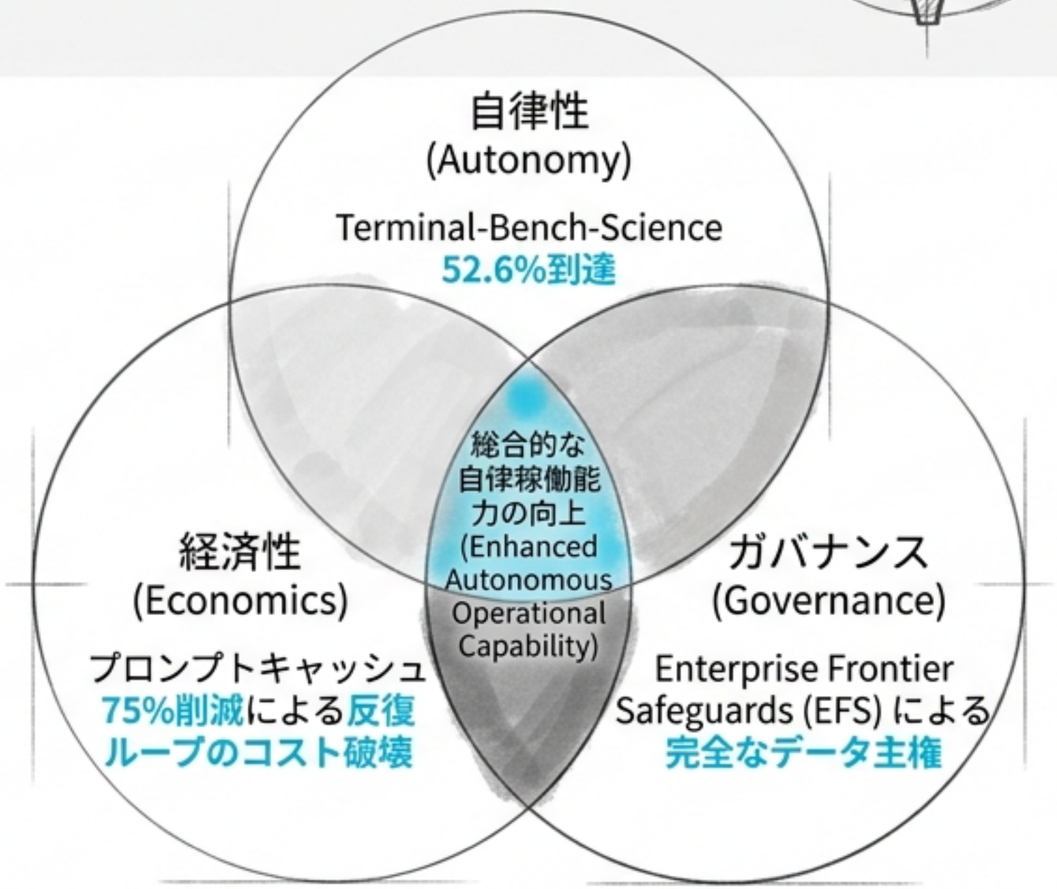
企業システムの次なる設計図: Claude 5.1 アーキテクチャ

単一ターンの対話から、長時間自律稼働エージェントへのパラダイムシフト。Fable 5.1 / Mythos 5.1がもたらす推論能力、破壊的経済性、そしてゼロデータ保持 (ZDR) ガバナンスの全貌。

