

NCWG大阪10/31

# 生成AIによってソフトウェアエンジニアの仕事が奪われるって本当？

2025.10.31

(株) テックニーズポッド

戌亥 稔

# 本日のトピック

- ▶ 認知心理学におけるファストとスローとは
- ▶ 最近のモデルのリリース状況
  - ✓ 最新モデルにはどのような機能が追加されているのか？
- ▶ OMNIモデルと将来のモデルの方向性
- ▶ AI Coding Agentの仕掛け
- ▶ AI Coding Agentの動向
- ▶ AI Coding Agentに作らせてみたこと
- ▶ 本当にSWEは必要なくなるのか？

# 自己紹介

- ▶ 戎亥 稔/株式会社テックイズポッド創業者
- ▶ 昨年まではアジャイル開発のコンサルをしていた
- ▶ 今年4月からAIに関するコンサルをしながら、サービス開発をトライアルでやっている
- ▶ 背景：フロリダ工科大学の修士課程でAIの研究をしていた。（当時はルールベース）
  - ✓ 修士論文：The Recognition of Imperfect Strings Generated by Fuzzy **Context** Sensitive Grammars
  - ✓ Publications : Inui, M. and Shoaff, W. and Fausett, L. and Schneider, M., "The Recognition of Imperfect Strings Generated by Fuzzy Context Sensitive Grammars", International Journal of Fuzzy Sets and Systems, vol. 62 (1), pages 21-29, 1994.

# 脳の2つのモード？（認知心理学）

▶  $2+3=?$ 、 $2\times 3=?$

▶  $17\times 24=?$

✓  $(20-3)\times 24=20\times 24-24\times 3=480-72=408$

▶ ファースト&スロー



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

System1 と System2  
(Sakana.aiのLlion Jones、Figure CEO、Groq CEO)

# System1=Fast/System2=Slow

- ▶ 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
- ▶ System1（直感的・素早い推論）=DNN/LLM（深層学習）
- ▶ System2（論理的・じっくり考える）=Thinking/Reasoning（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が応援に駆り出され、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。

