



Treasure AI Studio

The **Agentic Experience** Platform

Treasure Data, Chief Architect

Sadayuki Furuhashi

About me

- Sadayuki Furuhashi
 - @frsyuki
- Co-Founder of Treasure Data. Chief Architect
- Data-oriented OSS - MessagePack, Fluentd, Digdag, Embulk
- Treasure Data
 - Cloud Datawarehouse
 - Customer Data Platform
 - Agentic Experience Platform



本日のトピック

- Treasure AI Studio とは？
- 開発背景
- AI OS アーキテクチャ
- AI ネイティブ開発



Treasure AI Studio とは？

開発の背景



Live Demo

Data Analytics

Network Security

Package Installation

Mobile App



Treasure AI Studio 開発まで

- 2月 - Treasure Work のセキュアなホスティング版として開発開始
 - [plans/2026-02-17-basic-architecture/01-chat.md](#)
- 3月 - ウェブ、モバイル、デスクトップアプリまで動作
- 4月 - 本番環境にリリース
- 5月 - モニタリング、CI/CD、運用自動化
- 6月以降 - フリーサインアップ、外部コネクタ、スキル管理機能、定期実行など機能拡充



Tuesday, February 17th ▾



Ko Sato 16:25

joined #dev-agent-platform-ja. Also, Kazuki Tsujimoto and 2 others joined.



Ko Sato 16:26

- GitHub Repo → `treasure-data/tdq` で作っておきます
- SAML + Local API → hkdnet
- E2B → k-tsj
- UI-API Protocolo + Stream Design → sada
- UI → tsubasa

