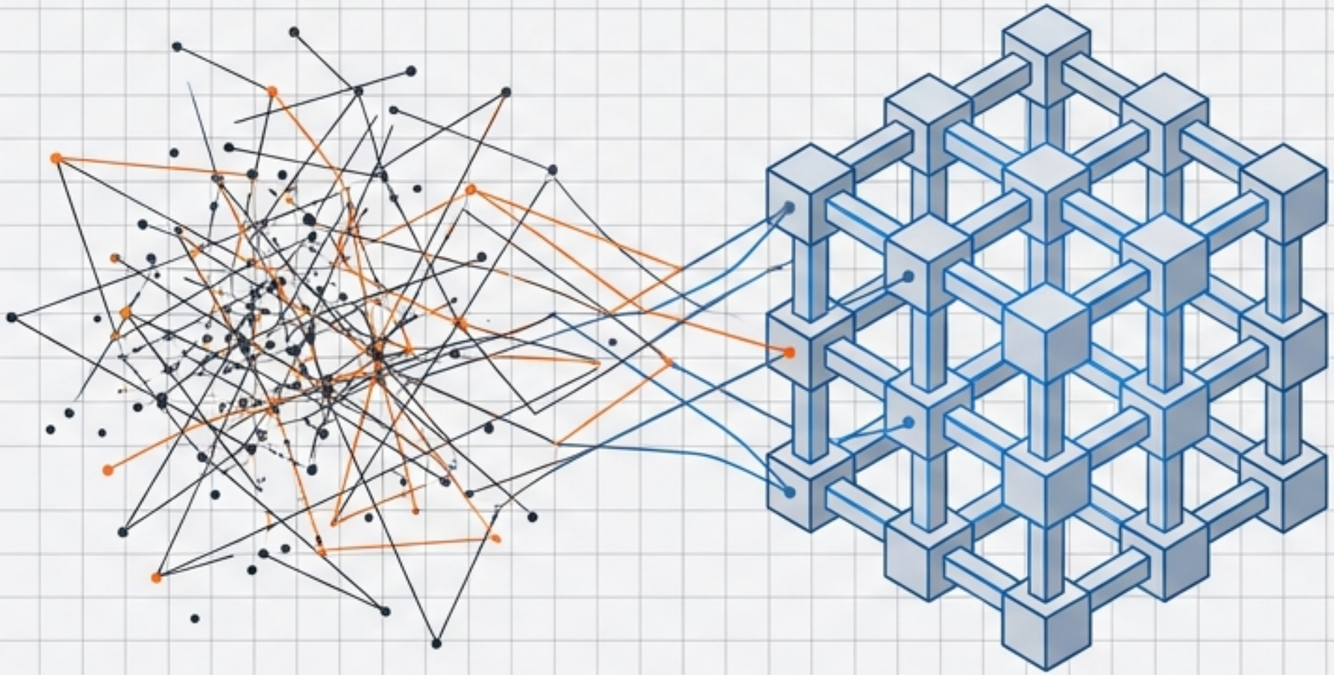
FEBRUARY 2026 UPDATE REPORT

OpenClaw: 実験から産業インフラへ

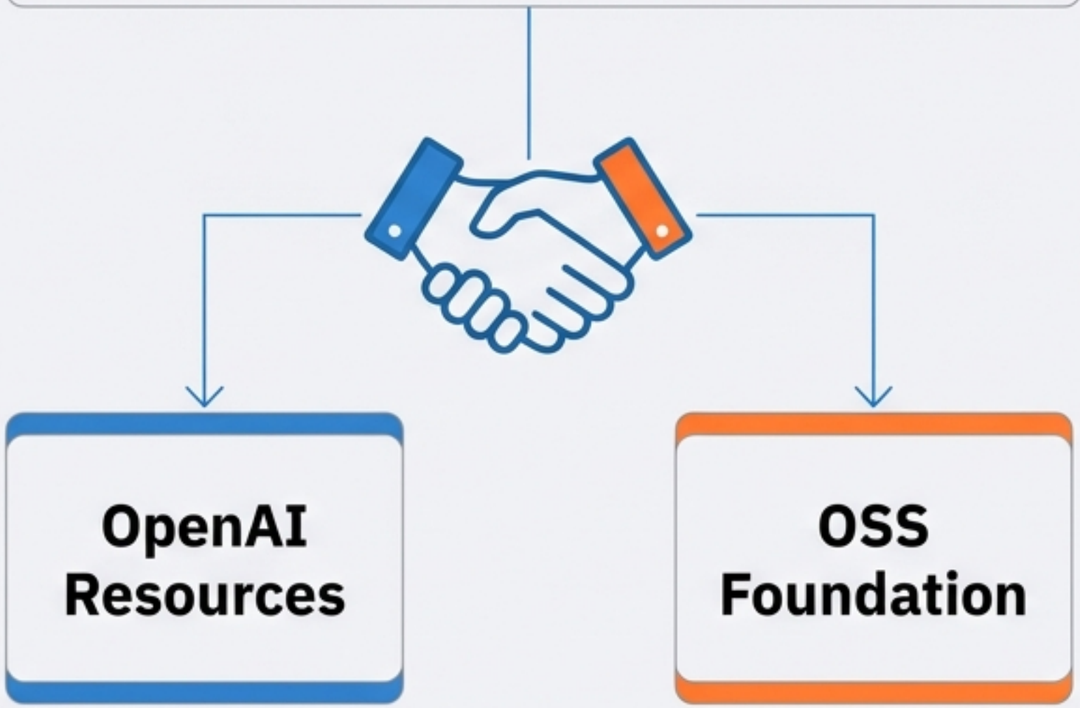
v2.12–v2.15 アップデート詳細分析と「Agentic Engineering」の未来



OpenClaw Project Update

Sam Altman: Peter Steinberger joins OpenAI to build personal agents.

OpenClaw will continue as an OSS Foundation.



戦略的転換点：OpenAI参画とFoundation化

- **Peter Steinberger氏の決断: OpenClaw創設者がOpenAIに正式参画。「Personal Agents」技術の加速を担う。**
- **OSSとしての永続性: 買収によるクローズド化ではなく、独立したFoundation（財団）としてOSS継続が決定。**
- **実務へのインパクト:**
  - **開発停止リスクの払拭: 「急に消えるかもしれない」という心理的ハードルが解消。**
  - **企業導入の加速: 社内PoCから本番運用への移行判断が容易に。**

