

新しいチャット

チャットを検索

画像

アプリ

Codex

GPT >

プロジェクト

プロジェクトを新規作成

sss

IDE CLI LLM活用法

Gitリポジトリエラー解消

枝分かれ・設計書SSOTの...

枝分かれ・枝分かれ・設...

枝分かれ・設計書SSOTの...

すべて表示する

仕上げの仕上げ

コーディング環境考察

あ

MCP統一

...もっと見る

あなたのチャット

設計書と監査レポート解説

設計監査成果物作成

MCP設計影響確認

ローカルGitHub運用改善

設計監査依頼事項

ツール統合提案調査

Git運用設計最適化

枝分かれ・MCPサーバー立て方

MCPサーバー立て方

枝分かれ・設計書作成ツール提...

枝分かれ・設計書作成ツール提...

MCPサーバーの料金体系

優しい鍋風レシピ

使い方の説明

CLI MCPで性能向上

GitHubリポジトリ表示問題

開発者モード説明

WSL起動トラブル解決法

CLI性能比較

会話のズレ修正方法

VS CODEとCursorとAntigravity

VSコード IDE機能

2026年目標整理

枝分かれ・開発運用強化調査

開発運用強化調査

枝分かれ・開発運用強化調査

hiroto katou

Plus

1. Research (最新情報取得)

- MCP/検索（例：Gemini CLIはMCP連携前提）で一次情報に当たる。 Google for Dev...

2. Freeze (根拠の固定化)

- 取得したページ/抜粋/要点を sources/ に保存し、**取得日・タイトル・要約・重要引用（短く）**を添える。

3. Write (設計書更新：分割・差分最小)

- CLIエージェントに「更新対象Part」「追加/置換する見出し」「根拠（sources）だけ」を渡して書かせる。

4. Verify (機械検証+証跡)

- リンクチェック、整合性チェック、禁止事項チェック等のVerifyを回してログを残す（あなたのSSOT思想と相性◎）。

5. Review (AIレビュー+人間承認)

- Codex CLIの /review のように、差分ベースでレビューを掛ける。 OpenAI Develop...

6. Release (コミット/タグ/証跡保持)

- “何を根拠に何を変えたか”が追える形で確定。

このフローで、ハルシネーションは「根拠の材料をSSOTに固定」することで構造的に抑えられます（気合ではなく仕組み）。

Model Context ... +1

4) 2026運用で“事故を減らす”必須ガード（MCP込み）

MCPはJSON-RPCベースのクライアント/ホスト/サーバ構造で、境界と権限設計が重要です。 Model Context ...

特に外部取得・APIキー・サーバ権限が絡るので、公式の **Security Best Practices**（段階的最小権限、スコープ設計など）は設計書に明記すべきです。 Model Context ...

また、MCPサーバのデバッグ/検証は **MCP Inspector** が公式に用意されており、接続・ツール呼び出し・レスポンス確認を体験的に行えます。 Model Context ... +1

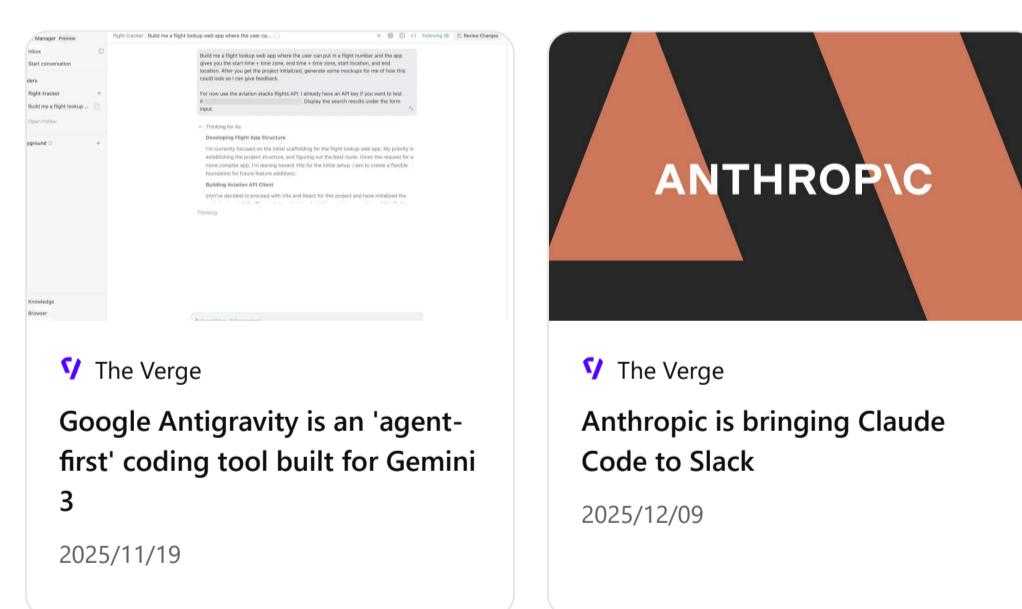
5) ツールの“最適役割分担”（おすすめ）

- 執筆（SSOT更新の主役）：Claude Code か Codex CLI
 - テスト実行→修正反復の運用Tipsが公式にあるのはClaude Code側。 Anthropic
 - 差分レビュー導線（/review）が明確なのはCodex CLI側。 OpenAI Develop...
- 最新情報の取り込み・外部連携（研究役）：Gemini CLI（MCP前提の設計） Google Cloud D...
- MCPサーバの検証：MCP Inspector Model Context ... +1
- IDE（編集・局所修正・レビュー）：VS Code / Cursor（ただしworktree等で衝突回避） Visual Studio Co...

最終まとめ（あなたの質問への答え）

- 今回のIDE + CLI中心は“最善に近い”。
- 精度をさらに上げる鍵は、**「最新情報をMCP/検索で取得→SSOTに固定→その根拠だけで書く→Verify/Reviewでゲート」**に寄せせること。 Model Context ... +2
- **ローカルLLMは補助（内部文書の加工）**に寄せるのが最適（最新情報確定には不利）。

最近のエージェント開発ツール動向（参考）



□ △ □ △ ... 🔍 情報源



ウェブを検索する

+ ⚙ Extended thinking ▾ 検索

0