2月17日
開発開始

2月27日
AI チャット動作

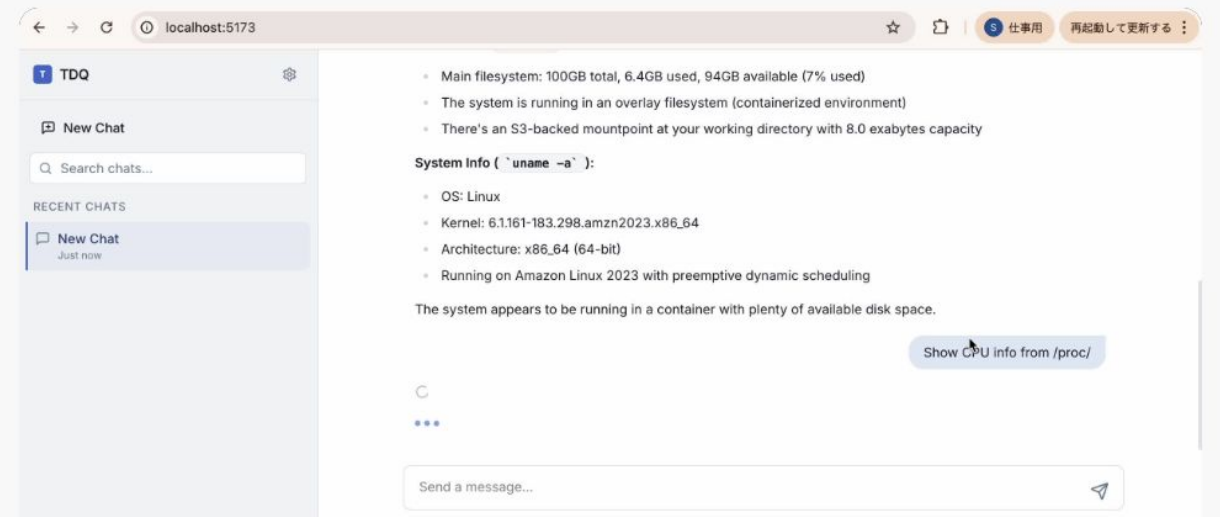


Sada Furuhashi 14:05

うごいたー！

sandbox-demo.mov ▾

Friday, February 27th ▾

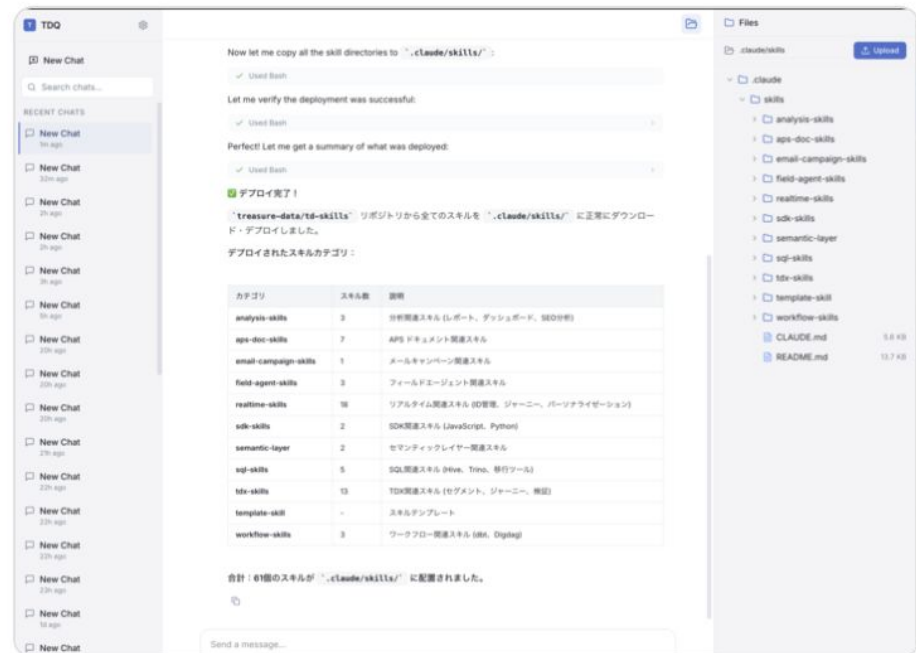




Tsubasa Tomoda VIP 15:02

Screenshot 2026-03-03 at 15.02.04.png ▼

Tuesday, March 3rd ▼



3月3日

デスクトップアプリ動作

ファイル アップロード/ダウンロード

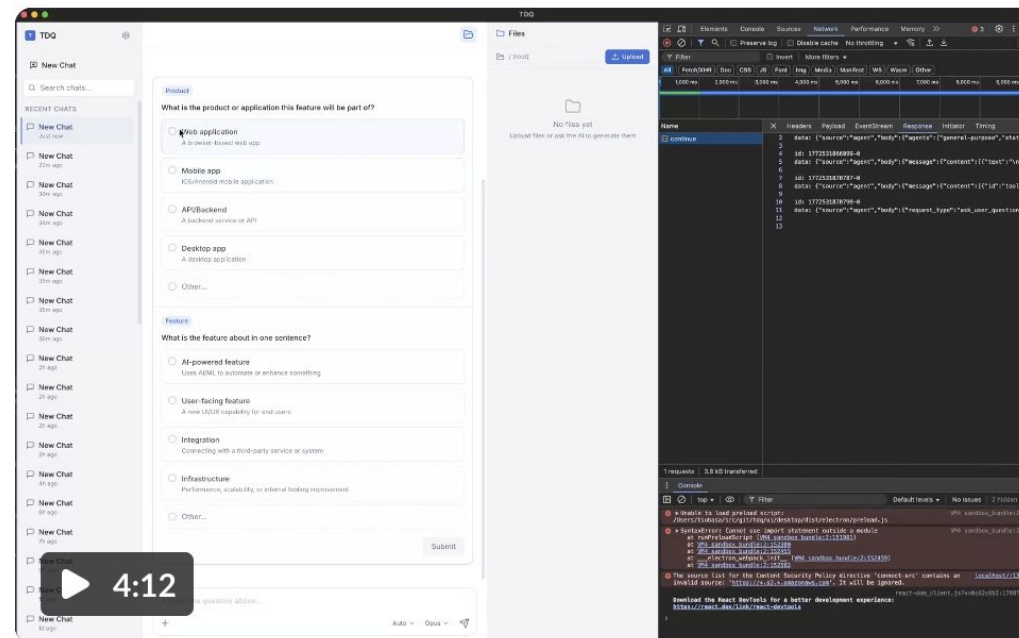


Tsubasa Tomoda VIP 19:03

Tuesday, March 3rd ▼

