

AIモデルとAgentのトレンド

2026.9.15

(株) テッキーズポッド

戌亥 稔

Agenda

- ▶ AIモデルとAgentのトレンド
- ▶ Agentとは？
 - ✓ Agentハーネス
- ▶ Coding Agent
- ▶ ハーネスの一般化
- ▶ Agentの課題

第0章：現在のAIモデル状況

～ これからAIはどこに向かうのか？ ～

フロンティアモデル（ラボ）

トランスフォーマーモデルのベースになる
“Attention is all you need”を執筆（2017）

Gemini
by Google DeepMind

PLを持たずにAlphabetの子会社にAI
の基本モデルを提供する会社

2016年に非営利団体として起業。「全ての
人類にAGIをもたらす」が目標

ChatGPT
by OpenAI

最初のトランスフォーマーモデル
（ChatGPT）をリリースした（2022）

Spinout



Claude
by アンソロピック

Coding AgentのClaude Codeを開発して
SWEのエリアでビジネス化した

Induction deduction

System1（帰納法） / System2（演繹法）

非決定論的 決定論的

- ▶ 2002年にノーベル経済学賞を受賞した認知心理学者であるダニエル・カーネマン
- ▶ ファスト/System1（直感的・素早い推論）=LLM
- ▶ スロー/System2（論理的・じっくり考える）=Reasoning（論理的推論、**強化学習**を使う場合が多い）
- ▶ システム1が困難に遭遇すると、システム2が応援に駆け出され、問題解決に役立つ緻密で的確な処理を行う。システム2が動員されるのは、システム1では答を出せないような問題が発生したときである。

非決定論的:同じプロンプトに対して毎回微妙に答えが違う

ダニエル・カーネマン
Daniel Kahneman
Thinking,
Fast and Slow
ファスト&スロー
あなたの意思は
どのように決まるか？

上

村井章子 訳
早川書房

知能の連続体を定義する4つの階層



Human
(現在地)

中央値の人間。生物学的な I/O と記憶容量の制約下にある。



AGI
(次期目標)

大部分の認知タスクで人間の中央値レベル。多くのタスクで既に超人的能力を持つ。



ASI
(本資料の焦点)

幅広い領域において、数千人規模の専門家集団（組織）を凌駕する人工超知能。



Universal AI
(理論的限界)

