PCC Workshop in Kobe 2026「AIによるHPC」

AIを活用したHPC構築・運用と AI時代に向けたデータ管理の高度化

HPCI共用ストレージ環境でのAIおよびAI Agent活用

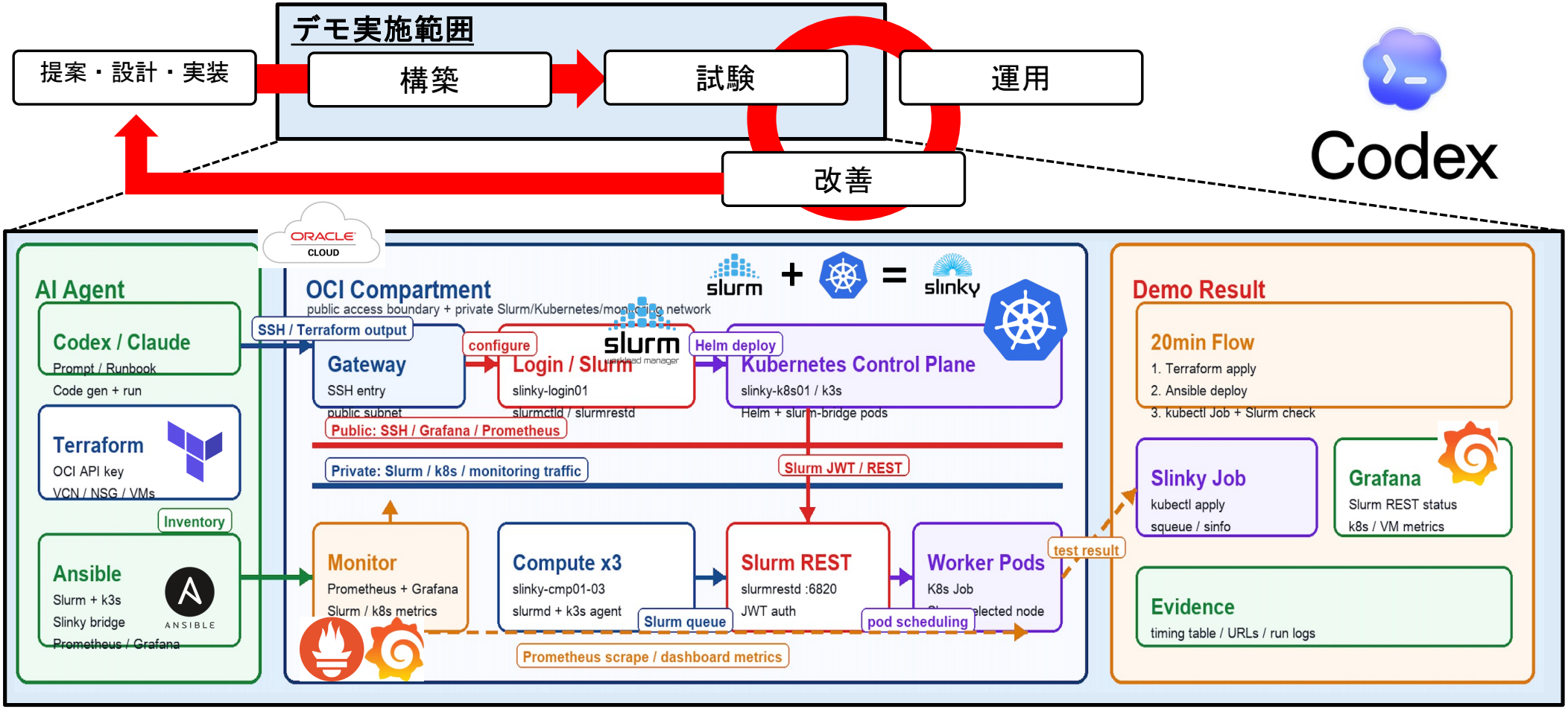
2026/07/03

理化学研究所
計算科学研究センター

運用技術部門
金山 秀智

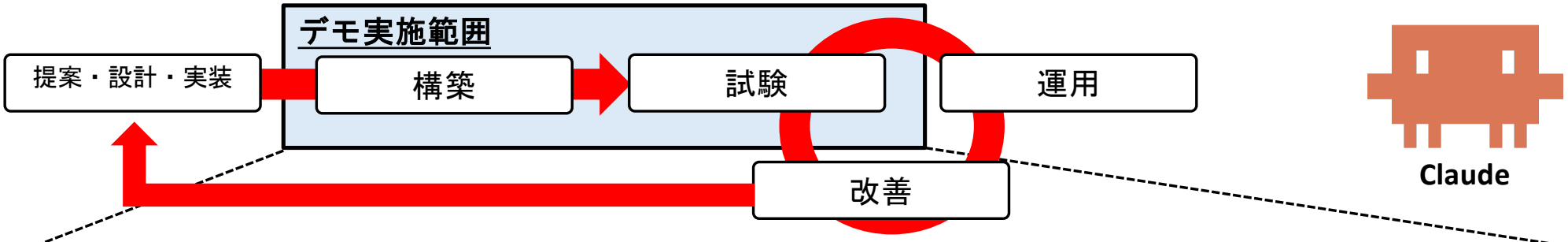
デモ1 : OCIを用いてSlinkyクラスタを作成 + 監視構築 + 試験を実施

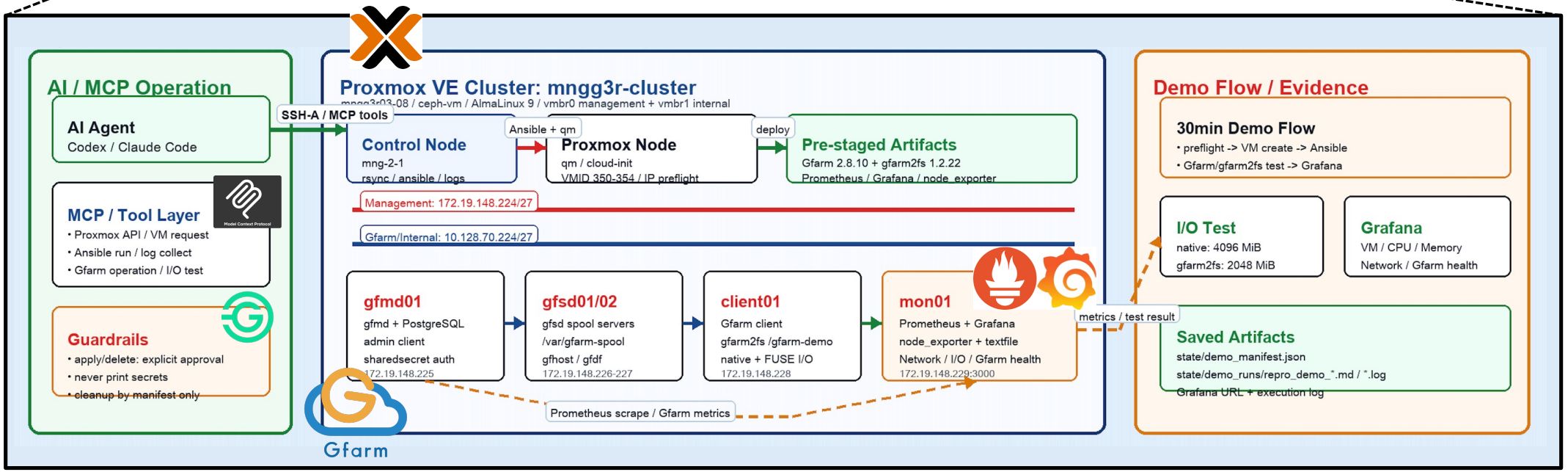
※ 実行時間の都合、デモを最初に実行します。



デモ2：Proxmoxを用いてGfarm試験環境とGfarm-MCPサーバ開発して試験を実施

※ 実行時間の都合、デモを最初に実行します。





自己紹介

金山 秀智 (Kaneyama Hidetomo)