なんかまだ全然だけどひとまずAskUserQuestion出来るの確認できました

Screen Recording 2026-03-03 at 18.57.05.mov ▼



3月3日

AskUserQuestionツール動作

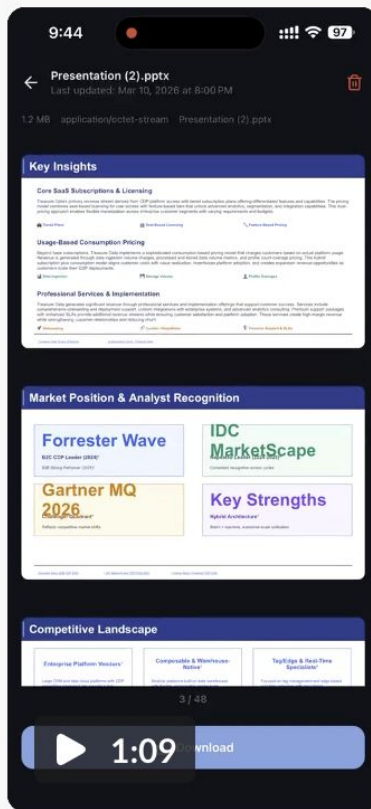


Tsubasa Tomoda VIP 09:47

Saturday, March 14th ▾

モバイルでもパワポ、ジャーニーちゃんが表示できるようになった

ScreenRecording_03-14-2026 09-44-06_1 ▾



3月14日

モバイルアプリ動作

アプリ内でパワポ描画



Kazuki Ota VIP 11:04

Monday, April 20th ▾

会場、こんな感じです

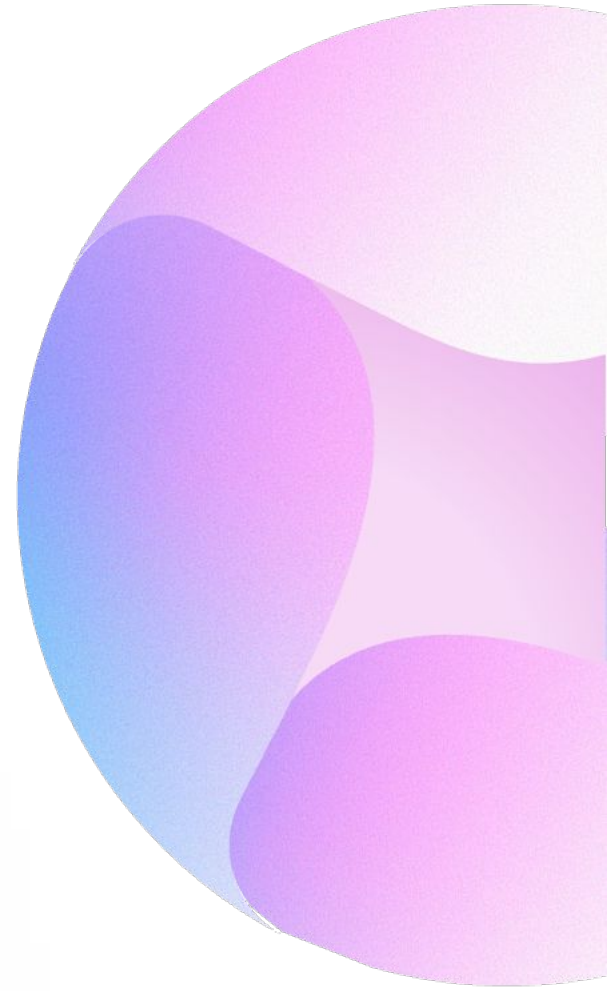
IMG_9574 ▾



