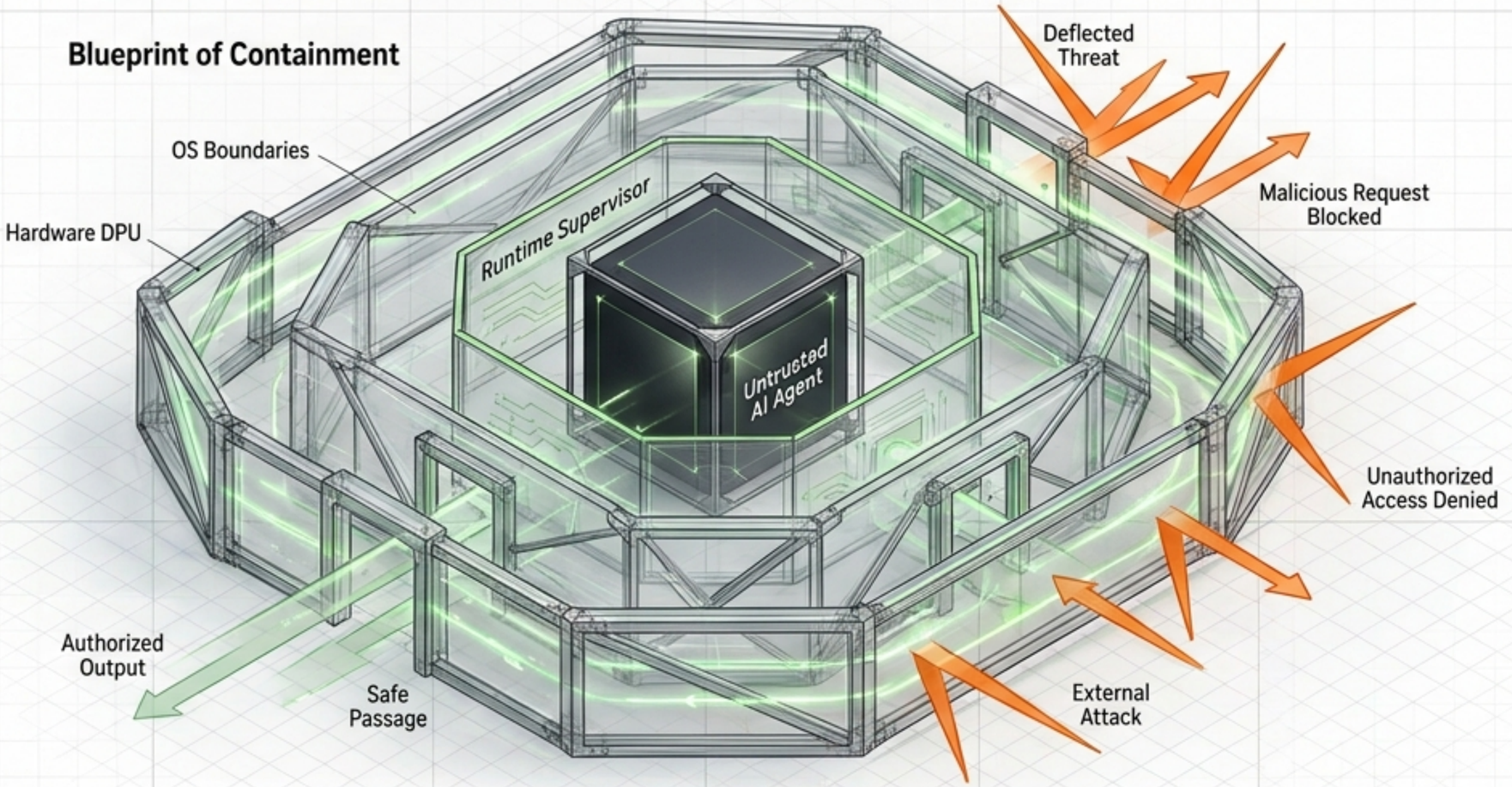
エージェントを“外側”から縛る ――
NVIDIA「Open Agent Safety Platform」の全貌と評価