理化学研究所 計算科学研究センター
運用技術部門　データ連携技術ユニット
先端技術ユニット

ストレージやシステムエンジニアリングを担当

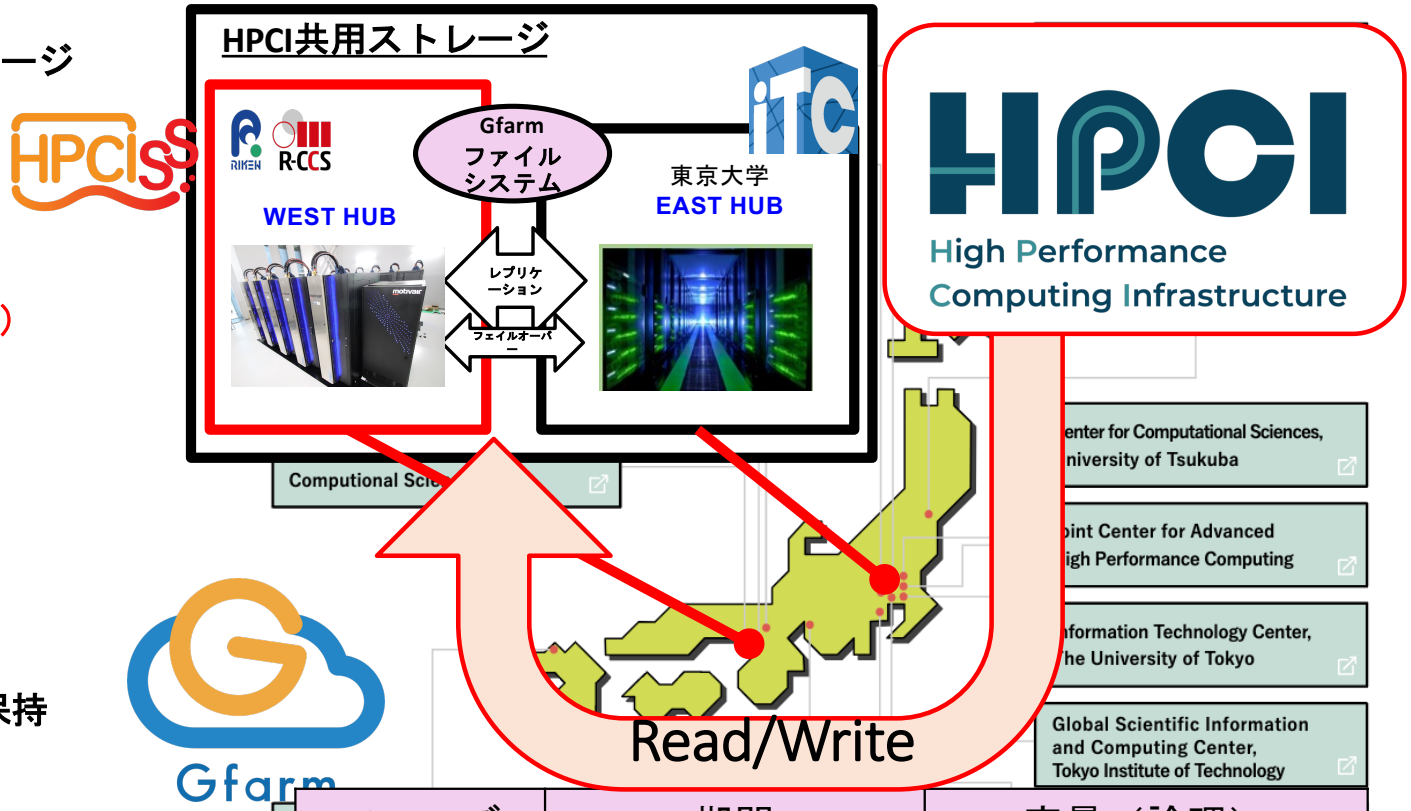
担当業務

- R-CCS HPCIおよびHPCI共用ストレージのエンジニアリング
- 富岳運用・保守
- 日中韓フォーサイトプロジェクト
- Fugaku Next ストレージ等々

HPCI共用ストレージについて

HPCIプロジェクトにおけるデータ共有ストレージ

- **目的:**
  - **研究データを保存**
  - **国内のスーパーコンピュータ間で研究データを共有**
  - **Open Science（研究データ公開・共有）**
- **運用:**
  - **東京大学**
  - **R-CCS**
- **運用期間:**
  - FY2012 -
- **レプリケーション:**
  - **東京大学とR-CCSの間で2レプリカを保持**
- **ファイルシステム:**
  - **Gfarmファイルシステム**
    - **開発: 筑波大学 建部先生**
    - **https://github.com/oss-tsukuba/gfarm**
- **現在R-CCS HPCI共用ストレージの”エンジニア”は私1人**
  - **(株)創夢殿 と (株)アルティネット殿と運用業務**
  - **NEC殿・DDN殿・DC ASIA殿でハードウェア保守**

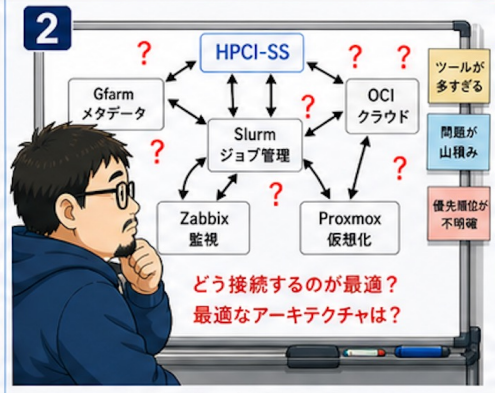


フェーズ	期間	容量（論理）
第1期	FY2012 – FY2018	10PB → 15PB
第2期	FY2018 - FY2024	45PB → 50PB
第3期(now)	FY2025 -	95PB+

