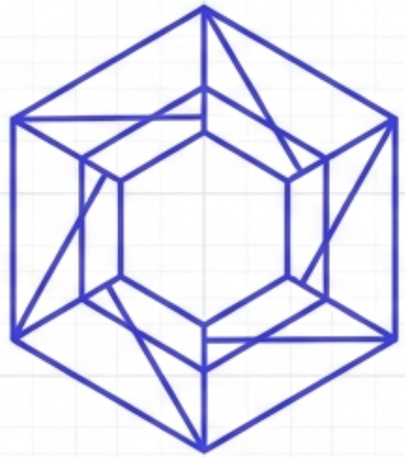
Clove: An Operating-System Runtime for Autonomous Agents

Moving beyond library abstractions to process-level isolation.



SYSTEM ARCHITECTURE & TECHNICAL MANIFESTO // V1.0

The Gap Between Demos and Infrastructure

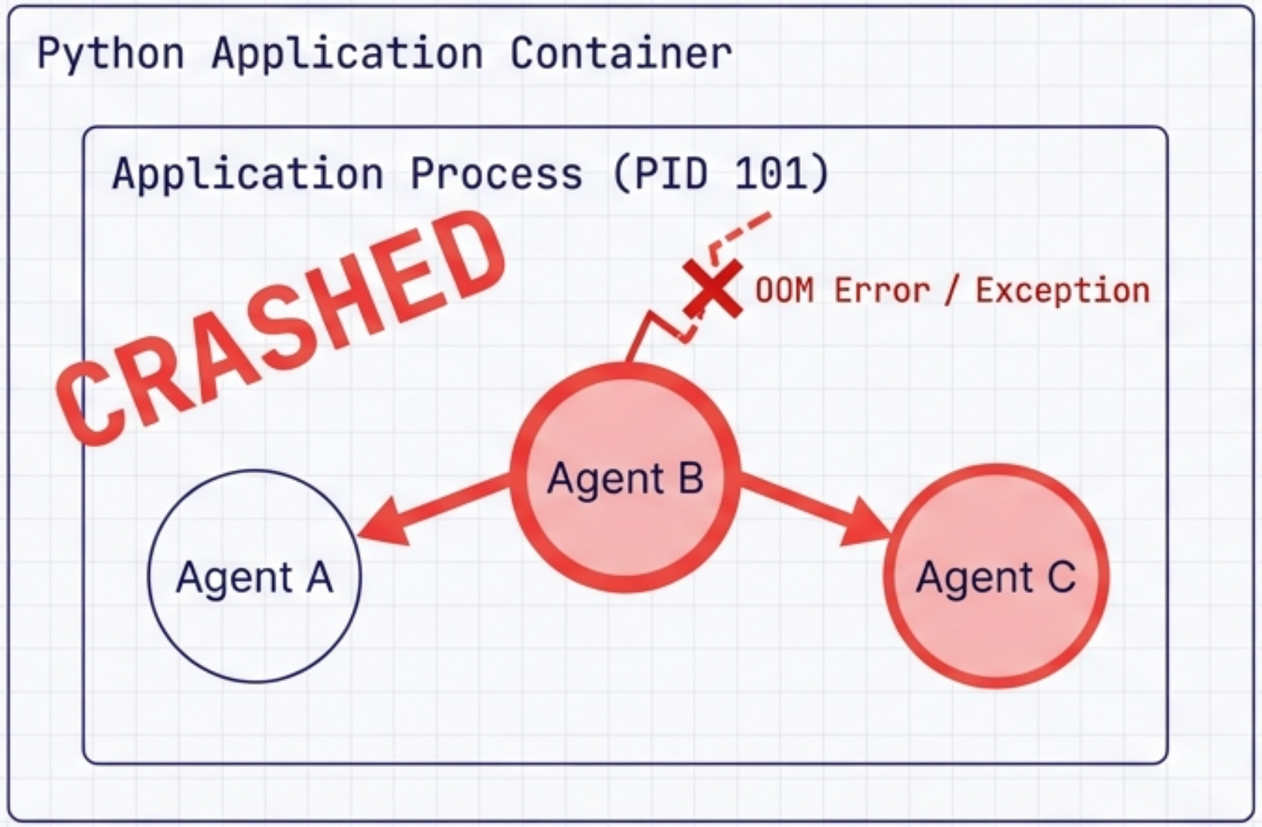
Modern agents are tasked with high-stakes execution, yet they run on "best-effort" library architectures.