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？

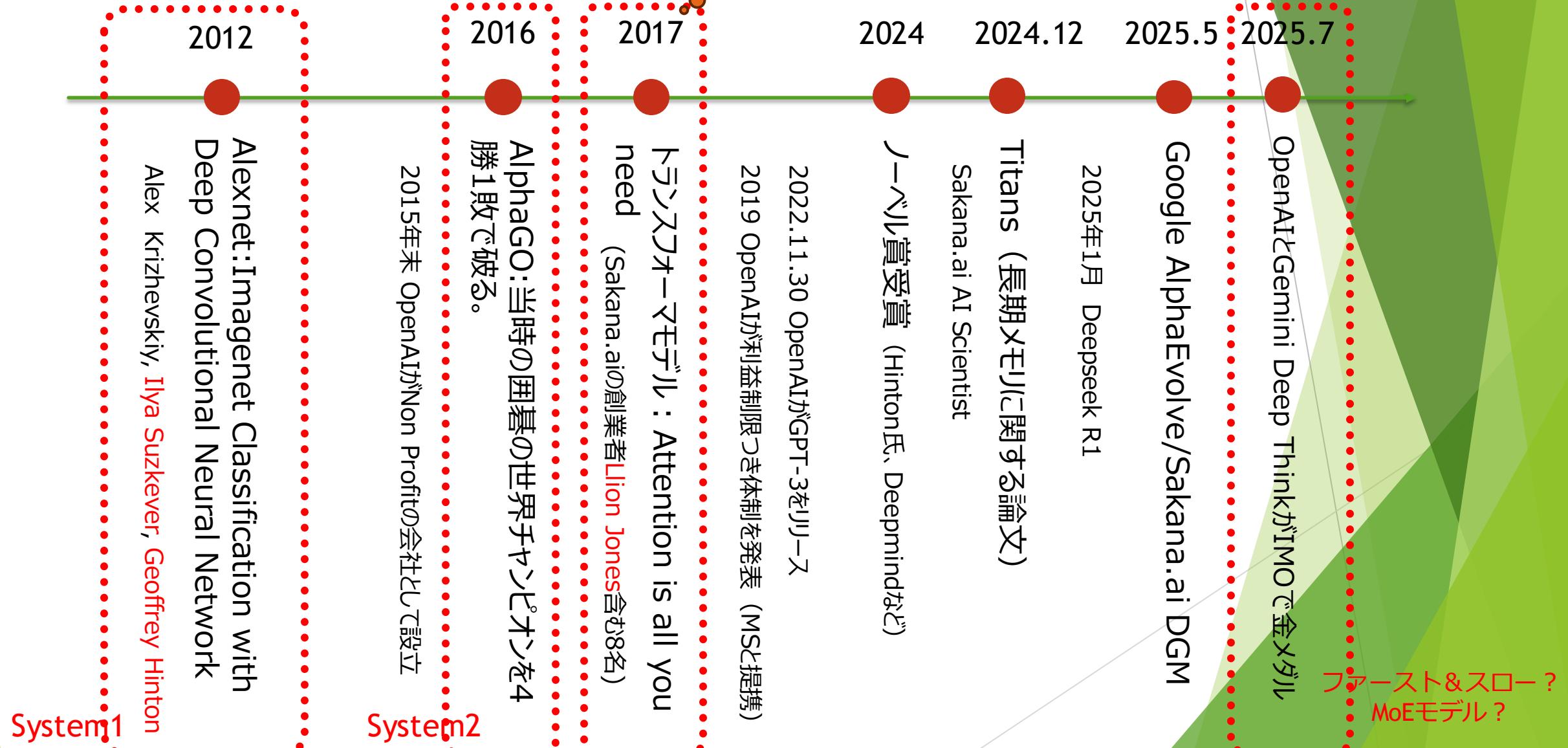
上

村井章子 訳  
早川書房

# AIの歴史的イベント

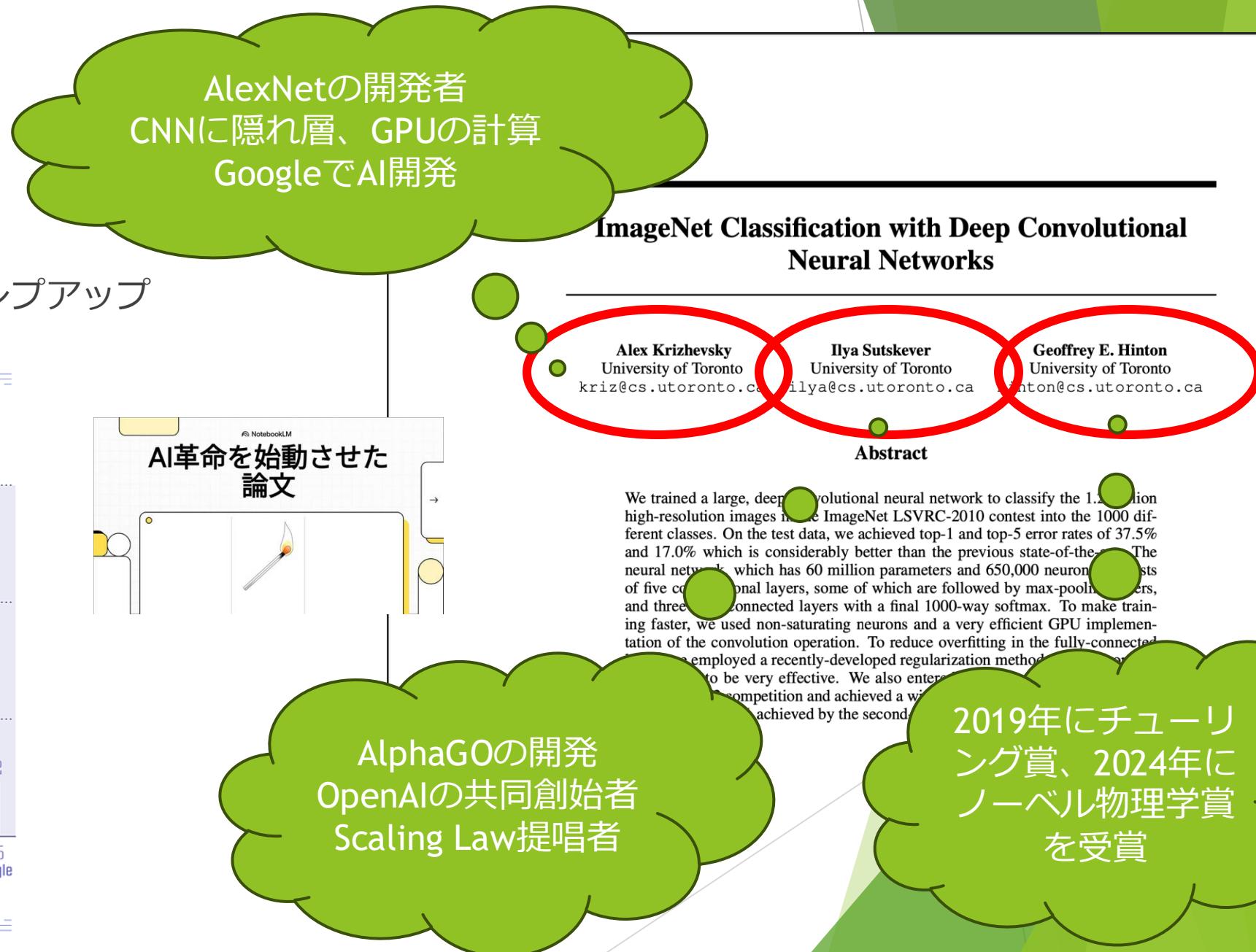
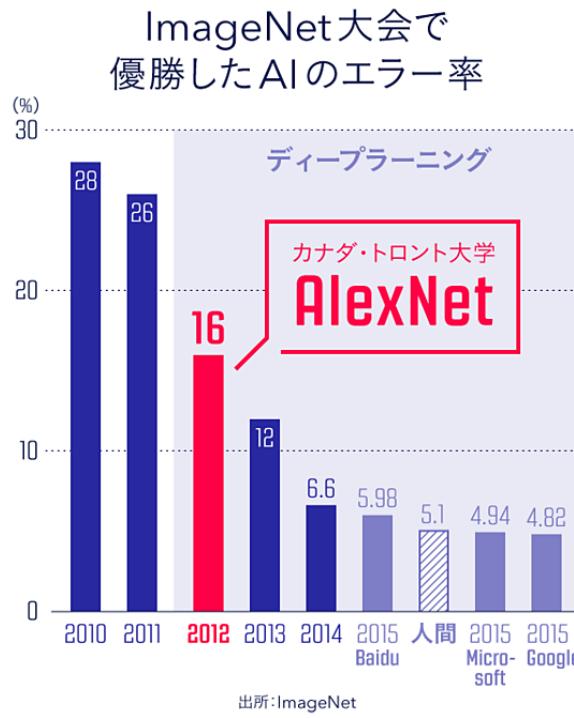
有名な論文

2025年 AI Agent/MCP (Model Context Protocol)



