

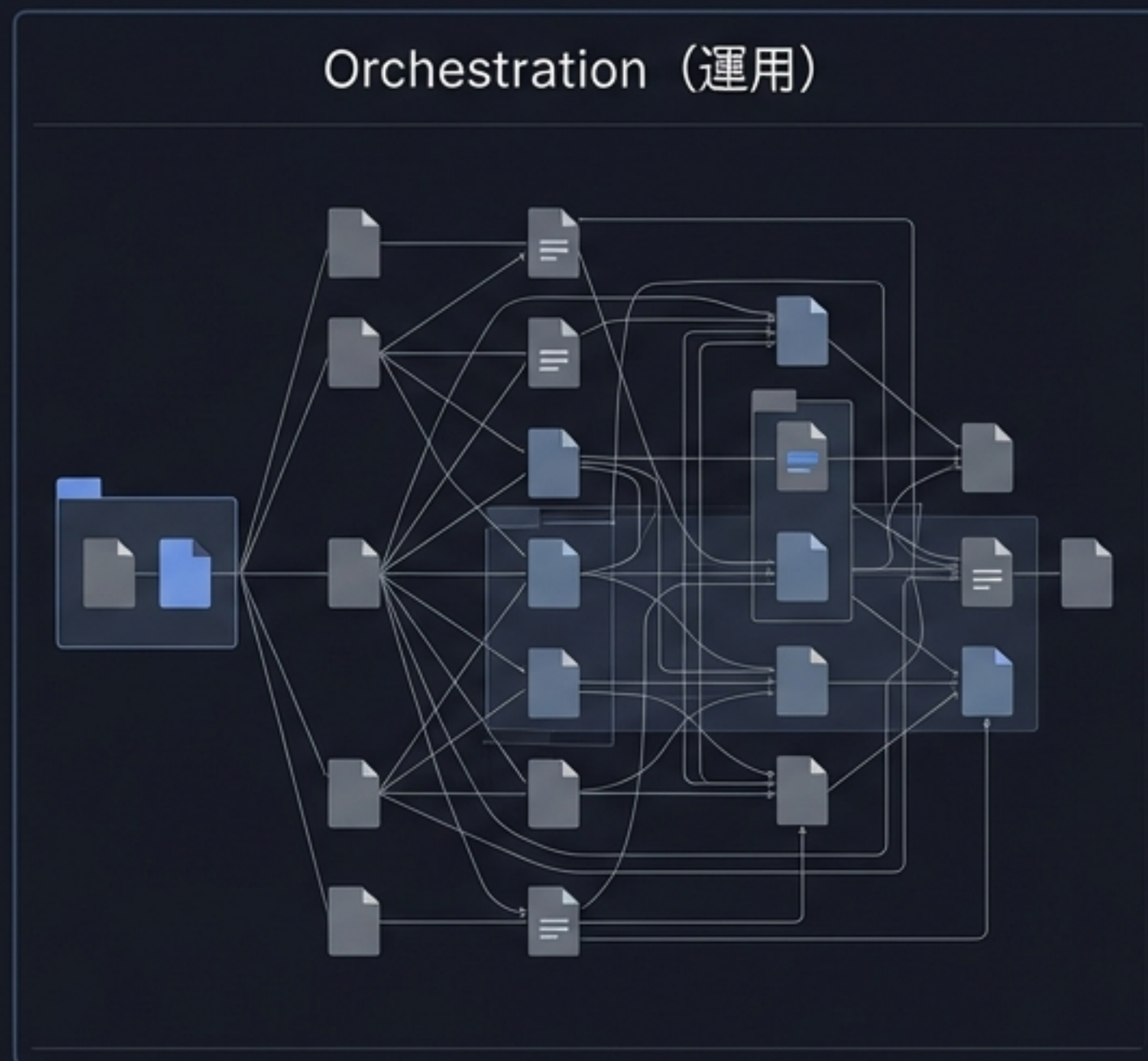