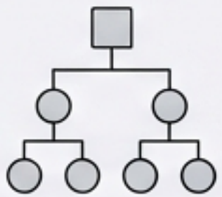
進化の4つの柱：モデル性能から「運用UX」へ

1. 組織・戦略

OpenAI支援とFoundation化による長期安定性。

2. エンタープライズ・セキュリティ

RCE/SSRF対策とサンドボックスの堅牢化 (v2.12)。



3. アーキテクチャ

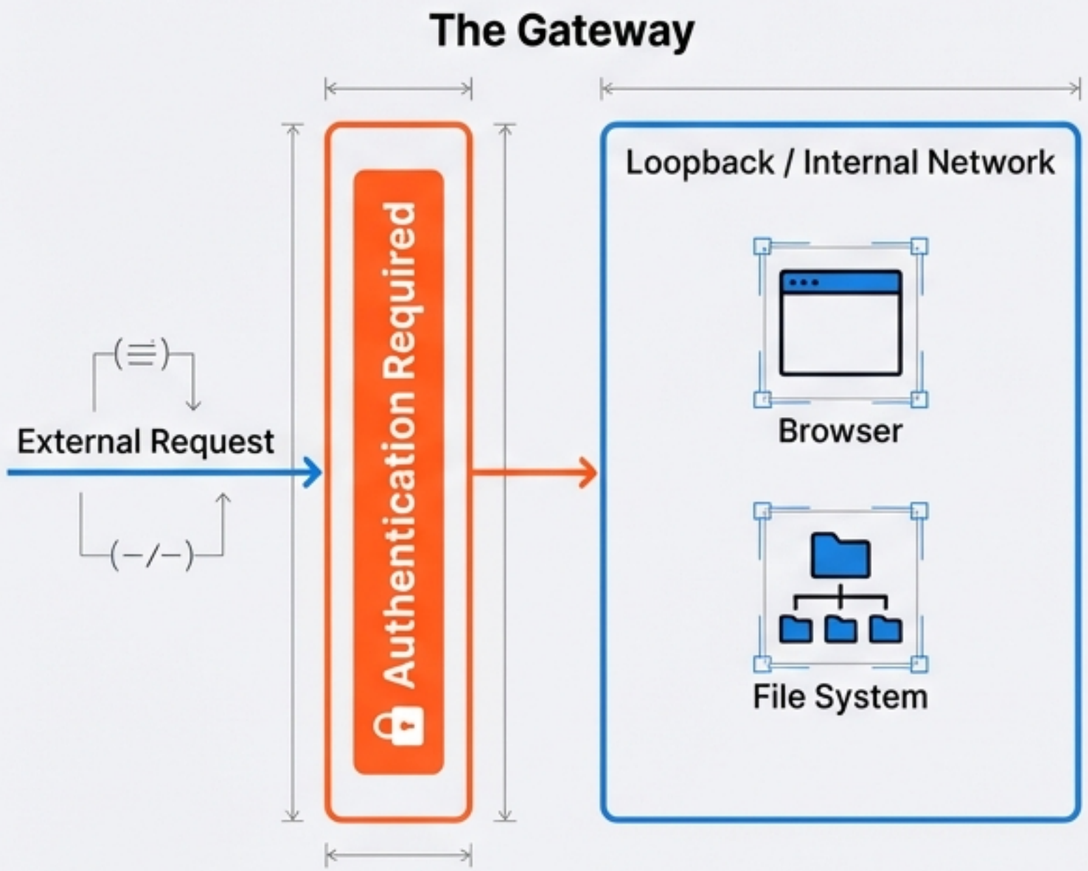
単体エージェントから「群知能（Swarms）」へ（v2.15）。

4. UX・インターフェース

テキストコマンドから「AppOps」へ（Discord Components v2）。

Insight: 競争軸は「どれだけ賢いか」から「どれだけ安全に業務に組み込めるか」へシフトしている。

v2.12 セキュリティ刷新：致命的な脆弱性の排除



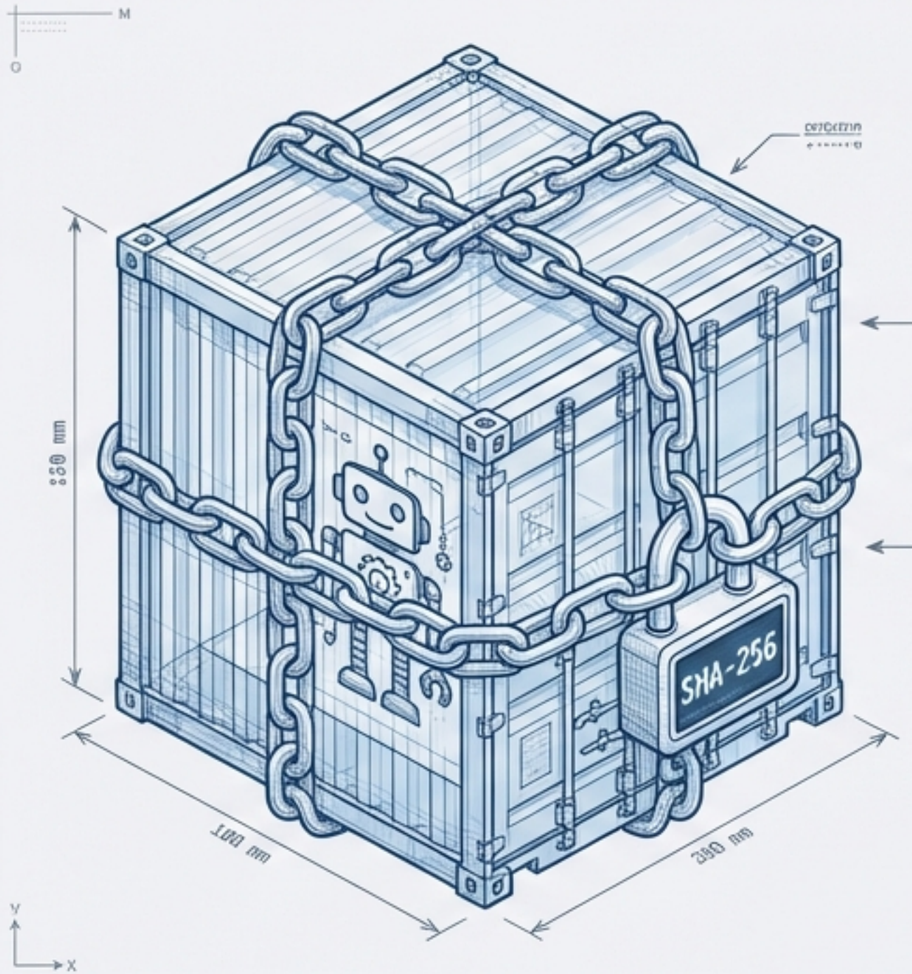
1. **RCE（Remote Code Execution）対策**
   ブラウザ制御のループバック接続に認証を必須化。
   外部からのワンクリックRCEリスクを遮断。