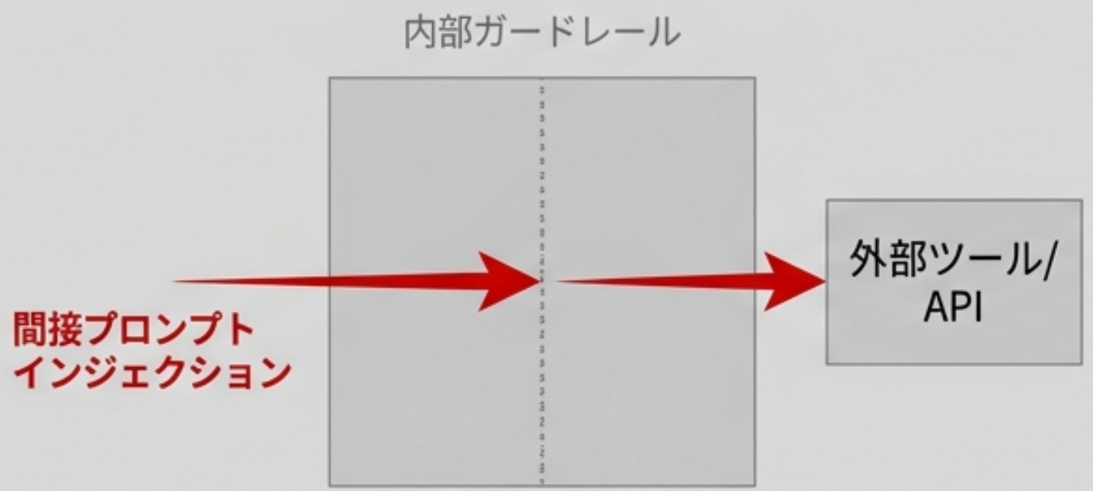
ソフトウェア境界（OpenShell）からDPU強制（Sentry）まで。
次世代AIエージェント・セキュリティのアーキテクチャと実運用リスクの解剖