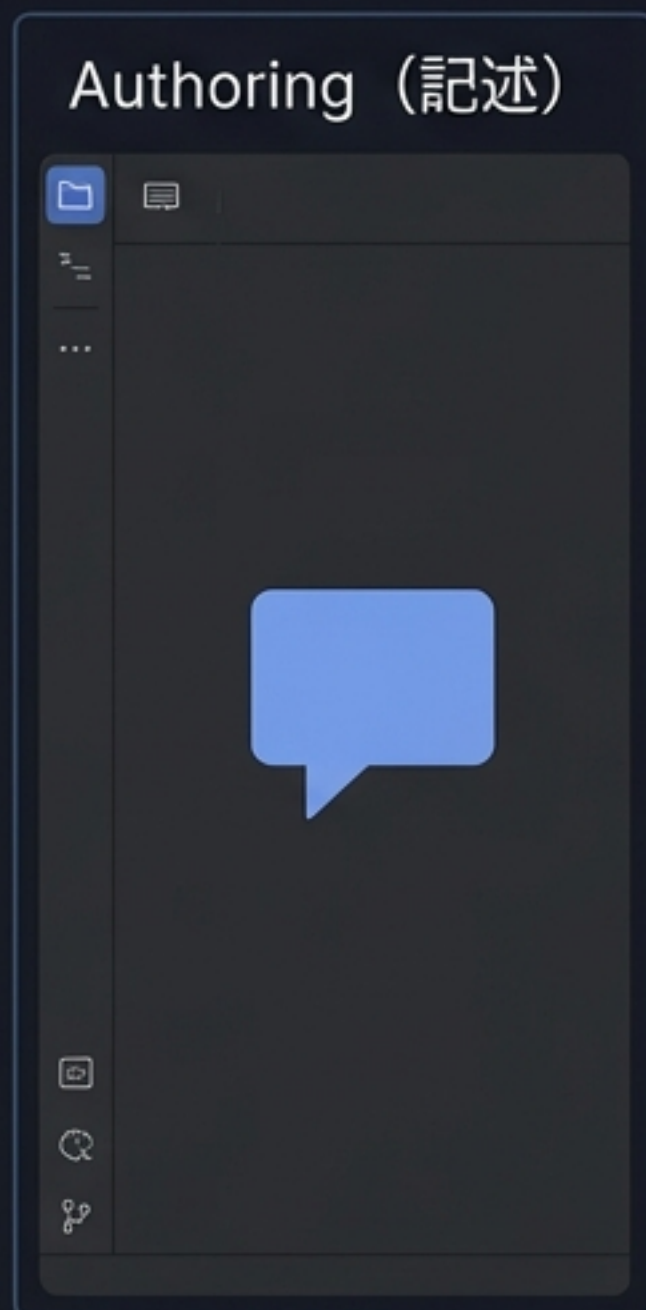
```
> cline init --infrastructure
```