EXPECTED RESPONSIBILITY		ACTUAL FOUNDATION
- Execute financial trades	REQUIREMENT → IMPLEMENTATION	- Python threads / coroutines
- Control robotics hardware	REQUIREMENT → IMPLEMENTATION	- Shared memory space
- Operate enterprise production systems	REQUIREMENT → IMPLEMENTATION	- Async tasks
- Continuous, unsupervised reasoning	REQUIREMENT → IMPLEMENTATION	- Try/Catch error handling

CRITICAL RISK: When an agent crashes, loops infinitely, or behaves adversarially, the entire system is at risk.

The “Library-Level” Failure Mode

Current frameworks rely on shared state. One failure cascades to the entire application.



Infinite Loops

System hangs, no preemption.

Memory Leaks

OOM Killer terminates parent process.

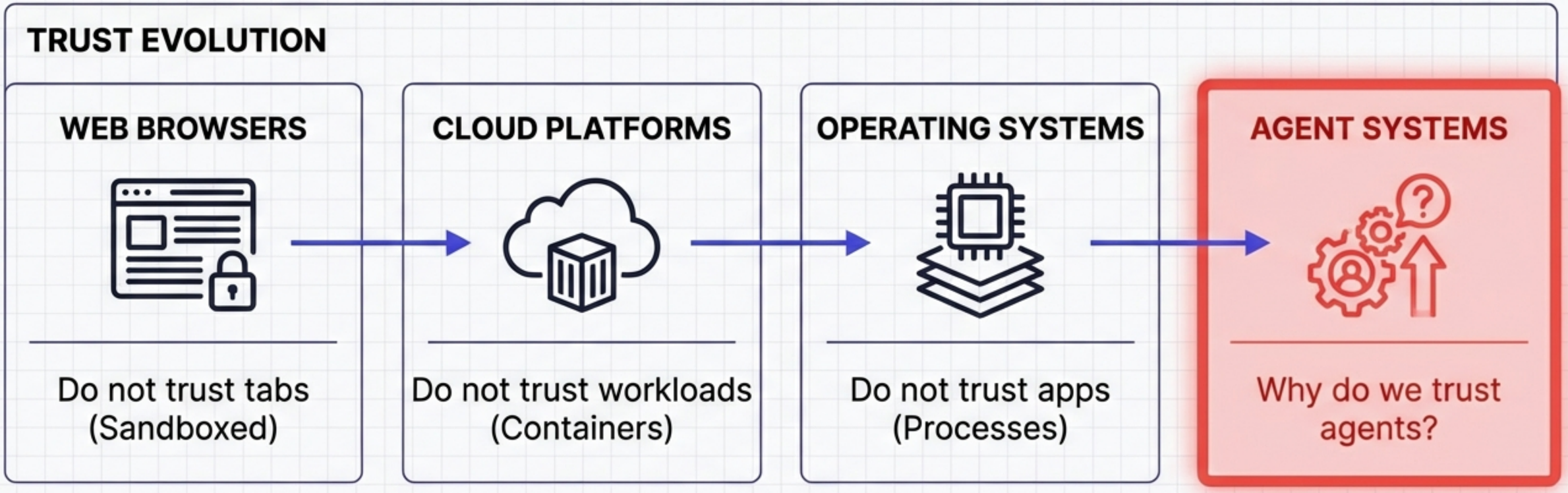
Token Exhaustion

Uncapped budget blowouts.

This is not an AI problem. This is an Operating Systems problem.

