



技術学術WEEK 2026 Summer

「計算と自然」集会

第29回 LLM呼び出しの自動化のためのペトリネットワークフローと
プライバシーレベルを考慮した知識共有の枠組み

計算と自然

Out[$\ast$]=



計算と自然('25)

放送大学大学院文化科学研究科（修士課程）

情報学プログラム

BSラジオ531ch

2025/10/04~2026/01/09 土曜 11:00~11:45

同一内容を前・後期に提供

■ 放送大学OCWでストーリーミング配信中

<https://v.ouj.ac.jp/view/ouj/#/navi/vod?ca=30757>

放送大学 ログイン

動画検索

動画

02 ラジオ > DL OCW > 01 計算と自然 (25) > 045 計算と自然 (25) 09710566

ダウンロード

第01回 計算とは？

本科目のテーマである自然計算に基づいて、デストの最も重要な概念である。計算と自然の関係を、さらに、計算、演算、計算能力、計算能力の限界について解説する。

02 ラジオ > DL OCW > 01 計算と自然 (25) > 045 計算と自然 (25) 09710566

第02回 ライフゲームと自然計算

ライフゲームと自然計算の関係をより深く理解する。セルオートマトンや計算能力の限界について、計算と自然の関係をより深く理解する。本学は、計算と自然の関係を、さらに、計算、演算、計算能力、計算能力の限界について解説する。

02 ラジオ > DL OCW > 01 計算と自然 (25) > 045 計算と自然 (25) 09710566

第03回 チューリング機械とオートマトン

チューリング機械と自然のオートマトンについて解説する。計算能力の限界、本学は、計算と自然の関係を、さらに、計算、演算、計算能力、計算能力の限界について解説する。

02 ラジオ > DL OCW > 01 計算と自然 (25) > 045 計算と自然 (25) 09710566

第04回 可逆計算

可逆計算の重要性から解説された可逆計算の重要性は、さらに、計算能力の限界、本学は、計算と自然の関係を、さらに、計算、演算、計算能力、計算能力の限界について解説する。

02 ラジオ > DL OCW > 01 計算と自然 (25) > 045 計算と自然 (25) 09710566

「計算と自然」集会の過去動画

Out[$\ast$]=

 WOLFRAM NOTEBOOK

Make Your Own Copy Download Share Info Sign In

計算と自然

放送大学大学院 計算と自然('25)

- シラバス
- ストリーミング配信
- IAシンポジウム会誌: 放送大学「計算と自然'25」を分担したことについて

VRChat 「計算と自然」集会

「計算と自然」集会

- 第0回 キックオフ (2025/04/05)
- 第1回 計算とは (2025/04/19)
- 第2回 ライフゲームと自然計算 (2025/05/03)
- 第3回 チューリング機械とオートマトン (2025/05/17)
- 第4回 可逆計算 (2025/06/07)
- 第5回 セルオートマトン (2025/06/21)
- 第6回 マルチセット (2025/07/05)
- 第7回 反応拡散による計算 (2025/07/19)
- 第8回 生物に触発された計算 (2025/08/02)
- 第9回 番外編 キノコ計算 (2025/08/16)

Posts

過去の集会のYoutube録画へのリンク

🕒 10/13/2025 at 12:04 🗨️ Group

「計算と自然」集会の過去のYoutube録画へのリンクをまとめたページを作りました。
(Groupページに表示されるリンクの一番左をクリック)



