

OpenAI

Configured for Autonomy

Making Enterprise Agents Useful and
Safe at Scale

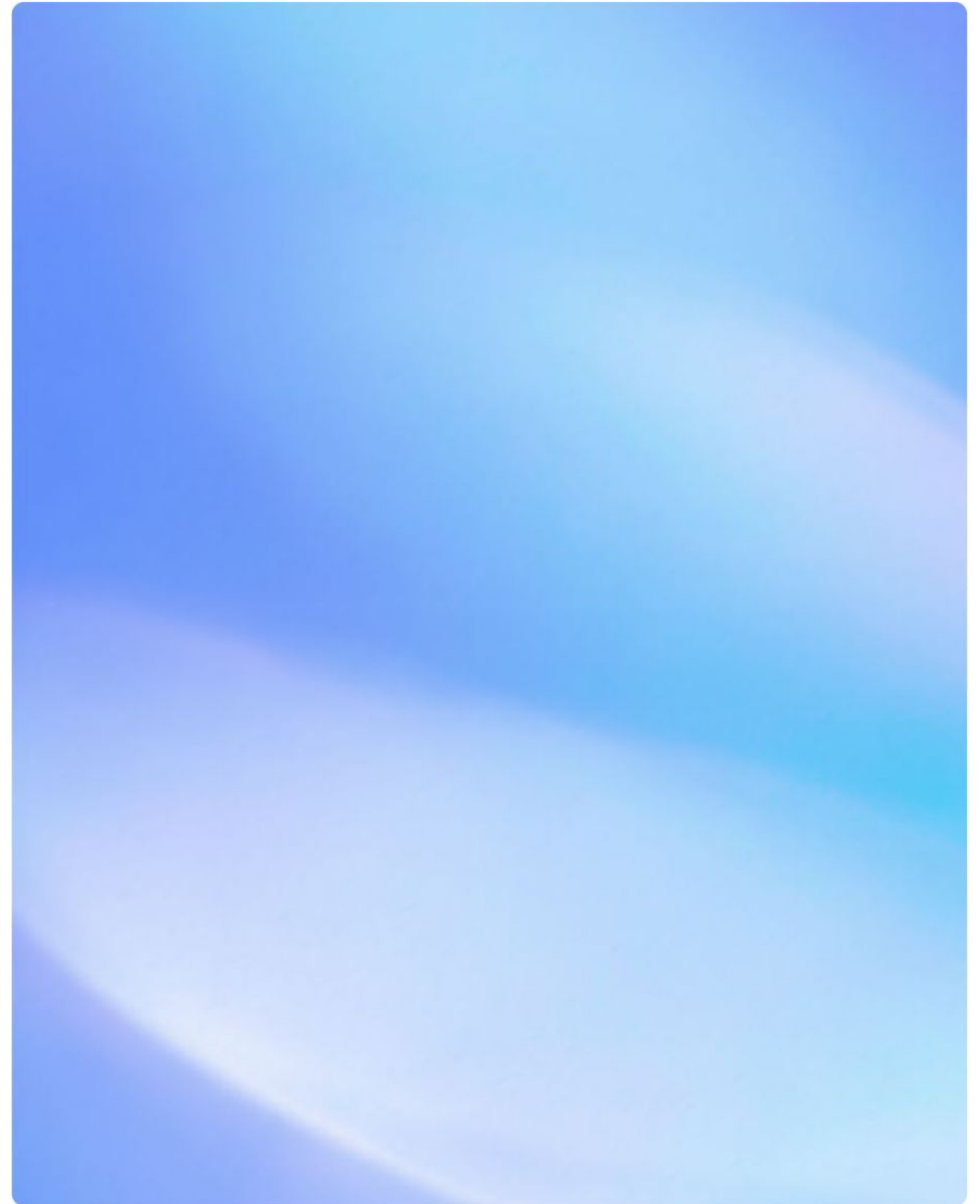
AGNTCon