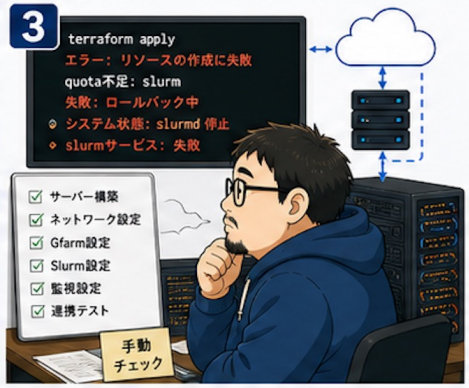
AIによる業務改善

AI活用前

調査に埋もれる



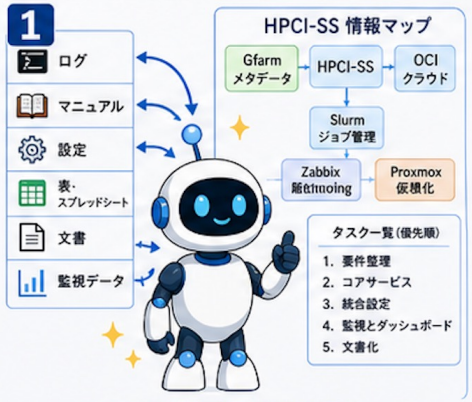
設計が止まる



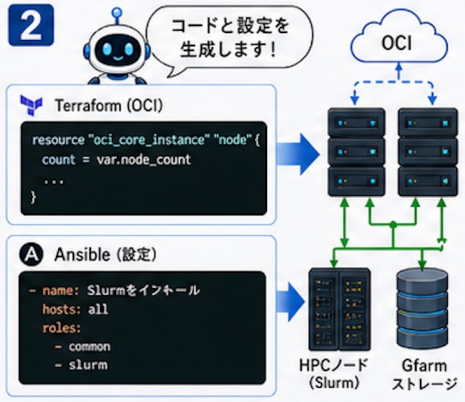
構築・試験が止まる

文書化が遅れる

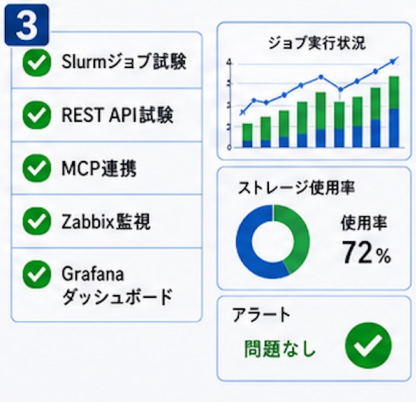
AI活用後



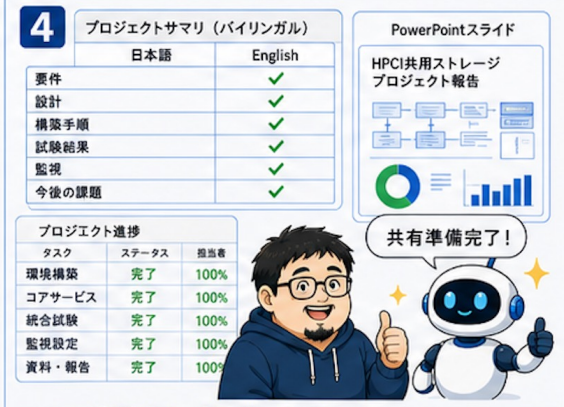
AIが情報を整理



コード生成で構築



試験と結果を自動化



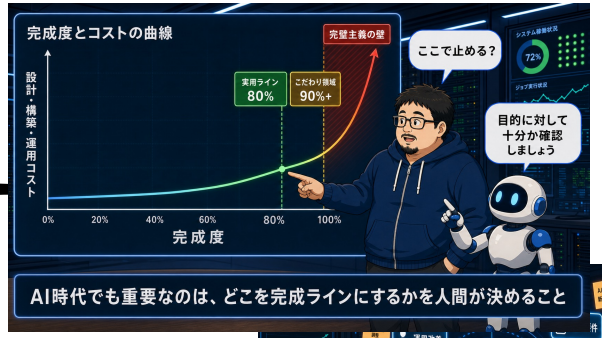
資料と報告を作成

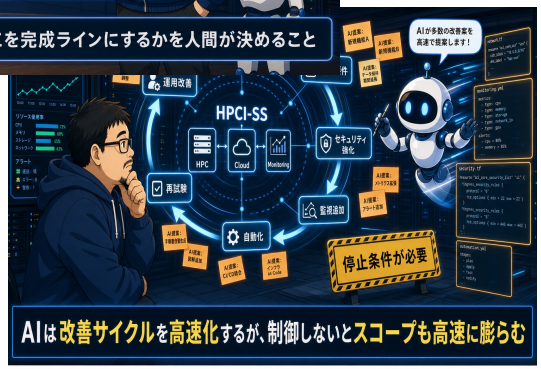
システムエンジニアの『あるある』とAI活用

- 完璧なシステムを求めがち/求められがち
  - どこまでも設計・構築・運用コストが増加(そして進まない)
- サービスレベル(例: 24/365対応)
  - 現実的には個人(部門や組織)のやる気や責任者に紐づきがち

- 属人化による弊害
  - 自動化・効率化を進んだことで、途中参加の難易度Up
  - セキュリティなどの即応性がルールを決めても担当に紐づく
  - 構成管理ツールによるコード化のあるなし
  - 使い古されたスクリプト群
  - 監視環境のアップデート
- 新たなチャレンジへの「ためらい」
  - プロジェクトの途中の環境変更への躊躇
  - 旧環境との接続を大事にして更新に臆病になる
  - レガシーな手法の選択
- ドキュメントの負荷
  - 自動化・コード化 vsドキュメント作成
    (効率化したはずがドキュメント作成作業が増えるパターン)
  - そして更新されないドキュメント

AIを使うことでコスト減にはなっても抜本的な改善ではない(？)





AIを使うことで大きく改善すると期待している範囲

AI Agentの簡単な利用設定

基本的な利用コマンドやオプションです。

項目	設定内容	利用箇所
Model / Effort	基本は最高モデル、xhigh/pro相当を指定 トークンを減らす際(特に構築や難しくない調査などではモデルやEffortを減らします)	全て
Mode	Plan	ドキュメント作成時の他ファイルや情報読み出し
	Auto	最終的な設計や構築・運用業務実施時
CLAUDE.md / AGENTS.md	日本語/英語(言語指定)、commit禁止、上書き禁止(backupルール)、Ansible等のコード化を強要、アクセスに関するwhite/blackルールの指定	プロジェクトルールを指定。 運用情報を外部に出力することが怖いので厳しく指定。このため現状はGitの利用については不可にしています。
skills	-	開発時でgitを使うときなどはdiffなどを使用しますが、活用できていないです。(運用チェックなどではskillsを作成して確認作業の効率化などを検討すべきかもしれません。)
--continue/--resume	-	中断した後の読み出し(claude code)

例: CLAUDE.md 抜粋

```
# Claude Code Project Instructions
このファイルはClaude Code向けのプロジェクト指示である。作業開始時に必ず読むこと。

## 1. 最優先ルール
- 日本語で出力する。ユーザーが英語成果物を明示した場合のみ、その成果物を英語にする。
- 利用可能な最高モデルを使う。思考設定は `xhigh`、または利用可能なら `pro` 以上相当を使う。
- 元プロジェクトは直接変更しない。
- 編集は作業コピー内に限定する。
- 実行内容は./log.d/以下に必ず記録する。
- 秘密鍵、APIキー、token、webhook URL、tfstate内の機密値を出力しない。
…
## 2. Git操作
- ユーザーの明示指示なしに `git commit` を行わない。
- ユーザーの明示指示なしに `git add`, `git push`, `git tag`, `git merge`, `git rebase` を行わない。
- 自動で行ってよいのは `git status`, `git diff`, `git log`, `git show` などの読み取り操作のみ。
- `git reset --hard`, `git checkout -- <file>`, `git clean` は明示指示なしに行わない。
…
## 3. 設定ファイル変更前のバックアップ
設定ファイルを変更する前に、必ずバックアップを作成する。
形式:
```text
<ファイル名>.%Y%m%d-%H%M.bk
```
対象:
- Terraform: `*.tf`, `*.tfvars`
- Ansible: inventory, `group_vars`, `host_vars`, playbook, role, template
- 設定: `*.json`, `*.toml`, `*.ini`, `*.conf`, `*.yml`, `*.yaml`
- Slurm, NFS, SSH, firewalld, Prometheus, Grafana, systemd関連設定
5. 自動実行してよい操作
…
```

基本的な利用ルールや制限を記載します。

運用利用や状況によっては、gitの利用自体を禁じることも多いです。

Backupルールやログの保存ルールなどを明記してます。

White/Black listなどを記載しています

AI Agent活用: ストレージおよびHPC関連調査・試験(2026年04月～06月)

期間	プロジェクト	利用AI/サービス	必要技術	AIで生成・支援した成果	納品物/成果物
2026/04	OCI Slurm + MCP + SlurmRest 調査	Claude Code Codex	MCP, OCI, Terraform, Ansible, Slurm, Slurm REST API	設計書、実装・構築書、Terraform/Ansibleコード、OCI APIによるテナント設定調査・改善、MCP/REST試験	設計書、構築書、Terraform、Ansible、試験仕様書、試験結果まとめ、Runbook
2026/05	Gfarm metadata / Vector search DB(基本設計)	Claude Code Codex	Gfarm metadata DB, pgvector, BGE-M3, OpenAPI/MCP, ACL	Metadata catalogとVector DBの分離、ACL pre-filter、自然言語検索/API方針の整理	要件、基本設計、API/MCP案、metadata search設計
2026/05	ネットワーク帯域改善調査	Claude Code(Opus/Sonet 4.7,4.8)	iperf3, TCP congestion, LACP, PFC, Host Pause	ログ整理、ボトルネック候補整理、次アクションの洗い出し	調査メモ、分析表、改善候補リスト
2026/06	Slurm bridge/Slinky環境調査	Codex 5.5	Slurm, Kubernetes, slurm-bridge, OCI, Terraform, Ansible,OpenMPI, Prometheus, Grafana	小規模構成再構築、REST job、MPI job、Slurm exporter/Grafana dashboard整備	計画書、実行ログ、時間計測表、14 panel dashboard、show_demo_urls、再実行プロンプト
2026/06	Gfarm(local engeenring & version up試験向け)	Claude Code(Opus)	Proxmox API, Ansible, Gfarm 2.8.10, gfarm2fs 1.2.22, Prometheus/Grafana	VM作成、Gfarm/gfarm2fs構築、大容量I/O試験、Network/gfmd/gfsd/I/O可視化	設計・実装仕様、artifact作成手順、Ansible playbooks、試験仕様、Grafana dashboard