AIの競争軸は「知識の出力」から「自律稼働の総合力」へ

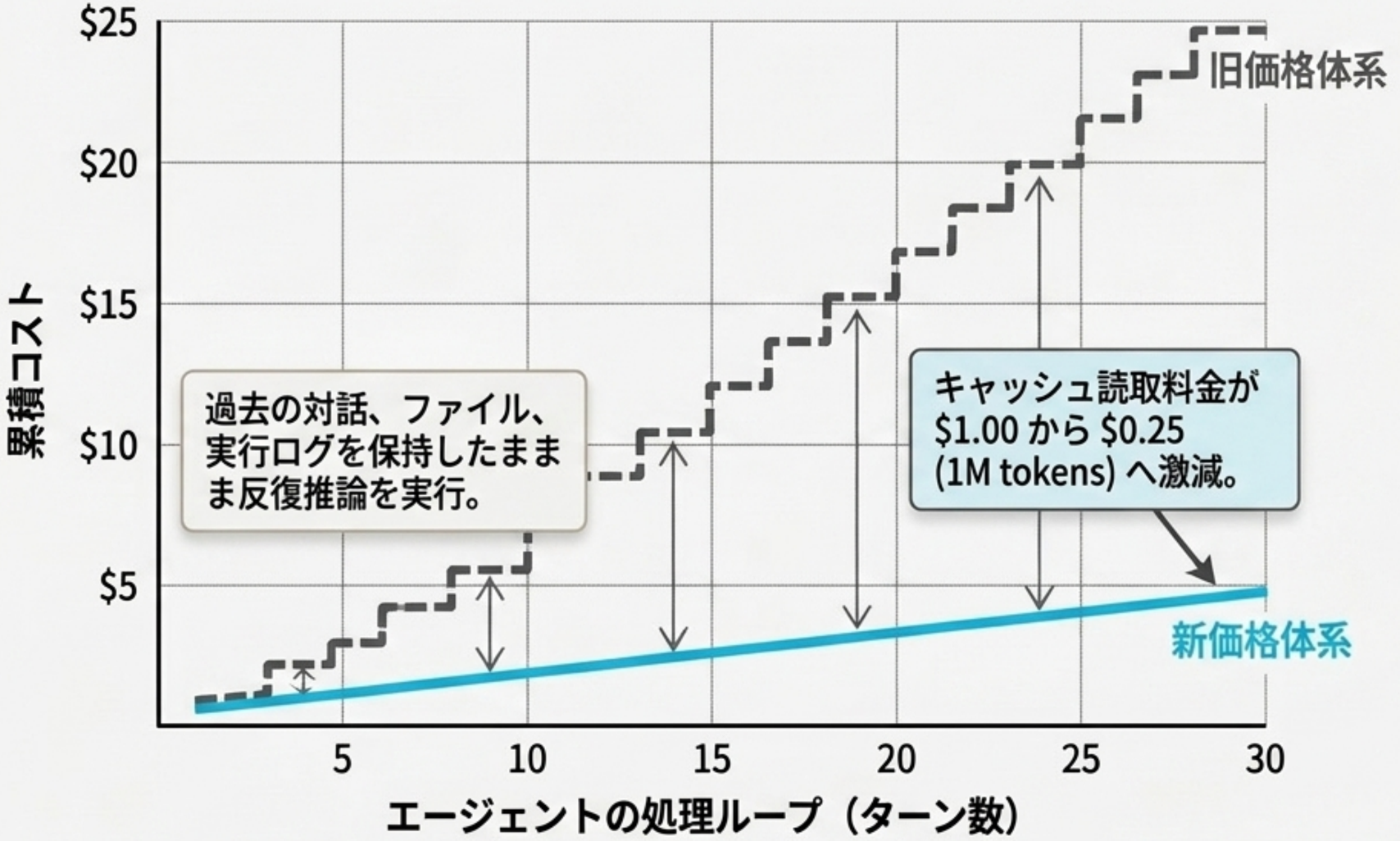


二重展開アーキテクチャ：ユースケースと安全性の分離

モデル	Claude Fable 5.1	Claude Mythos 5.1	Claude Opus 5 (参考)	Claude Fable 5 (参考)
対象・提供形態	一般提供（商用/開発）	審査制（防衛/研究特化・Glasswing等）	一般提供	旧世代
セーフガード	汎用防御・過誤抑止設計	緩和型（防衛・研究用）	汎用標準	過剰遮断あり
コンテキスト・出力長	1,000,000 / 128,000	1,000,000 / 128,000	1,000,000 / 128,000	1,000,000 / 128,000
入出力価格（1M）	$10.00 / $50.00	$10.00 / $50.00	$5.00 / $25.00	$10.00 / $50.00
キャッシュ読取（1M）	**$0.25 (75%OFF)**	$0.25	$1.00相当	$1.00