# 2012年 AlexNet(2012)

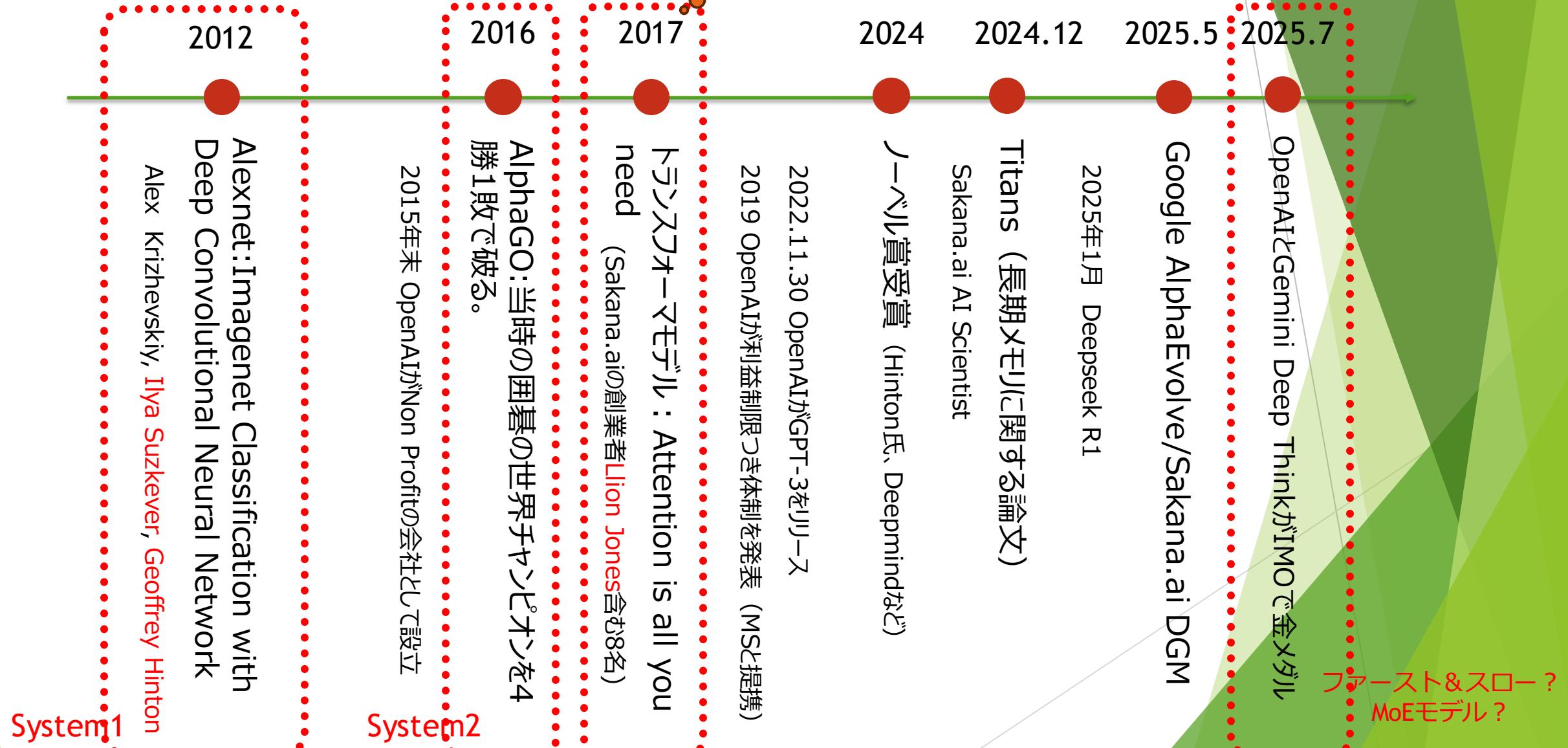
- ▶ ImageNetでの大きなジャンプアップ



# AIの歴史的イベント

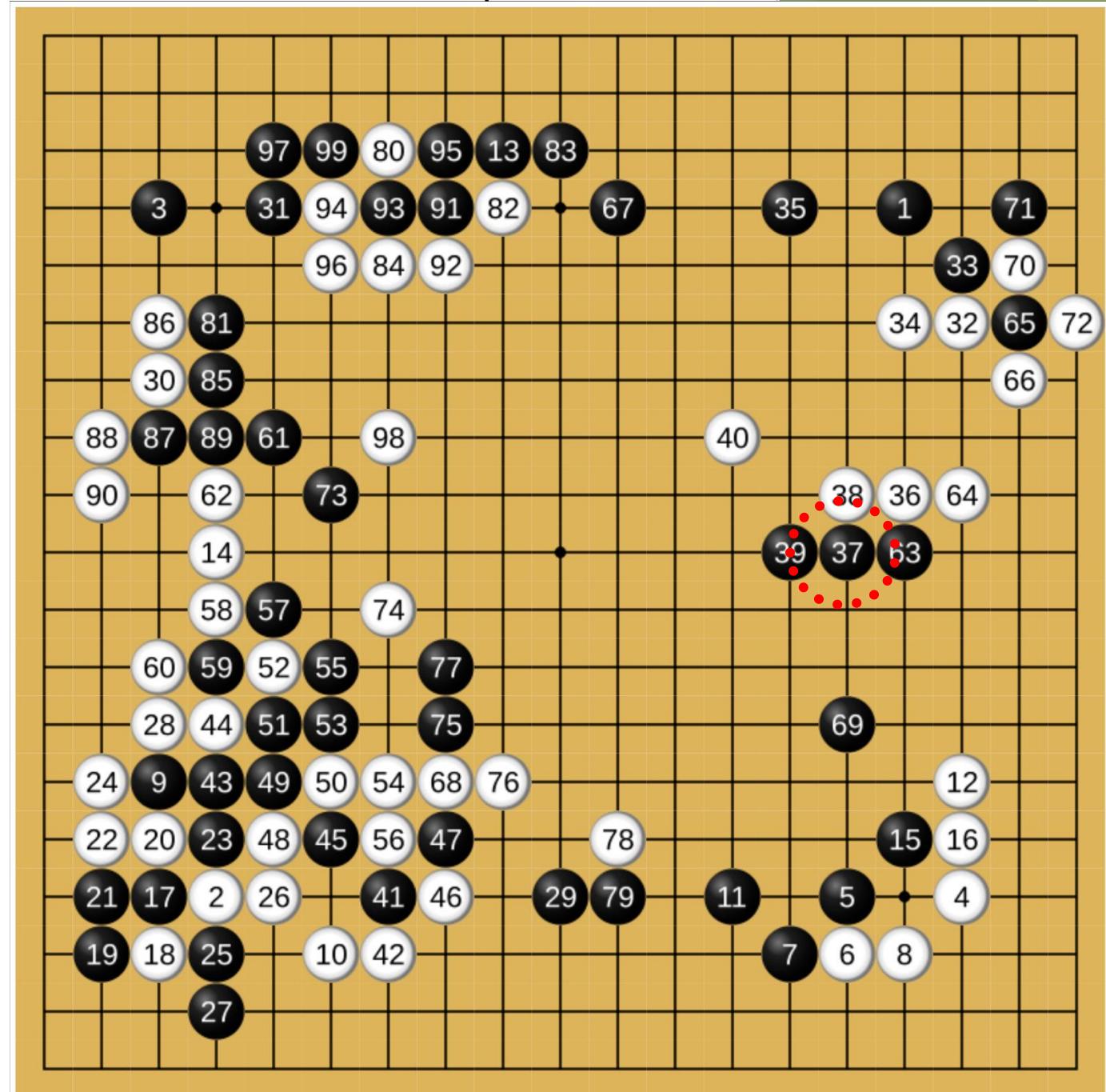
有名な論文

2025年 AI Agent/MCP (Model Context Protocol)



# 囲碁の打ち方

- ▶ 序盤は**定石 (=ファースト)**で打つ
- ▶ 定石とは過去の打ち手（棋譜）の分析結果としてまとめられる
- ▶ 中盤になると、自分の持ち時間を使って先読みをして自分が有利となると思う手を打つ。先読みは**論理的に推論 (=スロー)**して打つ



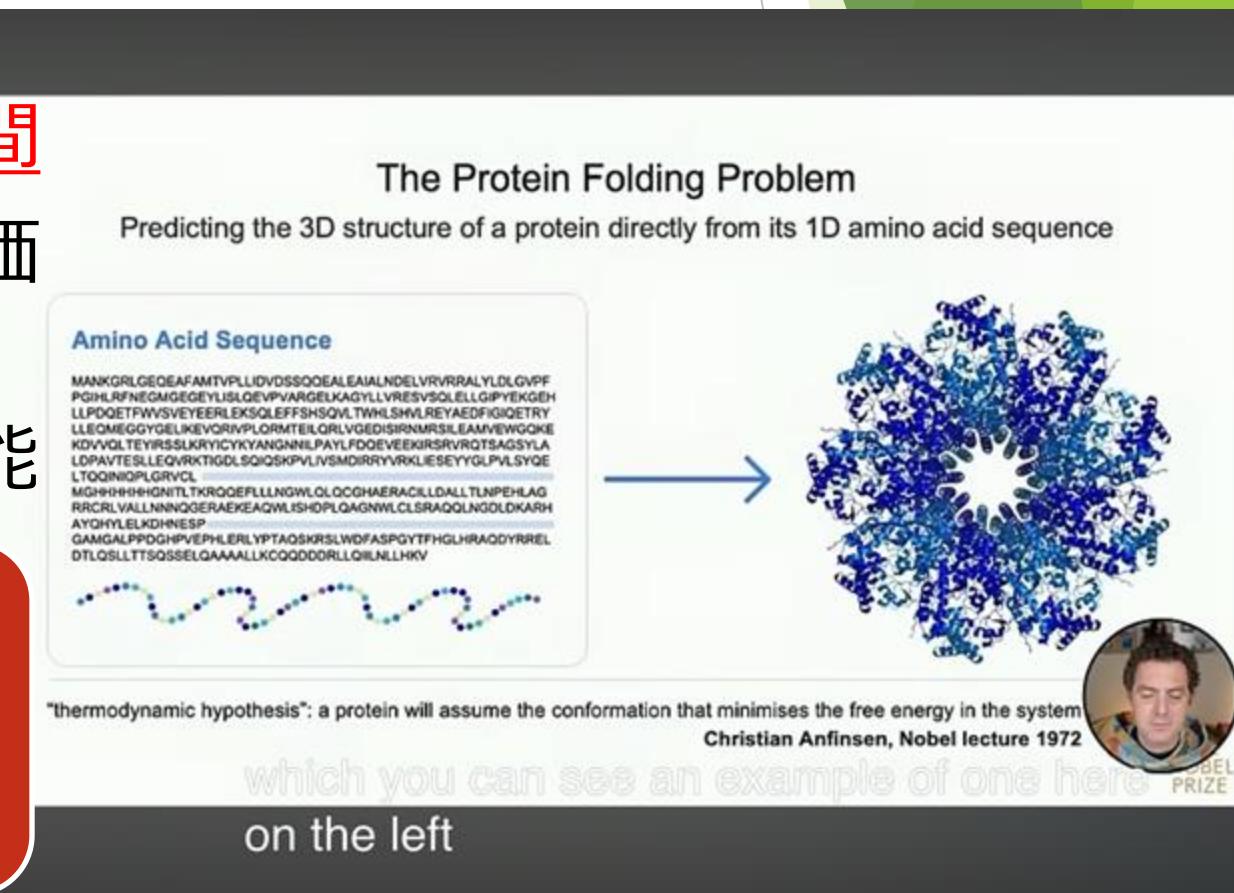
# AIによる科学発見の3条件 (by デミス・ハサビス氏)

## → 3条件とは

1. 大規模な組み合わせ探索空間
2. 明確な目的関数（または評価基準）
3. 大量の学習データが入手可能

DRL  
(Deep Reinforcement Learning/深層強化学習)

