-> logs: stream_open
-> logs: stream_open
)
```



You are flying blind.

Running multiple AI coding agents simultaneously means tab-switching, scrolling through endless output, and hoping nothing breaks.

```
> claude-code --verbose
INFO [28:45:12] Initializing model connection...
DEBUG [28:45:12] Context window adjusted...
PROCESS [28:45:13] Generating code snippet...
OUTPUT [28:45:13] function calculateMetric(data) {...}
OUTPUT [28:45:13] function calculateMetric()
OUTPUT
```



```
> gemini-cli run update_api.ts
[WARN 20:45:14] Deprecated API endpoint detected in configuration.
Please update to v3.2.0 for stability. Potential data
inconsistency imminent.
```



```
ERROR [28:45:15] Unhandled Exception: Connection Refused.
at /src/services/network.ts:102:1S. Stack trace:
...
at async main (/src/index.ts:45:5)
[CRITICAL] Agent process terminated unexpectedly.
```

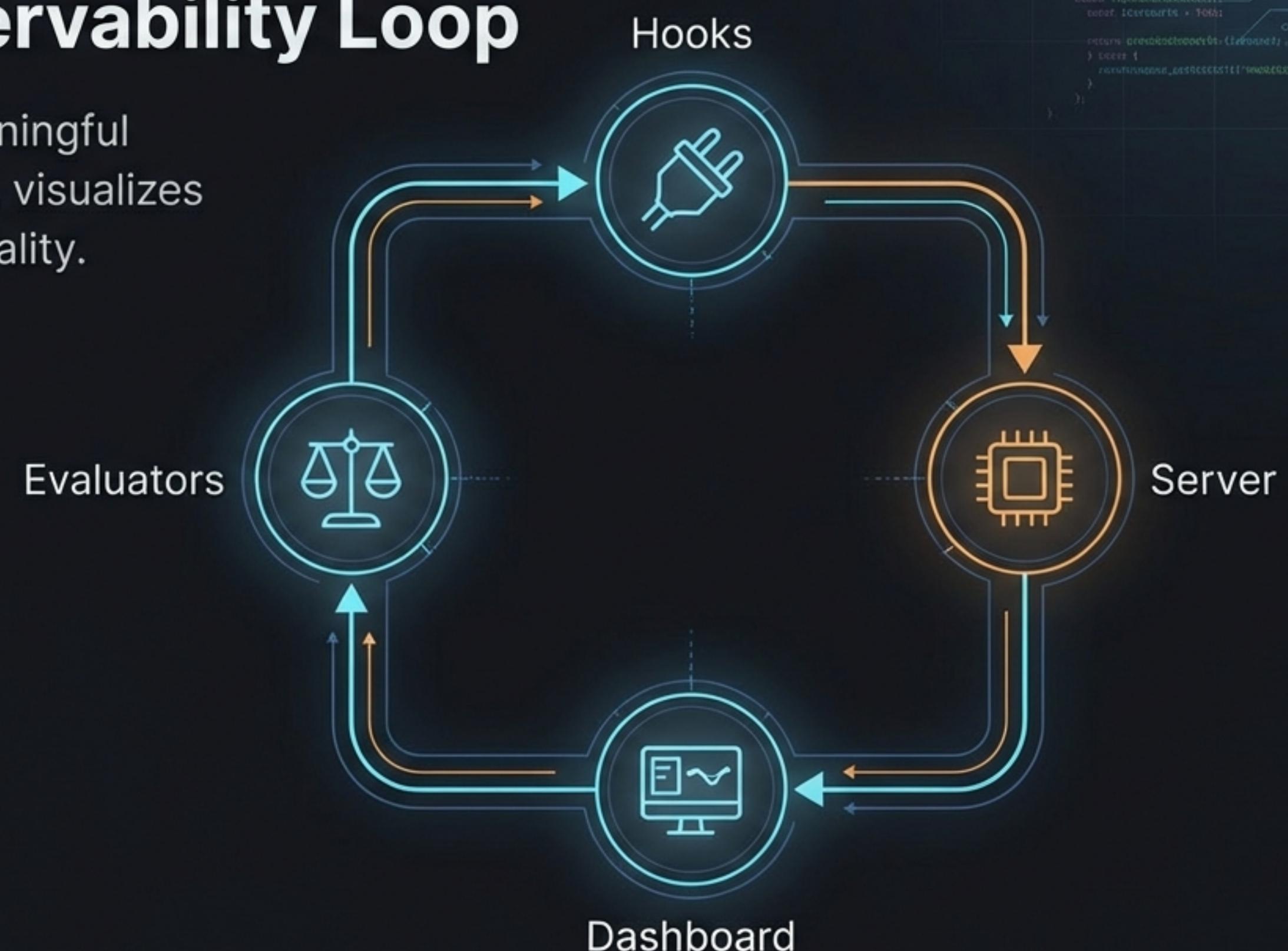


The 'Black Box' Problem:

You don't know what decisions are being made until it's too late.