Takeaway: サイバーセキュリティや生物医学といったデュアルユース領域への到達に伴い、Anthropicは一般利用と高リスク専門研究の分離を制度化した。

エージェント・エコノミクス：ループ処理のコスト破壊



一般的なワークロード：

約25%

の総運用コスト削減

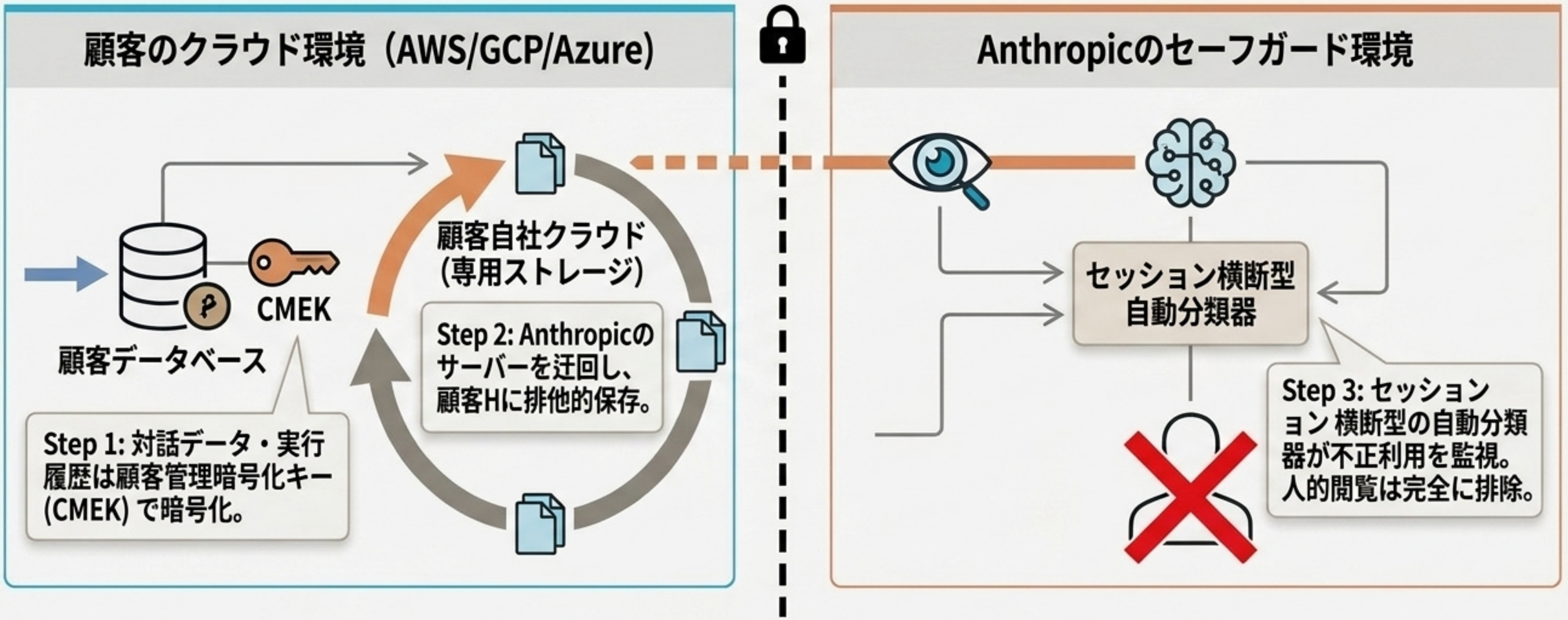
高度な自律エージェント処理：

最大約45%