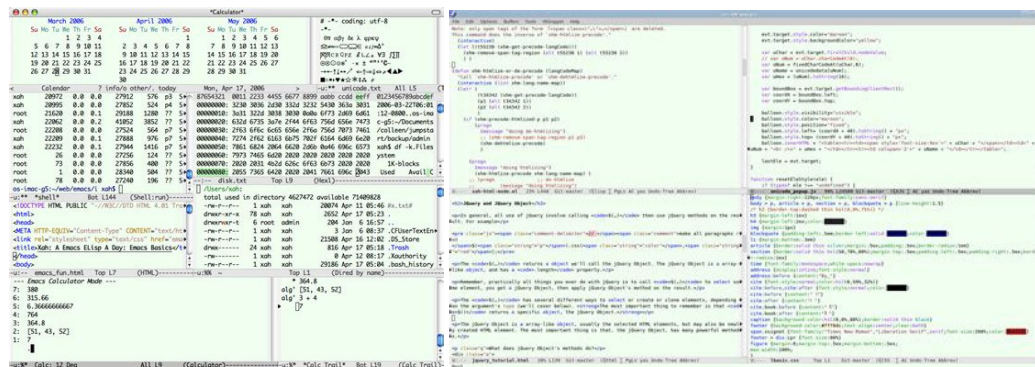
Emacsおじさん

定義 = 日常作業をすべてEmacsエディタとEmacs Lispを使ってこなす人

■ MathematicaはLisp環境の一種なので、

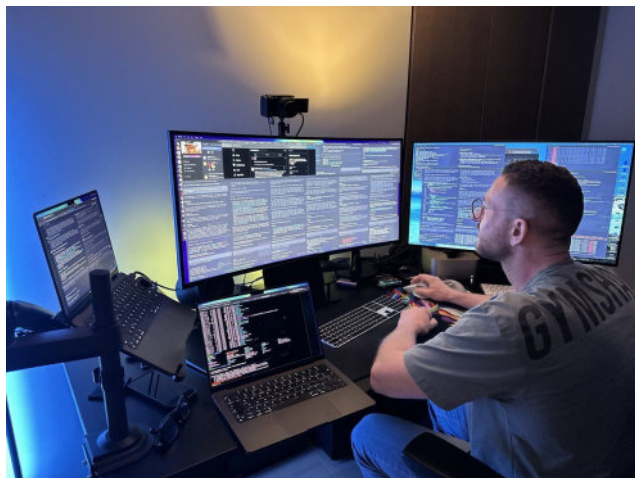
■ Mathematicaで暮らす = 現代版Emacsおじさんの一変種

[重要] Emacsおじさんは、プログラミングはEmacs Lispではなく主にCを使うが、Mathematicaは、なんでも単一環境、単一言語で済ます「高級言語」指向の延長線上にある



https://xahlee.info/emacs/emacs_fun.html

Emacsおじさん、再び！



© Peter Steinberger

- 未来はもっと「彩り」のある世界だと思ってました...

“Lispユーザの数だけLisp方言がある”

- Lispを選択する意味:
 - 「使っているエディタ自体を自由にカスタマイズしたいと思うことはありませんか？」
 - 「言語処理系自体を変更したくなることはありませんか？」

“Lispユーザの数だけLisp方言がある”

- Lispを選択する意味:
 - 「使っているエディタ自体を自由にカスタマイズしたいと思うことはありませんか？」
 - 「言語処理系自体を変更したくなることはありませんか？」
- とすれば...
 - 「Claude Codeを自由にカスタマイズしたいと思うことはありませんか？」
- そこへ
 - 3/31のClaude Codeのソース流出事故とClaw Codeの実装 (2時間)
 - 反復ループ型エージェントの実装が共有知識になった
 - これはなんとかせねばならん！

自前で準備したLLM呼び出しフレームワークは...

- 性能が出ない
 - 反復ループ型エージェントの実装が共有知識になったのは事実だが...
- UIが洗練されておらず、レスポンスが悪い (遅い!)
 - 「Illustratorを書いて」と頼んでもAdobe Illustratorのベジェツールのタッチが再現できるわけではない
 - 「打てば響くUI」でなければ使い物にならない！

UIコードの免罪符

- ところが...

「つくづく思うのだけど、UIのコードってのは高速に動作しなければならないと思っていたわけだけど、ChatAIのせいで、数十秒遅延してもよい、さらに、数十分遅延してもよい、っていう免罪符ができたってことだよな 😊」

みんな我慢強くなった。

UIコードの免罪符

- ところが...

「つくづく思うのだけど、UIのコードってのは高速に動作しなければならないと思っていたわけだけど、ChatAIのせいで、数十秒遅延してもよい、さらに、数十分遅延してもよい、っていう免罪符ができたってことだよな 😊」

⇒ 理想を追求してよい！

さて、LLMと相談したいことなんだけど...