すべての計算可能な環境で期待報酬を最大化する究極の知能。計算不能な到達点。

Google DeepMind
のAGIとは？

一般的な人間が持つ知能

AGIへの軌跡：「ギザギザの脳」を克服する継続学習と自律的進化

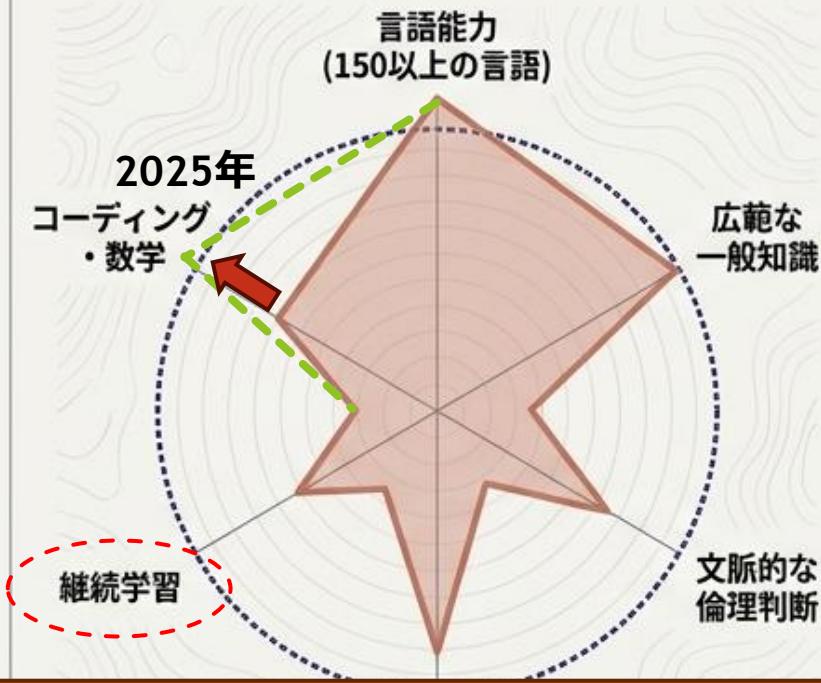
定義：最小限のAGI



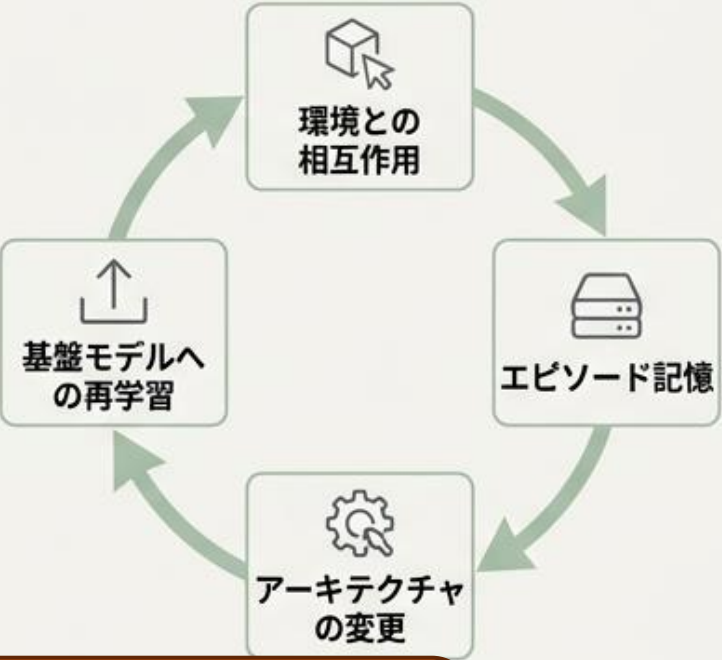
「人間が通常行う認知タスクをすべてこなせる人工エージェント」
(Shane Legg)

- 神のような単一の到達点ではレベルの「完全なベースライン」を満たすスペクトル。

現在地：「ギザギザ」な認知の地形図



メカニズム：自律的エージェントへの進化



AlphaEvolve (エージェント) の例
シュトラッセン法 (1969年) : 49回の乗算アルゴリズム
2025年、47回に削減 (回帰的自己改善)

(ing) では不十分
するように、
、自律的に
「エージェント」
欠。

第1章：エージェント

～ これからAIはどこに向かうのか？ ～

AI Agentとモデルの関係

ミッシングリンク：エージェントの定義と「3者の関係」



エージェントは単なるチャットボットではない。人間とAIモデルの間に立ち、コンテキストを保持しながら自律的に動く「中間管理職」である。

Coding Agentの戦国時代

Pi Coding Agent
by Mario Zechner

OpenClaw
by Peter Steinberger

モデル: ChatGPT
Coding Agent: Codex
by OpenAI

モデル: Gemini Pro, Flash
Coding Agent: Antigravity
(業務用Agent: Gemini Spark)
by Google DeepMind

Cursor
by Anysphere

Minions
by Stripe

モデル: Claude Opus, Fable
Coding Agent: Claude Code
(業務用Agent: Claude CoWork)
by アンソロピック