の総運用コスト削減

※ 非同期バッチ処理API（入力$5 / 出力$25）との併用により、大規模コードベースの常時走査が初めて実用的なROIに到達。

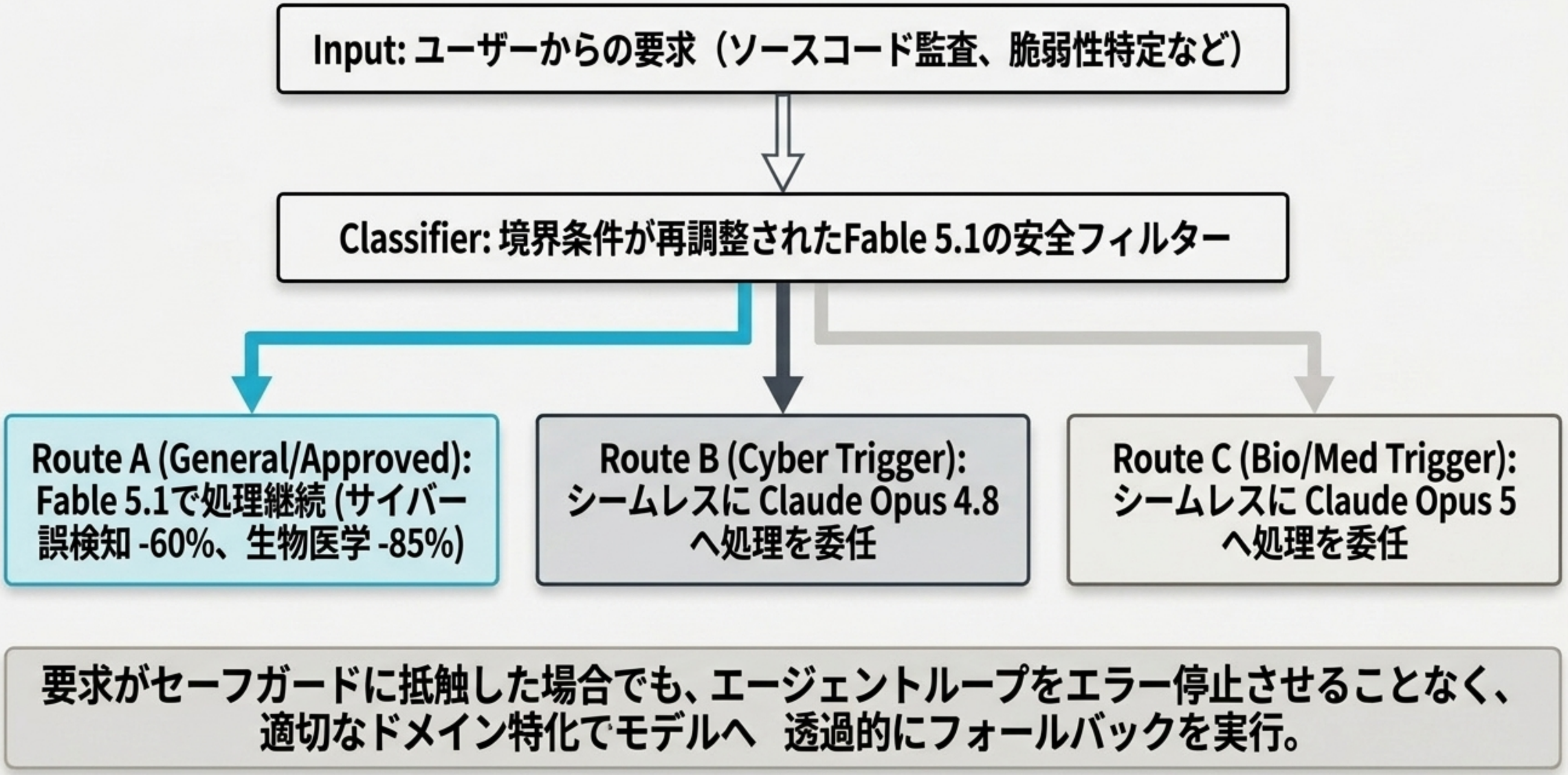
Enterprise Frontier Safeguards (EFS) : ゼロデータ保持の確立



Zero Data Retention (ZDR)