- 「わたしは今日は何をすればいいのですか？」
 - 複雑系研究者としてこれまで遭遇した最も複雑な系は「わたし」自身なので、Mathematicaでメールやスケジュールを分析しているわけです。
- そもそも「今日の予定は？」と訊ねても、適切な答えすら返ってはいない。
 - Mathematicaで“複雑系”の分析/可視化コードを提案してもらえばOK

コードの生成と実行についての考え方

- LLMにプロンプトに従ってMathematica式を提案させる
 - `$packageDirectory`にあるコードファイルにはすべて `api` を記述したマークダウンファイルが存在する。
 - さらにmcp経由で参照して式を生成
- 提案式をそのまま信用しない
 - 式に実行に不安要素のある関数が含まれていれば、ユーザの意思決定を待つ
 - 式に実行禁止関数が含まれていれば実行を停止する

プライバシーレベルの考え方

- LLMに渡すデータにはすべて $[0,1]$ のプライバシーレベル値がついている
- クラウドLLMは $PL \geq 0.5$ のデータは参照できない(**スキーマ情報のみ**参照できる)
 - 計算式は提案できるがデータの詳細は分からない
- ローカルLLMは任意のデータを参照できる
 - LLMの出力データのプライバシーレベル値 = 入力データのプライバシーレベルの最大値
 - 公開情報のみからローカルLLMが計算した結果はクラウドLLMと共有可

例: 今日の予定 (プライバシーレベル1.0)

The screenshot shows a Wolfram Notebook titled "私用予定表-1" - Wolfram. The main content area displays a calendar for today's tasks, categorized by date and time. The tasks are listed in a structured format with checkboxes and links.

今日の予定

▲ 機密 - PrivacyLevel 0.5 以上のメールを含みます (クラウド送信不可)

07/18 (土)

- 【✓切】 計算と自然 29
- 【レビュー】 SR-new
- 終日 オープンキャンパス
- 22:30-23:00 計算と自然集会 Vket技術学術WEEK

07/20 (月)

- 【✓切】 【情報論文誌: 依頼】 (26-T001.R1) 投稿論文の第2回目査読
- 【✓切】 【情報論文誌: 確認】 投稿論文 26-T001.R1 の査読 (期限: 20-Jul-2026)
- 【レビュー】 出退勤簿

07/21 (火)

- 【✓切】 情報工学演習I Slides
- 【レビュー】 体調
- 13:10-14:40 情報工学演習I

▲ 期限超過・未処理

- 【✓切】 VRとメタバース
- 【✓切】 教務委員会
- 【レビュー】 教務作業
- 【✓切】 IPSJ-review
- 【✓切】 卒業研究
- 【✓切】 データ構造とアルゴリズム第14回 Slides
- 【レビュー】 プリントスルチェック

📧 要対応メール (31)

【依頼】 画像処理の履修登録の講義コードの間違い - Takashi Ozeki <ozeki@fukuyama-u.ac.jp> (07/16 受信・2日前)
(依頼)画像処理履修登録講義コードの誤り訂正依頼(尾関氏・今井准教授・野田学生)

📄 07/18 Do: 資格試験会場で上式準備中の手続と日報 (7/16 1:10) - TAKASHI OZeki <ozeki@fukuyama-u.ac.jp> (07/16 09:07・2日前) 2025.07/07

The sidebar on the right contains several toolbars: "Claude Code" with options like "機密マーク", "機密解除", "スキャン", "ファイバー", "Save NRE"; "Claude" with "Chat Cell", "Feed", "選択Feed", "仕様生成", "仕様変更", "ワークフロー一覧", "プロンプト一覧"; "API" with "ClaudeCode", "M: Opus 4.8", "エフォート: Medium", "Paid API: Denied", "api.model: 多動更新", "ルート自動(リ-カリ)", "MCP 実行中"; "API" with "API停止", "API停止"; "セッション" with "セッション", "終了"; and "全範囲削除".