Agents Must Be Treated as Untrusted Processes

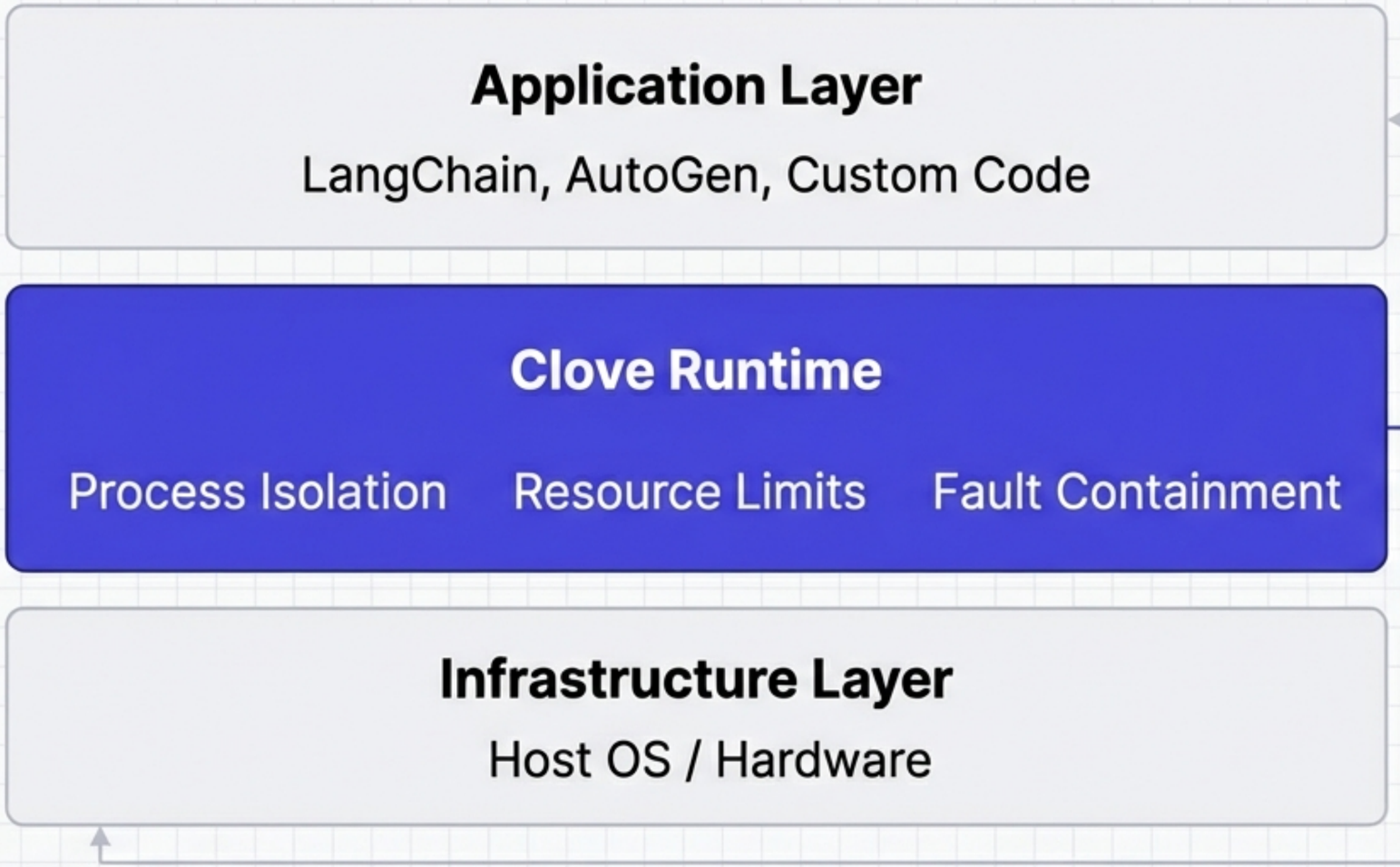
Agents are no longer assistive tools. They are active system participants.



Best-effort execution is unacceptable. **Autonomous** systems require **scheduling, observation**, and **strict limits**.

A Microkernel Runtime, Not a Framework

Clove sits below the application layer, providing the guarantees frameworks miss.