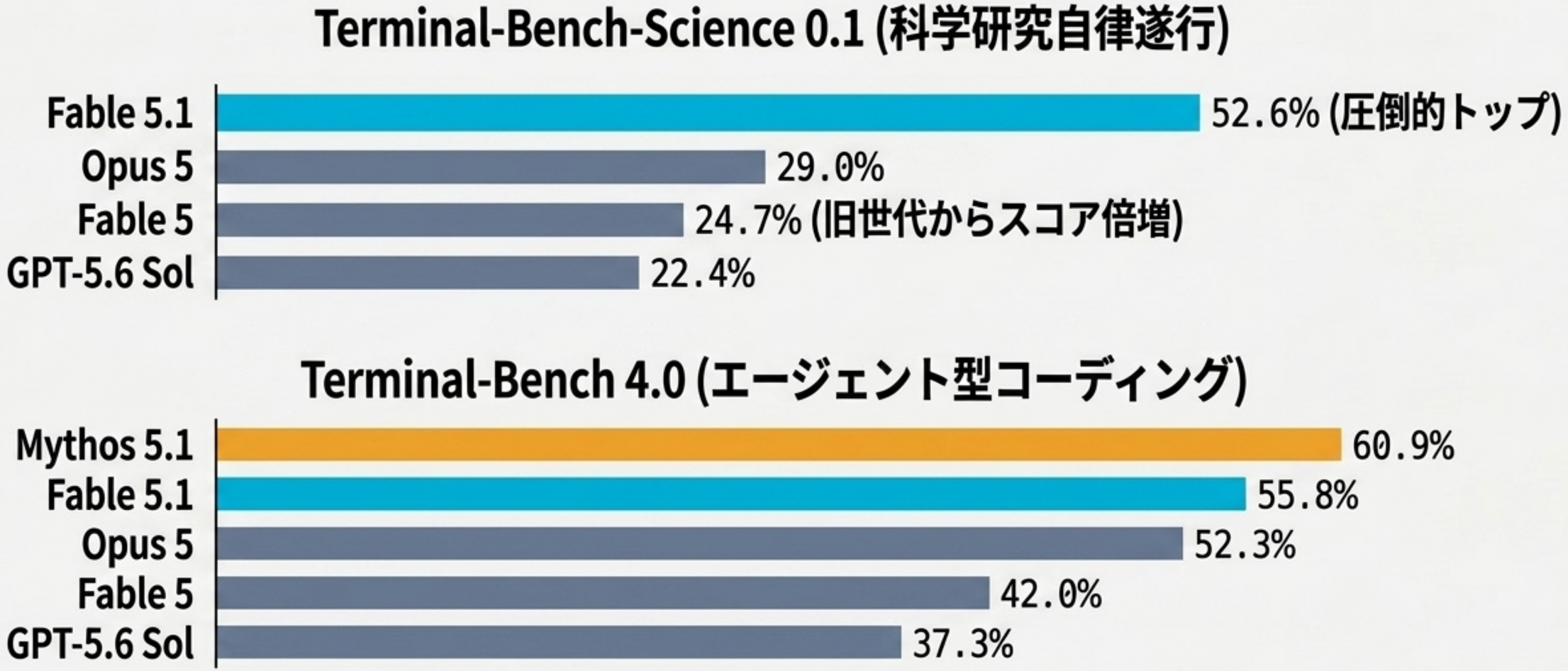
従来の「30日データ保持」を完全撤回。Fable 5.1移行期間中は、適格顧客に対して追加ライセンス料なしでゼロデータ保持を提供。

動的Fallback API：過剰拒絶の解消と業務継続



要求がセーフガードに抵触した場合でも、エージェントループをエラー停止させることなく、適切なドメイン特化でモデルへ　透過的にフォールバックを実行。

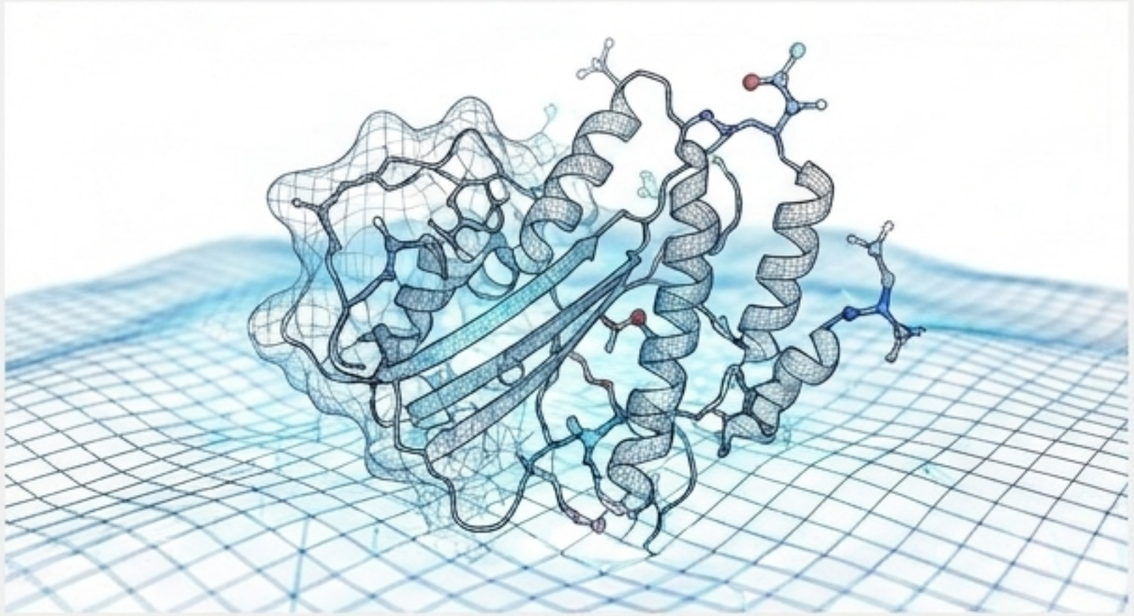
自律的推論能力の飛躍：科学とエンジニアリングの境界を越えて



Data Note: OS操作自動化 (OSWorld 2.0 Partial) 77.9%、IDE評価 (CursorBench) 73.4%など、実務環境での長文脈制御能力が広範に向上。