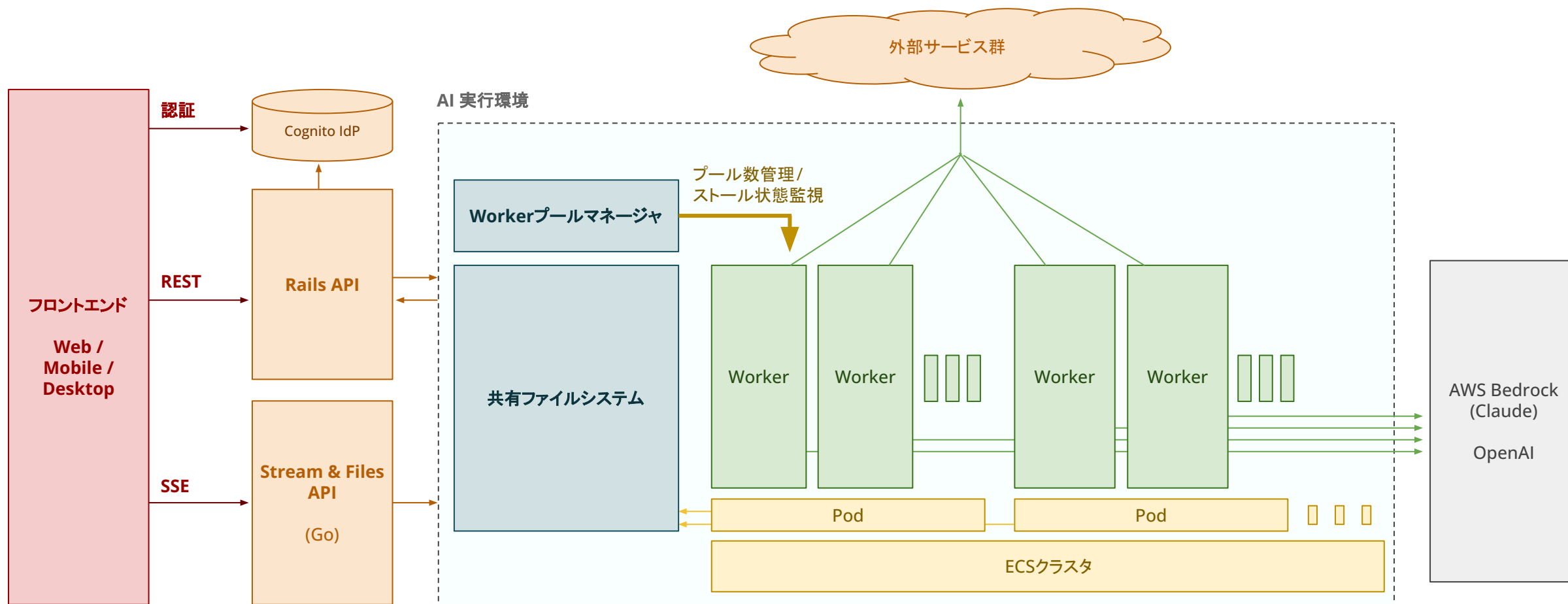
4月20日

ハンズオンイベント
本番環境にリリース

システムアーキテクチャ

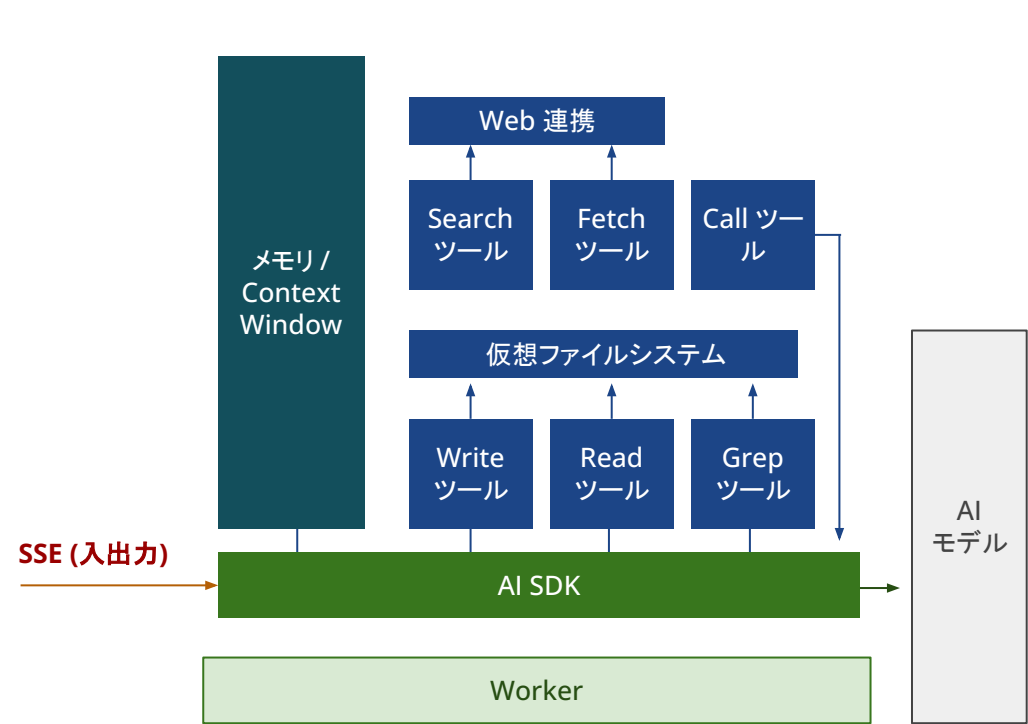


システムアーキテクチャ

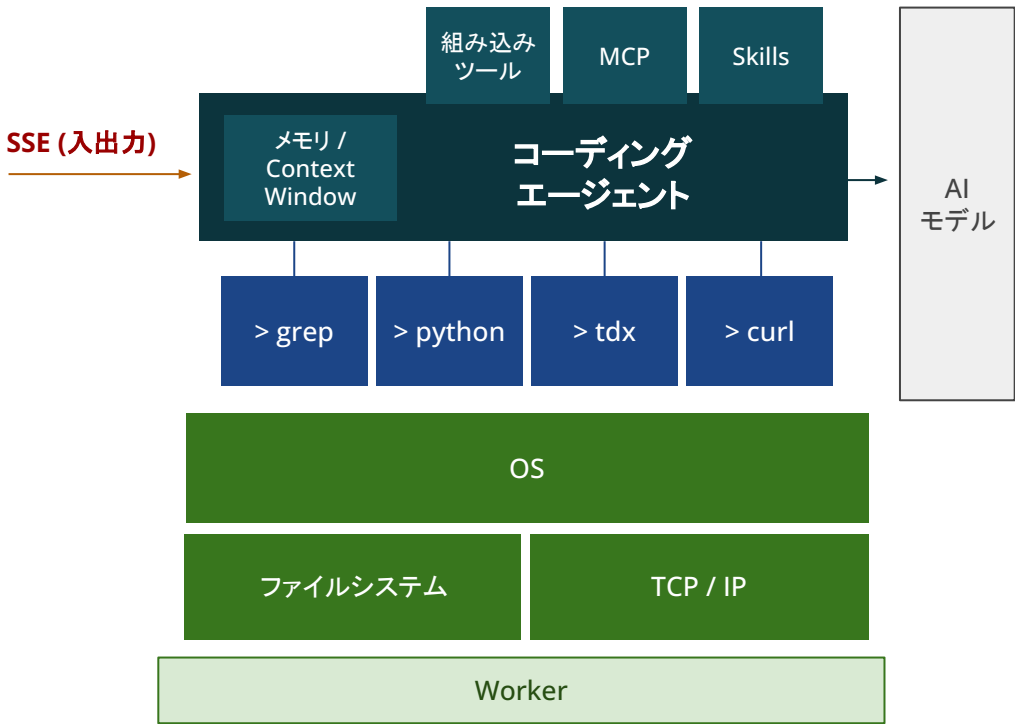


AI 制御方式

AI SDK 制御

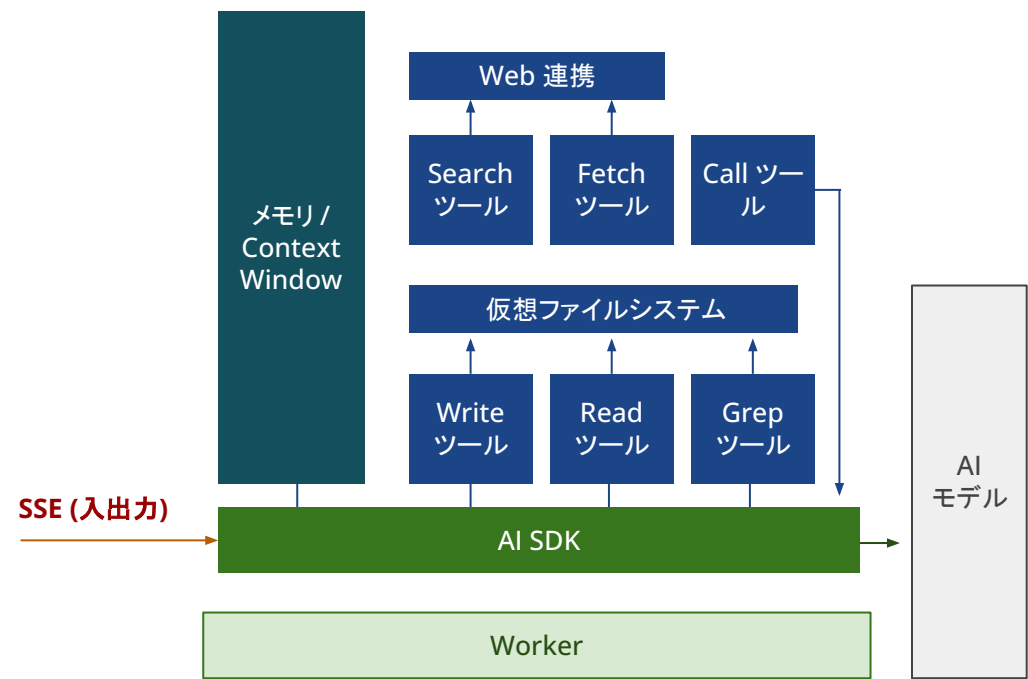


コーディングエージェント制御

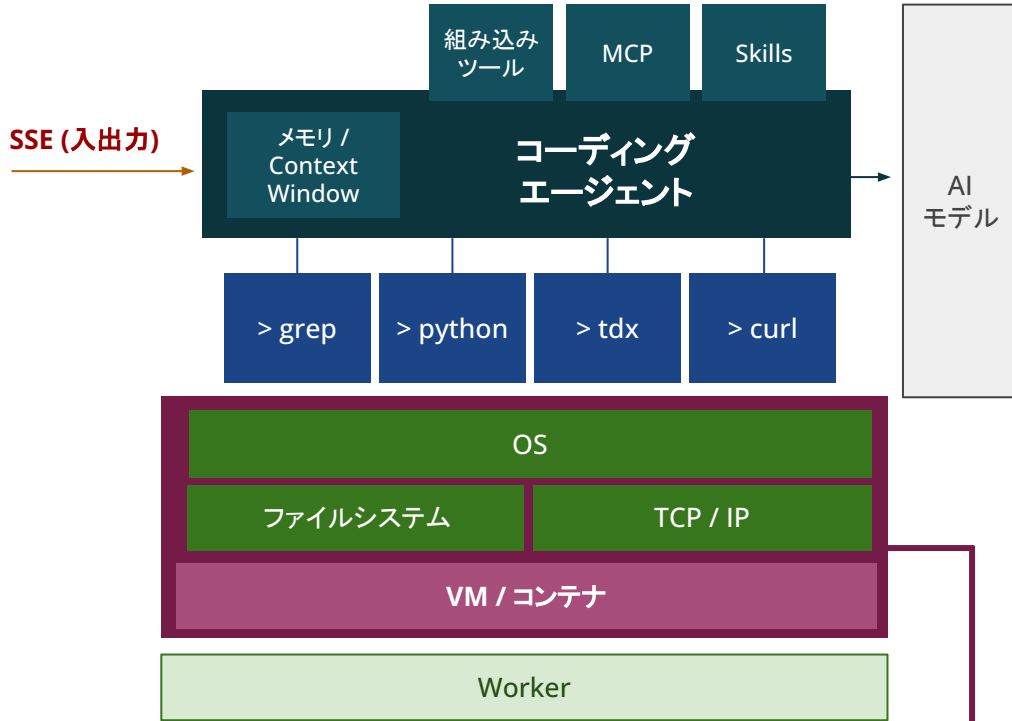


AI 制御方式

AI SDK 制御



コーディングエージェント制御



AI OS

AI 制御方式

- AI SDK 制御
 - LangChain、Vercel AI SDK、MAF、Strands Agents など
 - AI は常にツールを通して外界とインタラクション。AI に機能を追加するには まずツール開発
 - 信頼できるソフトウェアスタックを積み上げる
- コーディングエージェント制御
 - Claude Agent SDK、Codex SDK、OpenCode SDK など
 - 組み込みツールを通してファイルシステム、ネットワーク、各種ランタイム にアクセス
 - 機能を追加するには Skill や MCP を開発。または AI がスクリプトを書いたり OS 機能を勝手に使用
 - 信頼できないソフトウェアスタックを制御
- AI OS
 - 信頼できないソフトウェアスタックをセキュアに実行
 - クレデンシャルを AI に渡さずに実行(ゼロトラスト原則)
 - AI の挙動を記録し、危険な挙動は止める
 - AI が自律的に問題解決できるように CLI、ランタイム、ドキュメント を提供
 - データを保存・共有・バックアップする仮想ファイルシステム