AI Agent活用: 仮想基盤・モニタリング改善(2026年04月～06月)

期間	プロジェクト	利用AI/サービス	必要技術	AIで生成・支援した成果	納品物/成果物
2026/05-06	Proxmox基盤構築	Claude Code Codex	iDRAC/Redfish, Proxmox VE, Ceph, HA, vDPA/SR-IOV	iDRACに作業用アカウントを作成し、Redfish/VirtualMediaでProxmox OSを遠隔インストール。Ceph設計・性能試験、vDPA/SR-IOV試験を実施	設計書、構築手順、性能試験ログ、vDPA/SR-IOV調査、AI Agent向けRunbook
2026/05-06	k8s/ECK/Logging	Claude Code Codex Ollama	Kubernetes, ECK, Elasticsearch, Kibana, Logstash/Filebeat	k8s導入、Elasticsearch移植、ログ基盤の試験、監視・移行手順の整理	Kubernetes manifests、ECK移行手順、試験ログ、運用メモ
2026/05-06	Zabbix 7.4 + Gfarm監視	Claude Code(Opus)	Zabbix, TimescaleDB, Gfarm plugin, Ceph/K8s監視	Zabbix 7.4移行、Gfarm/Ceph/K8s監視、dashboard/trigger整理	監視設計、移行手順、dashboard、plugin/prototype、実行ログ
2026/06	Grafana tenant/self-monitoring	Claude Code(Opus/Fable) Ollama	Grafana API, Shibboleth, HPCI eppn, account CSV	tenant org provisioning、datasource/dashboard配備、自己監視の整理	同期スクリプト、設定手順、dashboard JSON、運用ログ

活用事例1: OCI Slurm + MCP / REST / 監視

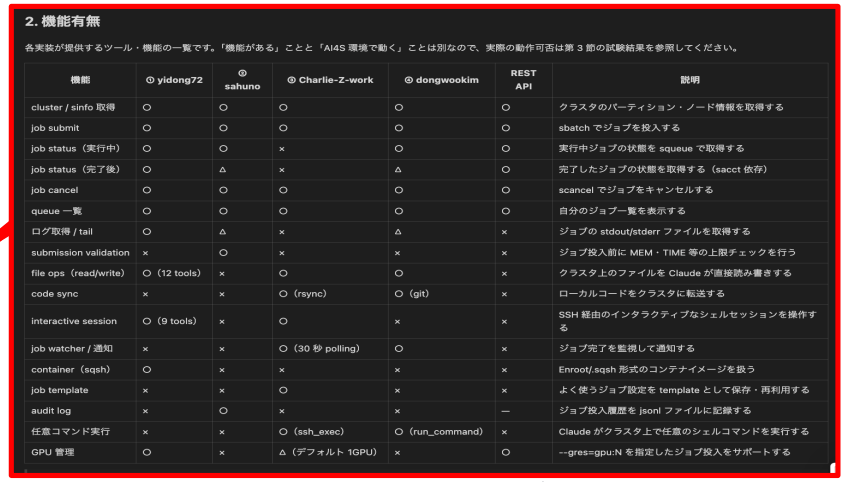
Goal:

- Slurm-MCP試験のため、小規模HPC検証環境を短期間で設計・構築・試験する。
- 4つのopen-source Slurm MCP実装を比較し、利用可能性を評価する。
- 他の業務と兼業して1週間以内に完了させる

- 試験環境の構築はクラウド(OCI)を利用
- 本番導入を考えて全てコード化

設計検討と文書作成 → OCI Compartmentの設定・構築 (Terraform) → Slurmクラスタ構築 (Ansible) → Slurmクラスタ試験 → Slurm MCP・RestAPI試験/報告書作成

AI Agent作業	具体的な成果
設計 / 構築仕様	作成文書: 要件定義、インフラ設計書 実装仕様、API仕様 構築手順、試験計画
Terraform / Ansible生成	OCI VCN/subnet/compute + Slurm/Munge/NFS/MCP/REST設定
OCI API調査	tenant/network設定の確認と改善
試験	34 MCP tools、Slurm CLI、REST job、MPI job
報告	試験仕様書、実行ログ、結果まとめ、再試験プロンプト



2. 機能有無

各実装が提供するツール・機能の一覧です。「機能がある」ことと「AI4S 環境で動く」ことは別なので、実際の動作可否は第 3 節の試験結果を参照してください。

機能	yidong72	sahuno	Charlie-Z-work	dongwookim	REST API	説明
cluster / sinfo 取得	○	○	○	○	○	クラスタのパーティション・ノード情報を取得する
job submit	○	○	○	○	○	sbatch でジョブを投入する
job status（実行中）	○	○	×	○	○	実行中ジョブの状態を squeue で取得する
job status（完了後）	○	△	×	△	○	完了したジョブの状態を取得する（sacct 依存）
job cancel	○	○	○	○	○	scancel でジョブをキャンセルする
queue 一覧	○	○	○	○	○	自分のジョブ一覧を表示する
ログ取得 / tail	○	△	×	△	×	ジョブの stdout/stderr ファイルを取得する
submission validation	×	○	×	×	×	ジョブ投入前に MEM・TIME 等の上限チェックを行う
file ops（read/write）	○（12 tools）	×	○	○	×	クラスタ上のファイルを Claude が直接読み書きする
code sync	×	×	○（rsync）	○（git）	×	ローカルコードをクラスタに転送する
interactive session	○（9 tools）	×	○	×	×	SSH 経由のインタラクティブなシェルセッションを操作する
job watcher / 通知	×	×	○（30 秒 polling）	○	×	ジョブ完了を監視して通知する
container（sqsh）	○	×	×	×	×	Enroot/.sqsh 形式のコンテナイメージを扱う
job template	×	×	○	×	×	よく使うジョブ設定を template として保存・再利用する
audit log	×	○	×	×	—	ジョブ投入履歴を jsonl ファイルに記録する
任意コマンド実行	×	×	○（ssh_exec）	○（run_command）	×	Claude がクラスタ上で任意のシェルコマンドを実行する
GPU 管理	○	×	△（デフォルト 1GPU）	×	○	--gres=gpu:N を指定したジョブ投入をサポートする

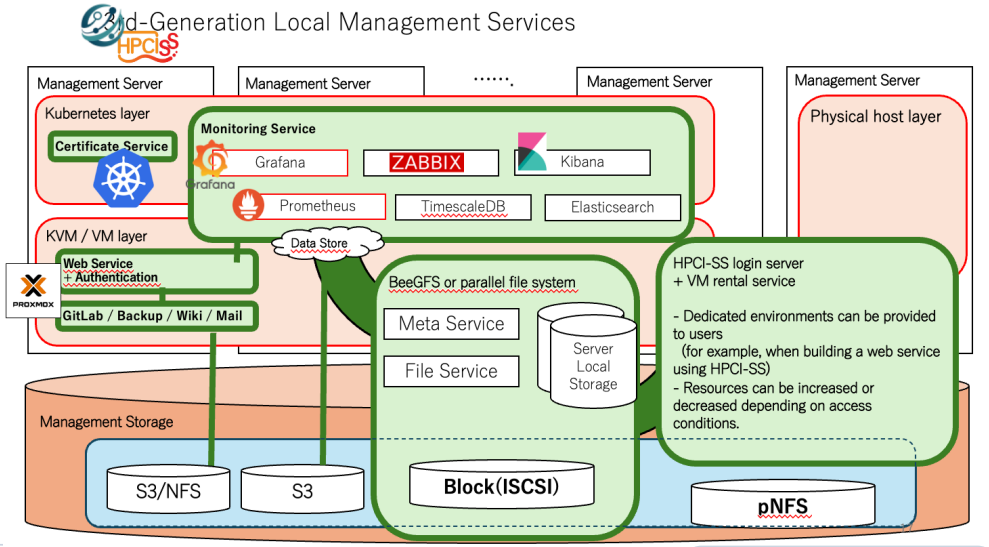
Result:

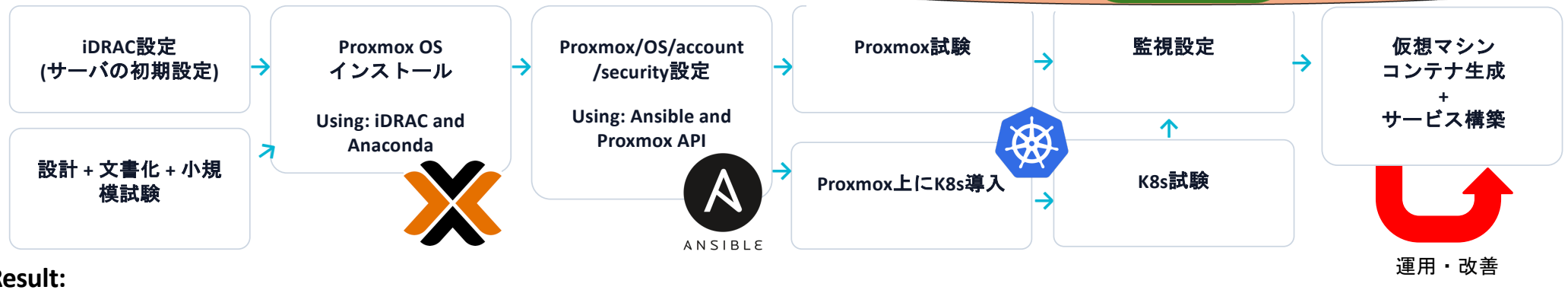
- 他作業しながら3日(設計1日、構築1日、試験1日)で完了→ このため調査範囲を拡張してREST APIについても追加調査
- 構築時のエラーなどは多々あったが、OCIコンパートメント内の環境やFirewall、インスタンスなどを丸ごと作り直せるので最初からのやり直しなどでAI Agentが思考して解決することが多かった(NFSの構築部分やSlurmの設定等の一部は構築中に助言)

活用事例2: Proxmox + Kubernetes

Goal:

- HPCI共用ストレージ向けの管理基盤を構築する。
- 仮想マシン基盤としてProxmox、コンテナ基盤としてKubernetesを導入。
- 今後の運用等でAI Agentが仮想マシンやコンテナ環境を自動で構築・設定・運用できるように、AI-agent-readyな文書を整備。





Result:

- セキュリティからAI向けのアカウント整備などは行ったものの、問題なく環境構築を実施し、試験を行い報告
- 生成したドキュメントを、追加でAI Agentに食わせると、仮想マシンやコンテナ環境およびサービスが自動構築
- サービスがAPIなどに対応していれば、AI Agentがサービス内の設定更新や調査(例えば監視環境のエラーなど)も実施
- 一方で (1) 本番運用でのアカウントの分離やセキュア化が大きな課題, (2) ドキュメント作成が甘かったのか性能面で追加調査を実施(運用技術部門内部での雑談で発覚 → ドキュメントのダブルチェック等は重要と感じました)

AIを活用しやすいHPC運用基盤とは

AI Agentを使って業務をしている感触(2026/07時点)

- 情報が十分ある基本的なシステム構築や運用作業は、AI Agentの現行モデルの性能で十分できてしまう(?)
  (ハードウェア交換は今のところ人間が必須。複合的な障害などではトンチンカンな回答はある)
- システム構築では、最初のドキュメント作成(要求仕様書/構築仕様書/構成図/環境情報/
  作業手順書/試験手順書…等々)をAI Agentと整備することがとても重要
  → 現時点ではエンジニアのスキルや能力が発揮できる場所と考えています。

- 構築・運用業務をAIに全てを委ねるならSSHでもRestでもMCPでもクラウドネイティブでも良い気がするが……
  - 権限付与の仕組みが弱い環境(例: SSHのみ)や、Adminしかないサービスだと AI Agent利用のセキュリティや制限が心配
    (AI Agentを利用するのは「危ないから禁止」という指示1つで使えなくなってしまう？)
  - このためAPI Keyなどで認可設定がより自由なCloude Nativeな環境のほうがAIとの相性がよく見えています。
    ※ ただし「Cloude Native環境自体をAI Agentで構築する際の権限は大丈夫なのか？」は堂々巡りなので
    クラウドサービス自体の運用でのAI活用事例にはとても興味があります。
  - 一方、ローカルLLMの活用を進めることで、運用時のセキュリティリスクを減らせるとは思っています。
    ※ 次のページに記載していますが、性能が低いモデルで検討する場合に、サービスや環境自体の
    セキュリティは良いのかといった疑問もあります。

AI Agentを運用に組み込む場合、
プロジェクトのSRO(Service Level Objective)やセキュリティ規定に沿った形で
AI Agentを活用しても問題ない環境を整備する必要があると考えています。

AI Agent利用への制限・批判をどうするか？ → パネルにskip

AI Serviceを利用することに対してセキュリティや完全性などの点で批判がありますが、
競争性やコスト面を考えるとAI Agentを利用しない設計・構築・運用は致命的と考えています。

(1) ClaudeやCodexのような外部サービス利用時の機密流出
- 設計などを行う際のIPアドレスやサービス設計などが流出するとセキュリティリスクになる可能性があります。
- 一方で、Claude Mutes等の最新のモデルで人間が把握していないセキュリティホールを多数発見されている状況で、それよりも性能が低いモデルでシステム設計や検討を行って、セキュリティは担保されるのでしょうか？

(2) ローカルLLM・ローカルAI Agentサービスのセキュリティと性能
- 開発や運用・構築についてもセキュアな部分へはローカルLLM/サービスの適用は必須と考えています。(例えばZabbixアラートの調査やユーザ情報を含むログの調査等)
- 一方で、グローバルなネットワークとの接続がないと最新の情報を担保できない(Deep Research)問題があり、ここでも性能とセキュリティがトレードオフになっています。
  ネットワークとつなげると、どうしても外部への情報流出リスクがでてしまうため、
  セキュリティレベルが下がってしまいます。(私はここの部分について悩んで答えが出ていません)

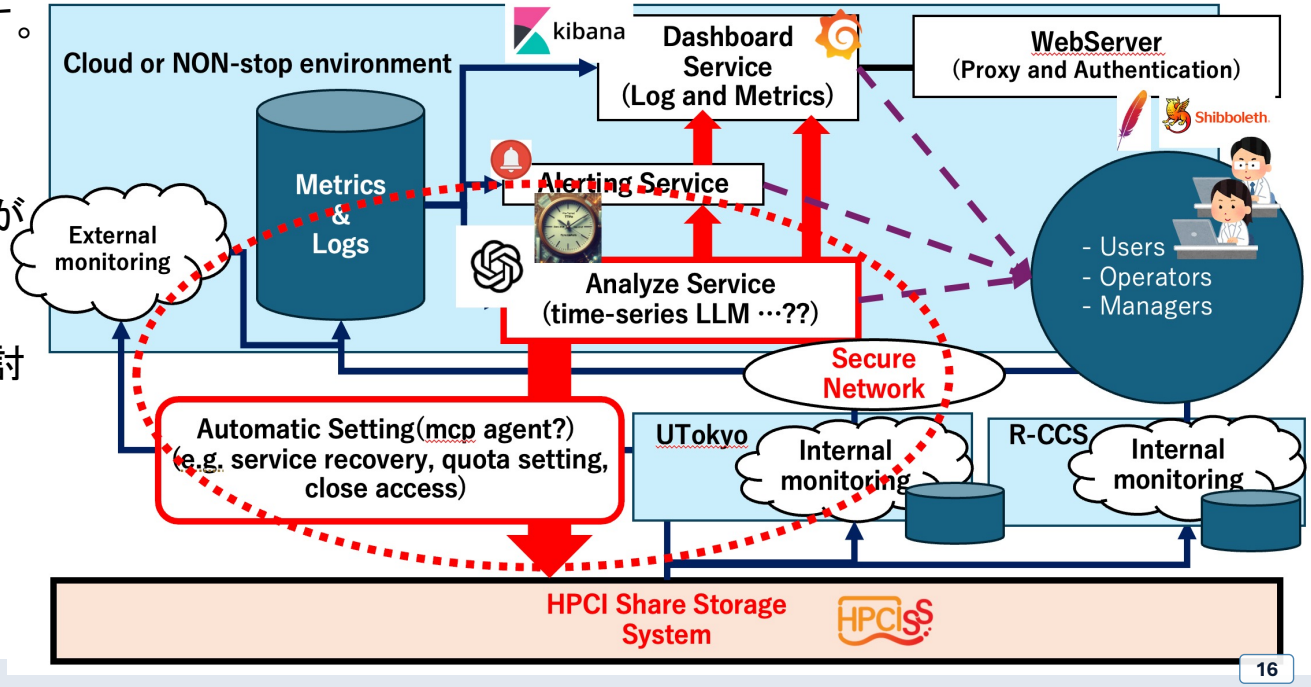
(3) 責任の所在や問題発生時の応対
- AIで何か問題が起きたときにAI Agentを利用している担当者が責められる場合「AI Agentを利用しません」以外の方法がなくなる危険性があります。
- 利用自体ができなくなると、コストの増大や競争性が不足して、PJが立ち行かなくなる可能性があります。
- なので、事前にAI Agent利用するための基盤環境の構築は大変重要と考えています。