Viyat Bhalodia

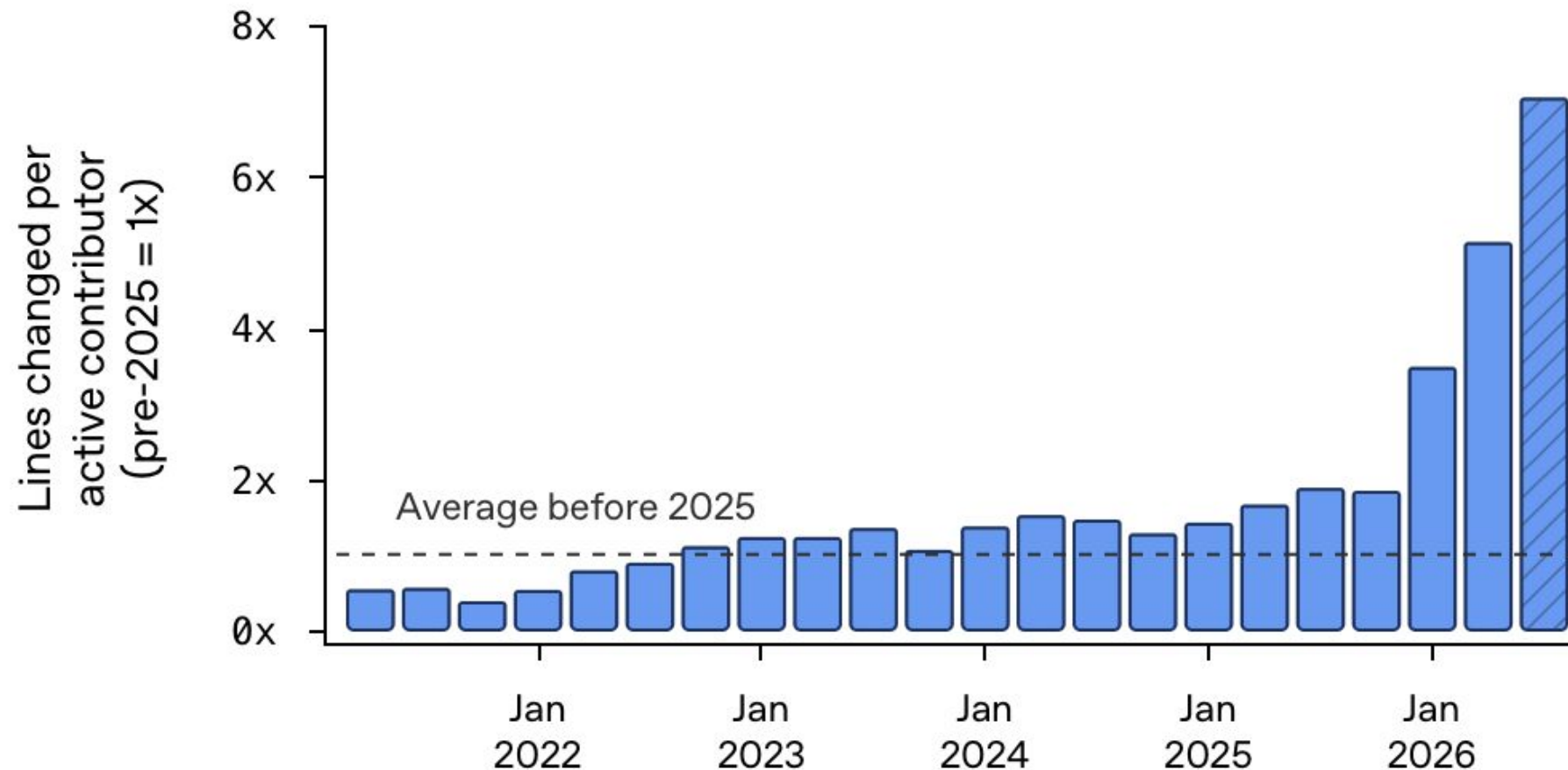


Casey Silver



How do you ship code this fast and still keep it secure?