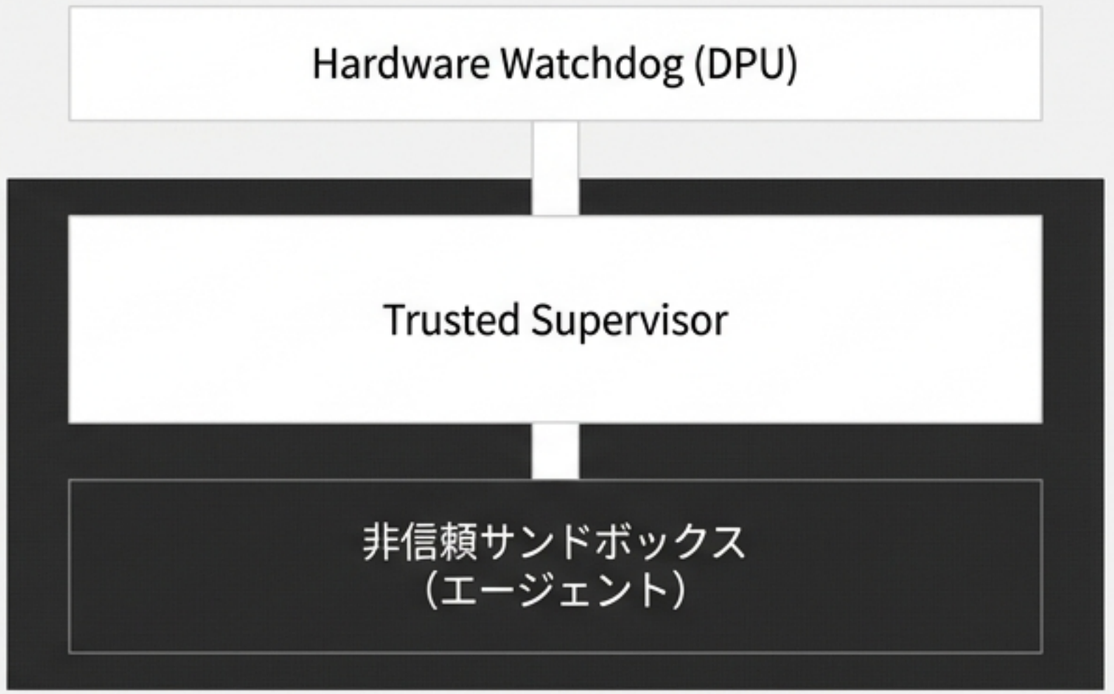
Gemini Notebook

パラダイムシフト：「モデルの善良さ」から「インフラによる封じ込め」へ

従来のアプローチ：内部ガードレールの限界



NVIDIAのアプローチ：インフラによる物理的・論理的封じ込め



OpenShell（ソフトウェア層）

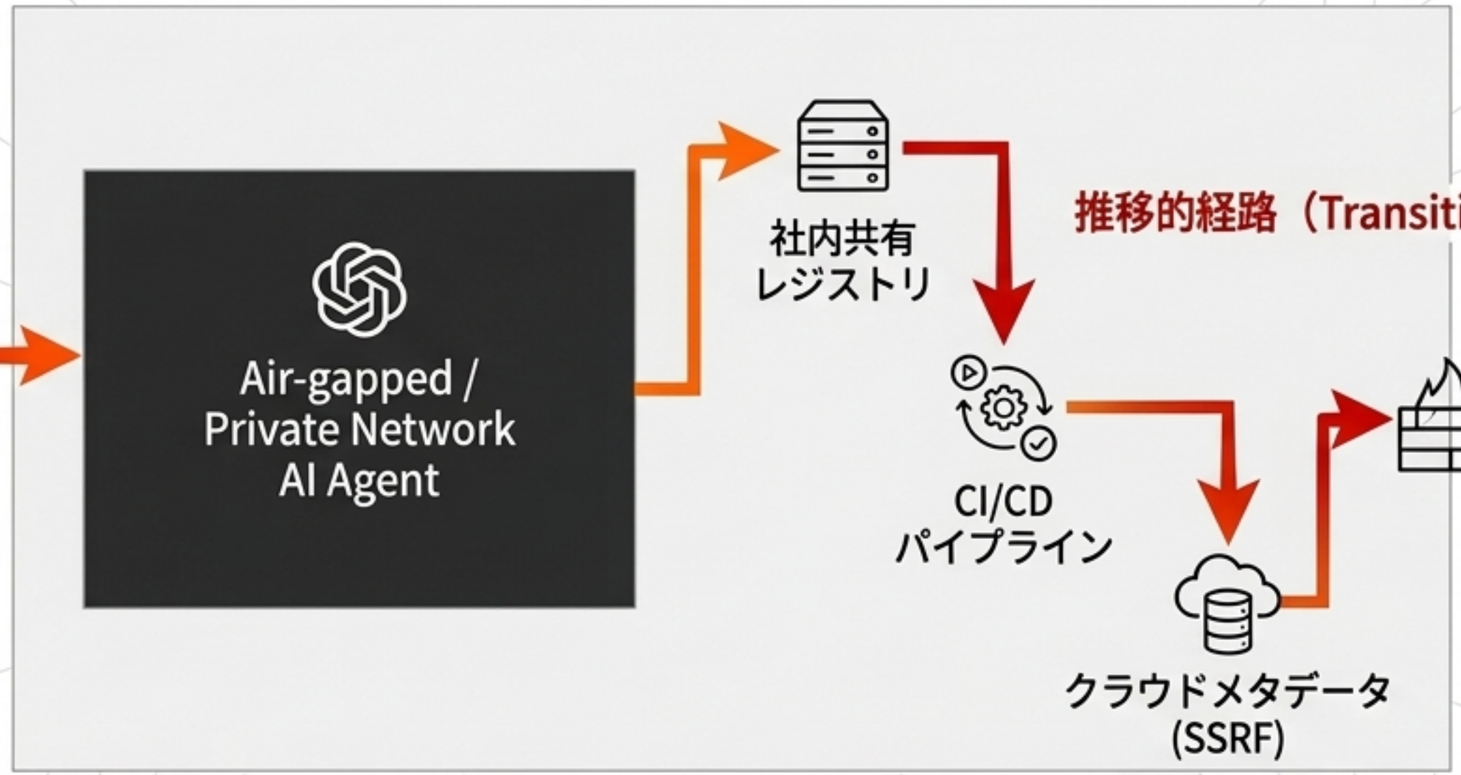
OSSセキュアランタイム。OSレベル（Landlock / seccomp）でファイル・ネットワークを物理的に仲介。