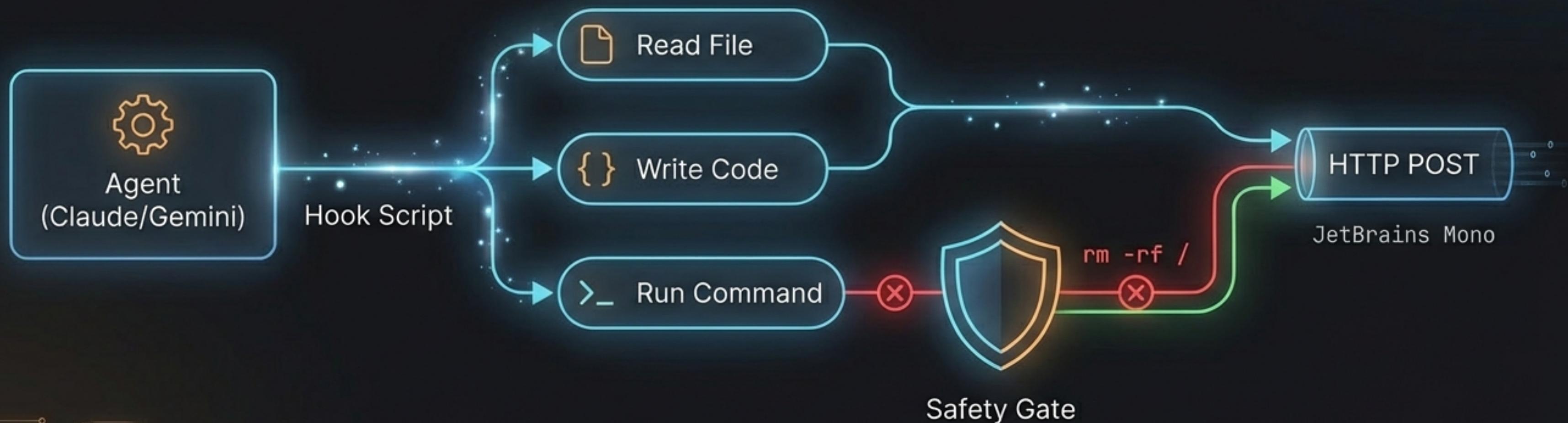
A Complete Observability Loop

The system captures every meaningful event, broadcasts it in real-time, visualizes the workflow, and judges the quality.



The Ears: Hooks & Safety

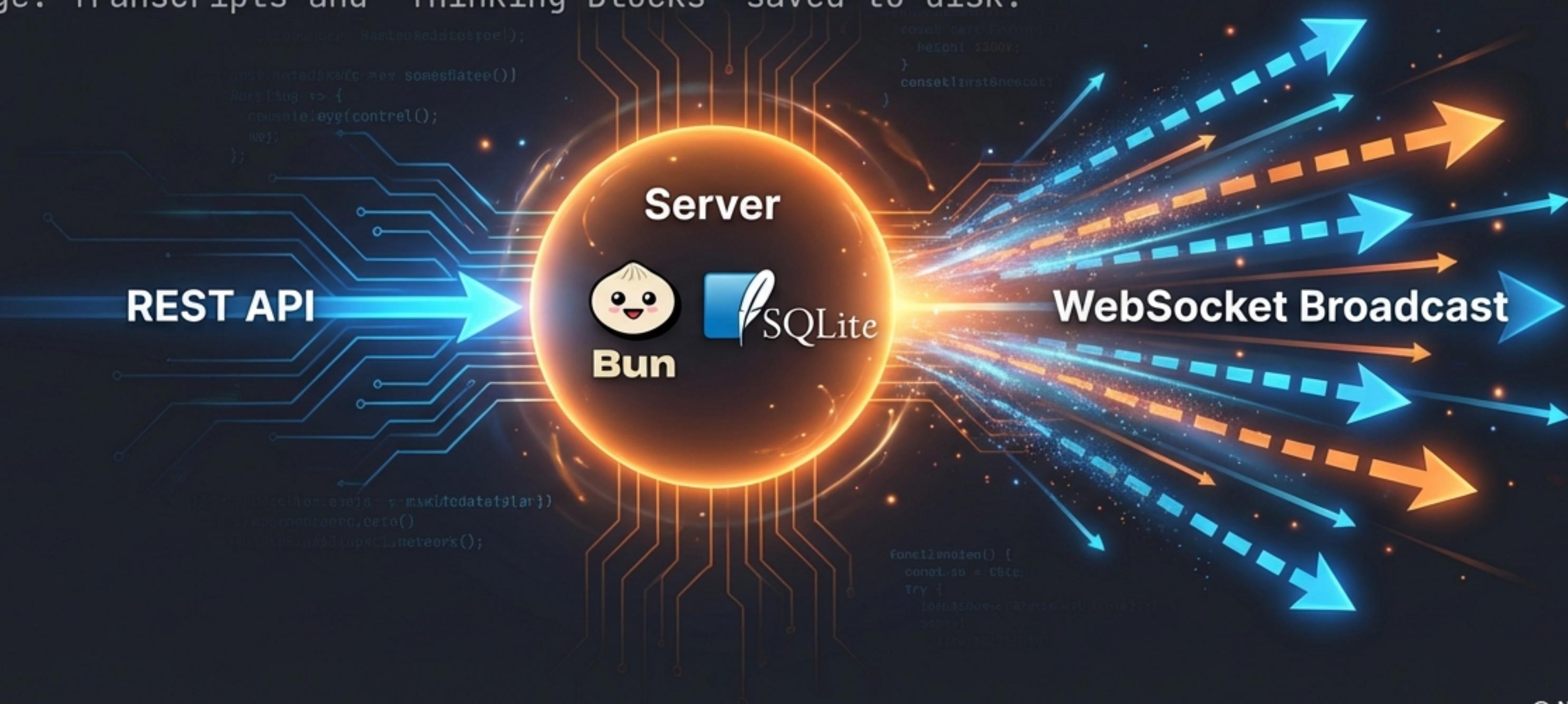
Python scripts plug directly into agent hooks.