Live Demo

Connector Credentials

Namespace Isolation

sudo & seccomp



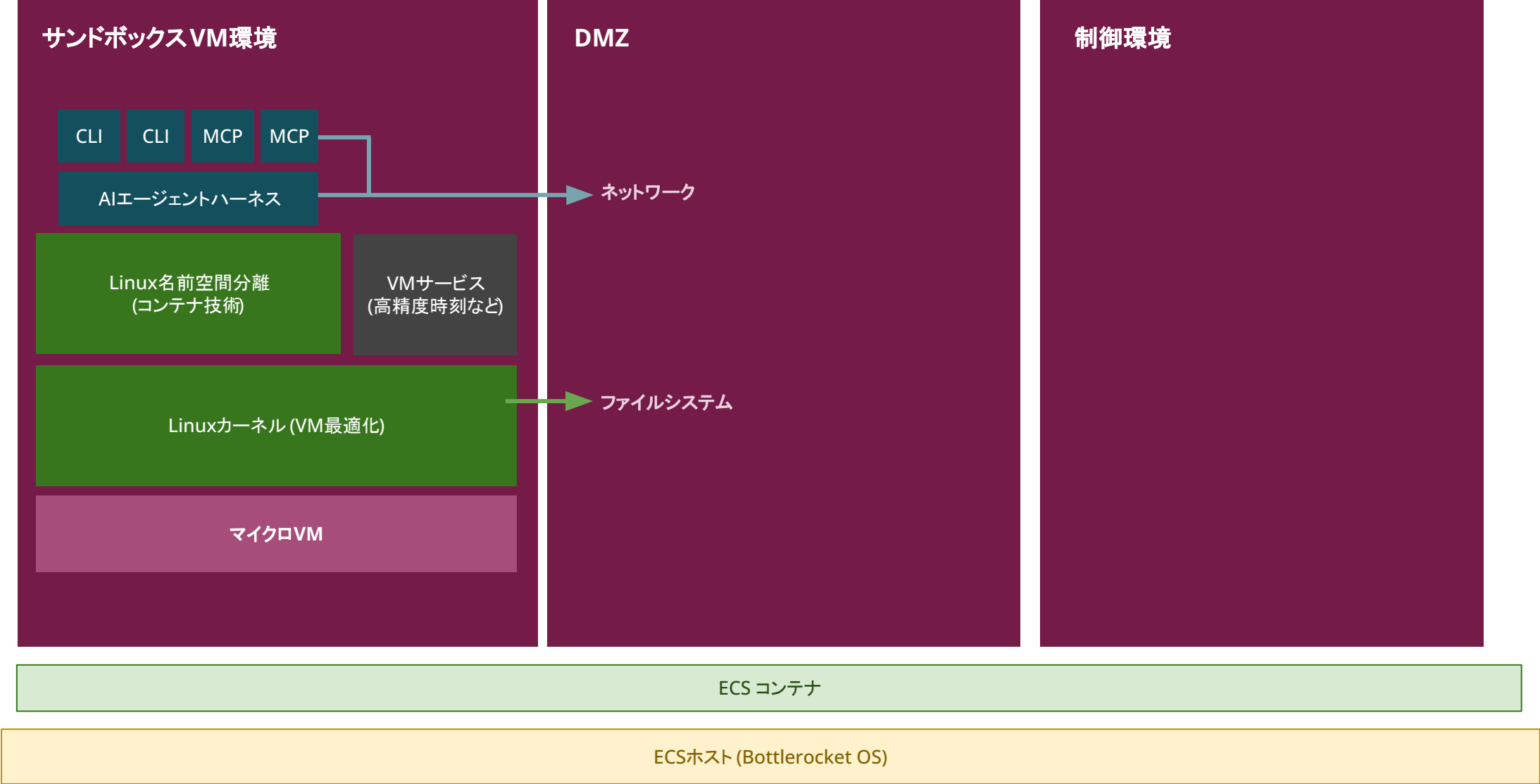
AI OS - 物理サーバ・オーケストレーション

- AWS ECS
 - Lambda で Worker プールマネージャを実装
- ARM 64 メタルインスタンス
 - コンテナではなく VM で分離が必須、しかし通常の EC2 インスタンスでは Nested VM になってしまう
 - Intel Nested VM も試したが安定せず
- Bottlerocket OS
 - コンテナの実行に最適化された OS
 - コンテナの実行以外が SELinux で封じられている
 - 起動後に設定変更ができない(イミュータブルOS)
 - 過去の CVE-2019-5736 などのコンテナ抜け脆弱性に耐性がある

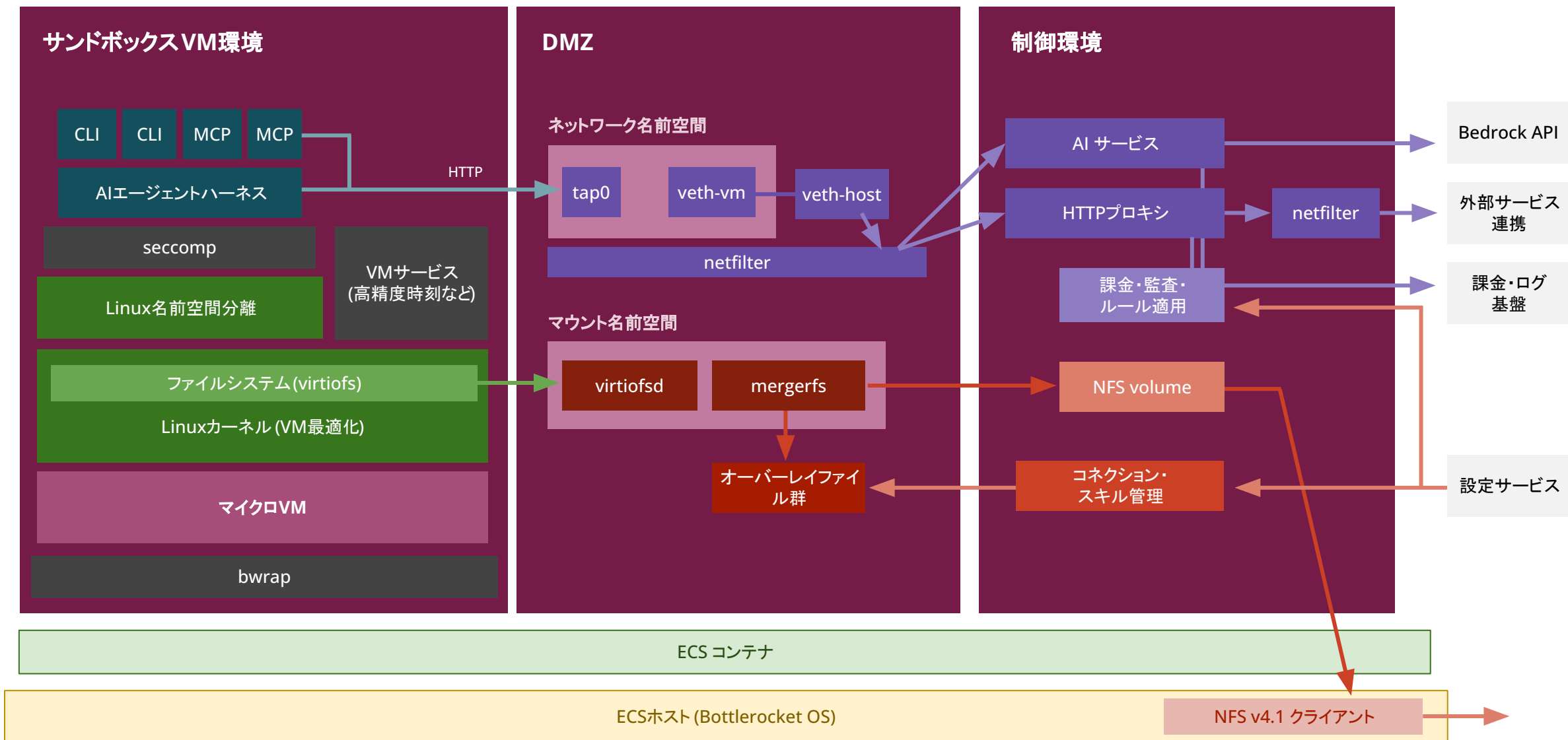
ECS コンテナ

ECSホスト (Bottlerocket OS)