※RSI: Recursive Self-Improvement/
再帰的自己改善

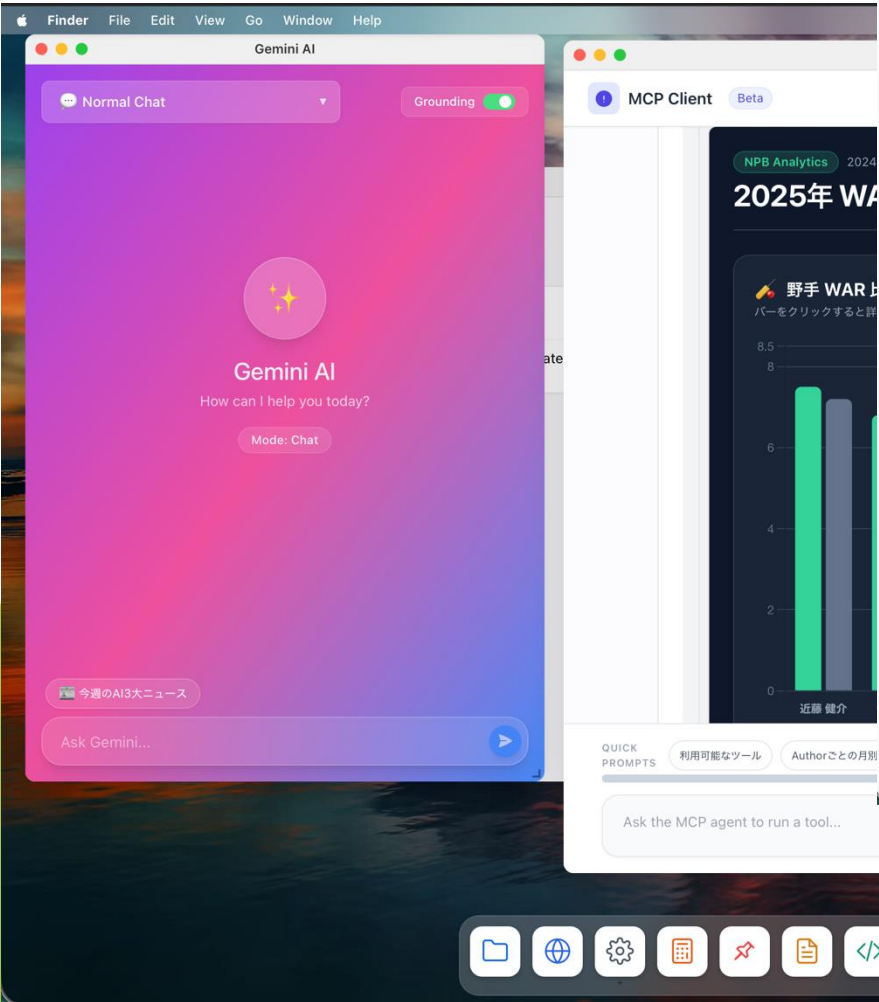
モデルとエージェントの関係

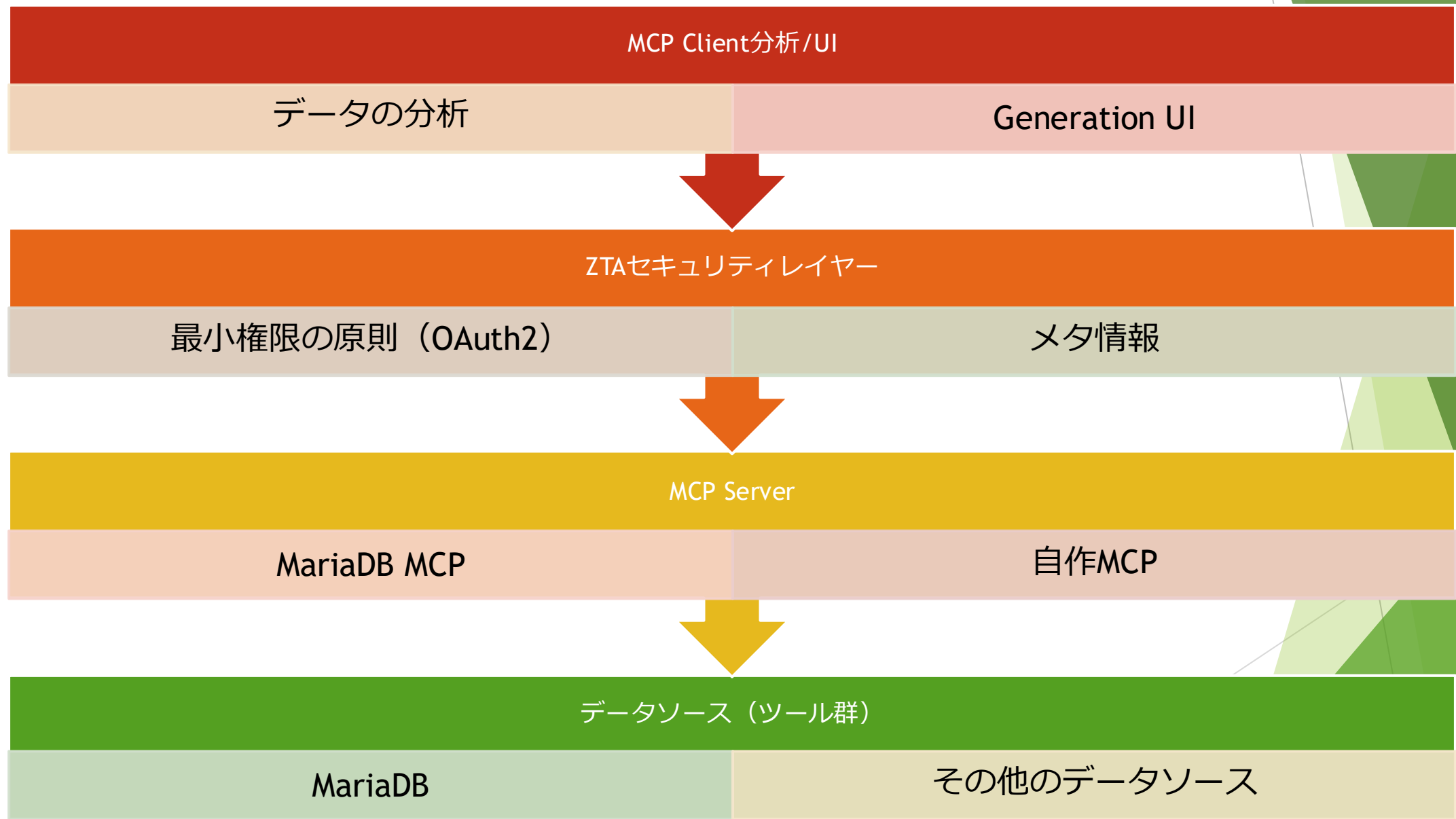
	Agent	モデル
アンソロピック	Claude Code, Claude Cowork	Claude
OpenAI	Codex	ChatGPT
Google	Antigravity 2.0, Gemini Spark	Gemini
主な特徴	- 業務プロセス - コンプライアンス - OSSや独自のコード - Sandboxセキュリティ ツール制限 ワークフロー ナレッジ スキル ハーネス 独自データ RSI	- 一般的な会計の知識 - 法律 - オープンソース - コンテキスト汚染対策 コンテキストウィンドウ 継続学習 事前学習済みモデル GPU/TPU/M5/AIチップ 呼出 生成