Cline CLI 2.0: AI補助から「開発運用基盤」へ

コードを書く機能から、エージェントを指揮するインフラストラクチャへ

From Writing Code to Directing Intelligence

「サイドバー」の限界：IDEは記述（Authoring）の場所である



1. Scope: サイドバーのチャットは「1ファイル、1カーソル」の制約を受ける。
2. Scale: 20ファイルに及ぶマイグレーションやリファクタリングを一括処理できない。
3. Isolation: CI/CDパイプラインやシェルスクリプトと分断されている。

「構文 (Syntax) ではなく、意図 (Intent) を管理せよ。」

Cline CLI 2.0: Unix哲学への回帰

AIを「ブラックボックス」から「インフラストラクチャのプリミティブ」へ。



```
$ git diff | cline -y "Review this..."
```

Unix Compliant

パイプ処理 (|) やリダイレクトが可能。

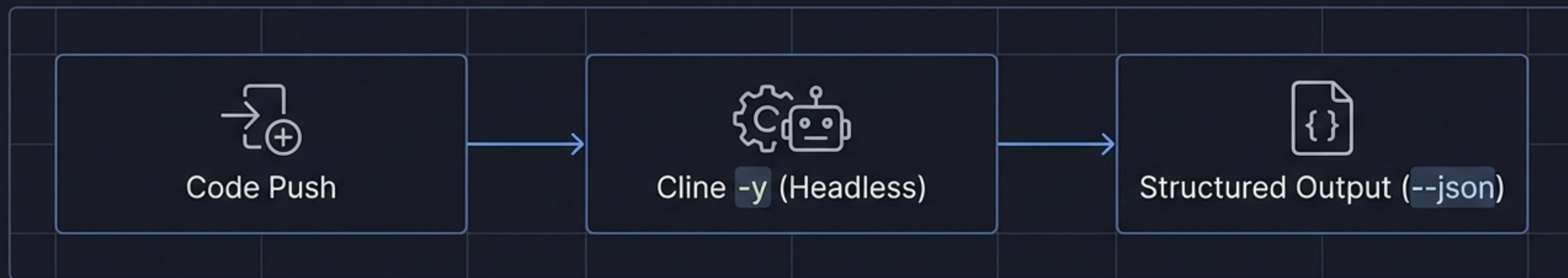
Composable

grep, awk, git とシームレスに結合。

Extensible

既存のシェルスクリプトやCIワークフローに組み込み可能。

Headless Mode: AIをCI/CDパイプラインに組み込む



PR作成時の自動リスク診断

```
Review: git diff origin/main  
| cline -y "Review this..."
```

リリースノート自動生成

```
Docs: git log | cline -y  
"Release notes..."
```

依存パッケージの脆弱性スキャン

```
Audit: cline -y "Check  
CVEs..."
```

人間が寝ている間に、エージェントがコードを修正し、テストを通す。

"Plan & Act" Loop: 思考と実行の分離

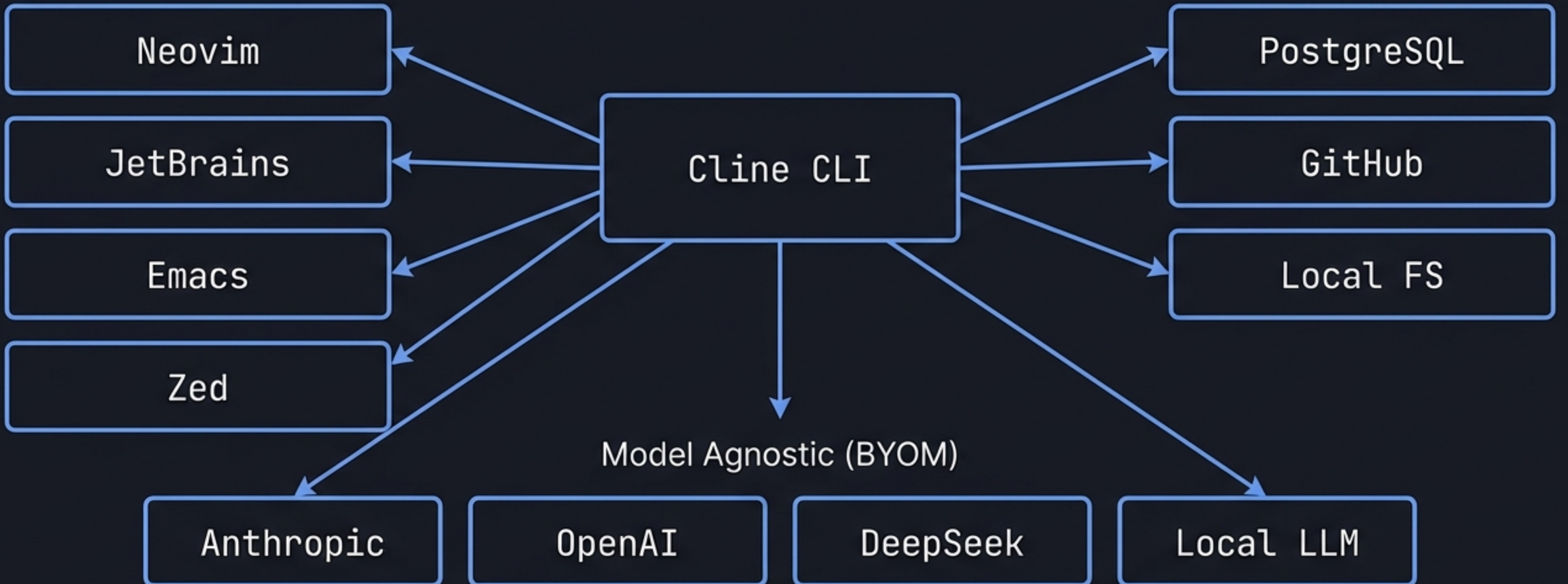


確率的なトークン予測 (Autocomplete) ではなく、論理的な行動計画。

自由と拡張性：ツールロックインからの解放

ACP (Agent Client Protocol)