時系列LLMによる運用支援

障害分析などを目的に時系列LLMの導入を整備中です。(AI Agentを用いて分析環境の構築を検討中)

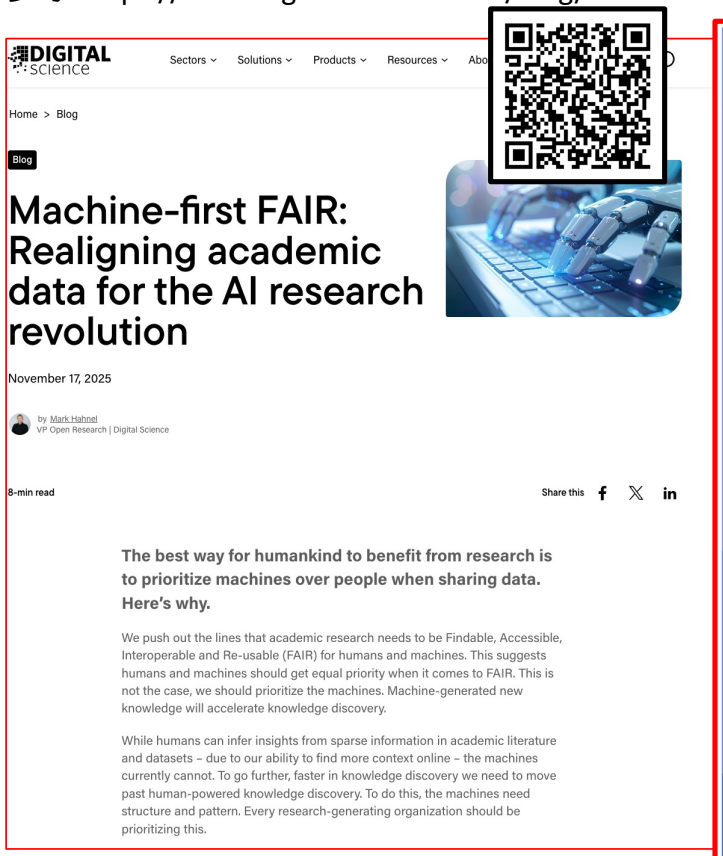
時系列LLM:
　時系列分析向けに設計されたモデルであり、ビッグデータ分析や大規模データ処理で利用されている。
　例：IBM Granite - https://huggingface.co/ibm-granite/granite-timeseries-ttm-r1

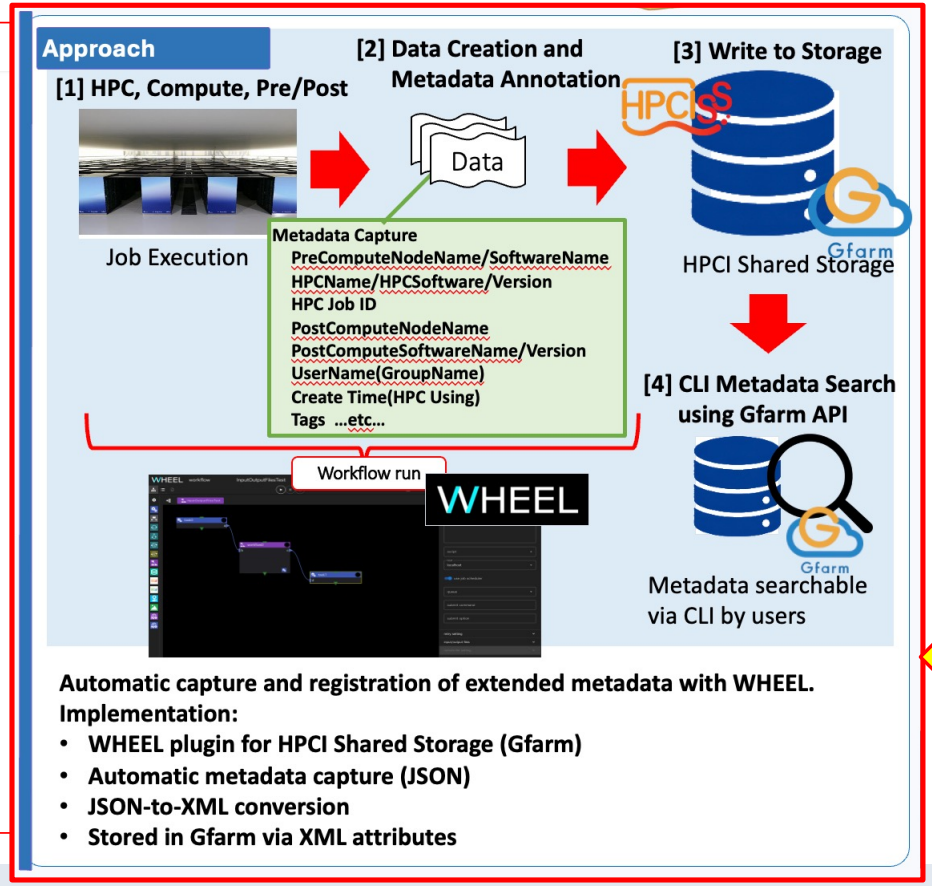
- モニタリング情報を活用した自動分析環境の整備に時系列LLMの導入を考えています。
- 障害予兆や気がついていない改善箇所の発見、ユーザの利用予測等への活用を考えています。
- 容量予測などでは活用が見込まれますが障害予兆などでは、メトリクス情報の収集精度の改善などを行う必要がある事がわかり、現在調整や環境整備を検討しています。

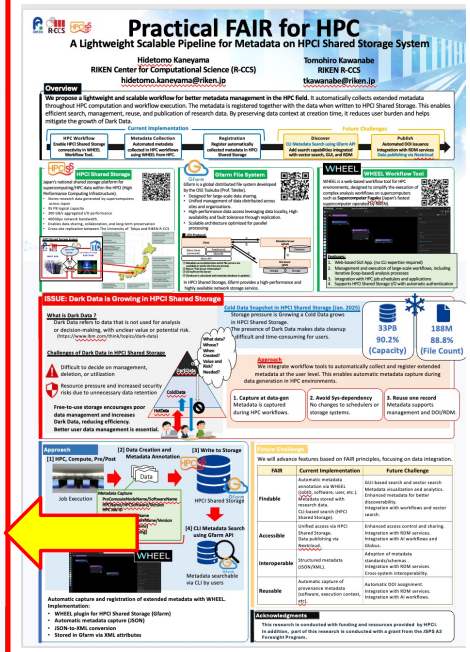


データ公開環境の整備および研究データへのメタデータ自動付与

データの取扱いは”人”ではなく”AI Agent”からの利用に変化していくと思っています。
また保存データの公示性を高めるためにもOpen Scienceと連携するためのストレージを目指します。
参考: https://www.digital-science.com/blog/machine-first-fair-academic-data-for-the-ai-research-revolution/