Across the company, engineers are shipping code faster



The lethal trifecta

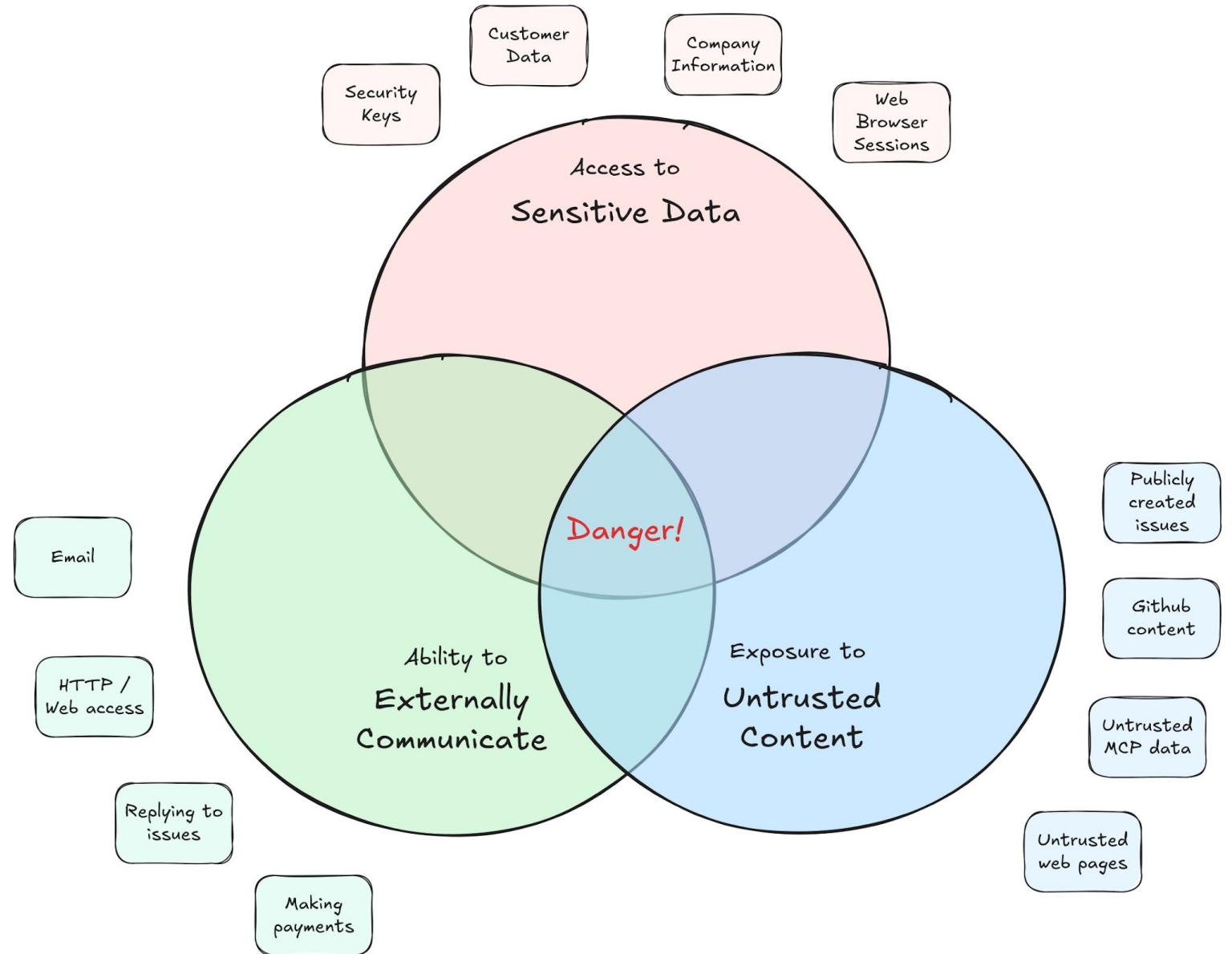


Diagram: Kornv Sietsma · Concept: Simon Willison

Sandboxing

Sandbox	Endpoint or cloud
Filesystem	Give it the files it needs.
Network	Limit where it can connect.
Process execution	Limits apply to child processes.