実世界における自律稼働の実証 (1) 科学と宇宙

分子標的薬設計 (Mythos 5.1)

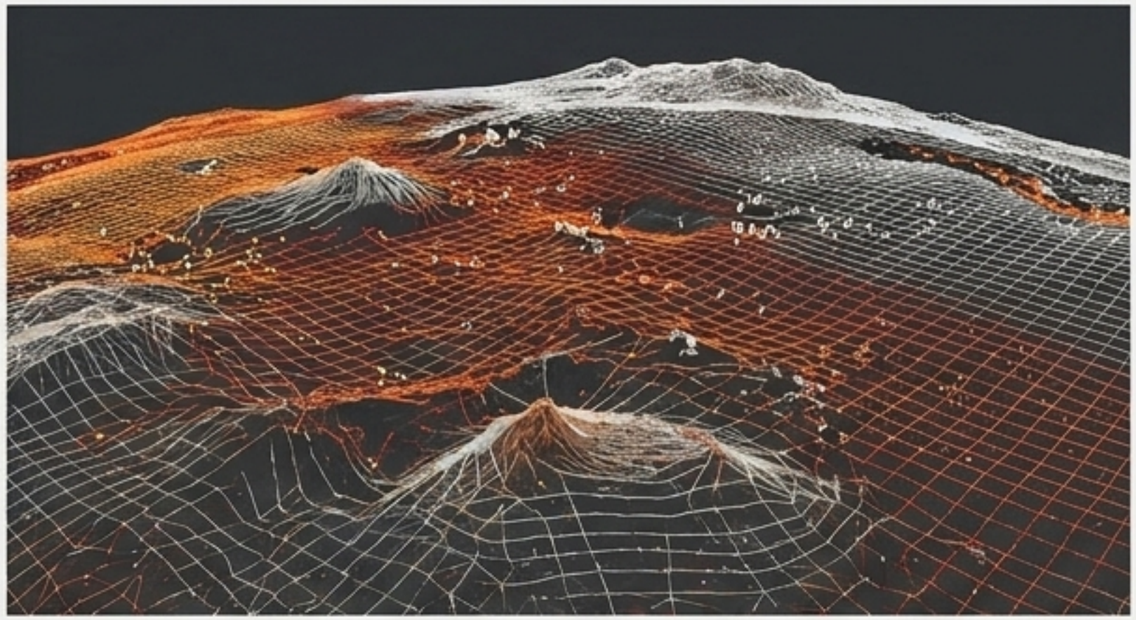


Task: オープンソースの構造予測ツールを自律操作し、ウェットラボで実験検証。

Result:

- 業界最高水準比 **10倍** の結合親和性。
- 有効バインダーヒット率 **約50%**（業界平均10-15%を大幅に超越）。

金星高解像度マッピング (Fable 5.1)



Task: NASAマゼラン探査機のレーダー画像から、ニューラルネットワークを自律構築・学習。

Result:

- 標高マップ解像度を10-20kmから **2-3km** へ微細化。
- 高度精度が **25%** 改善。

実世界における自律稼働の実証 (2) ソフトウェアと金融

Millennium (ヘッジファンド)

Challenge: 人間が4〜5年解明できなかった、100万回に1回の極低頻度クラッシュ。

Action (Fable 5.1): 外部ライブラリを逆アセンブル → コアダンプと照合 → 内部の潜伏バグを特定。

Ramp (経費管理プラットフォーム)