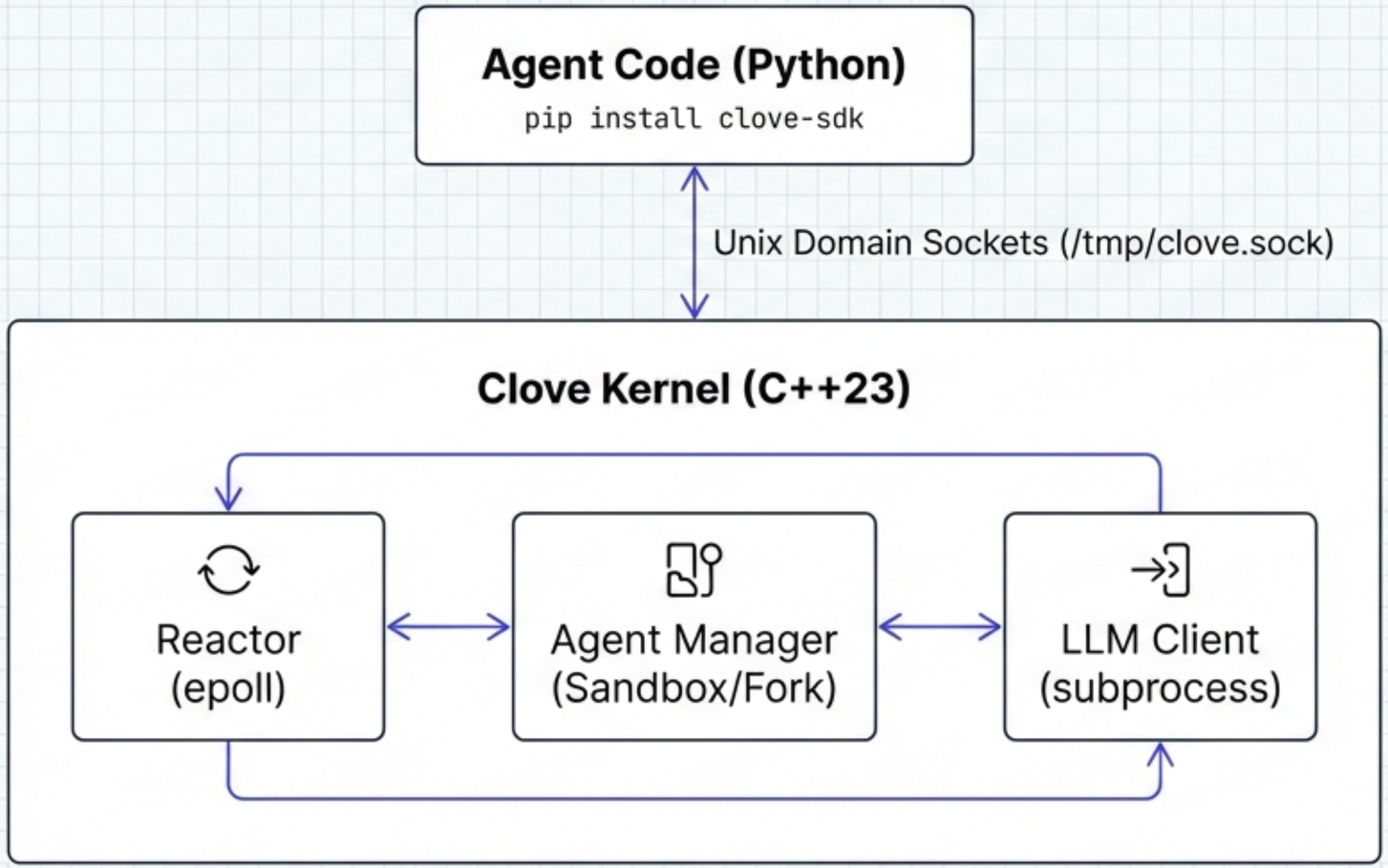


Clove is **NOT:**
- A prompt library
- A workflow DSL

Clove **IS:**
- A substrate for isolation
- An enforcer of limits

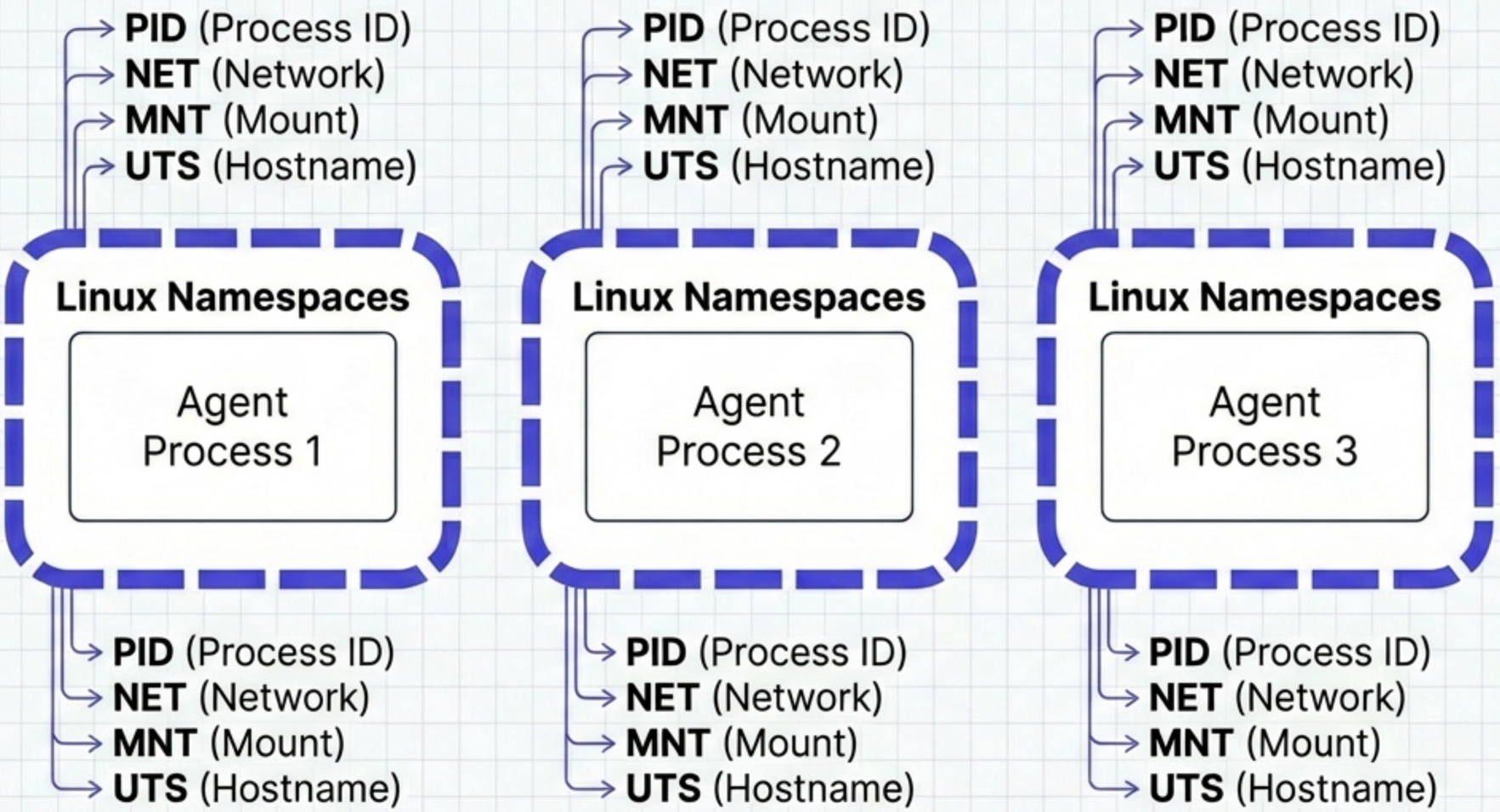