Credentials

Keep the service token outside the sandbox

If a token is in an environment variable or a .env file, the agent's code can read it.

What the agent sends

Destination

`git.example.com`

Authorization header

`Bearer <placeholder>`

Issue body

`<placeholder>`

What the proxy sends

Approved destination

`git.example.com`

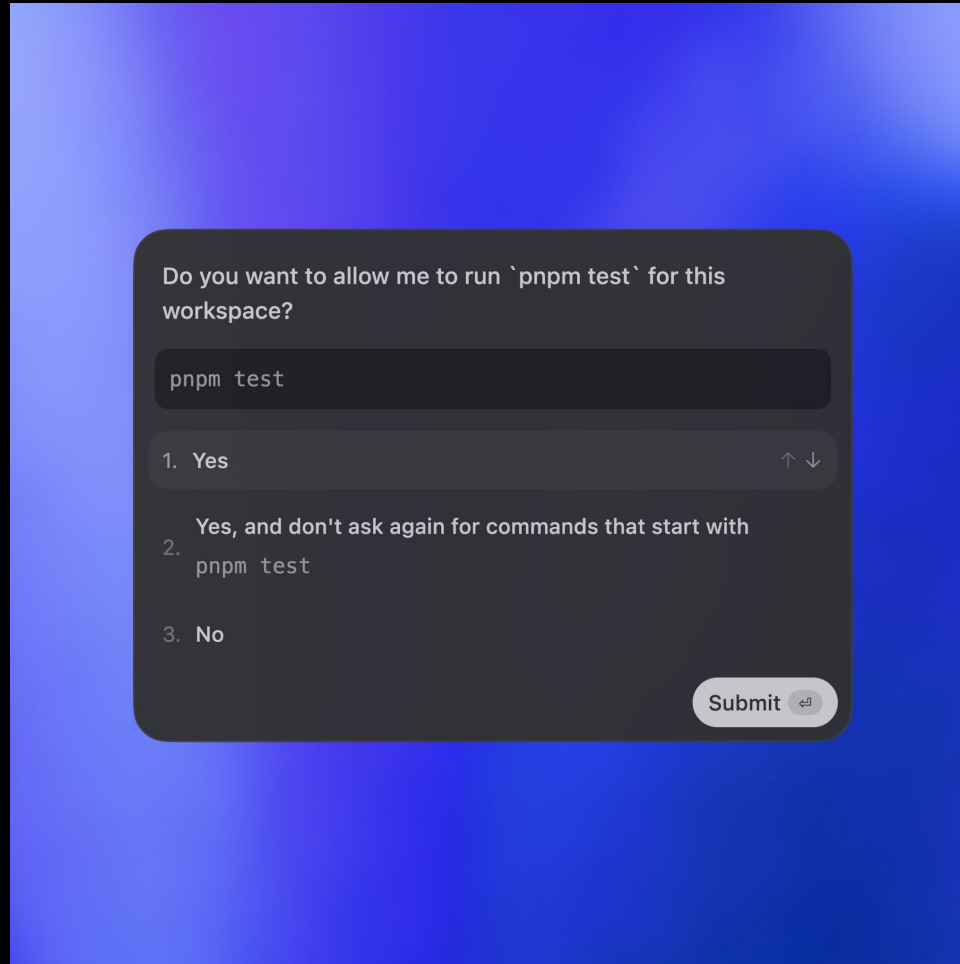
Authorization header

`Bearer [service token]`

Issue body

`<placeholder>`

Approval fatigue weakens security



Let the LLM review permission requests

Finding a flight and committing to a nonrefundable fare are two different decisions.



