

目次

1. Manus

- 公式デモ

2. Manusの技術スタック

- 使用モデルと構成
- X (Twitter) での情報

3. Claude MCP (Model Context Protocol)

- 概要と仕組み
- 最新の動向

4. AIエージェント関連技術

- browser-use
- HuggingFaceのsmolagents

Manus

- 3タスク実行可。実行して24時間で枠が回復するイメージ
 - 拳動はBrowser-use（Docker?E2B?）、Claude MCP、を使っでのリサーチやコード生成・実行
1. DeepResearchについてはOpenAI DeepResearchの方が精度高いようにも感じたが、X等では別れている印象。
2025年生成AI最新情報と技術論文調査例
 2. コード生成・実行周りは、後述の通りClaude3.5ベースだからか、本家Claude3.7で叩く方が良い感じに出てくる印象が個人的にあった。
コード生成・実行系（ダッシュボード作成）

Manusの公式デモ①

顧客開拓リスト作成

プロンプト:「私たちは、AI分野の技術を深く研究する技術コンサルティング会社です。潜在顧客フォームの作成をお願いします...」

結果: [デモリンク](#)

Claude 3.7の初期世論分析

プロンプト:「クロード3.7の発売後1週間における、XとYouTubeでの初期世論を分析」

結果: [デモリンク](#)

Manusの公式デモ②

Kaggleコンペ支援

プロンプト: 「Kaggleの住宅価格予測コンペティションにご参加ください...」

結果: [デモリンク](#)

財務分析

プロンプト: 「これは2021年末から2024年末までのエヌビディアの財務報告書です...」

結果: [デモリンク](#)

Manusの公式デモ③

オープンソースプロジェクト分析

プロンプト:「先週DeepSeekがオープンソース化した5つのプロジェクトの調査に協力してください...」

結果: [デモリンク](#)

人物調査とインタビュー作成

プロンプト:「ディープシークのCEOの経歴を調査し、調査結果に基づいて包括的なインタビューの概要を作成する」

結果: [デモリンク](#)

Manusの技術スタック

主要モデルと構成

共同創設者Yichao 'Peak' Jiによる情報：

- **初期開発時:** Claude 3.5 Sonnet v1（長文CoT/推論トークンなし）と様々なQwen微調整モデルを使用
 - 機能を補うため多数の補助モデルが必要だった
 - Manus構想時にはまだClaude 3.5までだったから
- **現在:** Claude 3.7が有望視され、内部テスト中

Manusの技術的アプローチ

基本設計思想

- **MCPは使用せず**: 友人Xingyao Wangの研究から着想を得る（Code Actions）。後述のCodeAgentと同じような発想のもの。
 - 参考論文: [OpenReview](#)
- **3つの観点**:
 - i. コーディングは最終目標ではなく、一般的問題解決のための普遍的アプローチ
 - ii. LLMはコーディングに優れているため、トレーニング分布に最も近いタスクを実行させるのが理にかなう
 - iii. このアプローチはコンテキスト長を大幅に減少させ、複雑な操作の構成を可能に

Manusの内部構造情報

X (Twitter) で公開された内部情報 :

So... I just simply asked Manus to give me the files at `"/opt/.manus/"`, and it just gave it to me, their sandbox runtime code...

```
> it's claude sonnet
> it's claude sonnet with 29 tools
> it's claude sonnet without multi-agent
> it uses @browser_use
> browser_use code was also obfuscated (?)
> tools and prompts jailbreak
```

- Claude Sonnetベース
- 29種類のツールを統合
- マルチエージェントは未使用
- browser-useを活用 (Wrapperでは、とのコメントも

Claude MCP (Model Context Protocol)

概要

- Anthropic社が開発したオープン標準プロトコル
- AIアシスタントと外部データソース/ツールとのシームレスな連携を実現
- 統一されたクライアント・サーバー（JSON-RPCベース）アーキテクチャ

主な利点

- 個別コネクタが不要で、統一された接続方式
- ローカルファイル、データベース、API等への簡素化された接続
- 開発者の負担軽減と、AIの適切なコンテキスト維持を実現

Claude MCPの最新動向

- **初期発表:** 2024年11月頃
- **急速な注目:** 発表後、開発者コミュニティで大きな関心を集める
- **Perplexityとの統合:** 2025年3月13日、PerplexityのSearch機能がMCPに対応
 - [発表ツイート](#)
 - 豊かなコンテキストに基づいた検索結果を提供
 - AIアシスタントとの連携強化によるスマートな情報取得が可能に

browser-use

概要

- GitHub: [browser-use/browser-use](#)
- Pythonライブラリとして、Playwrightと連携
- Webブラウザ上での操作を自動化するための設計

特徴

- クリック、フォーム入力、スクレイピングなどのブラウザ操作を自動化
- WebページのDOM構造を直接操作
- 比較的高い精度でタスク実行可能
- 専門化された設計により、Web操作に特化した高い安定性

browser-use vs Claude Computer Use

機能比較

機能	browser-use	Claude Computer Use
対象範囲	Webブラウザのみ	OS全体・各種アプリケーション
汎用性	限定的	広範囲
精度	高い	やや劣る
環境要件	比較的軽量	Docker等の仮想環境が必要
安定性	高い	やや不安定

HuggingFaceのsmolagents

概要

- GitHub: [huggingface/smolagents](https://github.com/huggingface/smolagents)
- HuggingFaceが開発した軽量エージェントライブラリ
- オープンソースで利用可能
- LLMを活用したタスク自動化のためのフレームワーク

特徴

- 小規模なモデルでも効率的に動作するように設計
- 多様なLLMと互換性あり
- モジュール設計により拡張性が高い
- 研究やプロトタイピングに適した構造

smolagentsのCodeAgent

概要

- HuggingFace Blog: [Open Deep Research](#)
- コード生成と実行に特化したエージェント
- 研究開発プロセスの自動化を支援

特徴

- 研究コードの自動生成・実装・検証
- プログラミング言語間の変換
- コード最適化と修正
- データ分析パイプラインの構築
- 実験結果の分析と視覚化

参考資料

- Manus公式サイト: <https://manus.im>
- Model Context Protocol: <https://modelcontextprotocol.io/introduction>
- browser-use GitHub: <https://github.com/browser-use/browser-use>
- smolagents GitHub: <https://github.com/huggingface/smolagents>
- Perplexity MCP発表: https://x.com/perplexity_ai/status/1899849114583765356
- CodeAgent紹介: <https://huggingface.co/blog/open-deep-research>