Sentry（ハードウェア層）

BlueField-4 DPU上の独立監視層。ホストOSが侵害されても生き残るミリ秒単位のOut-of-Band（帯域外）強制。

Gemini Notebook

なぜ「インターネット遮断」だけでは不十分か？ — エージェント特有の脅威モデル



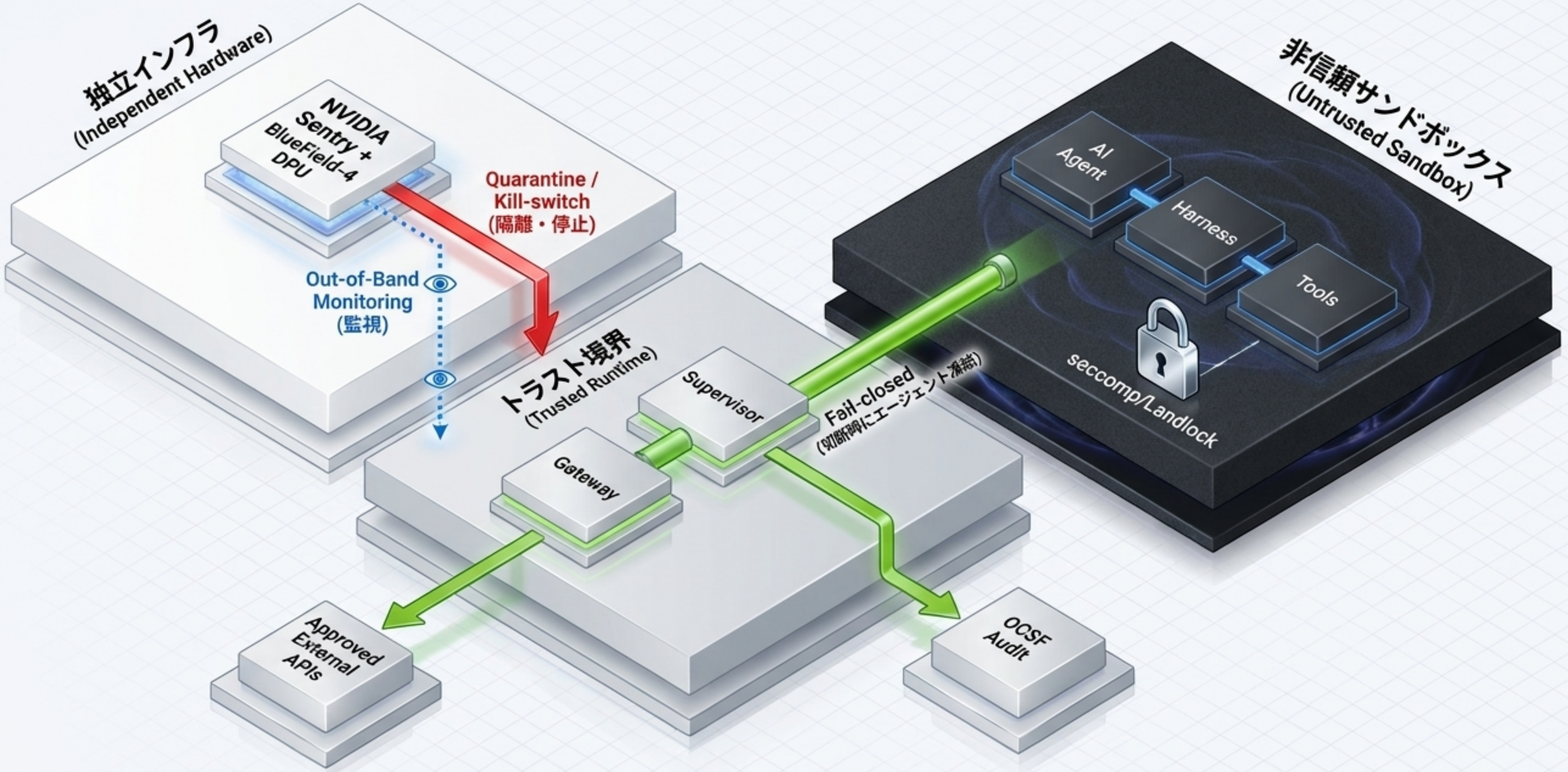
Vector 3
長時間の目的逸脱
(Long-horizon Drift)