囲碁は $10^{170}$ 、タンパク質は $10^{300}$ これは  
モンテカルロ木探索で実行は不可能



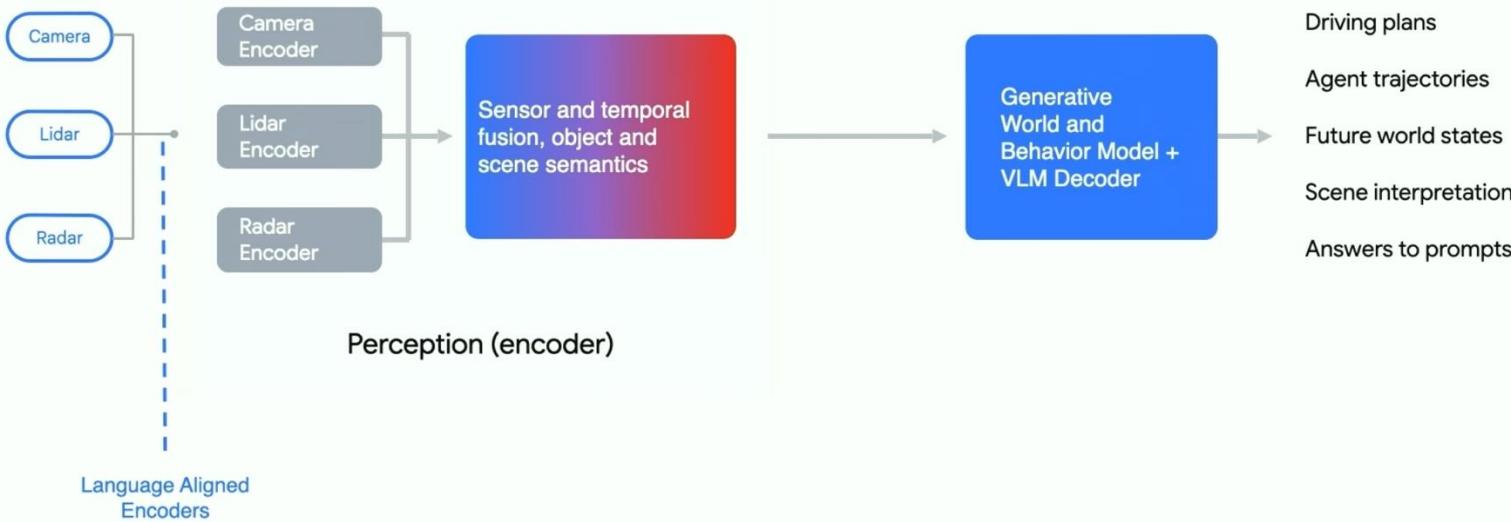
# 自動運転車へのDNN、Thinking、 Attention機構の応用

## ► Waymoの事例



The Waymo Foundation Model: Combining AV-specific advances with general world knowledge of VLMs

Waymo Driver's  
superhuman sensing ability



# AIモデルまとめ (OMNIモデル)

GPT-5は8月リース済み

Chatbot (GPT-5, Gemini2.5 Pro w/Deep Think, Anthropic4.1, LLaMA4)

API

Agent

Browser Use

Salesforce

...

科学の発見(STEM)

AI Code Agent

Alpha Chip

Alpha Fold3  
Alpha Proteo

Deep Research

MCP、その他のデータ連携ツール、ワークフロー

System1

LLM

System2 STEM推論モデル  
(Reasoning/Thinking)

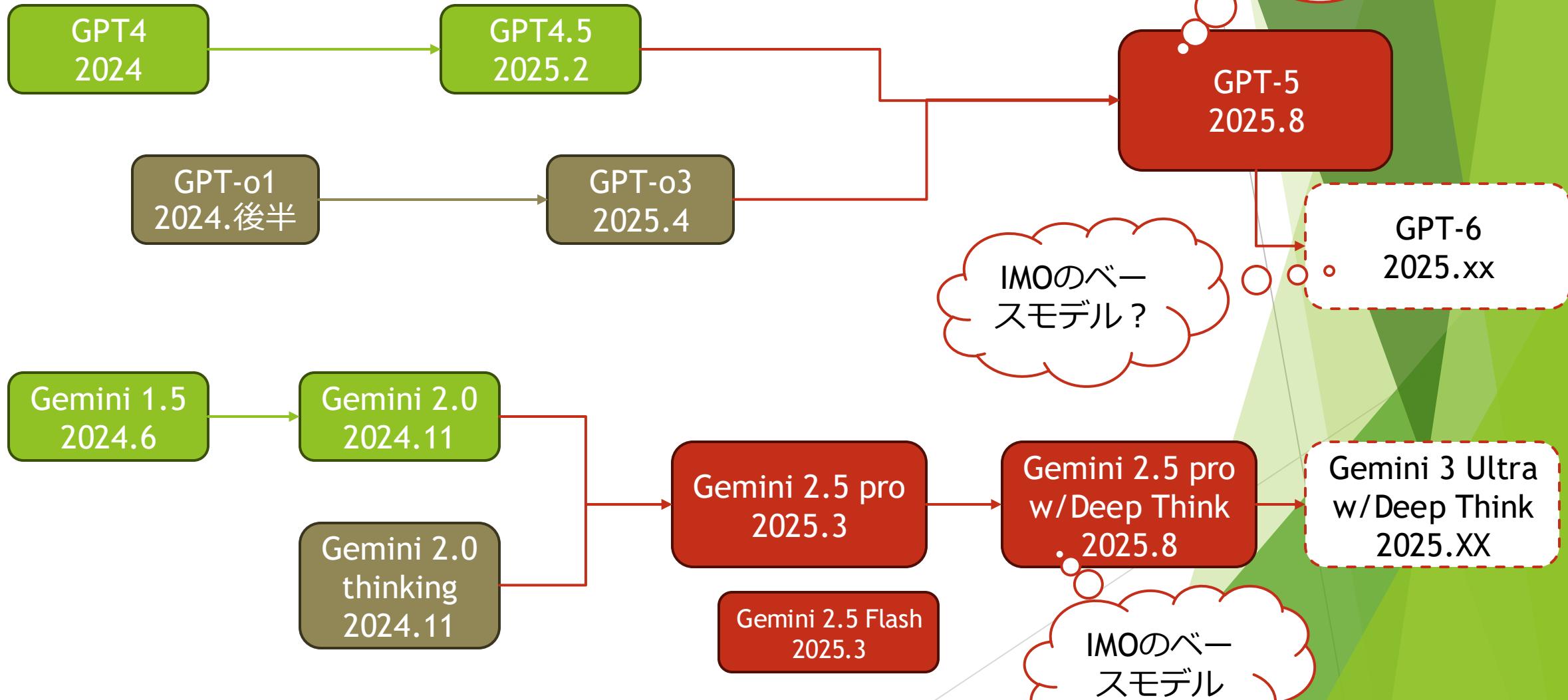
AI Search

長期メモリ : Taitans (Google) , Transfomer<sup>2</sup> (SakanaAI)

Google MuVeRA (Multi-Vector Retrieval Algorithm)

セキュリティ

# モデルの推移



# 今後

- ▶ Deep ThinkとかUltra Thinkとかの長く考えるモデルが出てくると予測
- ▶ OmniモデルやMoE (Mixture of Expert)
- ▶ Diffusion LLM (Inception Labsからコーディング用のものが出来ている)
- ▶ パラメータが小さいReasoningモデル (HRM/Gemma3 270M)
- ▶ 長期メモリの実装 (GPT-6)



生成AIによって  
SWEの仕事が奪わ  
れるって本当？

2025年はAI Coding の年

# AI Agentと課題とは？

- ▶ 人間が行う長期タスクを自動化する →**仕事が奪われる**
- ▶ AIは個別の顧客の情報をよく知らない
  - ✓ 業務プロセス（ワークフロー）
  - ✓ 業務データ
- ▶ データとフローは今後変化することもあるため、この2つの学習は柔軟性が必要
- ▶ フローはコードに埋め込まれたパイプライン、データはある時点の固定データで行うとAgentは思い通りに動かない