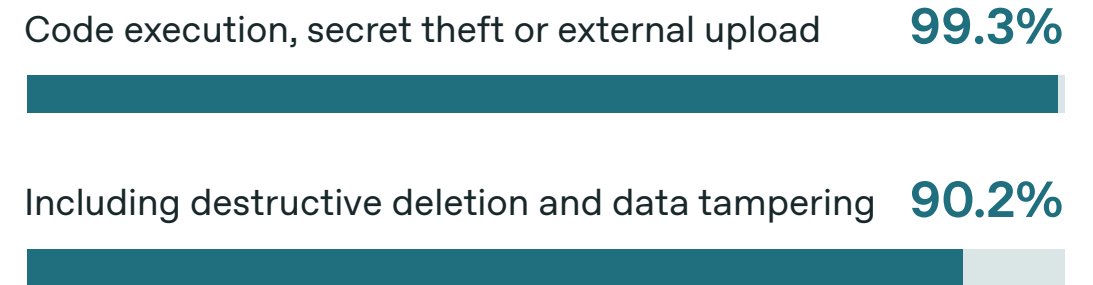
Why use an LLM reviewer?

How often did someone have to step in?



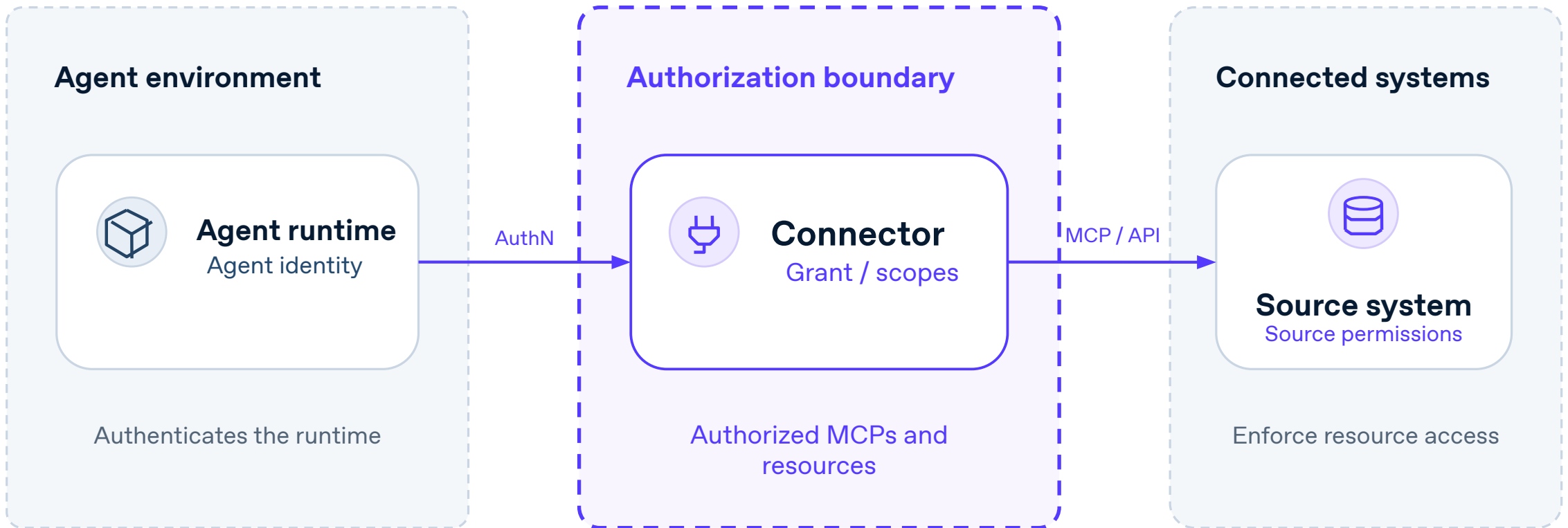
In one case, auto-review denied 7 of 720 requests. The agent found a safer way forward in 4 cases. The other 3 needed a person.

What did the reviewer block in testing?



Authorization is the last isolation boundary

Runtime identity and resource authority need to be granular



Granularity of control is dependent on the source system

Delegated access vs. agent specific identity

The identity model determines whose authority the agent carries

Delegated user access

AUTHORITY

Acts through a user's permissions

RISKS

Can leak data and otherwise abuse broad access the user maintains.