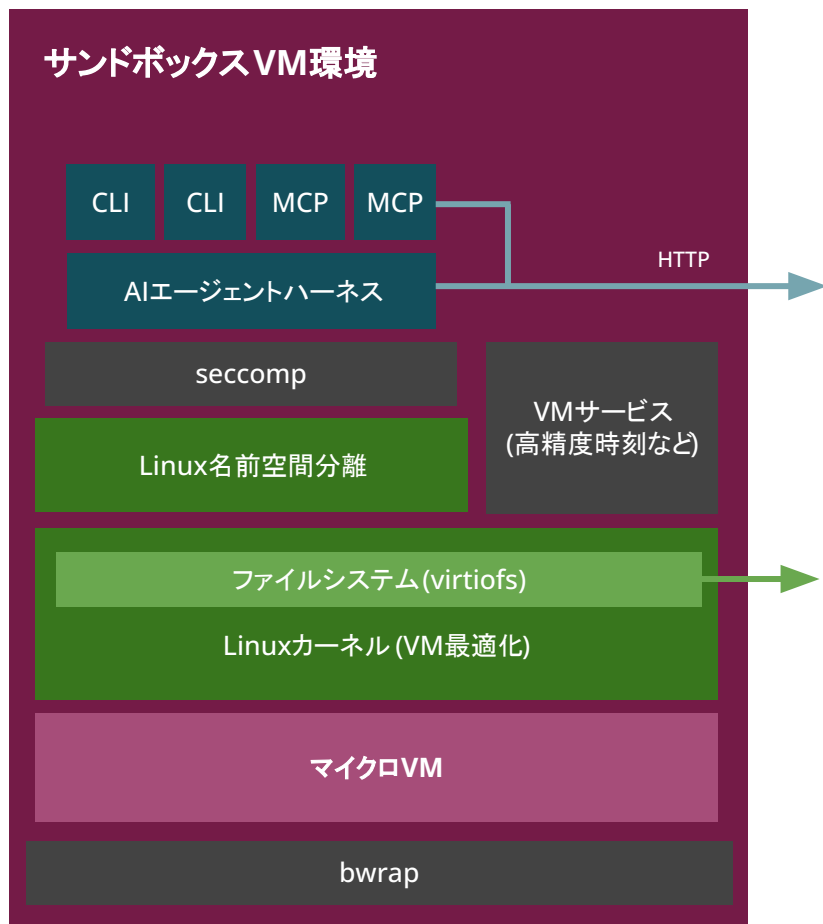
AI OS – Worker アーキテクチャ



AI OS - Worker アーキテクチャ

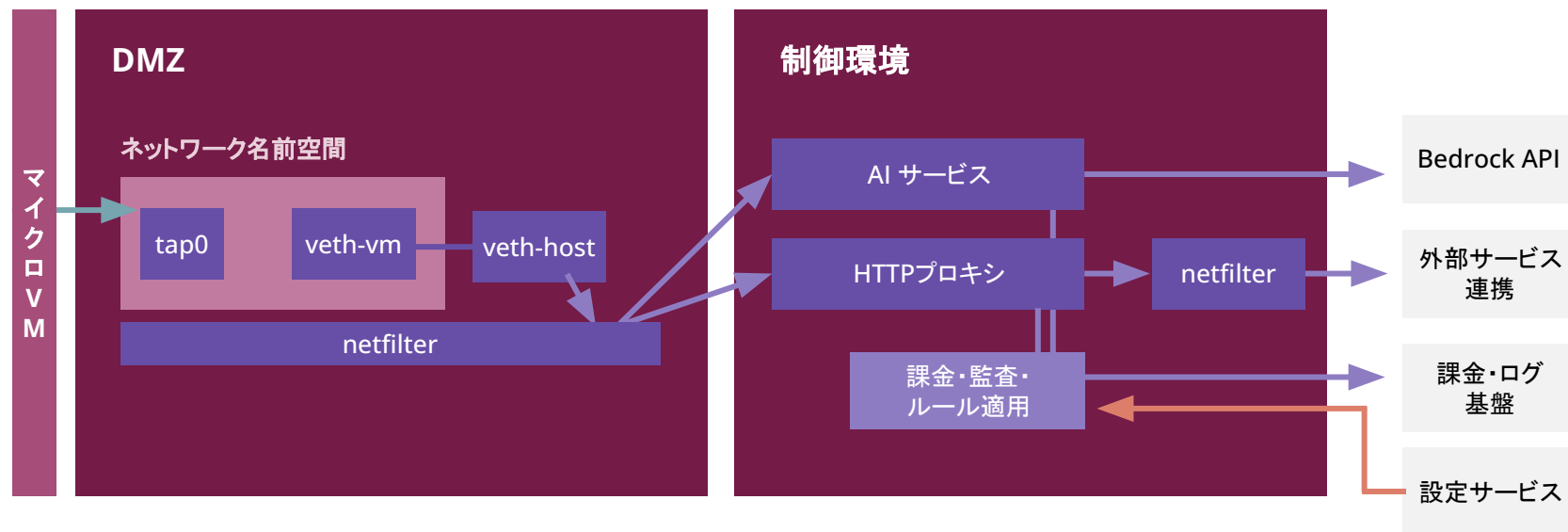


AI OS – サンドボックス VM環境



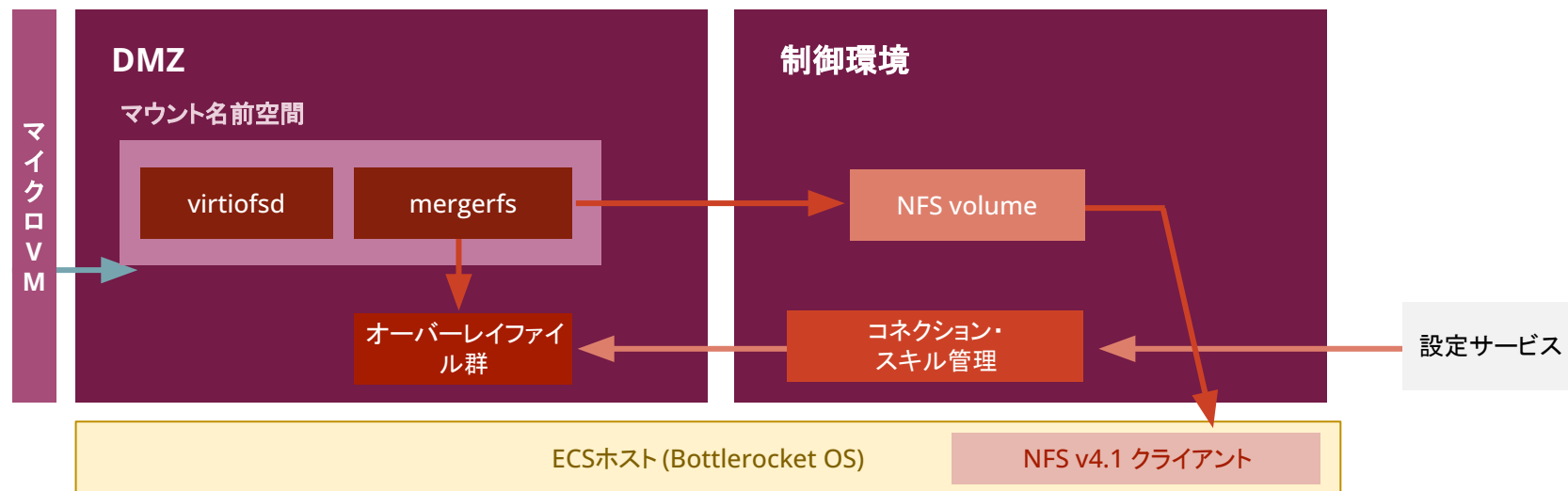
- AI エージェントハーネス
 - 乗っ取られる前提
 - root 権限も奪取される前提
- seccomp
 - AI エージェントが動作する最低限のシステムコールだけ許可
- Linux 名前空間分離
 - マウント名前空間やPID空間を隔離。適用したらrootでも抜けられない
- マイクロVM + Linuxカーネル
 - デバイスドライバや一般的でない機能が無効化されたカーネルを起動
 - CPU、メモリ、ディスク消費を制限
- bwrap
 - マウント名前空間をホスト環境から分離。root でも抜けられない
 - 必要なファイルだけ分離環境内に bind mount。/proc や /dev も分離
 - seccompも適用。マイクロVMが動作する最低限のシステムコールだけ許可。一度適用したら root でも抜けられない

AI OS - ネットワーク分離



- マイクロVMを分離されたネットワーク名前空間 (net ns) で起動
 - 通信はすべて net ns 内の tap0 から出力
 - マイクロVMを抜けられたとしても net ns 内からの出力
- net ns から外への通信は必ず veth-host を経由
 - AI サービスへの通信は通過
 - その TCP 通信は netfilter で宛先を捻じ曲げて HTTP プロキシへ
- HTTP プロキシで宛先ドメイン名にマッチして遮断
 - HTTP プロキシからの通信は UID でマークして VPC NAT Gateway へ

AI OS - ネットワーク分離



- マイクロVMに virtiofsd でファイルシステムを共有
- virtiofsd は分離された名前空間で実行
 - virtiofsd を奪取されても制御環境に到達できない
- 名前空間内にはmergerfs で構成されたファイルシステムだけを共有
- mergerfs で共有ファイルシステム (NFS) にオーバーレイファイル群をかぶせる
 - オーバーレイファイルがあればそれを読む、そうでなければ NFS から読む挙動
- オーバーレイファイルとしてスキルやコネクション情報を配置
 - 設定変更されたらオーバーレイファイルを即時追加・更新

AI OS にさらに必要な機能

- ネットワーク通信の自動分類と遮断