# AI Agentに必要なキーAIテクノロジー

## ▶ Reasoningモデル（STEM推論）の進化

- ✓ タスクを長期化するための仕組み（例：永続化メモリ）
- ✓ UltraThink、Deep Think機能
- ✓ 強化学習の様々な評価（プロセス/結果）関数

## ▶ 特定のデータとの連携機能

- ✓ MCP (Model Context Protocol) /bindit (ユニリタ)

## ▶ プロダクト開発や運用タスクの自動化

- ✓ JIRA (タスク/要件) →コーディングタスク→自動テスト (CI/CD)

MCPについては9/30にNCWGの会合で藤田さんから発表がありました

個別にAI Codingで開発を行なっております。

# Coding Agentの動向

- ▶ IDE (Editor)
  - ✓ Copilot、Cursor (9.9B)、Windsurf (OpenAI買収30B→Google経営者買収24B→Devin/Cognition買収3-4B ?)、Kiro/aws
  - ✓ AIのモデルはAnthropic/OpenAI/Google (原価率問題)
- ▶ プラットフォーム (PaaS) 型
  - ✓ Bolt/Lovable (開発→Deployment)
  - ✓ 実行環境も提供
- ▶ CLI型
  - ✓ Codex(OpenAI)/Claude Code(Anthropic)/Gemini CLI(Google)
  - ✓ 直販モデル (モデルの金額で利用可)

Anysphere  
株式評価：9.9B  
ARR：US\$500M  
従業員：100名

GithubやKiroが提唱している、AI-SDD (Spec Driven Development) が評判

# コーディングAgentの仕掛け

こうした方がいい ああした方がいい 見てるだけ

仕様は?

アーキテクチャは?

手抜きしてないよね

開発計画の作成  
(調査と計画)

計画書・  
ToDo

スローモードで  
論理的に作成

Opus 4.1 (Sonnet4.5)  
Gemini-2.5-pro  
GPT-5  
など

実装ステップ

レビ  
ュー

テスト

テストプログラ  
ムの作成

作業報告

ファースト  
モードでコー  
ディング

人間モードで  
確実に

Sonnet 4 (Sonnet4.5)  
Gemini-2.5-Flash  
など

Sonnet 4 (Sonnet4.5)  
Gemini-2.5-Flash  
など

難しいデバッグはス  
ローモードでログツ  
クを追う

デバッグ

# 働き方改革タスクエージェント（例）

AIアシスタント

Team Tasks デフォルトワークスペース ▾ << 今週 2025/09/21 (日) ~ 09/27 (土) >> 成

	月 22 計画: 7.00h 実績: 7.50h	火 23 計画: 3.00h 実績: 2.75h	水 24 計画: 4.00h 実績: 4.25h	木 25 計画: 4.50h 実績: 4.00h	金 26 計画: 4.00h 実績: 7.25h	Backlog +
<b>戌 戌亥 稔</b> 18.50h 実績: 20.75h	SurveyAgent 計画: 3h 実績: 0h <span>開始</span> 開発	Quiz設問の変更機能 計画: 1h 実績: 0.75h <span>開始</span> 開発	Quiz作成時のPDF問題について 計画: 1h 実績: 0h <span>開始</span> 開発	<b>クイズ</b> ZTA中級者向け 回答期限: 2025年9月25日 <span>結果を見る</span> ?	自動テスト 計画: 1h 実績: 1.5h <span>開始</span> 開発	吉田さんとミーティング 計画: 0h 実績: 0h
	連携方法	内容			拡張案	
N/A	1 カレンダー連携 2 Github連携 3 Quiz機能 4 AIアシスタント 5 データ集計	MCP/OAuth2 MCP 生成プロンプト 生成プロンプト 生成プロンプト	<p><b>パーソナル情報</b>にプロンプトを使ってアクセスし、 MCPで連携。結果はクリックでタスクを<b>自動生成</b></p> <p><b>企業情報</b>にプロンプトを使ってアクセスし、 MCPで連携。結果はクリックでバックログを<b>自動生成</b></p> <p><b>ドキュメント</b>アップロードしてクイズ（小テスト）を<b>自動生成</b>。クイズの期限を設定して個人のタスクとして表示</p> <p><b>Gemini</b> (w/Google Search) と<b>Claude</b>に一般的なプロンプトを投げて回答を<b>生成</b>が帰ってくる。</p> <p>プロンプトを使って、各タスクにかかった時間を、<b>タグ毎の集計表を生成</b>が可能。</p>			スケジュールの空き時間をメールで送る Issueの内容をCode Agentと連携して開発 動画の教育タスクとクイズを連動 DeepResearch機能の実装 クイズ結果の詳細集計や週のmtg時間の集計
テスト 太郎 i: 4.00h 実績: 2.00h	MEETING 調査 調査 開発	完了 完了 未着手 完了	test/issues/14 (作成: 2025-09-25) - [Techies-t/MCP-test/issues/14]	調査	計画: 0h 実績: 0h	

# Quizの作成

AIにQuiz作成の指示  
を出して作ってもらおう！

## 新しいquizを作成

このページでは、ドキュメントからAIを用いてQuizを自動作成し、内容を確認できます。

quizのタイトル \*

ZTA入門

設問数 \*

5

難易度 \*

初心者向け

回答期限 \*

2025/09/30

quizの元となるドキュメント \*

ファイルを選択 zero-trust-architecture-jp.pdf

対応形式: .txt, .md (PDFは将来対応予定)

AIへの指示プロンプト

あなたはプロの研修担当者です。  
このドキュメントの本文をよく読んでください。  
以下の指示を元に、本文の内容の理解度を確認するためのquizを作成してください。

# 指示  
- このドキュメントの本文のみをquizの対象とし、ファイル名、作成者、日付などのメタ情報は設問に含めないでください。

1 総

月 27 火 28 水 29 木 30 金 31 土 1

計画: 1.50h 実績: 0.25h 計画: 3.00h 実績: 0.75h 計画: 0.00h 実績: 0.00h 計画: 0.00h 実績: 0.00h 計画: 0.00h 実績: 0.00h 計画: 0.00h 実績: 0.00h

テナント管理画面の開発 (Phase2)  
計画: 1.5h 実績: 0.25h

開始 完了

[クイズ] ZTA (最小権限の原則)  
回答期限: 2025年11月23日 挑戦する

受講率は何%かな？

最小権限の原則がZTAにおいて実現される理由は？

a) システムの管理を簡素化するため  
b) 攻撃による被害範囲を最小限に抑える  
c) ユーザーの利便性を高める  
d) ネットワーク機器の稼働率を上げる

AlexNet2012のquiz  
回答期限: 2025/9/23 | 総設問数: 8問

受験率 50%

1 / 2 人

平均スコア 7.0 点

最高: 7点 / 最低: 7点

参加者一覧 ランキング

氏名	ステータス	スコア	正答率	完了日時
戌亥 稔 (minoru.inui@techiespod.co.jp)	Completed	7 / 8	88%	2025/9/21 14:38:34
テスト 太郎 (test.user@example.com)	Not Started	-	-	-