2. **SSRF（Server-Side Request Forgery）対策**
   `input_file` / `input_image` に対するAllowlist導入。
   内部ネットワークスキャン（169.254.x.x等）の拒否ポリシー適用。

3. **セッションハイジャック対策**
   `hooks/agent` エンドポイントでの `sessionKey` 上書きをデフォルトで拒否。

WARNING: 運用中の環境は v2.15+ への即時アップデートを推奨。

サンドボックスの堅牢化と「決定論的安全性」



Config

Hash(Config + Order)

Order

= Unique ID

SHA-1 → SHA-256:
サンドボックスID生成ハッシュを強化。構成の衝突攻撃を暗号学的に排除。

コンテナエスケープ防止:
危険なDocker構成（Bind Mounts, Host Networking）を構成レベルでブロック。

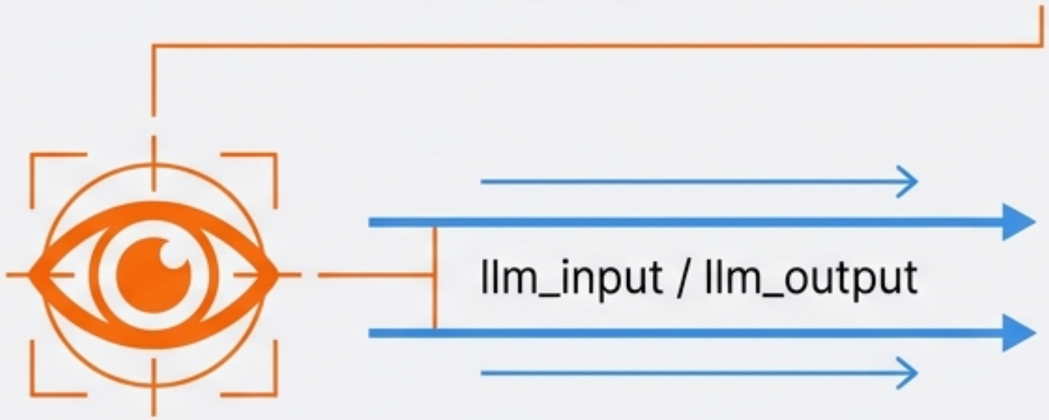
構成順序の保存:
配列順序をハッシュに含めることで、微細な構成差異による意図しないコンテナ再利用を防止。

