Grimlock: Guarding High-Agency Systems with eBPF and Attested Channels

1st Workshop on Operating Systems Design for AI Agents (AgenticOS), co-located with ASPLOS 2026

ROBLOX

Authors

Qiancheng (Mark)

Wenhui

Gan

Sheng

Biao

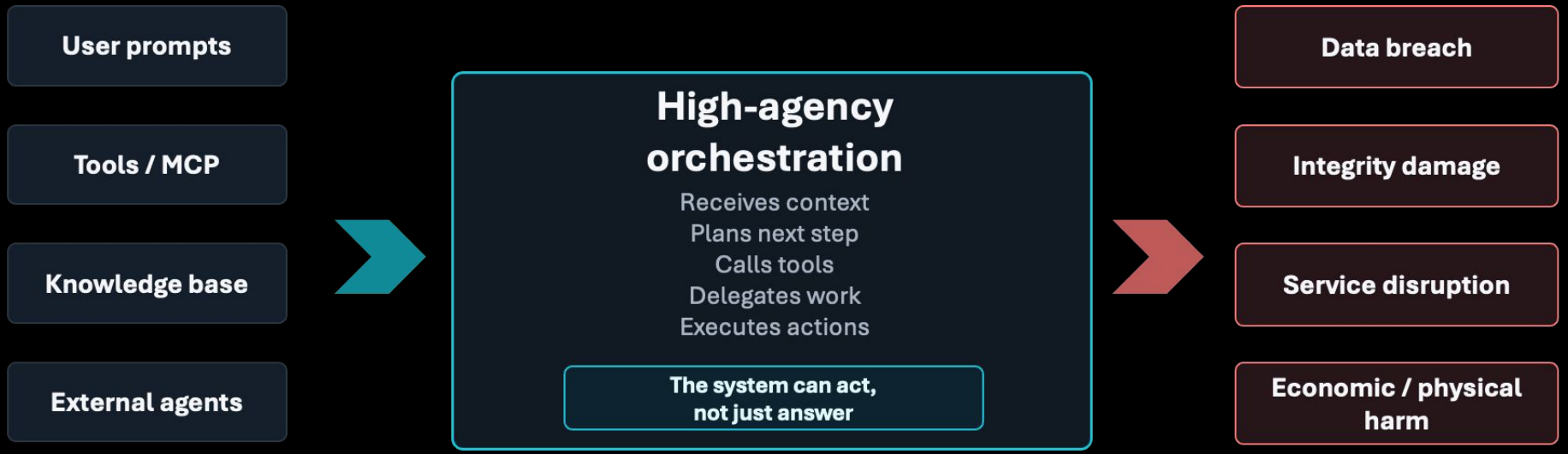
David

Shawna

Rob

Why high-agency changes security?

Orchestration code can call tools, spawn subtasks, and delegate across machines and clouds



So what breaks today? What needs to change?

What breaks today

Identity, authorization, provenance, and delegation drift into the least trustworthy layer: agent application code.

- checks are duplicated across orchestration paths
- enforcement is inconsistent and hard to audit
- a buggy or malicious runtime can try to bypass or escalate scope

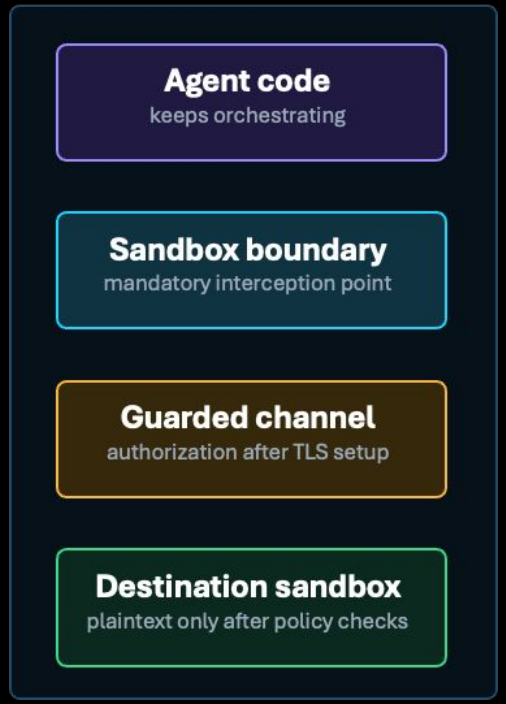
The key idea

Let agents keep the flexibility.
Move trust enforcement into a guarded substrate they cannot bypass.

No-bypass	Channel-bound	Least privilege
all sandbox traffic must traverse the guard	authorization artifacts bind to one established channel	delegation propagates scoped, auditable permissions

How Grimlock restores the boundary?

Two design principles: mandatory mediation at the sandbox edge, then authorization bound to the live channel.