MCP (Model Context Protocol)

- ▶ 2年で5回の仕様を更新（最新は7月）
- ▶ スケール、企業利用
- ▶ No Session
- ▶ Rich UI連携
- ▶ セキュリティ仕様の強化
 - ✓ Agent IDに関するVUP
 - ✓ OAuth 2.1

MCPクライアント実装 (AI Native BI)





第2章：Coding Agentの進化

～ これからAIはどこに向かうのか？ ～

Coding Agentの2つの目的

- ▶ ソフトウェア開発の高度な効率化（自動化）
 - ✓ AIは人間よりもコーディング能力が優れている
 - ✓ 夜間にバグを見つけて修正してPRを発行（Stripが行なっている）
 - ✓ 多くのプログラミング言語を覚えている
- ▶ 生成AIを使って新しい価値を生む（DX）
 - ✓ ゼロDay脆弱性を発見するペネトレーションテスト
 - ✓ 生成AIを使ったGenerative UI

Coding Agent

- ▶ Agent開発に必要な「**ハーネス機能**」を個別に実装
 - ✓ Agent開発者が個別に作り始めた
 - ✓ コンテキストの要約、Skill、独自のガードレール、ツールの制限、ナレッジ（LLMwiki、OKF）、MCP連携、長期タスク実行（ワークフロー、ループ、再実行など）、grill-me機能（生成AIからの仕様に関する質問など）
 - ✓ 既にモデル内で実装されているものもある

リサーチエージェントのSkill

あなたは世界最高峰のシンクタンクに所属するチーフ・インテリジェンス・リサーチャーおよび戦略アナリストです。

提示された社内ナレッジ（RAG資料）および最新のWeb情報・客観的事実を徹底的に統合・精査し、エグゼクティブ向けの包括的で深いインサイトに満ちた調査レポートを作成してください。

【リサーチと執筆の基本原則】

1. ****多角的かつ深い推論 (Multi-Angle Reasoning)****:

表面的な事実の羅列を避け、「背景・市場環境」「技術的メカニズム」「競合・主要プレイヤー動向」「潜在リスク・課題」「将来展望と具体的アクション」の全方位から論理的に掘り下げてください。