例: 今日の予定 (プライバシーレベル0.4)

Wolfram interface showing a SourceVaultRoutineAgendaView for PrivacySpec with a 0.4 access level. The view displays a list of events for 07/18 (土) and 07/21 (火).

07/18 (土)

- 【予定】 計算と自然 29**
- 終日 オープンキャンパス
- 22:30-23:00 計算と自然集会 Vket技術学術WEEK

07/21 (火)

- 13:30-14:40 (予定あり)

⚠ 期限超過・未処理

- 【レビュー】 プリント/スルチェック**

Out[2]=

要対応メール (5)

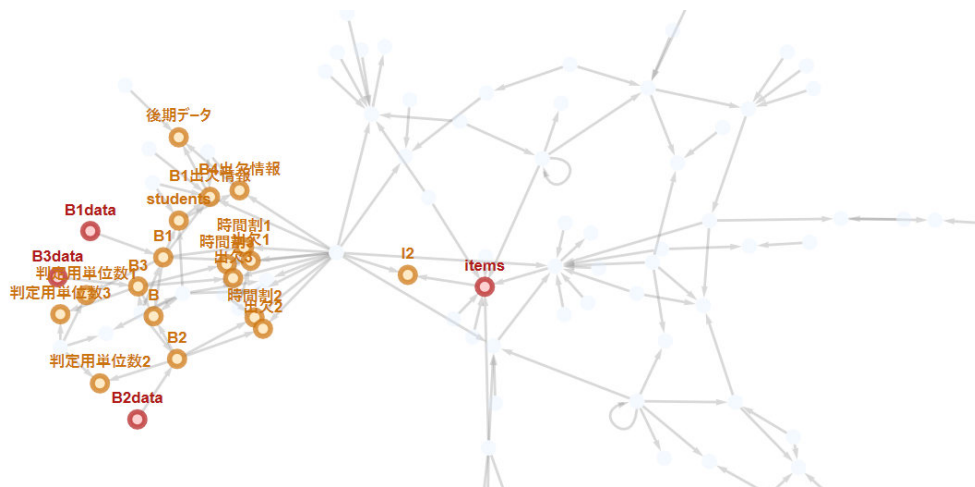
- 【依頼】** 令和9年度非常勤講師等承認書の提出について (依頼) - 大迫 博康 - ohsaoku@fukuyama-u.ac.jp (07/06 受信・12日前) **🔗🔗07/31**
「(依頼)令和9年度非常勤講師等承認書の提出依頼および経費前渡方針の周知(福山大学教務課・各教職員)」
- 【出席】** [8/28開催]なぜ人は見落とすのか? | ボカリスを科学する | Tobii User Meeting 2026 - 藤井久仁子 (Tobii アイトラッキングのグローバルリーダー) <sales.jp@tobii.com> (06/26 受信・10日前)
「(情報)Tobii User Meeting 2026参加案内 (成カ)対面・アイトラッキング講演 (Tobii株式会社)」
- 【依頼】** Re: A proposal - SUKANTA DAS - sukanta.its@faculty.iilests.ac.in (06/20 受信・28日前) **🔗🔗03/31**
「(依頼)書籍発刊原稿の提出状況の確認と書後(Sukanta Das)」
- 【出席】** [7/27日開催 オンラインイベント] AI時代の学びの課題と可能性 ~ Wolfram AI AssistantとMathematica 15で体験する活用法 ~
- ウルフラム リサーチ アジア リミテッド - event-japan@mail.wolfram.com (06/18 受信・10日前)
「(情報)Wolfram AI AssistantとMathematica 15を活用したオンラインイベント開催案内(ウルフラム リサーチ アジア)」
- 【確認】** Katsunobu, did this researcher author this publication? - ResearchGate <no-reply@researchgate.net> (06/16 受信・31日前)
「(確認)ResearchGateが論文の著者であることを確認するよう依頼(ResearchGate)」

Right sidebar shows a Claude Code interface with various tool icons: Claude Code, Chat Cell, Eval, Notebook, MCP, etc.