System Architecture

A minimal, opinionated runtime constructed around a C++23 event loop.



Sandboxed by Design

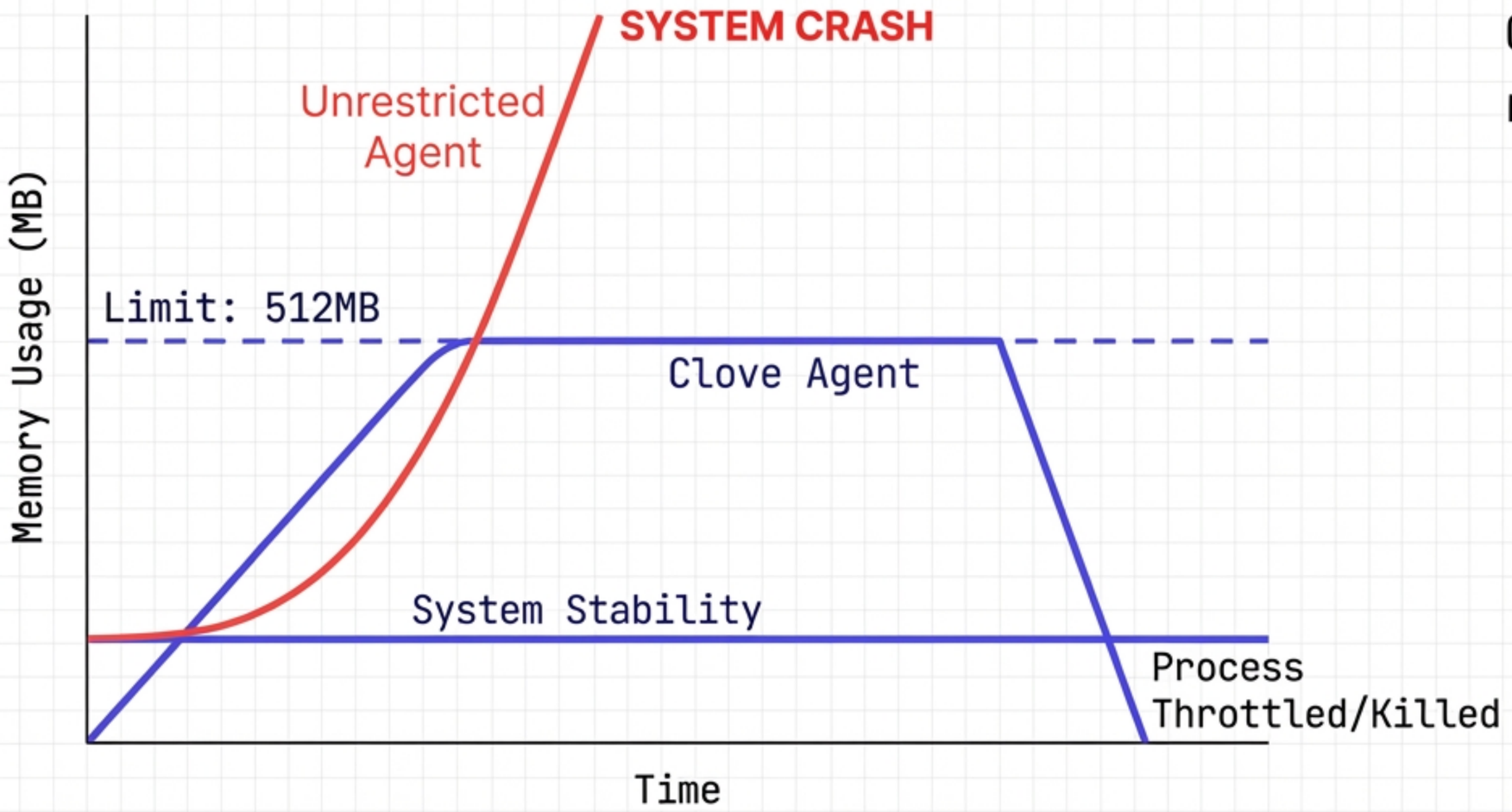
Agents run as independent OS processes, not threads.