⚠ **LLMが騙されて危険なコマンドを発行しても、システム側（ランタイム）で物理的に拒否するガードレール。**

可観測性とコンプライアンス

```
[INFO] [2024-07-25T10:15:30Z] User 'alice'
initiated request. API_TOKEN=[REDACTED]

[ERROR] Stack trace:
  ...
  File "main.py", line 45, in process_request,
    token = '[REDACTED]'
```

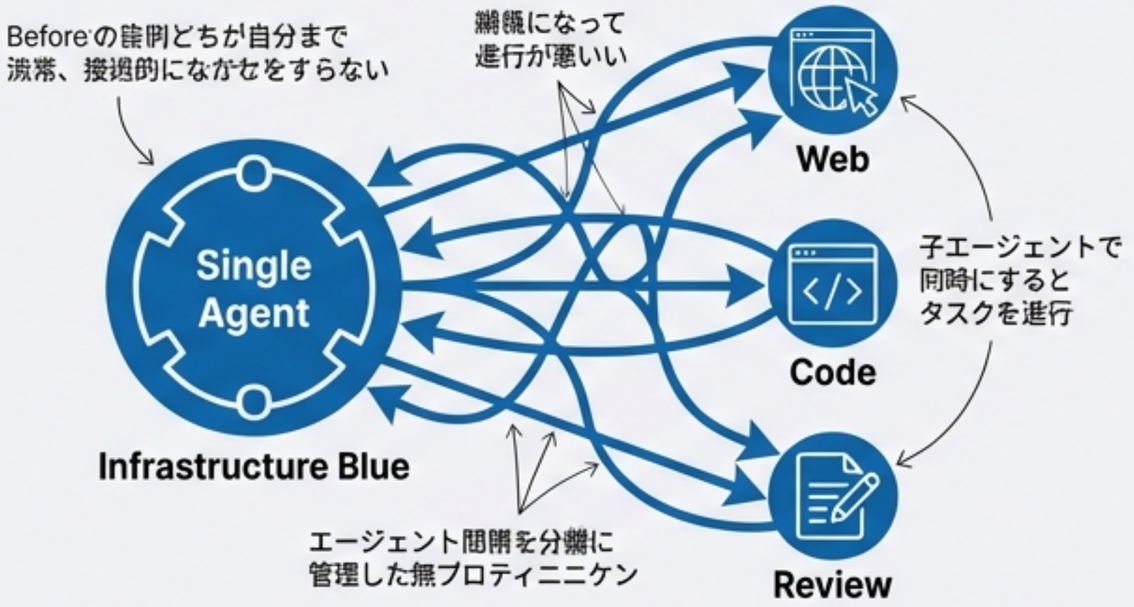


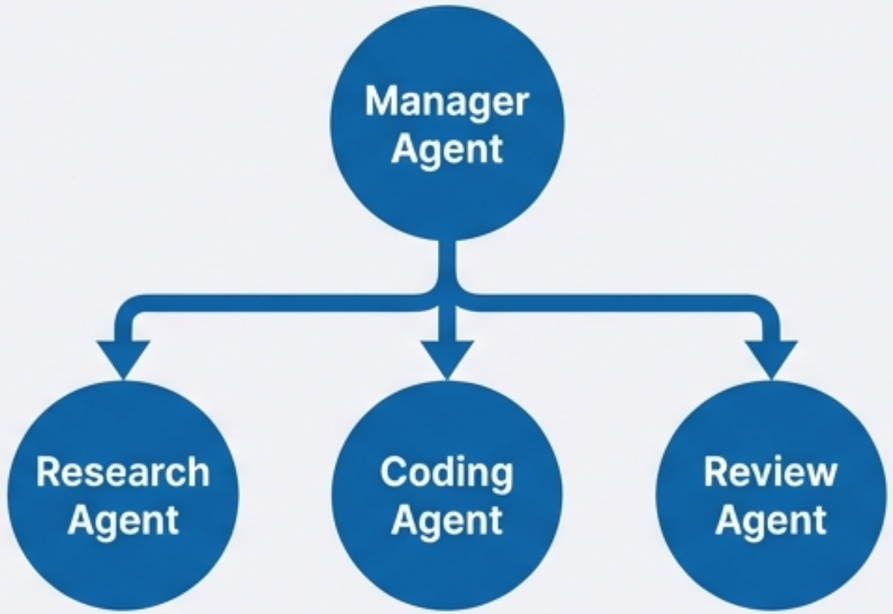
- 🪝 **新フック `llm_input` / `llm_output`**: 入力プロンプトと生成結果を外部プラグインで傍受・監査可能に。機密情報の混入チェックやポリシー違反検知に活用。
- 🚫🐞 **ログの衛生管理 (Log Sanitation)**: エラーログやスタックトレースに含まれるAPIトークン（Telegram Bot Token等）を自動的にRedact（黒塗り）処理。
- 🛠️ **プレフライト検知**: シェルインジェクション（Python内のBash変数混入など）を実行前に静的解析し、エラー・ループを防止。