Force11 poster 2026/06

まとめ　+デモ確認

- 競争性やコスト減を考えるとシステムの設計・構築・運用ではAI Agentは必須と考えています。
- 利用時のセキュリティとの兼ね合いには今後も苦労しそう
  - コマーシャルなサービスの性能や機能が良いので今後も活用したい
  - 一方でローカルなLLMを活用する場面も増えているので内部導入がどんどん増えていく
- AI利用のベストプラクティスを捉えきれていない(変化が早い…)
  - 少なくとも現在はドキュメントの作成に力を入れることでうまくいくことが多いです。
    – ですので、いろんな方の最新の使い方を参考にしたい、情報を共有したい
- データの取扱いはAI Agentに置き換わると思っています。
  - このためストレージのトレンドも変わっている
  - 今まで以上に早いサイクルでの対応が必要

HPCI Shared Storage

Gfarm

Thank you

[1] AIによってHPCは、今後どのように進化し恩恵を与えるか

1 改善サイクル

改善サイクルが一層早くなることが考えられます。
このためにはシステム設計時点でアップデートや機能追加をサービス停止せずに行える環境は必須に思います。

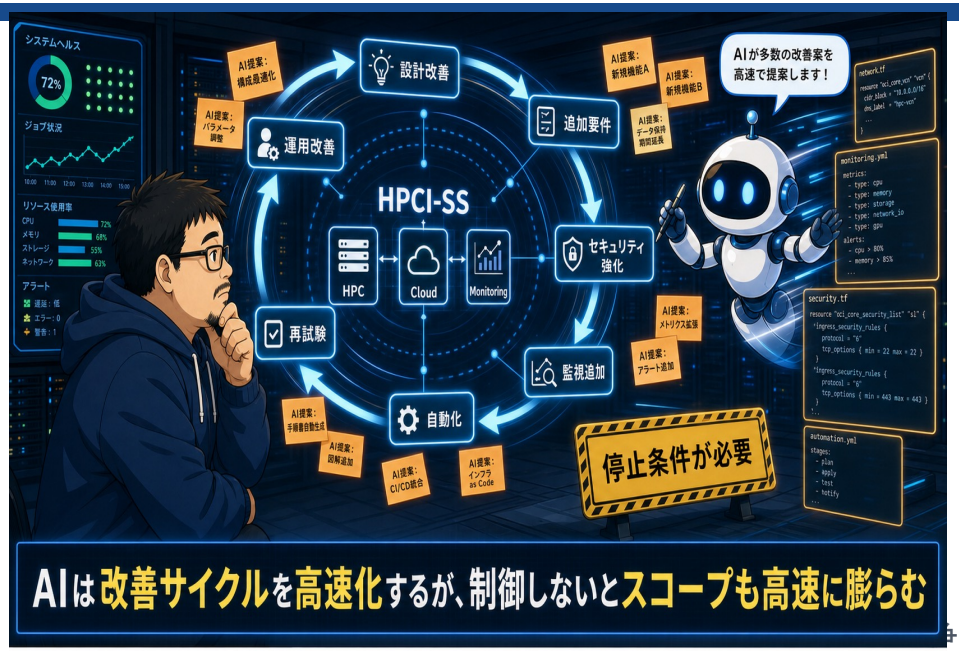
2 利用者の門扉を広げる

既にOpenOndemandやk8sなどで、CLIやジョブスケジューラの壁は低くなっていますが、AI Agentによりユーザアクセスがより容易になると考えています。

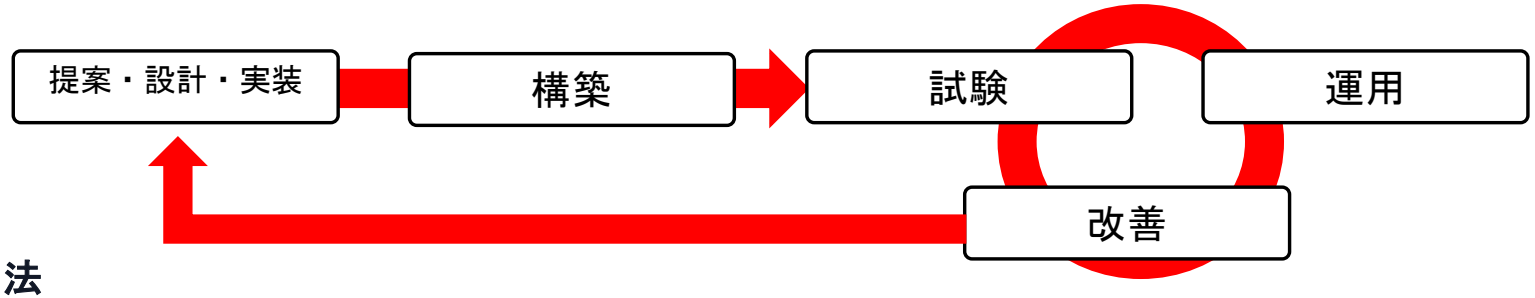
3 個性の出す場所は？

AI推論の要望に注視するとHPCの個性は？

ストレージとしては
・GPUの性能のボトルネックにならないような性能を確保し続ける
・AI Agentアクセス向けの環境整備を行い続ける
ことになる気はしているのですが....



[2] AI台頭に伴う技術的な課題



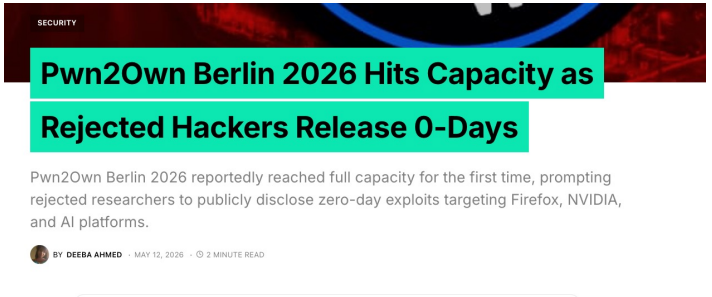
1 攻撃側のAI利用を防ぐ方法

脆弱性発見や攻撃の速度は人間の監視範囲を超えています。防御側でも高性能なAI活用が必要なのは明白ですし、
定期的なAIによるセキュリティチェックを要するなど攻撃から守るための運用手法が必要に思います。

https://hackread.com/pwn2own-berlin-2026-hits-capacity-hackers-0-days/

2 AI Agentリスクは設計から

情報漏洩や誤操作を理由にAI Agentの利用を禁止すると、競争性や開発速度、コスト面で立ち行かなくなります。
AIを常に活用するための基盤設計を行う必要があると考えています。
現時点では、特にパーミッション(権限)や認可部分を改善していく必要があると思っています。


SECURITY

Pwn2Own Berlin 2026 Hits Capacity as Rejected Hackers Release 0-Days

Pwn2Own Berlin 2026 reportedly reached full capacity for the first time, prompting rejected researchers to publicly disclose zero-day exploits targeting Firefox, NVIDIA, and AI platforms.