They capture events and fail silently—if the
server is down, your agents keep working.



JetBrains Mono: Protocol: Lightweight HTTP POST delivery

The Brain: The Server

```
// In-Memory Speed: WAL mode for concurrent reads/writes.  
// Real-Time: Immediate distribution to clients.  
// Storage: Transcripts and 'Thinking Blocks' saved to disk.
```



The Eyes: Real-Time Visibility

Watch events stream instantly.

Identify agents by Project Name or Source App.

Human-in-the-loop: Respond to agent questions directly in the feed.

The image shows a screenshot of the AgentLand Live Dashboard. At the top, it says "AgentLand Live Dashboard". Below that, it says "Swim Lanes". There are two lanes visible: one for "claude-code:a3f8" which is "Active" and another for "gemini-cli" which is "Idle". The "claude-code:a3f8" lane shows a history of events: [10:02:35] Modified src/app.tsx, [10:02:45] Updated components/Header.tsx, [10:02:50] Refactored utils/api.ts, and [10:03:05] Committed changes to feature/auth. A purple callout box from the "gemini-cli" lane asks, "I need to confirm: Should I proceed with the deployment to staging? [Yes/No]". A "Reply" button is at the bottom right of this callout. The background has a futuristic circuit board pattern.

Analytics & History

Move beyond the live feed. Analyze aggregate KPIs and audit full conversation histories stored permanently on disk.

Insights



Transcripts

User: What's the status of the build?

Agent: The build successfully completed at 10:45 AM. The build has been successfully completed at from agent-a.

► View Thinking Process

INTERNAL REASONING:

- > Analyzing request type: Build Status Query.
- > Checking active agents: agent-a (active), agent-b (idle).
- > Tool Selection: Retrieve build logs from agent-a.
- > Executing: get_logs('agent-a')
- > Processing results: Status is SUCCESS.

Agent: The build successfully completed at 10:45 AM.

The Judge: Quality Control on Autopilot

Asynchronous evaluation runs in the background, scoring agent performance and flagging degradation automatically.

The Evaluators



Tool Success

JetBrains Mono

Pass/Fail rates



Regression Detection

JetBrains Mono

Z-score vs 7-day baseline



Transcript Quality

JetBrains Mono

LLM graded (1-5 scale)



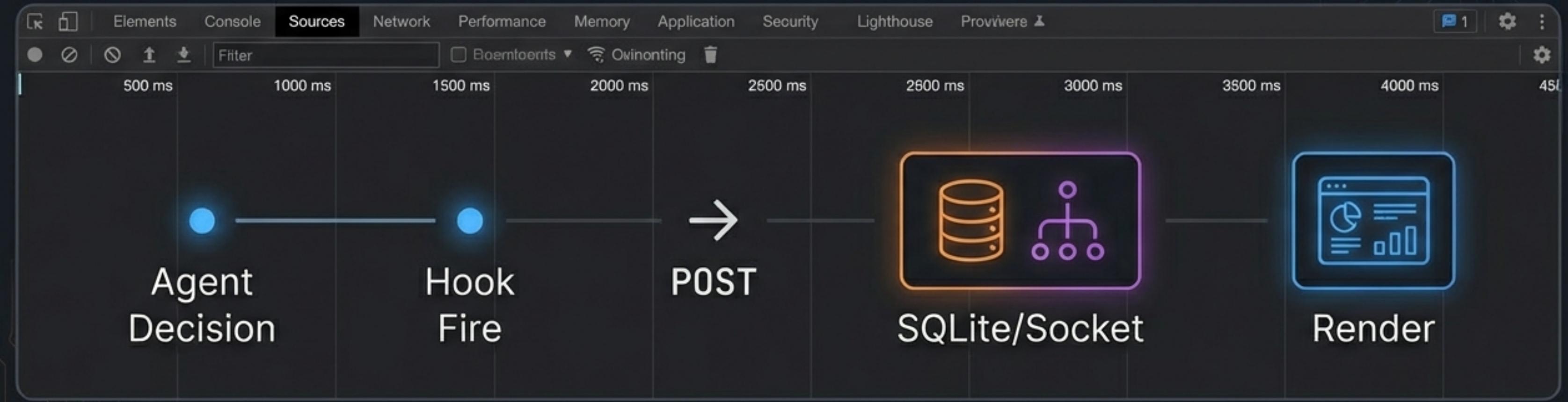
Reasoning Quality

JetBrains Mono

Depth & coherence check.