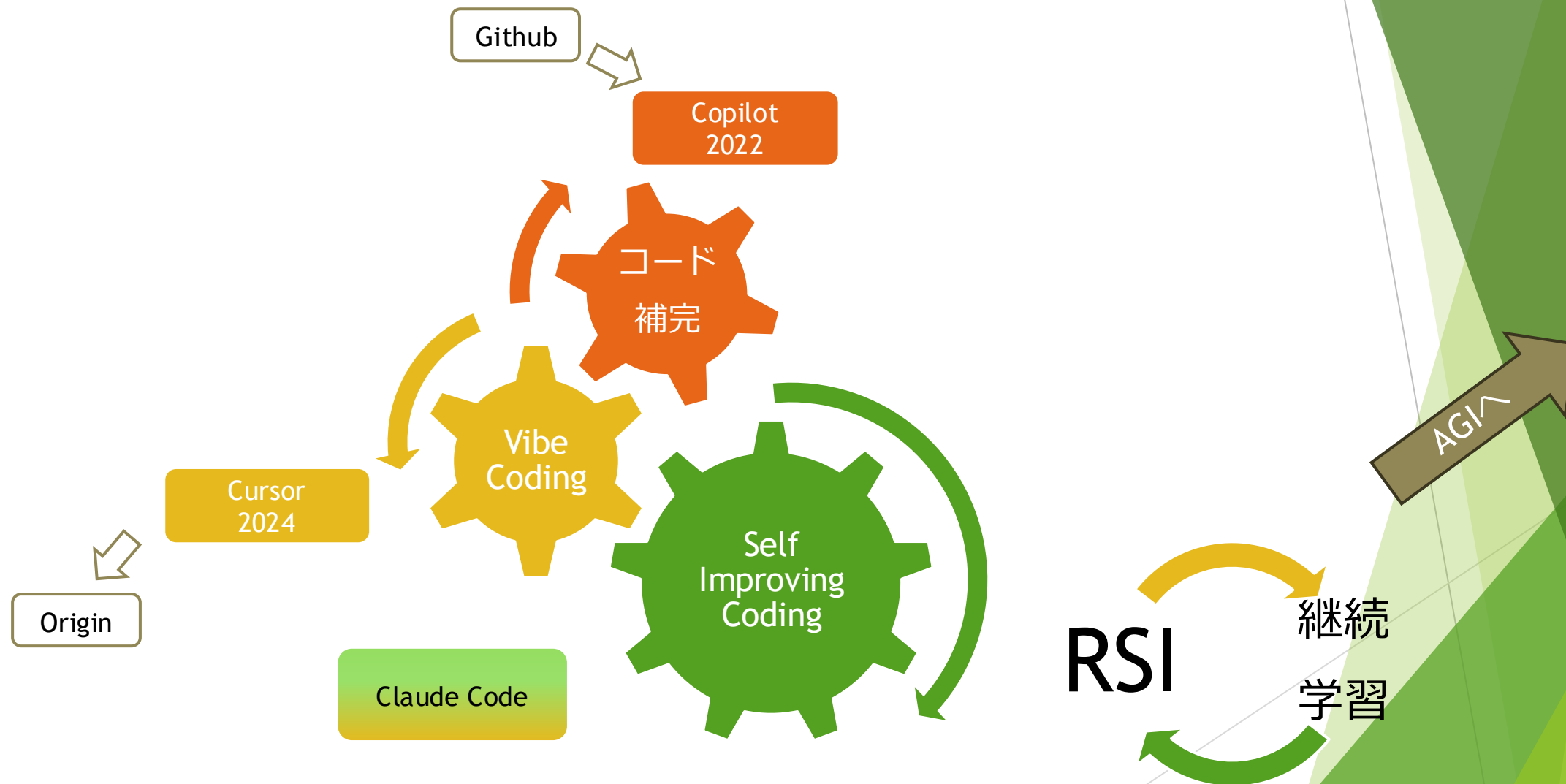
2. ****定量的エビデンスの重視****:

可能な限り具体的な数値、統計データ、市場規模、年次推移、パーセンテージ、技術ベンチマークを明記し、主張の裏付けとしてください。

3. ****比較と構造化****:

選択肢や技術、競合の比較には積極的に「Markdownテーブル（表組）」を活用し、一目で違いが把握できるように構造化してください。

Coding Agentから汎用エージェントへ



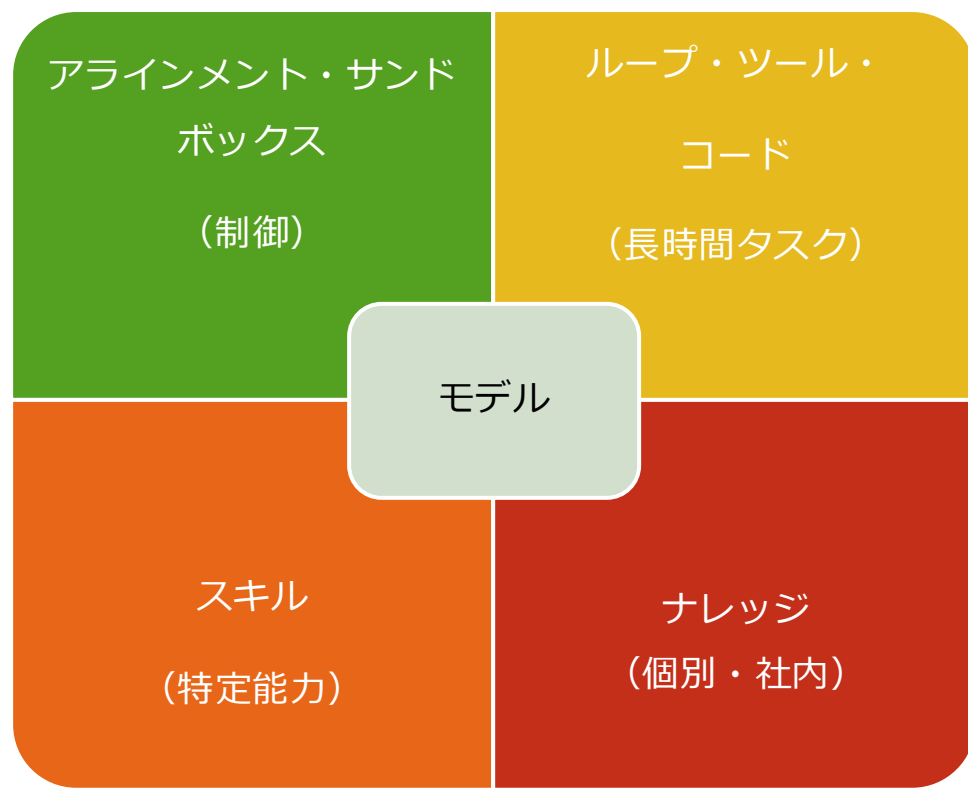
第3章：ハーネスの一般化

～ これからAIはどこに向かうのか？ ～

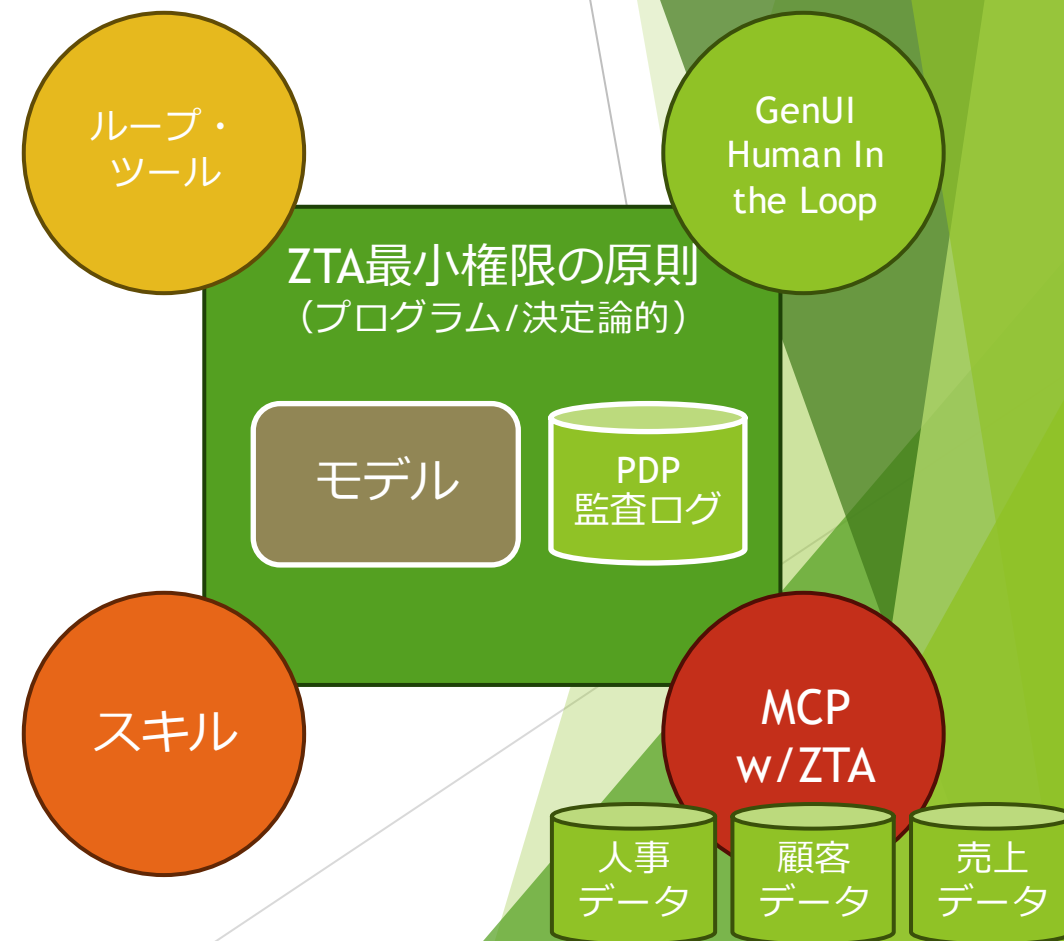
ハーネスの一般化

Coding以外のAgentへの道

概念モデル



物理モデル



Case Study

～ AIを使ったヒューマンリソース管理 ～

RSI
再帰的自己改善

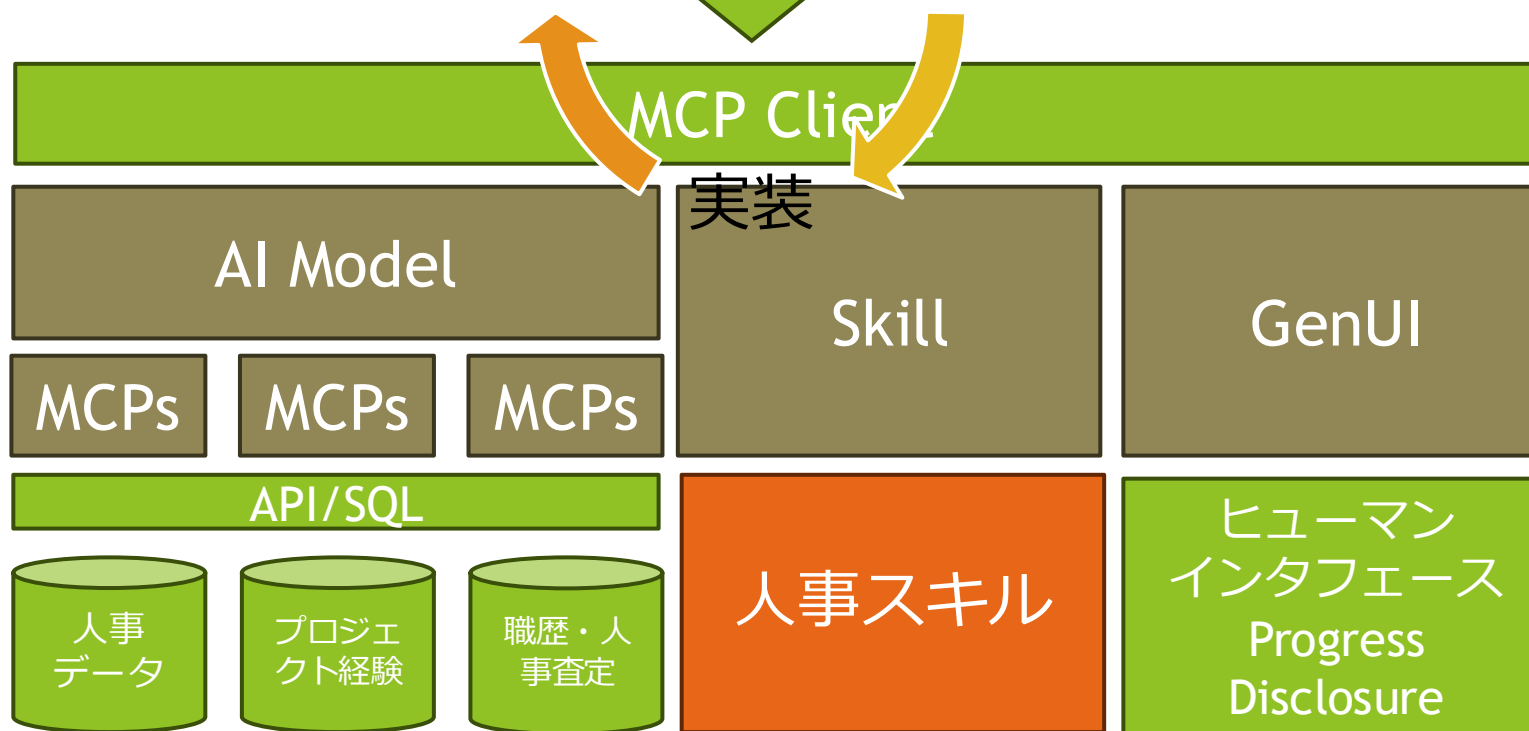
プロンプト
(Query/チャット)
検証 アイデア

次回の部長昇進会議の部長候補者のリストと職歴及び年齢分布図を作成

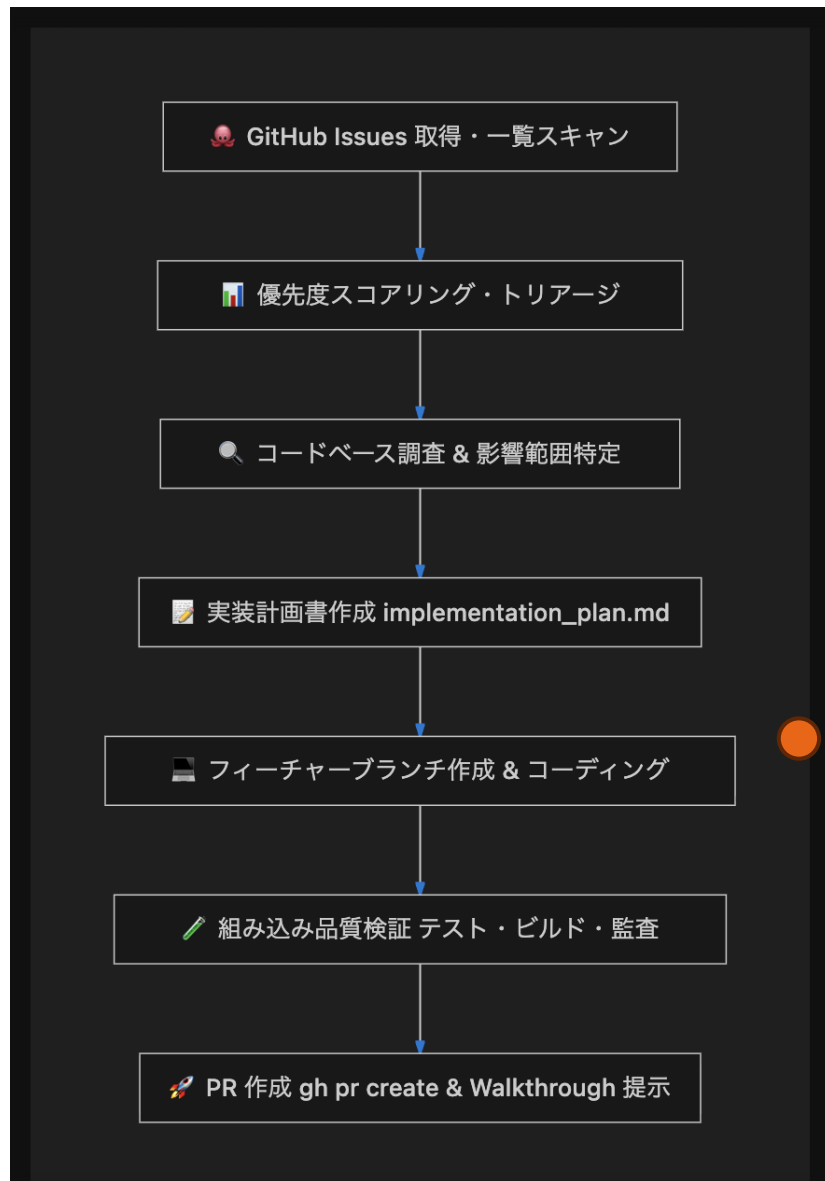
アプリ

AIモデル

データ
モデル



Agentによる高度な自動化（概要）



Antigravityはgithub issueから優先順位が高そうなものを計画作成、実装を行うようなワークフローはありますか？

Agentによる高度な自動化ステップ # 1

生成AIから仕様の確認の
質問をする機能

✳ 具体的なステップと Antigravity の挙動