BY DEEBA AHMED · MAY 12, 2026 · 2 MINUTE READ

3 ハードウェア導入サイクルへの懸念

AIストレージの流行りや要求のサイクルがとても早いです。
このため必要なアプリケーション機能やAIへのアプローチをハードウェアに依存しすぎると、現行の5年以上単一のハードウェアを利用するという導入サイクルでは長すぎるかもしれません。

[3] AI時代の新たな制度・社会・倫理的課題

1 AI活用格差を埋められない…

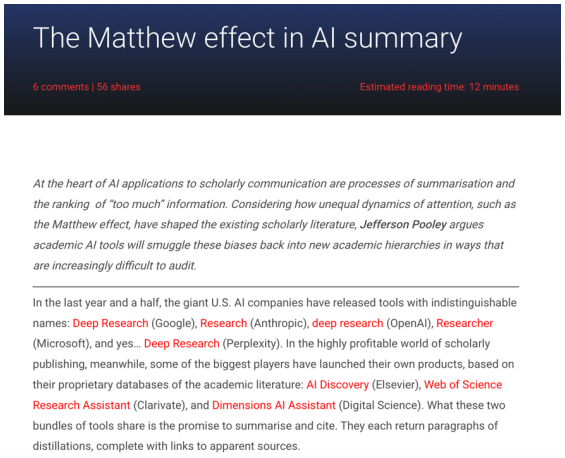
エンジニアリングでは、AIを利用できる人(会社が許可している稼働かも含めて)と利用できない/しない人で、業務効率に差が出ている印象。

2 人間の学習はむしろ重要

AIにない問いや設計判断、アイデアを出すには、人間側の知識や経験が重要と思っています。

一方後輩の育成や成長は、初心の仕事がAIで済んでしまうため実務をしながら学ぶことが難しくなっている印象が…

https://blogs.lse.ac.uk/impactofsocialsciences/2026/05/19/the-matthew-effect-in-ai-summary/



The Matthew effect in AI summary

6 comments | 56 shares Estimated reading time: 12 minutes

*At the heart of AI applications to scholarly communication are processes of summarisation and the ranking of "too much" information. Considering how unequal dynamics of attention, such as the Matthew effect, have shaped the existing scholarly literature, **Jefferson Pooley** argues academic AI tools will smuggle these biases back into new academic hierarchies in ways that are increasingly difficult to audit.*

In the last year and a half, the giant U.S. AI companies have released tools with indistinguishable names: Deep Research (Google), Research (Anthropic), deep research (OpenAI), Researcher (Microsoft), and yes… Deep Research (Perplexity). In the highly profitable world of scholarly publishing, meanwhile, some of the biggest players have launched their own products, based on their proprietary databases of the academic literature: AI Discovery (Elsevier), Web of Science Research Assistant (Clarivate), and Dimensions AI Assistant (Digital Science). What these two bundles of tools share is the promise to summarise and cite. They each return paragraphs of distillations, complete with links to apparent sources.

http://finance.biggo.jp/news/202604130032_Linux-Kernel-AI-Code-Rules-Responsibility

Finance ニュース、株式、企業を検索 Agent に質問 リアルタイム 市場 決算説明会 イベントカレンダー

Linux Kernel が AI 生成コードの利用を許可、ただし開発者が法的・技術的な全責任を負う

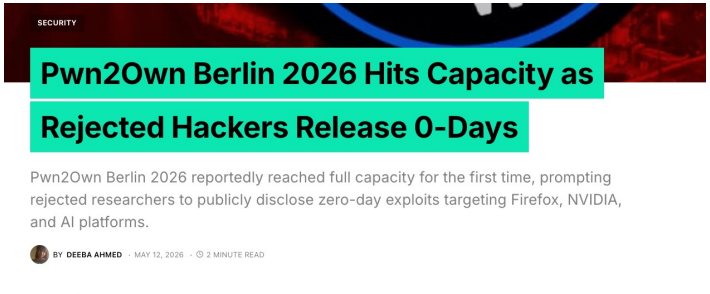
2026-04-13 09:33 (GMT+9) Google の優先する情報源に追加

ソフトウェア開発への人工知能の統合は、オープンソースコミュニティ全体で激しい議論を巻き起こしてきました。一部のプロジェクトが法的および品質上の懸念を理由に、AI 生成コードを完全に禁止する一方で、他のプロジェクトはその可能性を慎重に探っています。世界中の数十億台のデバイスで動作する基盤ソフトウェアである Linux kernel は、今回、決定論的かつ現実的なスタンスをとりました。新しく公開された方針文書において、プロジェクトのリーダーシップは、AI 支援による寄稿を許可するものの、コードを提出する人間の開発者に究極の責任を課すという厳格な枠組みの下でこれを認めることを明らかにしました。Linus Torvalds 氏自身が推進したこの動きは、カーネルの伝説的な安定性、セキュリティ、および法規制の遵守基準を損なうことなく、最新ツールの効率性を活用することを目的としています。

新方針：AI アシスタントへの条件付き歓迎

Linux kernel プロジェクトは、AI 支援による寄稿に関するルールを、公式リポジトリにホストされている

https://hackread.com/pwn2own-berlin-2026-hits-capacity-hackers-0-days/



SECURITY

Pwn2Own Berlin 2026 Hits Capacity as Rejected Hackers Release 0-Days

Pwn2Own Berlin 2026 reportedly reached full capacity for the first time, prompting rejected researchers to publicly disclose zero-day exploits targeting Firefox, NVIDIA, and AI platforms.

BY DEEBA AHMED · MAY 12, 2026 · 2 MINUTE READ

[4] 「AIによるHPC」が切り拓くべき未来

1 研究成果を早く使える形へ

研究成果などの汎用化がより用意になり、社会への貢献が早くなる可能性。
例えば、私の業務では解析、可視化、運用ツールといった論文などで発表されている機能やアルゴリズムを、AI Agentを活用して汎用化やシステム運用に独自に組み込むことが低コストになるのではないかと期待しています。

2 データへの責任

多くの人がAIを介して情報にアクセスする世界なので、
よりデータへのアクセス性や、メタデータ等を利用した検索性が必要になると思っています。
このためデータを管理するシステムを運用する身としては、
よりデータを利活用できる環境整備への責任を感じています。

https://www.nature.com/articles/s41586-025-09922-y

Article | Published: 14 January 2026

Artificial intelligence tools expand scientists' impact but contract science's focus

Qianyue Hao, Fengli Xu, Yong Li & James Evans

Nature **649**, 1237–1243 (2026) | Cite this article

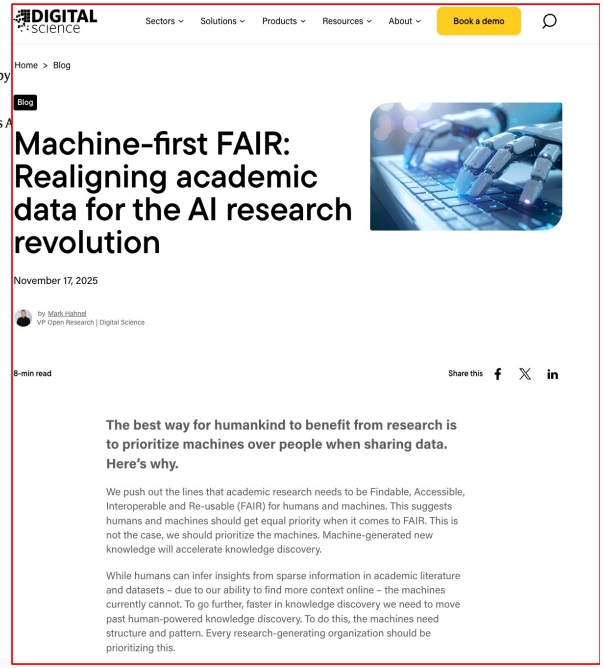
Save article

53k Accesses | **81** Citations | **1183** Altmetric | Metrics

Abstract

Developments in artificial intelligence (AI) have accelerated scientific discovery[1]. Alongside recent AI-oriented Nobel prizes[2,3,4,5,6,7,8,9], these trends establish the role of AI tools in science[10]. This advancement raises questions about the influence of AI tools on scientists and science as a whole, and highlights a potential conflict between individual and collective benefits[11]. To evaluate these questions, we used a pretrained language model to identify AI-augmented research, with an F1-score of 0.875 in validation against expert-labelled data. Using a dataset of 41.3 million research papers across the natural sciences and covering distinct eras of AI, here we show an accelerated adoption of AI tools among scientists and consistent professional advantages associated with AI usage, but a collective narrowing of scientific focus. Scientists who engage in AI-augmented research publish 3.02 times more papers, receive 4.84 times more citations and become research project leaders 1.37 years earlier than those who do not. By contrast, AI adoption shrinks the collective volume of scientific topics studied by 4.63% and decreases scientists' engagement with one another by 22%. By consequence, adoption of AI in science presents what seems to be a paradox: an expansion of individual scientists' impact but a contraction in collective science's reach, as A augmented work moves collectively towards areas richest in data. With reduced follow-on engagement, AI tools seem to automate established fields rather than explore new ones, highlighting a tension between personal advancement and collective scientific progress.

https://www.digital-science.com/blog/machine-first-fair-academic-data-for-the-ai-research-revolution/



26