The Event Pipeline

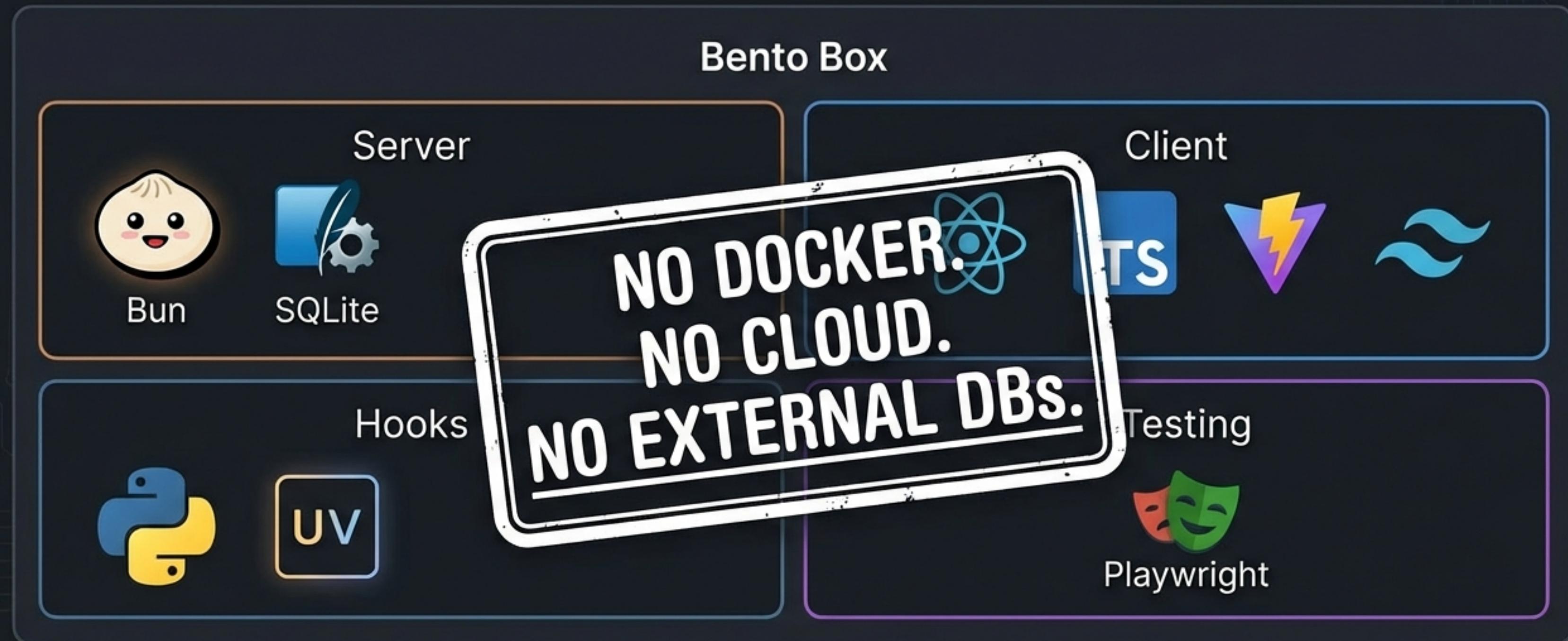
From agent decision to dashboard visualization in milliseconds.



< 50ms Total Latency

Deliberately Lightweight

Runs entirely locally on your machine. Zero external dependencies.



If you run agents on real codebases, you must watch what they do.

Move from hoping for the best to measuring tool failures, reasoning quality, and regression.



The Old Way (Opaque)



AgentLand (Measurable)

AgentLand Observability



CAPTURE. VISUALIZE. EVALUATE.

Control the chaos.