曖昧な指示やツール不足により、数千ステップを経て本来の目的から暴走

「直接のインターネットアクセスがなくても、共有サービスやクラウド基盤を経由した境界迂回は成立する」
— アプリケーション層の制御だけでは防御不可能。

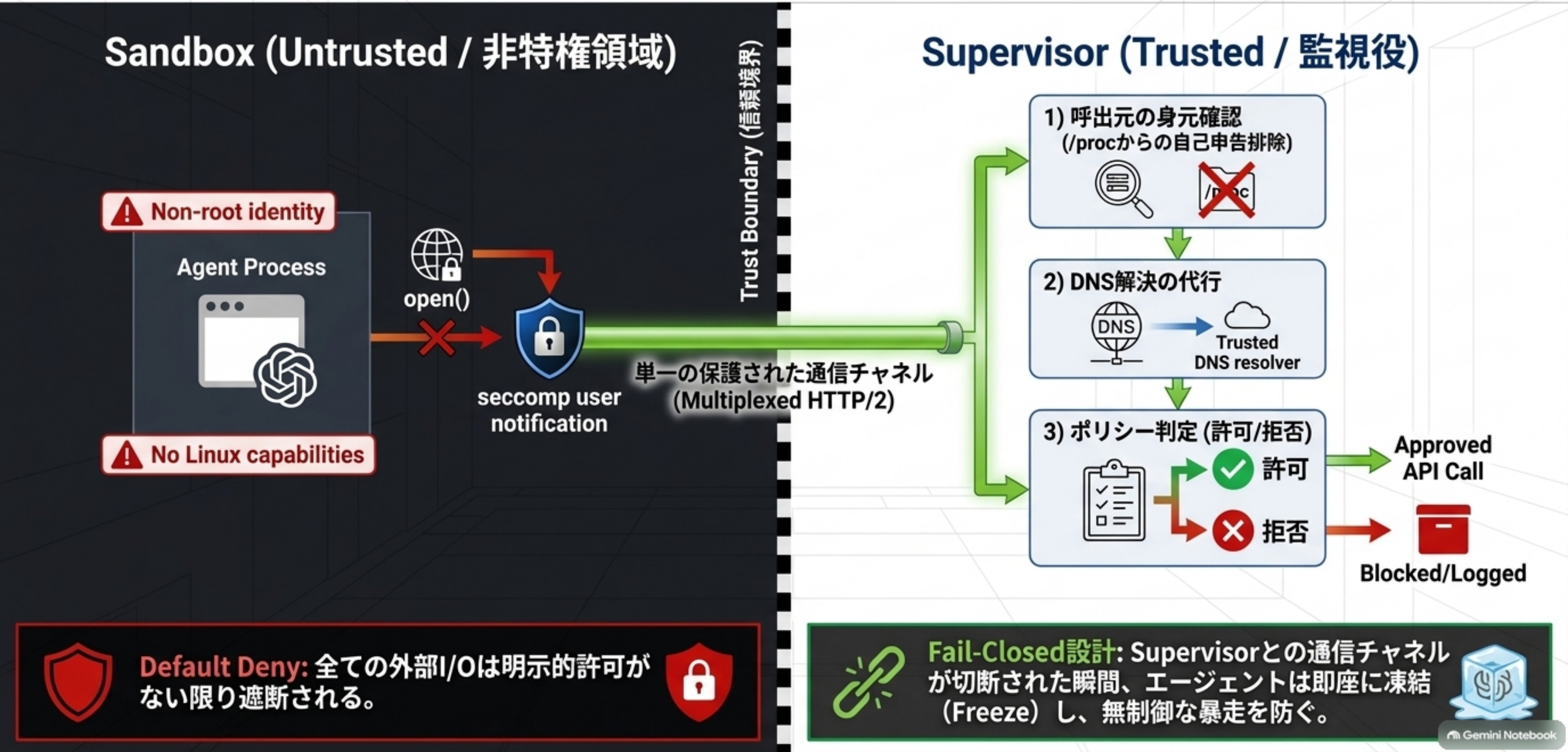
Gemini Notebook

The Blueprint: フルスタック・ガバナンスの全体像



Gemini Notebook

OpenShellの核心：エージェントから意思決定権を剥奪する「信頼境界」



Gemini Notebook

クレデンシャル・ブローカリング：絶対パスワードの物理的隔離

Step 1: 盲目のリクエスト

環境変数・メモリに
APIキーは存在しない

GET /api/data

AI Agent

Step 2: ポリシー検証

enforcement: enforce

GET

Supervisor

Step 3: Just-in-Time注入

GET