1. eBPF mediation at the sandbox boundary

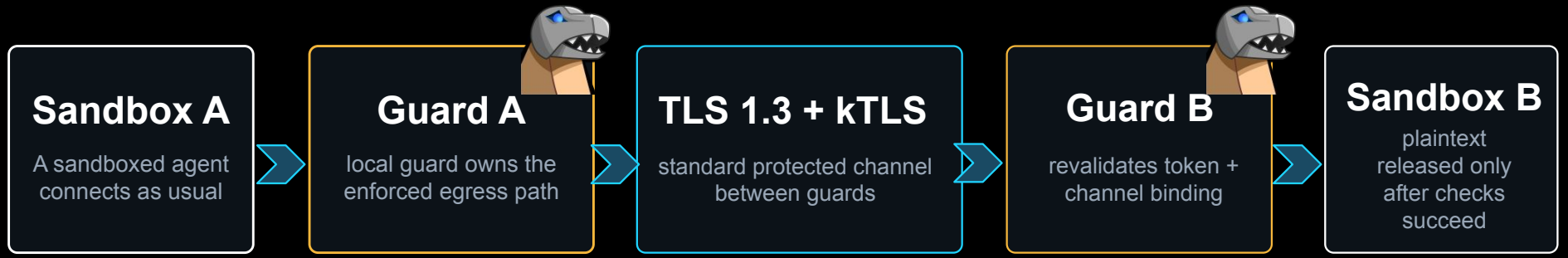
- intercepts sandbox ingress and egress
- forces traffic through the local guard
- gives us no-bypass enforcement below the agent

2. Post-handshake attestation + Scope Token

- uses standard TLS 1.3 as the transport base
- binds evidence and authorization to the established channel
- mints short-lived, least-privilege delegation tokens

How does Grimlock work?

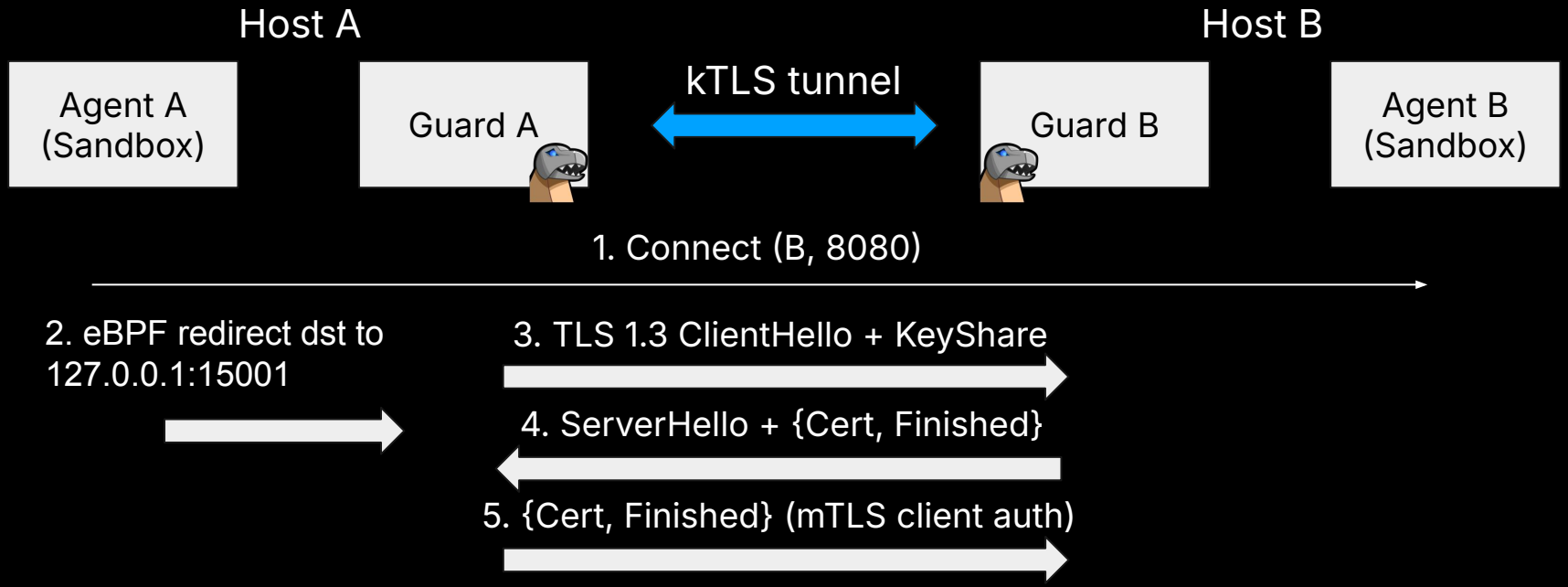
Each agent runs in its own sandbox; each host runs Grimlock as a local guard.



1. eBPF hooks redirect all sandbox traffic into the host-local guard
2. Guards establish a standard TLS 1.3 channel; kTLS handles record processing
3. Authorization happens after handshake completion, bound to that channel
4. Receiving guard validates and only then releases plaintext to the destination sandbox

A2A communication: what actually happens

Grimlock: A2A Communication (1/2)



Grimlock: A2A Communication (2/2)

7. Attestation Challenge (nonce_down, scope, audience)

9. Remote Verifier appraises + mints **Scope Token**: (scope, audience, expiry, H(cb))

8. Attestation Response (TEE Evidence committing to H(cb))

10. Scope Token + Application Data

11. Validate Token ✅

12. Release plaintext data (TLS terminated at Guard)

What Grimlock buys us?

Before

- Trust lives in app code
- Checks vary across runtimes and orchestration paths
- Delegation is broad, implicit, and hard to audit

With Grimlock

- Trust lives in guarded substrate
- Mediation is enforced below the agent
- Delegation becomes short-lived, scoped, and auditable

Takeaways

1. High agency is valuable, but it collapses security boundaries if trust lives in agent application code.

2. Grimlock restores that boundary with no-bypass mediation at the sandbox edge and authorization bound to the live protected channel.

3. Secure agent-to-agent communication, not asking the agent to become the security boundary.

Q&A