Guarantee: One agent cannot crash another. Shared memory is strictly forbidden.
If an agent fails, the damage is contained to its own namespace.

Deterministic Resource Control

Enforcing limits via cgroups v2 integration.

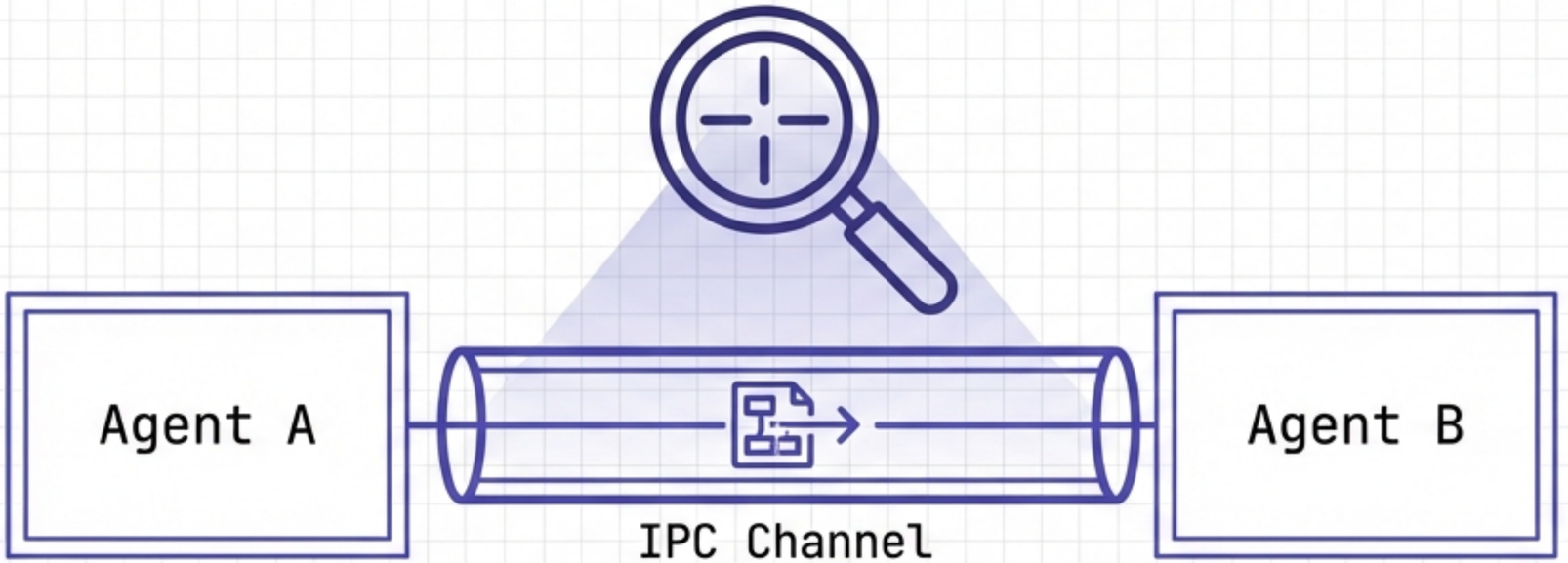


Controlled resources

- CPU Quotas
- Memory Limits
- PID Limits
- LLM Request Queuing

Explicit, Observable Communication

No hidden state. No implicit coupling.