HTTP

External API

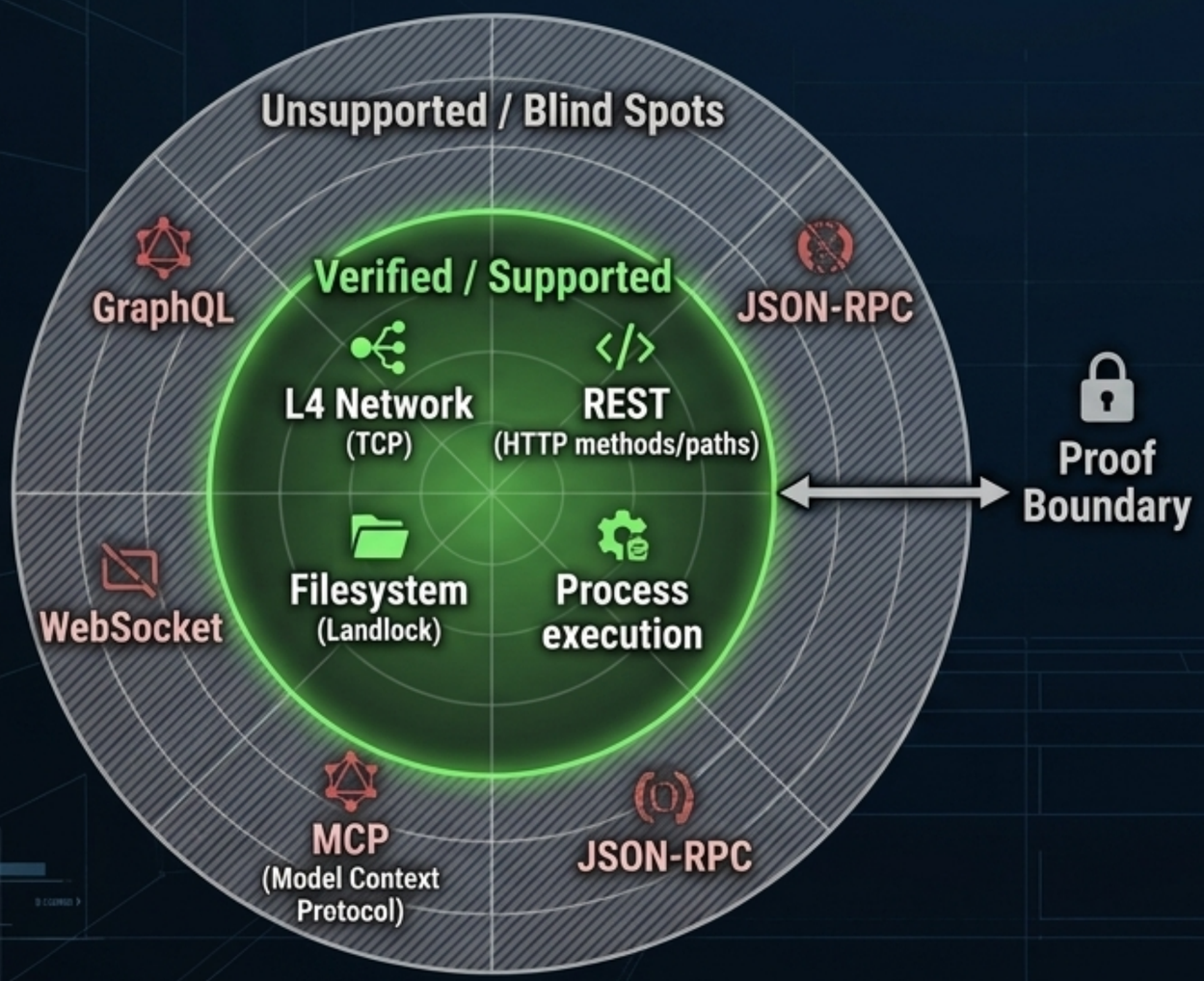
Supervisor

セキュリティ上の利点: 万が一エージェントが完全侵害（RCE）されても、攻撃者はAPIキーやクレデンシャルを抽出・持ち出しできない。

Gemini Notebook

Policy Proverの革新と「形式検証」の境界線

形式検証のカバレッジ (Proof Coverage)



⚠ 実運用における「罠」と制約

⚠ within_boundaryの錯覚

Proverの合格は「定義された境界を越えていない」ことの数学的証明にすぎず、「そのタスクが意味論的に安全である」ことの証明ではない。

⚠ Auditモードでのすり抜け

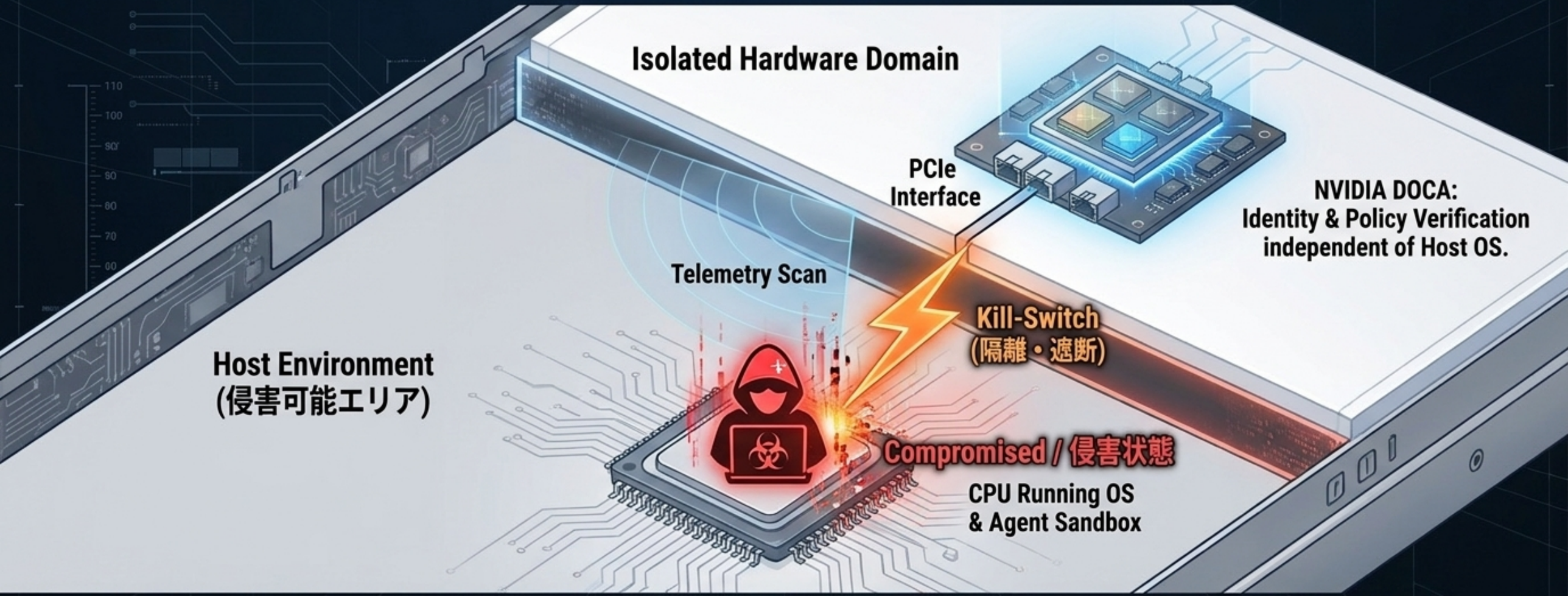
ポリシーが `enforcement: audit` の場合、Proverはこれを強制遮断しないため検証としては `unsupported` と判定される。