⚠️ **WARNING: 監査ログの設定ミスは機密情報の漏洩につながります。本番環境での適切な設定と運用を徹底してください。**

アーキテクチャの変革：単体から「群知能（Swarm)」へ

Before vs After

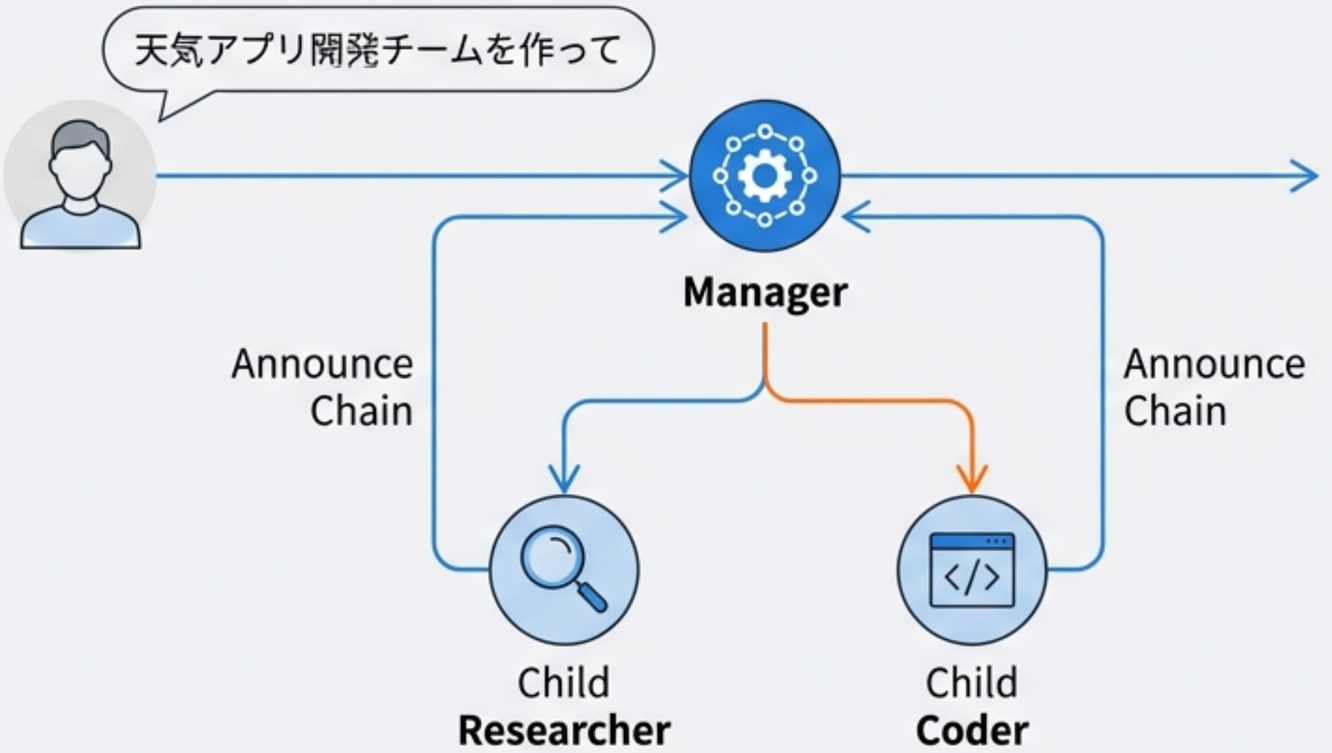




- コンテキストの純粋化: 役割を細分化することでハルシネーションを抑制。
- 並列処理: 複数の子エージェントが同時にタスクを進行。
- 権限の分離: 親は強力な権限を持つが、子には「Web閲覧のみ」など権限を絞って委譲可能。

⚠ WARNING: エージェント間通信と権限管理の複雑化には注意が必要です。スワーム全体の適切な監視とガバナンスを確立してください。

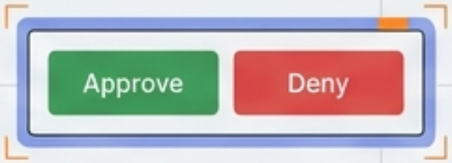
Sub-Agentsの実装と制御メカニズム



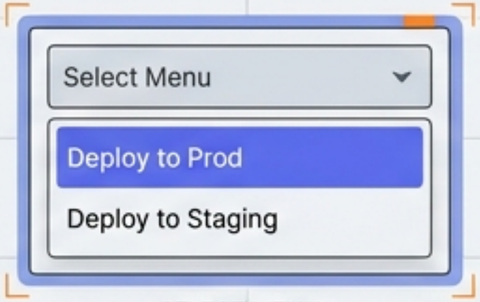

```
{
  "maxSpawnDepth": 2,
  "maxChildrenPerAgent": 5
}
```

- `maxSpawnDepth`：孫エージェントまでの生成を許可し、無限ループ（フォージェント制限、無限ループ（フォークボム）を防ぐ階層制限。
- `maxChildrenPerAgent`：1体のマネージャーが指揮できる部下の数を制限。

UXの進化：ChatOpsから「AppOps」へ



BUTTON COMPONENTS



SELECT MENU

MODAL FORM

- Discord Components v2: ボタン、セレクトメニュー、モーダルフォームの実装。
- 承認フローのUI化: 危険なコマンド実行の承認が「`approve`と入力」から「ボタンをワンクリック」に進化。
- ターゲット指定: 承認リクエストの送信先をDM、チャンネル、または両方に設定可能 (`channels.discord.execApprovals.target`)。
- Takeaway: エージェントは「チャットボット」を超え、GUIを持つ「アプリケーション」として振る舞い始めた。