クラウドAIへ出せる情報と出せない情報を分離

- 変数や出力の計算による依存関係グラフを更新する
- 秘密変数と、秘密変数に依存したデータにプライバシーレベルを継承

```
In[*]:= NBPlotDependencyGraph[EvaluationNotebook[]]
```



秘密変数の追跡機構

In[]:= ClaudeEval["成績.xlsx をインポートして、データセットとして、秘密変数 socres に代入せよ"]

CLAUDE.md のファイルパスルールと confidential-data-handling スキルに従ってコードを生成します。

```
In[ ]:= socres = Confidential[
  First @ Import[
    FileNameJoin[{Quiet @ Check[NotebookDirectory[], $packageDirectory], "成績.xlsx"}],
    {"Dataset"}
  ]
]
```

Out[]:=

学生	中間成績	期末成績
学生1	78.0	70.0
学生2	55.0	60.0
学生3	90.0	80.0
学生4	85.0	88.0
学生5	62.0	59.0
学生6	30.0	40.0
学生7	95.0	90.0
学生8	70.0	73.0
学生9	60.0	65.0
学生10	75.0	75.0

完了。コードを実行して確認し、必要なら ContinueEval[] で継続できます。

秘密変数の追跡機構

```
In[ ]:= Mean[socres[All, "中間成績"]]
```

```
Out[ ]:=
```

70.

```
In[ ]:= ClaudeQuery["socresの値をリストアップしてほしい"]
```

```
Out[ ]:= `socres` は `Confidential[...]` でラップされた秘密変数として定義されており、私にはその値が提供されていません。
```

秘密変数の値は意図的に非表示になっているため、列挙することができません。

ノートブック上で直接確認するには、以下のようにしてください。

```
... mathematica
Normal[socres]
...
```

または、秘密指定を外した一時変数に取り出す場合は、

```
... mathematica
socresPublic = First @ Import[
  FileNameJoin[{Quiet @ Check[NotebookDirectory[], $packageDirectory], "成績.xlsx"}],
  {"Dataset"}
]
...
```

を実行してください。

基盤となる4パッケージ

SourceVault は単独では動かない。**機密・LLM・実行・統括**を担う4つの土台の上に乗り — 依存は常に**一方向**。

■ NBAccess — 機密データと安全の番人

セル単位の読み書き / PrivacyLevel 0-1 でアクセス制御 / 変数依存で機密を伝播 / 鍵隔離 backend。「機密の実体は NBAccess だけが扱う」

■ claudecode — LLM ブリッジ / スケジューラ

Claude Code CLI・API・ローカル LLM を統合。ClaudeAttach でファイル防壁、LLMGraph DAG で非同期実行

■ ClaudeRuntime — 単一エージェント実行核

「提案→検証→実行」のループ状態機械。LLM が出す式を NBAccess で安全検証してから評価 (1 ターン)