CONSIDERATIONS

Carefully control where the agent can write, involve user reviews on writes, and monitor long horizon work.

Agent specific identity

AUTHORITY

Dedicated service accounts & workload scoped

RISKS

Less flexible and more complex to setup and maintain

CONSIDERATIONS

End-to-end support is still uneven across identity providers and source systems. Ownership is less clear.

The agent use case should inform your choice

Agent aware access patterns are uneven

Can the source system distinguish the agent acting on your behalf?

Agent acting for you

Uses your delegated access



Source system

Can it distinguish you from your agent (or even which of your agents)

DESIRED SOURCE SYSTEM
VIEW

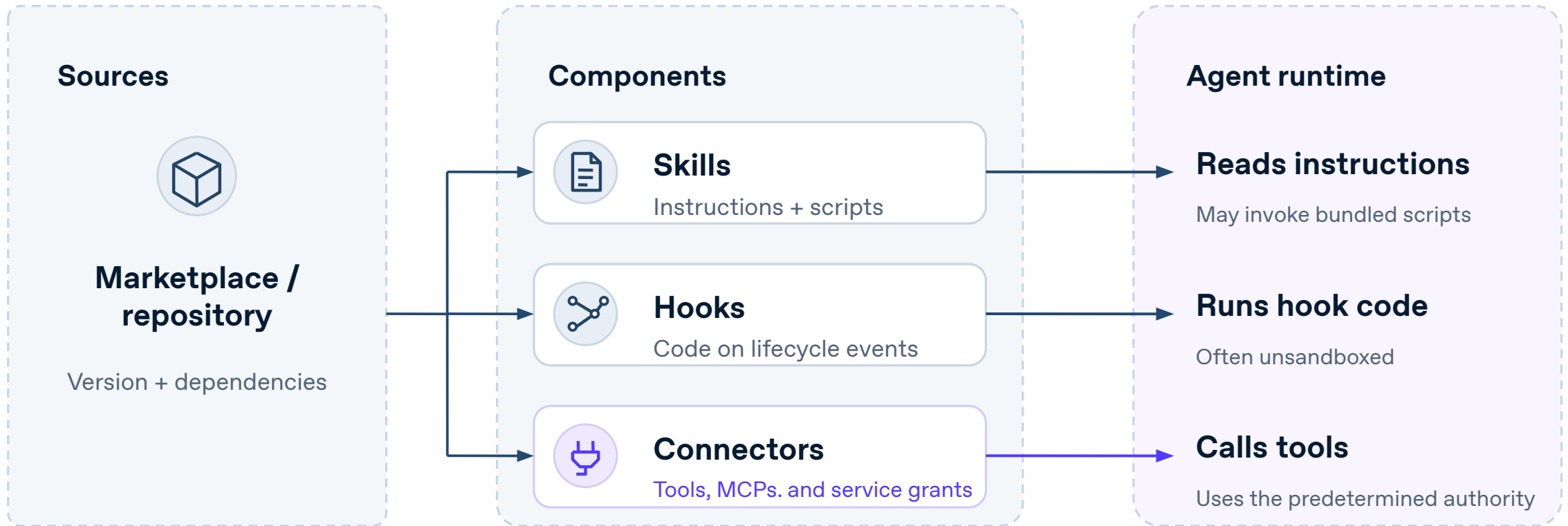
SLACK EXAMPLE

Sent using @ChatGPT

Slack retains human + app attribution. That alone does not identify the particular agent or thread.