運用品質の向上：日々の「摩擦」を取り除く

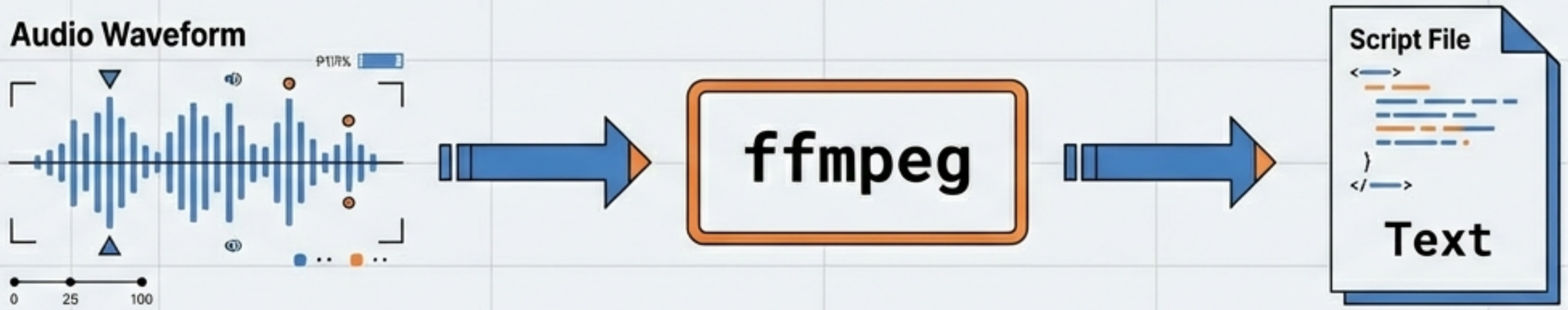
Telegram Polls: 意思決定をチャット内で完結。

Cron全文配信: 要約せず正確なログを配信（`delivery.to`）。

DMポリシー統一: Botからの不要なDM通知を防止（`dmPolicy`）。

TUIの安定化: 長時間セッションでのクラッシュ修正。

マルチメディアと実世界への適応



- 音声波形とプレビュー: Discordでのボイスメッセージ送信に対応。TTS（音声合成）ツールと連携し、エージェントが「喋る」体験が可能に。
- 画像・ファイル処理: ワークスペース内のパス処理（`MEDIA:`）が安定化。

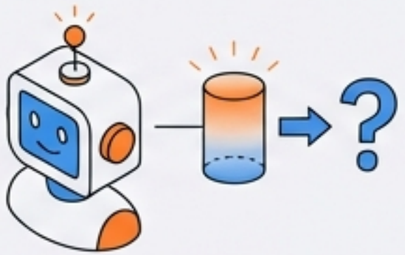
Case Study: 音声非対応のエージェントが、自力でファイルヘッダを解析し、`ffmpeg`で変換、APIを叩いて音声を翻訳した事例（Lex Fridman Podcastより）。LLMの創造的問題解決能力の証明。

哲学：「Vibe Coding」から「Agentic Engineering」へ

“You are not just a programmer.
You are a builder.” — Peter Steinberger

アプリはAPIになる

ブラウザ操作であらゆるWebサービスがAPI化される。



エージェントの視点

毎回記憶がリセットされる「疎外感」を理解して設計する。

魂のファイル (soul.md)

エージェント自身が自己記述するドキュメント。

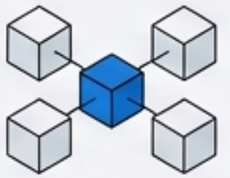
v2.12–v2.15 アップデート総括

SECURITY

SHA-256サンドボックス /
RCE・SSRF修正 / Docker構成ブロック

ARCHITECTURE

ネストされたサブエージェント /
群知能アプローチ

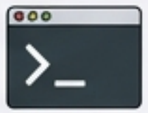


UX

Discord Components v2 /
AppOps化 / 音声・メディア対応

OPS

Cron全文配信 / DM制御統一 / TUI安定化

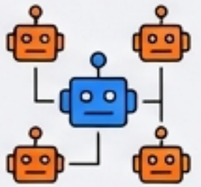


次のアクション：今すぐ実務へ適用せよ

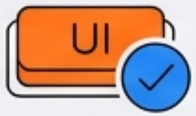
1. **UPDATE:**
v2.15+ へ更新（セキュリティ脆弱性修正のため必須）。

2. **AUDIT:**
トークンのローテーションと `dmPolicy` の見直し。

3. **EXPERIMENT:** 「チームを作成して」
「チームうプロンプトでサブエージェントン機能を試行。



4. **REFACTOR:** テキストコマンドによる承認フローを、
UIボタン（Components v2）に置き換え。



AIエージェントの未来は「待つもの」ではなく「組むもの」になった。