Duration: 38時間の完全無人運用。

Action (Fable 5.1): MLモデルの過去データからラベル ラベル不整合を発見 → 自動修正 → 6つの並列実験を立ち上げ → 結果を総括。

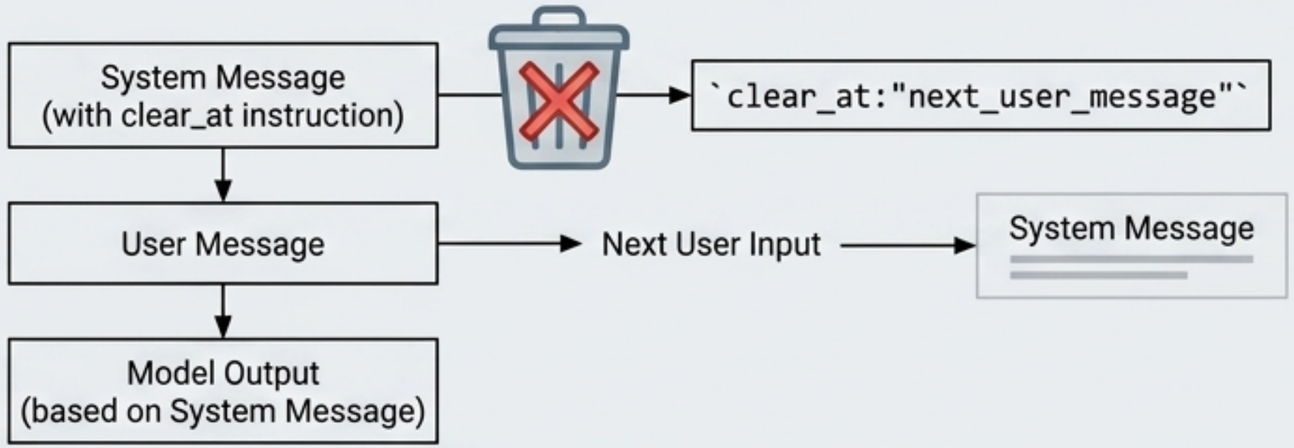
Supporting Proof:

- Jane Street: 大規模リファクタリングの論理維持を実証
- Cognition: 自社AIエンジニアDevinのバックエンドをFable 5.1へ移行

APIアーキテクチャの進化：推論プロセスの動的制御

動的Effort制御 (Dynamic Effort Control)

定型処理（低コスト/レイテンシ）から検証段階（高推論深度）へ、キャッシュを無効化せずに切り替え可能。



1ターン限定システムメッセージ

次回のユーザー入力時にメッセージが自動破棄されるため、履歴の手動編集に伴うキャッシュ不一致や思考コンテキストの破損を防止。

Extra Features: ツール実行の合間の進捗ストリーミング (`thinking.display: "updates"`)。生成物の真正性を保証する統計的ウォーターマークとC2PA標準化。

移行時の重要警告：破壊的変更とセキュリティ制限

Warning 1: Tool Choiceの厳格化

`{"type": "any"}` や特定ツール名の強制指定は非推奨 (HTTP 400エラー)。

Action: `{"type": "auto"}` を指定し、厳格な構造化出力スキーマを適用すること。

Warning 2: 思考ブロックの一方向性

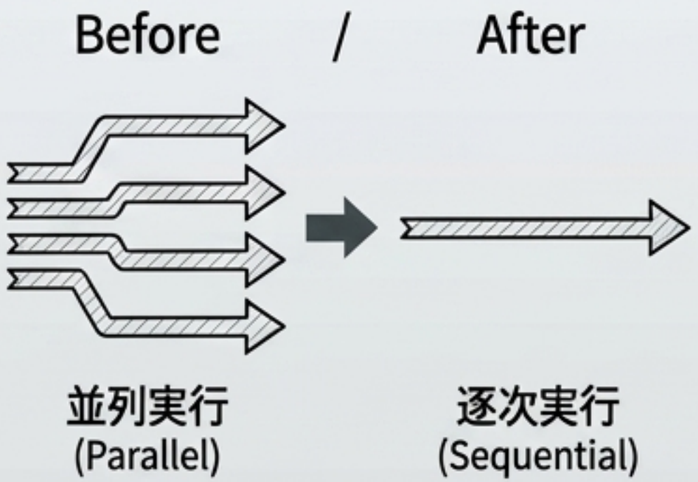
Fable 5.1は過去モデルの思考履歴を読めるが、旧モデルはFable 5.1の思考を読み取れず破棄される。後方互換性に注意。