The agent's supply chain

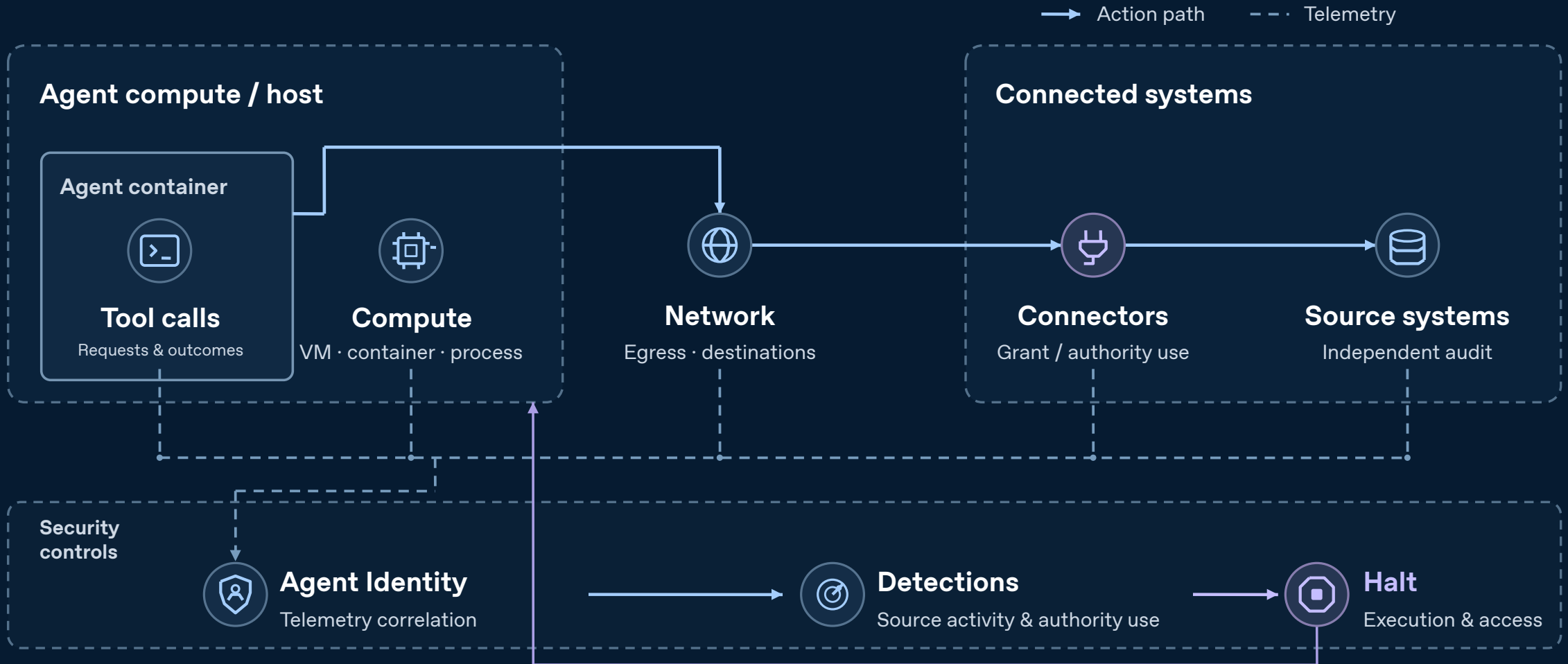
Skills, hooks, and connectors add instructions, code, and access



An update can change what the agent trusts, runs, or reaches.

Monitoring & Halting

Complete telemetry across the agent's action surface



Creating an agent specific response pipeline

Correlate logs by agent identity to detect and interdict suspicious behavior

CORRELATE LOGS

Agent Activity Logs

Egress proxy

Unusual outbound request

Sandbox / VM

Privilege escalation attempt

Connector

Sensitive data access

DETECT

Deterministic Rules

Policy violations and indicators of misalignment

Agentic review

Indepth, scalable review for escalation to a human

RESPONSE (examples)

Revoke connector access

Disable inference

Kill or recycle the compute

Apply a stricter computer & network controls

Takeaways

Defense in depth

We have the controls to
secure agents

Sandbox &
Isolation

Enforce deterministic execution
boundaries.

LLM Reviewer

Use contextual review to scale

Monitoring &
Halting

Build a monitoring and response
pipeline

Humans in the
loop

Filter human intervention for the
highest impact events