- All communication via **explicit** IPC syscalls.
- **Zero** shared memory.
- Every interaction is **auditable**, loggable, and **replayable**.
- Enables **post-mortem** analysis of failures.

Frameworks vs. Clove Runtime

Failure Scenario	Standard Frameworks	Clove Runtime
Infinite Loops	System Hangs	**Agent Throttled via CPU Quota**
Memory Leak	🚨 OOM Kills Application	**Single Process Terminated**
Malicious Code	Full System Access	**Sandboxed Namespace**
Shared State	Corruptible	**Impossible (IPC Only)**
LLM Access	Race Conditions	**Fair Queuing**

Clean Abstractions for Python

Complex kernel guarantees. Standard Python SDK.

```
[KERNEL] Initializing Clove Runtime v1.0...
[KERNEL] Agent spawned (PID 4022)
[KERNEL] Enforcing memory limit: 256MB
```

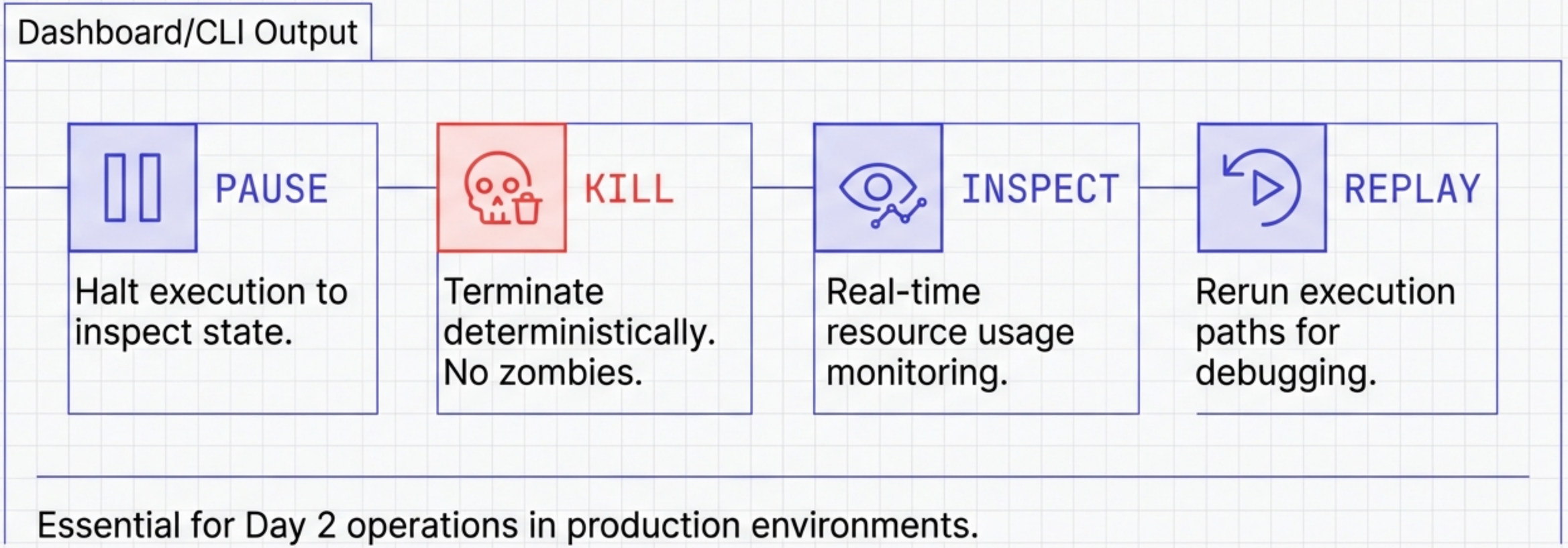
```python
from clove_sdk import CloveClient

with CloveClient() as client:
    # Spawn child agent with strict limits
    client.spawn(
        name='worker',
        script='worker.py',
        limits={'memory': 256*1024*1024, 'cpu_quota': 5000
    )
```

Execute sandboxed commands, spawn child agents, and query LLMs.

Turning Black Boxes into Operable Infrastructure

Clove provides operators with a control plane, not just logs.



High-Stakes Use Cases

CRYPTO & TRADING

JetBrains Mono

Latency matters. Capital is at risk. Framework-level failures are unacceptable.

ROBOTICS

JetBrains Mono

Physical safety. Multiple agents planning & perceiving. Needs reproducibility.