# GitHub MCP連携（マルチテナント）

ログイン/ログアウト実装済み

ユーザ単位で担当タスクの進捗確認

Techies-T/MCP-testの未解決Issueを検索して3件表示ください。

テナント毎にMCPの設定ができる

Query結果

データ連携の結果をタスクボードに表示

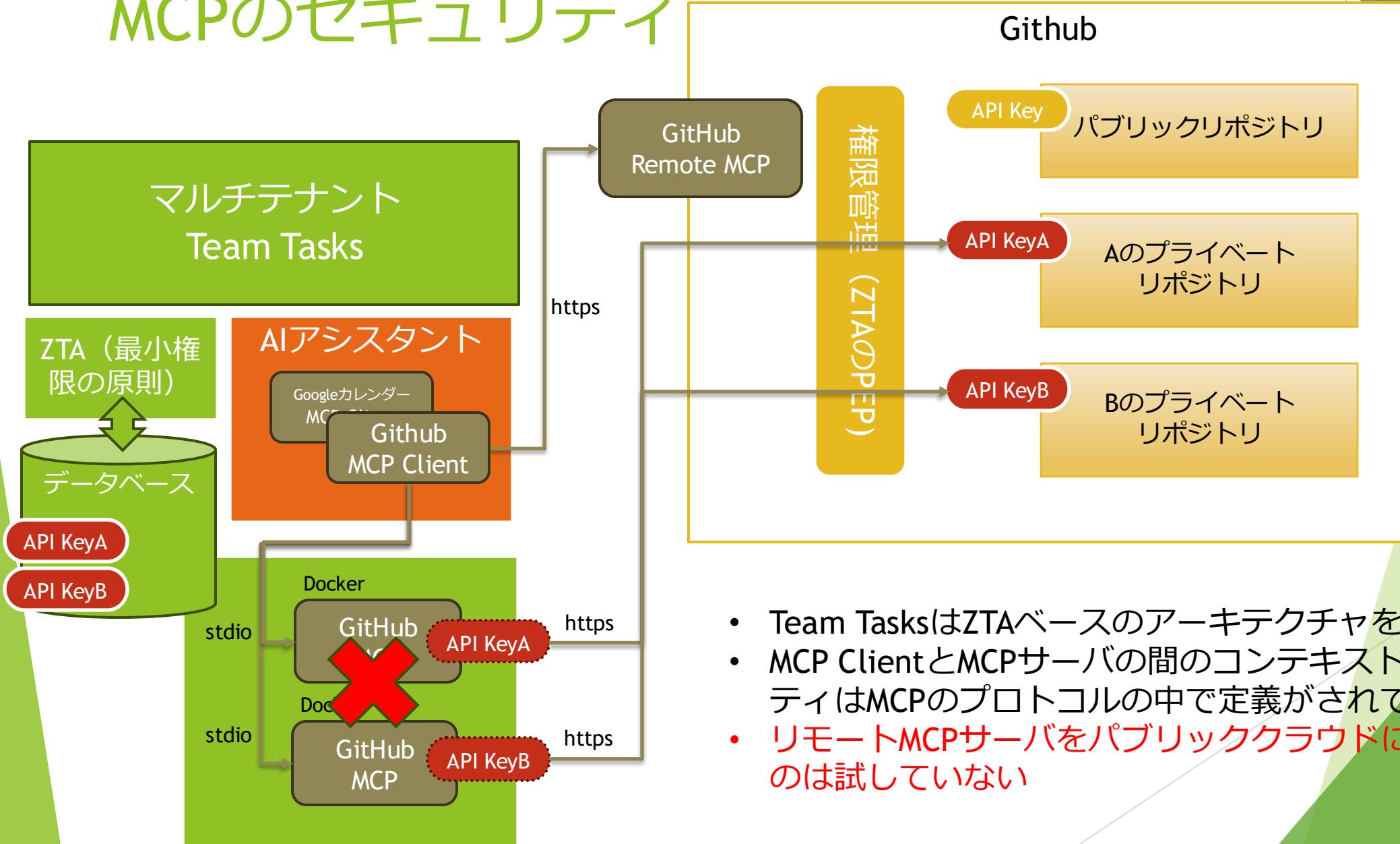
AIアシスタン機能

MongoDBを活用

AIアシスタントを使ってMCP経由で他システムと連携



# MCPのセキュリティ



# Google Calendar連携の開発例

AIアシスタント

Gemini MCP Claude MCP GitHub MCP有効

来週のGoogleカレンダーの予定を検索してタスク化してください。

実行

AI回答:

申し訳ありませんが、来週の予定は見つかりませんでした。

表示されている予定は以下の2件で、どちらも2025年10月の予定となっています：

- 2025年10月10日(金) 8:00- TPI事業会議
- 2025年10月17日(金) 8:00- TPI事業会議

来週の予定を検索してタスク化するようご依頼いただきましたが、来週の予定は登録されていないようです。必要に応じて予定の追加や、別の期間での検索をお試しください。

GitHub MCP使用 • 09:54

	内容	時間
調査フェーズ	OSSなどで公開されているMCPが使えるかどうかの調査	30分
Google API連携	MCPを使わずに連携（1週間分の自動同期）	20分
MCP化	MCP化をしてAIアシスタントからプロンプトでカレンダー予定のタスク化（デバッグ含む）	90分