MCP (Model Context Protocol)



AIツールマトリクス：競合ではなく「適材適所」

GitHub Copilot	Claude Code	Cline CLI 2.0
<p>Role: Speed (IDE内補完)</p> <p>Best For: 日常的なコーディング、数行のロジック作成。</p>	<p>Role: Quality (深い推論・設計)</p> <p>Best For: 複雑なアーキテクチャ設計、仕様の曖昧さ解消。</p>	<p>Role: Operations (自動化・運用)</p> <p>Best For: 並列実行、CI/CD、横断的な修正タスク。</p>

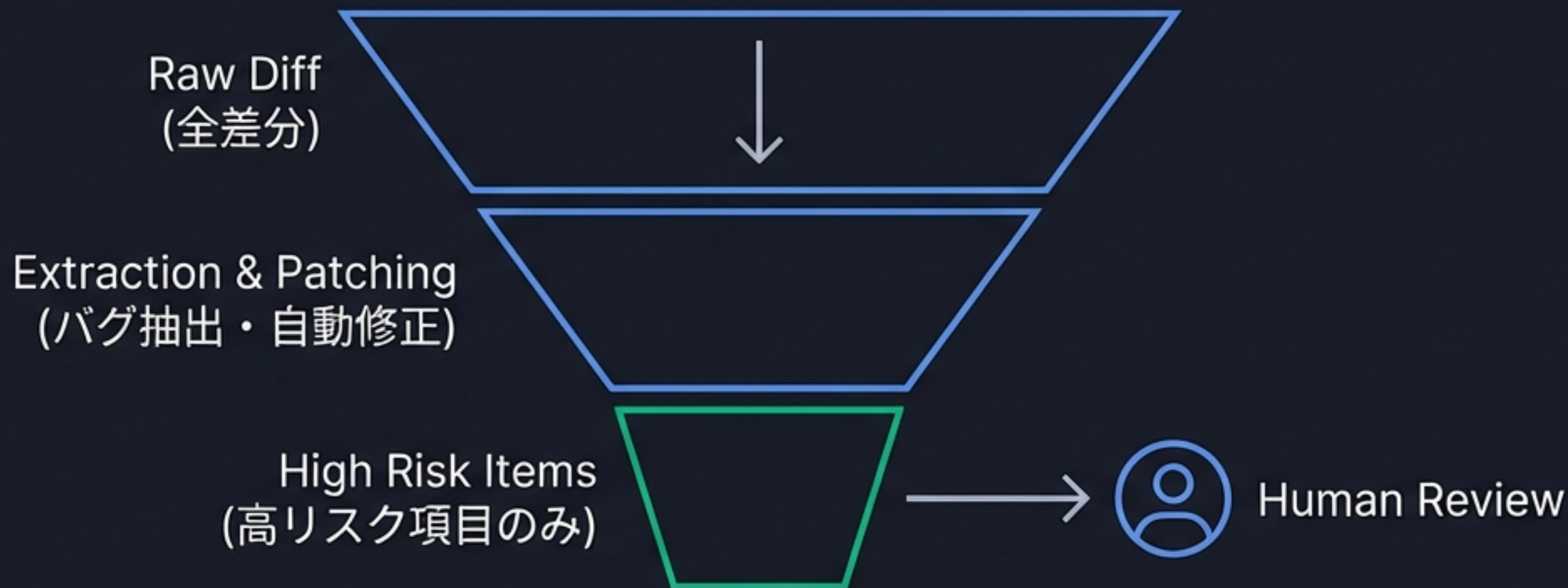
最強の分業フロー：「速度」と「品質」の両立



「Copilotで書き、Clineで回し、Claudeで決める。」

Killer Use-Case: PR前の「自動品質ゲート」

```
git diff origin/main | cline -y "不具合リスクと修正案を出して"
```