 - HTTPプロキシでガードレールを適用する
 - ユーザーの指示や意図に従った REST API 呼び出しでなければ許可を求める・遮断する
- ファイル更新のバージョニング・巻き戻し
- 複数ユーザー間でファイル共有
- AI とユーザーが同時に共同ファイル編集
- スキル・MCPのパッケージマネージャ
- スケジュール実行・通知
- etc...



AI ネイティブ開発



AI ネイティブ開発環境

- モノレポ構成
 - [monorepo + GitHub + CircleCI の制約、あるいはなぜ私は動的ワークフロー生成という黒魔術を採用するに至ったのか](#)
- 全コンポーネント並列開発 & ローカルテスト
- プランファイルをリポジトリにコミット
- デメリット: デプロイパイプラインの複雑化



AI ネイティブ開発環境

AI 以前

1. PoC
2. Confluenceにドキュメント作成
3. アーキテクチャレビュー
4. 各コンポーネント並列開発
5. コードレビュー
6. リリース

AI 以後

1. PoCしながら
plans/にドキュメント作成しながら
全コンポーネント開発して
ローカルで end-to-end テスト
2. アーキテクチャレビュー &
コードレビュー
3. リリース(スキルで自動化)

 **frsyuki** Change daily AOSS index to weekly ✓

0ae6f4d · 2 weeks ago

 History

Name	Last commit message	Last commit date
 ..		
 README.md	Change daily AOSS index to weekly	2 weeks ago

README.md  

Event Pipeline: 日次インデックスから週次インデックスへの変更

作成日: 2026-05-19 関連: plans/2026-04-07_opensearch-collection-sharing/README.md

背景

OpenSearch Serverless にはコレクションあたり **100 インデックス** の上限がある。現在 3 パイプライン (token-usage, worker-allocation, network-audit) が日次インデックス ({prefix}-YYYY-MM-DD) で 30 日保持されているため、最大 93 インデックスを消費する。

2-3 種のインデックスを追加する予定があり、日次のままでは上限を超える。

変更のきっかけ

Thank you

Let's build something **amazing**



Please visit us at: treasure.ai