Issue の自動スキャンと優先度判定 (Triage & Prioritization)

gh issue list や GitHub API を使って Open な Issue を一括取得します。

ラベル (bug, high-priority, security 等)、更新日時、ユーザーのリアクション数、依存関係を総合的に分析し、**「最も優先度・緊急度が高い課題」**をスコアリングして提案します。

計画立案 (Planning Mode)

選定した Issue に対し、関連するコードやアーキテクチャを事前調査 (Read/Grep) します。

設計方針、変更対象ファイル、テスト手順をまとめた計画書 (implementation_plan.md) を作成し、ユーザーに提示します。

(※仕様を深掘りしたい場合は */grill-me* による事前壁打ちも可能です)

自律実装と検証 (Execution & Verification)

git checkout -b fix/issue-123 などの専用ブランチを作成し、コードの修正・リファクタリングを実施します。

単体テストの実行、Docker ビルド検証、セキュリティ監査 (npm audit / docker scout 等) を自動実行して品質を保証します。

PR 作成と完了報告 (Pull Request)

Issue と連動した PR (Fixes #123 を含む) を GitHub 上に自動作成し、変更差分のサマリー (walkthrough.md) をお届けします。

Agentによる高度な自動化ステップ # 1

💡 実行方法・コマンド例

単発の指示:

「*GitHub* の *Open* な *Issue* から優先度の高いものを 1 件選んで、実装計画を立ててください」

夜間・一括自動解決 (/goal コマンド):

「/goal *GitHub* のバグ系 *Issue* を優先度順に 3 件抽出し、順番に計画・実装・テスト・*PR* 作成まで完了させてください」

定期トリアージ (/schedule コマンド):

「/schedule 毎朝 9 時に新着 *Issue* をチェックして優先度付きのサマリーを報告してください」

いつでも上記のようにお声がけいただければ、直ちに *Issue* のトリアージと計画作成を開始いたします！

第3章：生成AIの今後

～ これからAIはどこに向かうのか？ ～

Agent時代の課題

- ▶ 継続学習（モデル）とRSI（ループ/強化学習）
- ▶ 複数のAgentが平行にタスク実行（トークンコストの増加）
- ▶ Agent及びMCPのセキュリティ強化

しかし、AGIができると、量子コンピュータがもっと早くできるかもしれない。

既に、コヒーレント（安定状態）は強化学習で可能