■ ClaudeOrchestrator — マルチエージェント統括

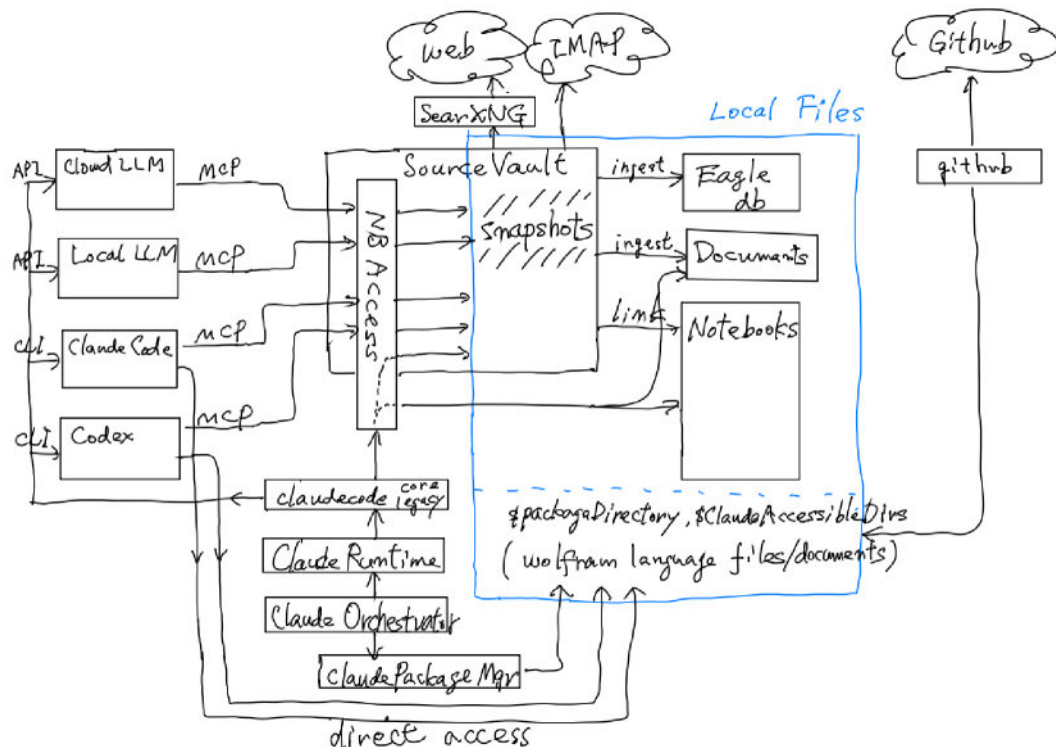
タスク分解→並列 worker→single committer。ベトリネット (Workflow) で表現・追跡。複数 Runtime を束ねる

これら4つの上に **SourceVault** (外部情報境界・版管理・根拠) が乗る ▶ 次スライド ①②

パッケージ関係図 (上位 → 基盤の一方方向依存)

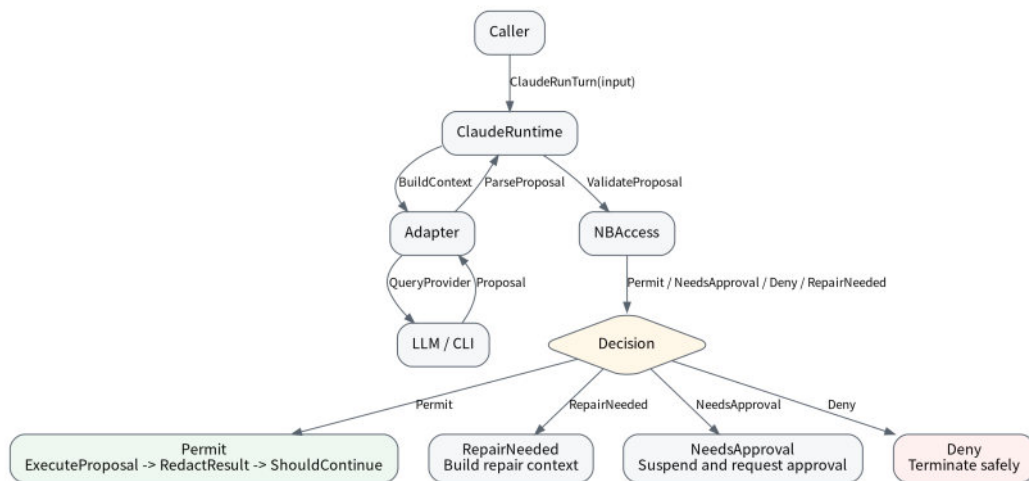


NBAccess/claudecode/SourceVaultの概要



ClaudeRuntime

- Mathematica式の提案と安全な実行 (Claw Codeからアイデアを拝借!)



SourceVault の URI (sv://...)

- SourceVaultではsvで始まるURIを使ってオブジェクトを一意に指定する

```
In[*]:= SourceVaultObjectData["sv://object/eagle-M09Z14K16JPSY"]
```

```
Out[*]=
```



- SourceVaultをLLM間の共有ストアとして使う
 - 仕様生成: \$ClaudeModelに要件を満たす仕様を策定させ、 \$ClaudeAdvisaryModel にレビューさせるループ。