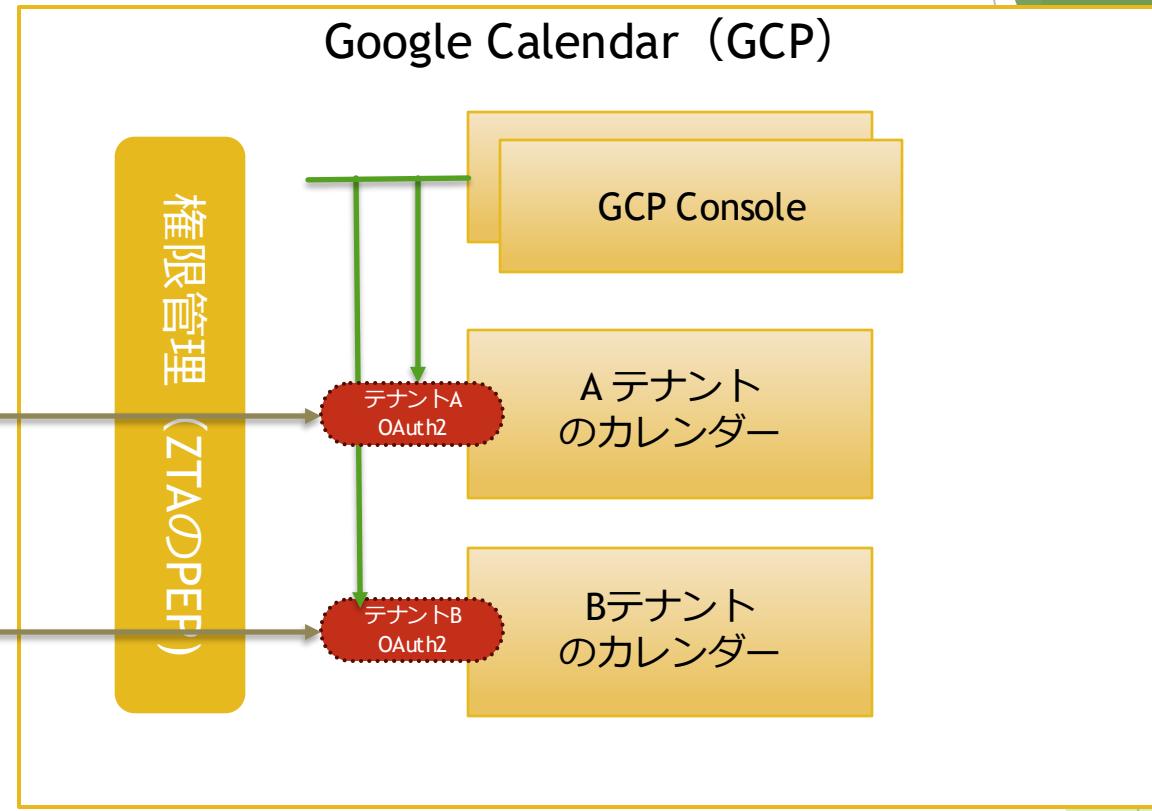
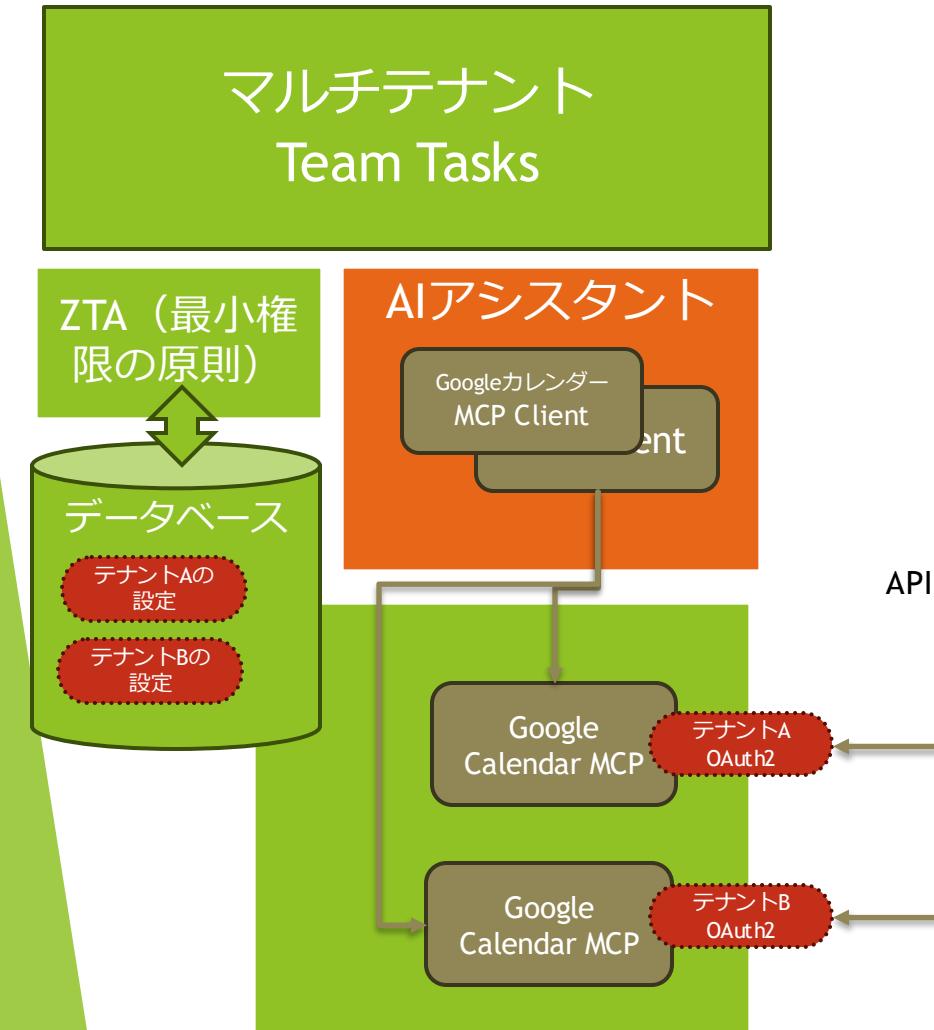
カレンダー予定 (2件)

全てタスク化

- TPI事業会議  
10月10日(金) 08:00 2人
- TPI事業会議  
10月17日(金) 08:00

コンテキストのリミット制限により複数日に分割して実装。AIの上位のライセンスがあると解決する (US\$200-300)

# Google Calendar



- Google Calendar APIを利用しAPI/Oauthのセキュリティで実装
- Githubと同様にMCP ClientとMCPサーバの間のコンテキストで利用
- ローカルに Google Calender専用のMCPサーバ機能を実装

# MCPのクエリー（デモ用）

- ▶ Techies-T/MCP-testの未解決Issueを検索して
- ▶ 今週のGoogleカレンダーを検索して、タスク化してください

# Coding Agentの最近の動向

AMDと協業

TPUも使ってみよう

NVIDIAと協業

- ▶ リリース済み、予定
  - ▶ 8月にGPT-5がリリース
  - ▶ 9月末Claude Sonnet 4.5がリリース
  - ▶ 10月にGemini 3.0 Proがリリース（発表）予定
- ▶ オムニモデルとなり、LLMとReasoningが合体すると予測
  - ▶ Claudeは企業向け、コーディングに強いイメージ
  - ▶ Geminiは企業向け、科学の発見（専門家モデル）に強み
  - ▶ OpenAIは消費者向け？、科学の発見（専門家モデル？）



事前学習よりも推論時の計算能力が必要となってきた  
いる

独自チップも開発しよう  
• Apple, MS, AWS

# Coding Agentの課題

## ▶ コンテキスト（文脈）

- ✓ 生成AIを使ってCodeを書くにはコンテキストを正しく保つ必要がある
- ✓ CursorのCEOは10万行のプログラムを一気に理解するには100万トークンが必要と説明している
- ✓ レートリミットで1日使えない場合もある

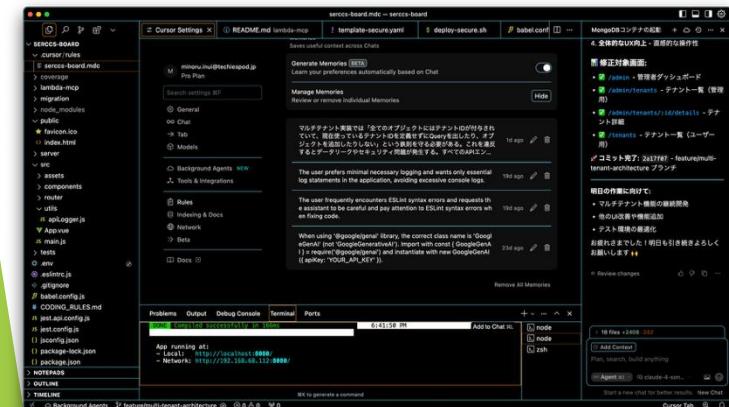
## ▶ モデルのコンテキストウィンドウ

- ✓ Gemini 2.5 Pro (1M Token)
- ✓ Claude Sonnet 4.5 (1M Token)

# コンテキストウィンドウ

プロンプト①

Gemini APIを使ってGithub MCPからデータをとるプログラムを書いてください



プロンプト②

Gemini API Keyの設定画面を作成してください

これだけではなくソースコードもコンテキストウィンドウとしてカウントされる  
(100万コードで大体1000万トークン)

コンテキストウィンドウ

システムインストラクション (Cursor機能)

cursor.rule(GEMINI.md, CLAUDE.md)  
Memory1  
Memory2 ...