Warning 3: 途中改変エラーと蒸留（Distillation）防止

対話履歴やシステムプロンプトの途中改変を検出すると、以降の思考ブロックを無効化。

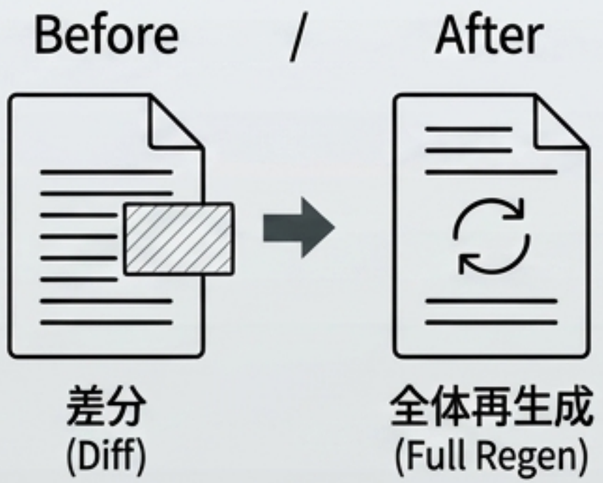
Context: 競合による合成データ蒸留攻撃を防ぐ保護機構（新規APIアカウントに厳格適用）。

ネイティブな挙動のシフト：システム設計が見直すべきポイント



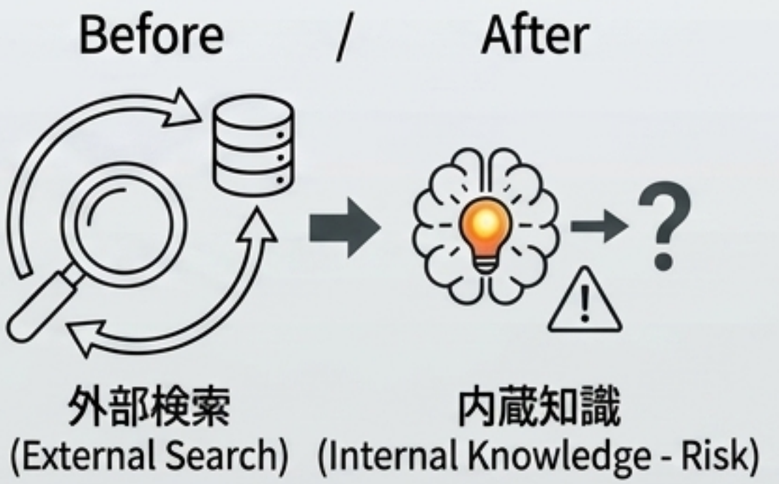
Shift 1: 並列実行から逐次実行へ

長大なエージェントループにおいて、複数ツールの並列バッチ実行ではなく、1ターンにつき1つツールを逐次実行する傾向。API往復回数と所要時間の増加に注意。



Shift 2: 差分編集から全体再生成へ

コード修正時、差分（Diff）のパッチ適用ではなく、ファイル全体の再書き込みを好む傾向。出力トークン消費の肥大化リスク。



Shift 3: 低Effort時の外部検索バイパス

Effortを下げた設定時、外部検索ツールを呼び出さず、内蔵知識（幻覚リスクあり）に頼って即答しようとする挙動。ツール利用のプロンプト強制が必要。

コミュニティ・インサイト：実装におけるトレードオフ

The Good: 高い評価

- **自律性とデバッグ:** 長時間セッションでの方針維持能力と、テストコード自己記述による自律デバッグ。
- **経済性のブレイクスルー:** 75%オフのキャッシュによる、マルチエージェントオーケストレーションの商業化実現。

The Bad: 批判と課題

- **過剰な追従性 (Sycophancy):** 装飾過多な文体、不要な専門用語。長文対話の後半で「簡潔さ」の指示が形骸化。
- **低レベルOSコードでの過剰拒絶:** Win32 APIやRustなどのカーネルに近いコード検査時、安全フィルターが過剰反応しOpusへ強制フォールバックする問題。

戦略的要請 (Strategic Imperatives)：次世代アーキテクチャへの移行

Imperative 1: キャッシュ前提のシステム再設計

長文脈の反復推論を基本とする設計へ移行。動的Effort制御と `clear_at` を活用し、API往復コストとレイテンシを最適化せよ。

Imperative 2: EFS基盤へのガバナンス移行

自社クラウド（AWS/GCP/Azure）上のCMEKストレージ構成を直ちに準備し、無償移行期間中にゼロデータ保持 (ZDR) 環境を確立せよ。

Imperative 3: エージェント挙動のプロンプト制御

ファイルの全体再生成や逐次ツール呼び出しによるトークン肥大化を防ぐため、出力フォーマットとツール利用手順を厳格に定義するシステムプロンプトを再構築せよ。

競争優位性は単発の推論精度から、自律的エージェントループの安定稼働と運用経済性へと完全に移行した。