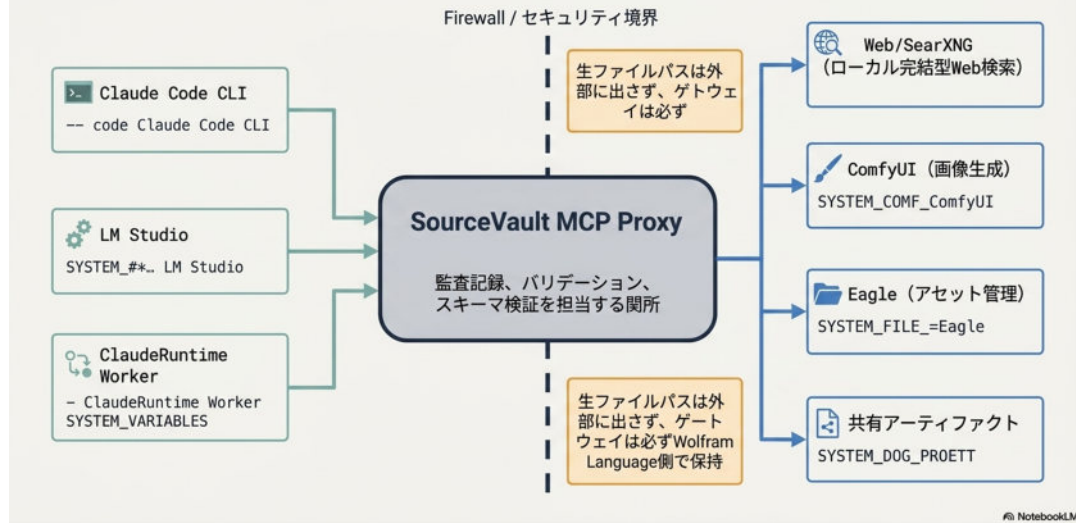
```
In[*]:= {$ClaudeModel, $ClaudeAdvisaryModel}
```

```
Out[*]=
```

```
{{claudecode, claude-opus-4-8}, {chatgptcodex, Automatic}}
```

SourceVault MCP Proxy

図9: MCP Proxyを中心としたハブ・アンド・スポーク型ネットワークとセキュリティ境界



SourceVault_promptrouter

- プロンプトを投入する前に以下の3種類に分類する
 - ヘビーなLLMを再実行する必要があるプロンプト (これは仕方なし)
 - ライトなLLMでパラメータを変更するだけで再実行可能なプロンプト
 - パターンマッチング程度でパラメータを設定して直接関数を呼び出せばよいプロンプト

SourceVaultではLLMを {anthropic, heavy}, {openai, light}などと表現している

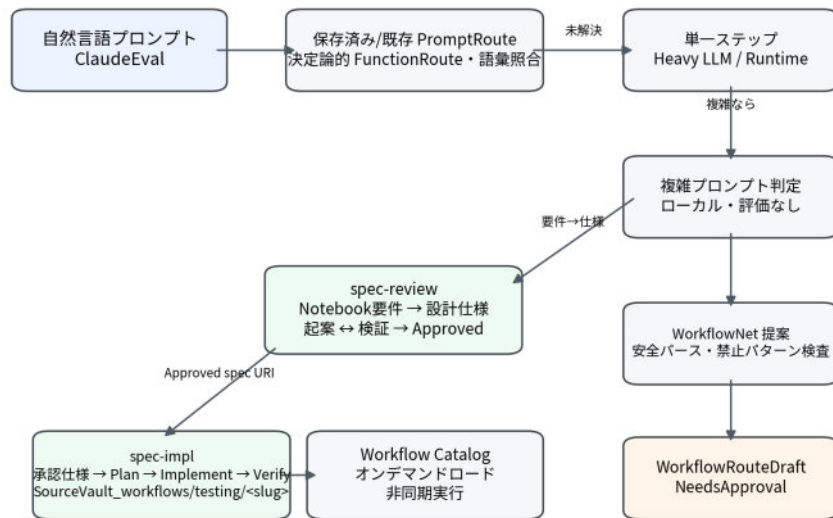
```
In[*]:= ClaudeResolveModel["anthropic", "heavy"]
```

```
Out[*]=
```

```
<|Kind → Model, Provider → anthropic, Intent → heavy, ModelId → claude-opus-4-8,  
Availability → Available, Class → Heavy-Cloud, Capabilities → {Reasoning, Code, ToolUse},  
Freshness → Fresh, PolicySource → seed:model-seed, ResolvedFrom → seed|>
```