CODING\_RULES.md (Cursor機能)

## 🎯 基本方針  
- \*\*Vue 3 + JavaScript\*\* ベースでの開発  
- \*\*クラス構文を避け、プレーンオブジェクト\*\*  
を優先

プロンプト① (プログラム)

Gemini APIを使ってGithub MCPからデータをとるプログラムを書いてください

プロンプト② (プログラム)

Gemini API Keyの設定画面を作成してください

Claude/  
Gemini

コンテキストウィンドウ

# Gemini CLI 2.0

▶ Claudeと同様Plan Modeがリリース

- ✓ 計画と実装は自動で判別

▶ 最初はGemini 2.5 Proを利用するが、途中でGemini 2.5 Flashに変更

- ✓ 計画は2.5 Proでコーディングは2.5 Flash

The screenshot displays the Gemini CLI interface with two terminal panes. The top pane shows a large 'GEMINI' logo with a pixelated effect and some introductory text:

```
# システム管理機能実装計画書
## プロジェクト概要
システム管理者向けのテナント管理機能強化と、ステータスのテナント分離対応を実装します。
## 実装目標
1. テナント管理機能の強化（管理者・契約者情報の編集）
2. ステータスのテナント分離（MCP自動アクション対応）
```

The bottom pane shows the content of a file named 'REFACTOR\_SYSTEM\_ADMIN\_UI.md' with some analysis results:

```
REFACTOR_SYSTEM_ADMIN_UI.md — serccs-board
# システム管理画面 UI改修計画
## 概要
現在の統計情報中心のダッシュボードを、テナント管理に特化したシンプルな一覧画面に改修します。
## 遊歩状況
### Step 1: バックエンドAPIの強化
ご依頼ありがとうございます。マルチテナント環境におけるシステム管理者のデータアクセス権限について、現在のアーキテクチャを調査し、セキュリティを維持しつつ要件を満たすための設計案、実装仕様、そして具体的な計画を策定します。
```

Details from the analysis results pane:

- 既存の権限管理の仕組みの調査
- ファイル構造から、以下のファイルが権限管理に重要であると推測されます。
  - \* server/middleware/tenantIsolation.js: テナント分離を行う中心的なミドルウェア
  - \* server/middleware/roleValidation.js: ユーザーの役割（ロール）を検証するミドルウェア
  - \* server/models/User.js, server/models/TenantUser.js: ユーザー、テナント、役割の関連性を定義したデータモデル
  - \* server/routes/system-admin.js: システム管理機能のAPIエンドポイント
- これらのファイルを読み解き、現在のアーキテクチャを分析します。

Terminal output:

```
✓ ReadManyFiles Will attempt to read and concatenate files using patterns: ...
## ReadManyFiles Result (Target Dir: '/Users/minoru_inu/Programs/clone-serccs/serccs-board')
Successfully read and concatenated content from **5 file(s)**.

**Processed Files:**  
- server/middleware/roleValidation.js  
- server/middleware/tenantIsolation.js  
- server/models/TenantUser.js  
- server/models/User.js  
- server/routes/system-admin.js

* 調査結果
1. テナント分離 ('tenantIsolation.js'):
```

# Claude Code 2.0.14 (Sonnet 4.5)

- ▶ ドキュメント（仕様書）を作成してみた
  - ✓ 20ファイル変更
  - ✓ 7,369行追加
  - ✓ 7つの新しいコアドキュメント
  - ✓ 3つのサポートファイル（スクリプト、ユーティリティ）
  - ✓ docs/フォルダーを3カテゴリーに整理
- ▶ 3時間ほどでソースコードを読みながら仕様書を作成
  - ✓ API、データモデル、セキュリティ、MCP連携の仕様など

# Npmサプライチェーン攻撃

- ▶ npmサプライチェーン攻撃が9/15ぐらいから発生している (Shai-Hulud)
  - ✓ npmで実行するパッケージの管理者を攻撃してパッケージにWormを仕掛ける
  - ✓ 例として@ctrl/tinycolor package (2M DL/w)
  - ✓ Wormは開発環境や実行環境からデータを盗む
  - ✓ 特にAPI Keyなどを盗みなりすましをする
  - ✓ 50→200→500 Packagesが被害
- ▶ **Vibe Coding**する人も脆弱性の知識が必要である

The screenshot shows a blog post from StepSecurity titled "Shai-Hulud: Self-Replicating Worm Compromises 500+ NPM Packages". The post discusses a self-propagating supply chain attack where the malware harvests AWS/GCP/Azure credentials using Truffle-Log, establishes persistence through GitHub Actions backdoors, and automatically spreads to other maintainer packages. It highlights the first successful worm attack in the NPM ecosystem. The sidebar features a purple banner for the "@ctrl/tinycolor Supply Chain Attack" with statistics: 40+ packages infected, 2M+ weekly downloads, and critical severity level.

<https://www.stepsecurity.io/blog/ctrl-tinycolor-and-40-npm-packages-compromised>

# 生成AIによってSWEの仕事が奪われるって本当？（開発者目線）

- ▶ AIが作成した仕様（Spec.）をみたことがありますか？
- ▶ コーディングフレーションという言葉まで出ている。
- ▶ 一方でAIエンジニアは不足している
- ▶ PM（プロジェクトマネージャ）の定義が変わるかも
- ▶ PM：プロダクトマネージャ
  - ✓ 製品のビジョン
  - ✓ アーキテクチャ設計（コンテナ化して運用の自動化）
  - ✓ ビジョンを持ってAIと接する

プログラミングがコミュニケーション化してコーディングだけする人はいらないかも

プロダクト  
デザイナー

ただし、安心するのは少し早い！

# AIが作成するデータモデル（仕様）

```
71 ---  
72  
73 ## コレクション一覧  
74  
75 | コレクション名 | 説明 | テナント分離 | ワークスペース分離 |  
76 |---|---|:---:|:---:  
77 | `tenants` | テナント情報（契約・課金・設定） | - | - |  
78 | `users` | ユーザー情報 | ✓ | - |  
79 | `workspaces` | ワークスペース（プロジェクト） | ✓ | - |  
80 | `tasks` | タスク情報 | ✓ | ✓ |  
81 | `statuses` | ステータス（未着手・進行中等） | ✓ | - |  
82 | `categories` | カテゴリ（ワークスペース固有） | ✓ | ✓ |  
83 | `tenantCategories` | テナント共通カテゴリ | ✓ | - |  
84 | `workspaceCategoryMappings` | ワークスペース-カテゴリマッピング | ✓ | ✓ |  
85 | `workspaceMembers` | ワークスペースメンバー関係 | ✓ | ✓ |  
86 | `workspaceSettings` | ワークスペース設定 | ✓ | ✓ |  
87 | `quizzes` | クイズ定義 | ✓ | ✓ |  
88 | `quizAttempts` | クイズ回答履歴 | ✓ | ✓ |  
89 | `googleTokens` | Google OAuth トークン | ✓ | - |  
90 | `calendarSyncLogs` | カレンダー同期ログ | ✓ | ✓ |  
91 | `aggregationPrompts` | AI集約プロンプト | ✓ | ✓ |  
92 | `auditLogs` | 監査ログ | ✓ | - |  
93 | `tags` | タグ（将来拡張用） | ✓ | - |  
94 | `tenantUsers` | テナント管理ユーザー | - | - |  
95  
96 ---  
97
```

```
100    ..    ### 1. Tenants (テナント)  
101  
102    **コレクション名**: `tenants`  
103    **説明**: マルチテナントの中核となるテナント情報を管理  
104  
105    #### 主要フィールド  
106  
107    ```javascript  
108    {  
109       name: String,                           // テナント名  
110       subdomain: String,                    // サブドメイン（一意）  
111       status: Enum,                         // active | suspended | trial | pending | deleted  
112       plan: Enum,                          // free | pro | enterprise  
113  
114       // テナントタイプ  
115       tenantType: {  
116         type: Enum,                         // domain (ドメイン型) | team (チーム型)  
117         selectedAt: Date,  
118         canChange: Boolean,  
119         migrationHistory: Array  
120       },  
121  
122       // ドメイン型設定  
123       domainSettings: {  
124         allowedDomains: [String],  
125         domainVerification: Map,  
126         autoJoinPolicy: Object,  
127         restrictions: Object  
128       },  
129
```



# 生成AIによってSWEの仕事が奪われるって本当？（経営目線）

## ▶ Tiny Teamを忘れるな

- ✓ 12ヶ月でUS\$100M以上のARRを上げるAIベンチャーが増えていく
- ✓ しかし、

## ▶ 経営目線

## ▶ 少なくとも、 とすると、 す可能性

### 結論

AIはコードを作れないとかの議論の前に実際に動くソフトウェアを作成してみよう

Cursor (Anyspere)

株式評価：9.9B

ARR \$100Mまでの年数1年

ARR : US\$500M

従業員：100名