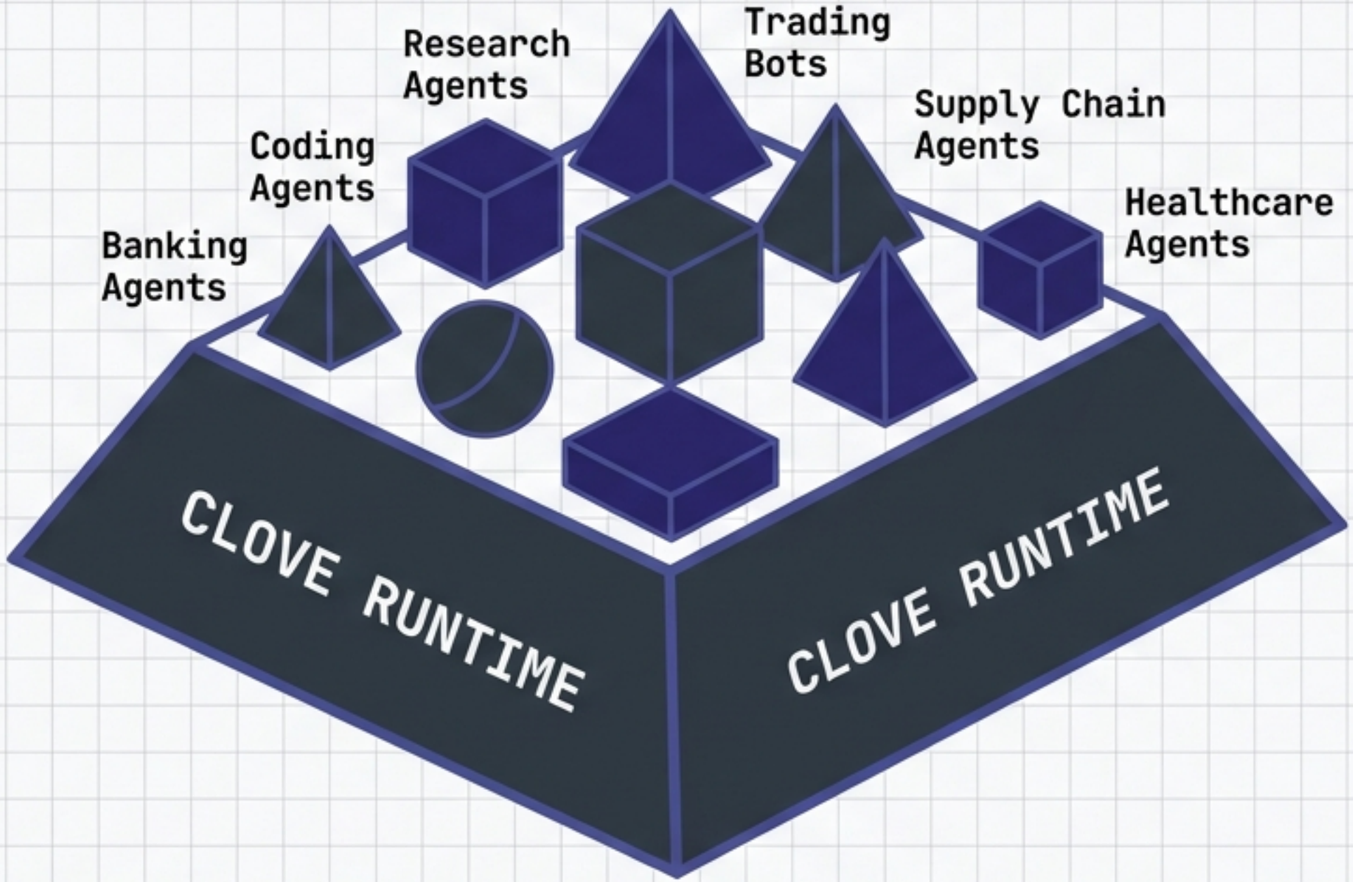
ENTERPRISE AI

JetBrains Mono

Compliance and auditability. Policy enforcement is non-negotiable.

The Standard Substrate for Autonomous Compute

The long-term vision.



Hard to build.
Slow to earn trust.
Impossible to displace.

Linux for Servers.
Postgres for Data.
Clove for Agents.

“Clove will not win hackathons. It will attract serious engineers.”

Start the Runtime

The problem is inevitable. The solution is structural.

DOCKER

```
docker run -d --privileged ghcr.io/anixd/clove
```

SDK

```
pip install clove-sdk
```

GitHub Repository | Documentation