レビュー時間を短縮し、人間は「重大な意思決定」のみに集中する。

Guardrails: 安全な運用のための防壁



Permission Control

CLINE_COMMAND_PERMISSIONS

`rm -rf` 禁止, `npm test` 許可



Directory Scoping

Target Isolation

-y 実行時は対象ディレクトリを
厳密に制限



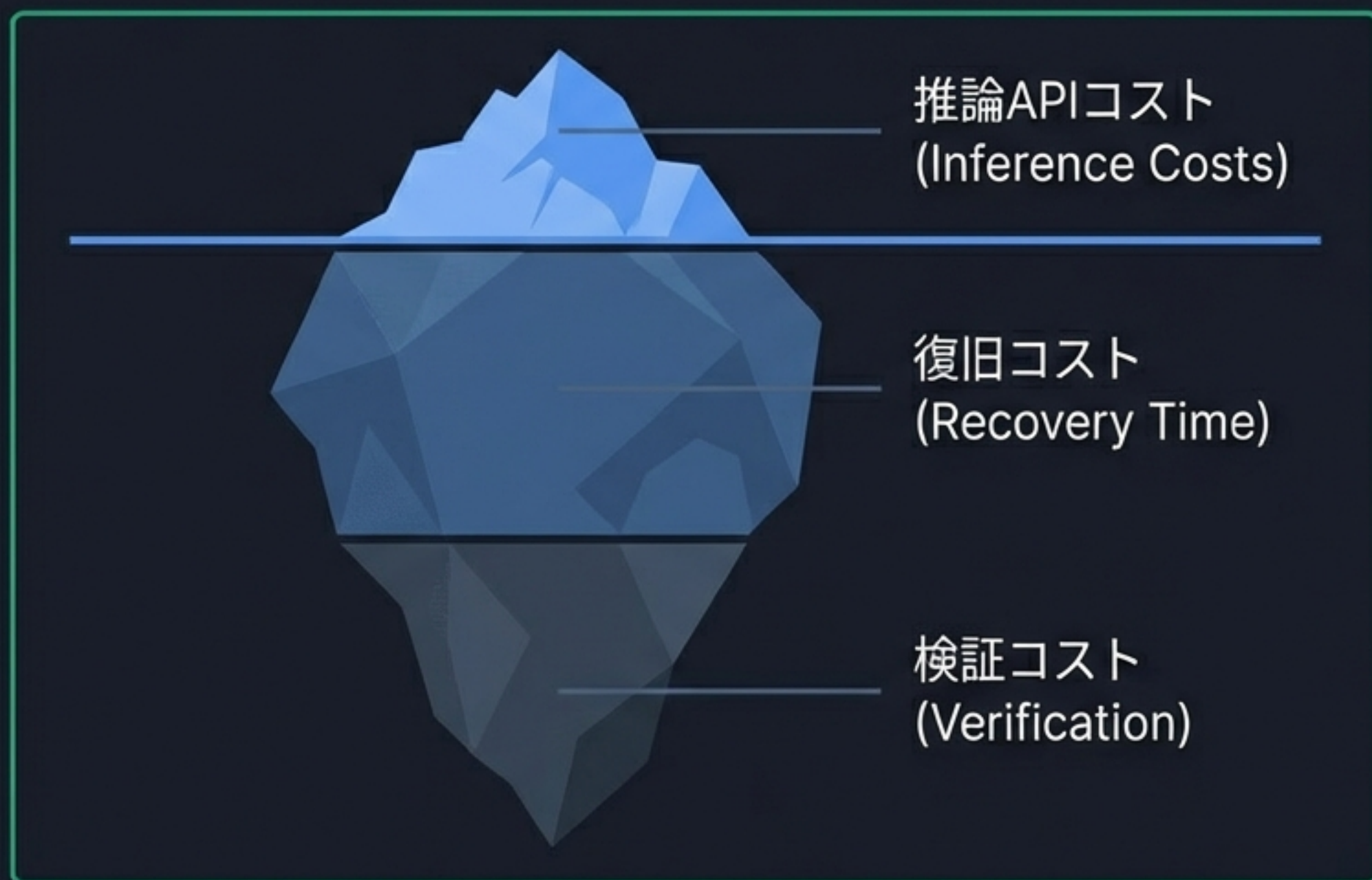
Timeout

Run Limits

無限ループ防止のためのタイム
アウト設定

CIでは「読み取り専用 (Read-only)」から始め、書き込みはサンドボックス内でのみ行う。

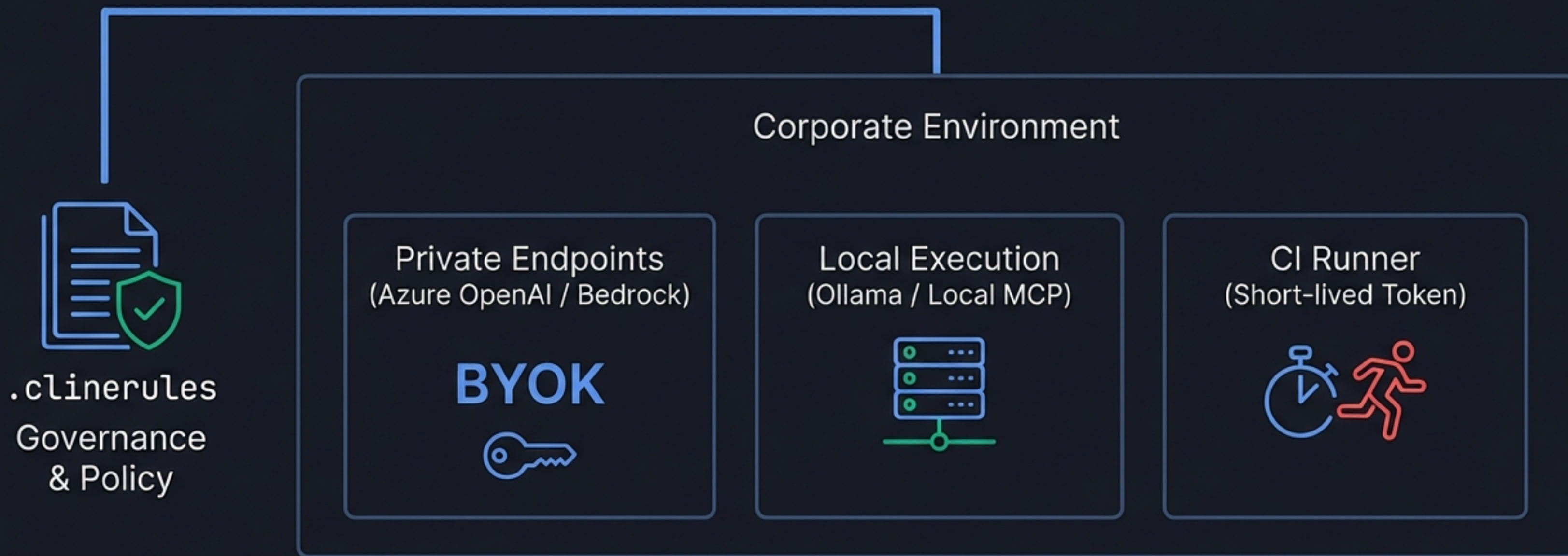
コストとリスクの管理：「無料」の罠



Mitigation Strategy

- ✓ .clinerules によるガバナンス
- 🔒 メインブランチへの直コミット禁止

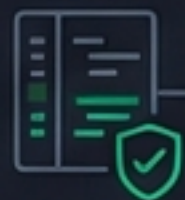
Enterprise Architecture: 組織導入の設計



明日から始める3つのステップ

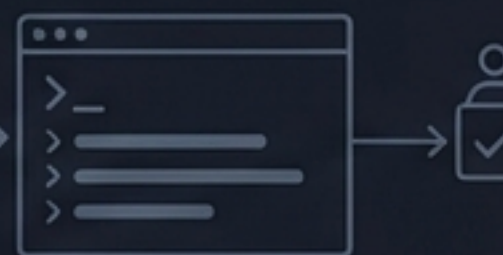
✔ PR Gate (Safe)

git diff を読ませるだけの
「読み取り専用」自動レビュー



⚙️ Bulk Fix (Semi-Auto)

定型作業をCLIで一括実行
(人間が確認)



🤖 Headless CI (Full-Auto)

ガードレールを構築し、
CIパイプラインに組み込み



開発の未来：コードを書く人から、知能を指揮する人へ



エンジニアは「構文」から解放され、純粹な「意図」と「価値」の創造に集中する。

The terminal is ready. Let's build.