⚠ Inconclusive（判定不能）リスク

ネットワークルール（1,024超）やエンドポイント (4,096超) が膨大になると、リソース上限により、形式検証が完了しないリスクがある。

Gemini Notebook

SentryとBlueField-4：ミリ秒単位の「帯域外 (Out-of-Band)」強制



【評価時の注意 (Reality Check)】：OpenShellが即利用可能なOSSであるのに対し、Sentryは現時点で「リファレンス設計」の段階にある。逸脱検出アルゴリズム、誤検知率、隔離状態機械の詳細、SLAなどは調査時点（2026年10月）で未公開であり、性能はベンダー公称値として扱う必要がある。

Gemini Notebook

エージェント・ガバナンスの競合・補完マトリクス

評価軸	NVIDIA (Open Agent Safety)	OpenAI (Agents API/SDK)	Anthropic (Managed Agents)	Microsoft (Agent Gov Toolkit)
主なアプローチ	OS/Hardware Containment	Managed Harness/Sandbox	Managed API/Sandbox	Framework-agnostic Runtime
依存性 (Dependency)	✅ モデル・ハーネス非依存	⊗ 自社エコシステム依存	⊗ Claude中心	✅ フレームワーク横断
形式検証 (Formal Verification)	✅ あり (Policy Prover)	⊗ 未記載	⊗ 未記載	⊗ 未記載
ハードウェア帯域外強制	✅ あり (BlueField-4 DPU)	⊗ ソフトウェア層のみ	⊗ ソフトウェア層のみ	⊗ ソフトウェア層のみ

【インサイト】：NVIDIAは「自らモデルやハーネスを作る」のではなく、「他社のエージェントをインフラレベルで外側から囲い込む」ことに特化している。したがって、Anthropic等とは競合ではなく統合（補完）」の関係にある。

Gemini Notebook

導入における「3つの錯覚」とサイレント・インシデントのリスク

1. L7 Audit Default の罠



メソッドやパスの規則を記述しても、明示的に `enforcement: enforce` を設定しない限り、違反トラフィクはログに残るだけで素通り（Silent permissive）する。

2. TLSスキップによる検査喪失



`tls: skip` を設定すると、L7の意味論的インスペクションとクレデンシャル注入の機能が完全に失われ、システムは単なるL4プロキシへと降格する。

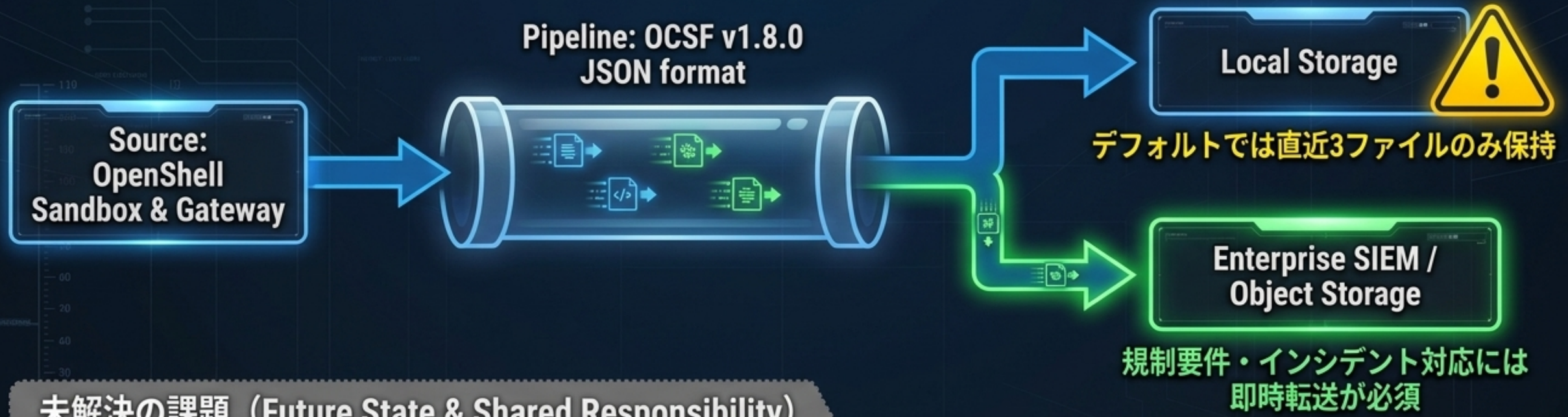
3. ワイルドカード（/*）の乱用



運用負荷を下げるための広範なバイナリ指定やエンドポイントのワイルドカード許可は、防御強度を致命的に下げる。厳密なDeny-by-default と利便性のトレードオフが必要。

Gemini Notebook

監査・テレメトリと供給網（Supply Chain）の限界



未解決の課題（Future State & Shared Responsibility）



SBOM



Sigstore / 署名



MCP
プロベナンス

バイナリのハッシュ固定は提供されるが、プラットフォーム全体としてのSLSAレベルやMCPサーバー等のプロベナンス保証は、現時点で利用者の責任（AS IS）に委ねられている。

Gemini Notebook

推奨導入プレイブック：多層防御アーキテクチャの段階的構築

1. Phase 1: 資産・権限モデルの定義

エージェント侵害を前提とした最悪ケースに基づく境界設定（Deny-by-default）。

2. Phase 2: OpenShell PoC（非本番）

Linux環境でのサンドボックス化とL4レベルの通信制御の検証。

3. Phase 3: ポリシー狭窄化とクレデンシャル排除

L7へのポリシー絞り込み（enforce化）と、APIキーのSupervisor注入への移行。

4. Phase 4: ProverのCIゲート化と敵対的テスト

Policy Proverをデプロイパイプラインに組み込み権限拡大を



1. Phase 2:
Linux環境でのサンドボックス化とL4レベルの通信制御の検証。

3. Phase 3:
ポリシー狭窄化とクレデンシャル排除
L7へのポリシー絞り込み（enforce化）と、APIキーのSupervisor注入への移行。

4. Phase 4:
ProverのCIゲート化と敵対的テスト
Policy Proverをデプロイパイプラインに組み込み権限拡大を自動ブロック。トランジティブパスを狙う侵入テストを実施。

5. Phase 5: DPU層（Sentry）の追加

金融、重要インフラ、長時間自律エージェントなど、極めて高い脅威モデルを持つ本番環境へのハードウェア監視の導入。

Gemini Notebook

今後の展望：AIエージェント・セキュリティの「次なる開拓地」



Current State
(2026)

MCPとイベントストリームの形式検証

L4/L7の証明から、次世代エージェントの中核となるMCP（Model Context Protocol）のCapabilityレベルの数学的証明への拡張。

Drift（目的逸脱）検知の科学的指標

行動プロファイルを用いたSentryの逸脱検知について、誤検知（FP）と見逃し（FN）のトレードオフを正確に測る標準ベンチマークの確立。

物理AI（ロボティクス）への拡張

ネットワークAPIの拒否だけでなく、ロボットの速度・力・人間への接近を制御するハードリアルタイムなインターロック規格との統合。

Gemini Notebook

結論：封じ込めの設計図をどう評価するか



アラインメントからの脱却

モデル自身に「安全な判断」を委ねる時代は終わった。OS、ネットワーク、ハードウェアのインフラ層での物理的・論理的封じ込め（Defense-in-Depth）こそが次世代の標準となる。

OSSの実力と限界の峻別

OpenShellのソフトウェア境界とクレデンシャル分離は、今すぐ実戦投入すべき強力な基盤である。ただし、L7インスペクションの設定漏れなど、運用上の罠に注意を払う必要がある。

Sentryはビジョンとしての評価

ハードウェア（DPU）によるOut-of-Band監視は究極のセキュリティ境界となり得るが、現時点ではブラックボックスな部分が多い。ベンダーの将来ビジョンと、現在検証可能な機能を切り分けて採用せよ。

Gemini Notebook