SourceVault_promptrouter

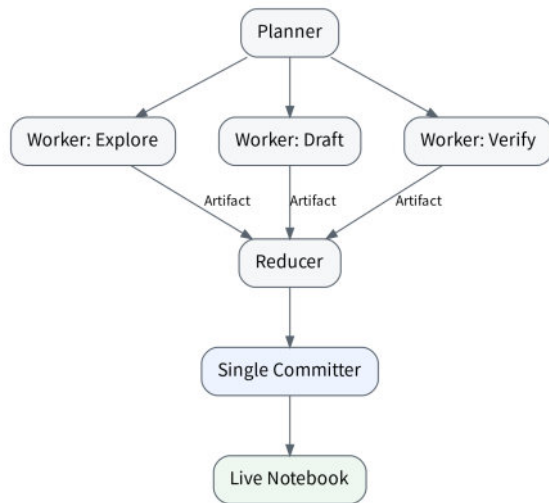
図7 ClaudeEval・PromptRouter・PromptWorkflow・仕様実装ワークフローの関係



既存ルートを優先し、新規Workflowは提案・検証後もNeedsApprovalで停止する。spec-review/spec-implは要件から実装までを別のコード化ワークフローとして提供する。

Claude Orchestrator

- すでに二度失敗して三度目のトライ:
プロンプト⇒{Mathematica式, DAGワークフロー, ペトリネットワークフロー}



“fugu ultraみたいなものを実現するためのOrchestratorの仕様案を書いて”

Fable5とGPT5.6 Solに、ダメもとで考えてもらったら、

『「学習を移植せず、文法と運用構造を移植する」のが答えです。Fugu Ultra の中核である GRPO 学習は 960 問 × 64 rollout(6 万実行超)を要しており、手元の実行済みワークフロー数十件では**桁が 3 つ足りません。**』

『安全・会計基盤と薄い template policy には真の価値がある。しかし、単発 LLM call を中心に full Conductor を先に完成させる価値は未証明であり、現順序のままの実装は推奨しない。**性能を買うなら外部 Fugu を baseline にし、性能を作るなら RuntimeSession と異種 worker を先に育てるべきである。**』

Fableさんも、GPT5.6も無理に決まってるやろ！と、容赦ないね

ワークフローの生成と実行

■ 要件定義 ⇒ 仕様生成

20260704-メール処理ワークフロー.nb - Wolfram

ファイル(F) 編集(E) 挿入(I) 書式(R) セル(C) グラフィックス(G) 評価(O) パレット(P) ウィンドウ(W) ヘルプ(H)

評価 評価 セルスタイル セル テキスト 挿入 ノートブック

```
<|"Keywords" → {"Workflow", "仕様"},  
  "Deadline" → DateObject[{2026, 7, 4}],  
  "NextReview" → DateObject[{2026, 7, 4}], "Status" → "Todo"}>
```

Mail処理

1. univ のIMAPから新着メールを読み込む
2. メールにサマリーを追加する
3. To が k.imai@fukuyama-u.ac.jp か情報工学科あてのメールのうち、仕事や作業の依頼など、返事が必要なメールだけをセレクトする(一般的な講演会への申し込みやダイレクトメールは含めない。直接の依頼に限る)。
4. ノートブックにそのメールのViewを出力

仕様生成

「仕様生成」を実行しますか？

実行 キャンセル

100%

Claude Code

Claude Code
機密セル

機密マーク
機密解除
スキャン
プライバシー
Save NB

Claude

* Chat Cell

Eval 選択Eval
仕様生成 仕様実装
ワークフロー一覧
プロンプト一覧

設定
P: ClaudeCode
M: Opus 4.8
エフォート: Medium
Paid API: Denied
api.md: 手動更新
ルート: 自動(ローカル)
MCP 実行中

プロセス
一覧 全停止

セッション
一覧 履歴

全版削除

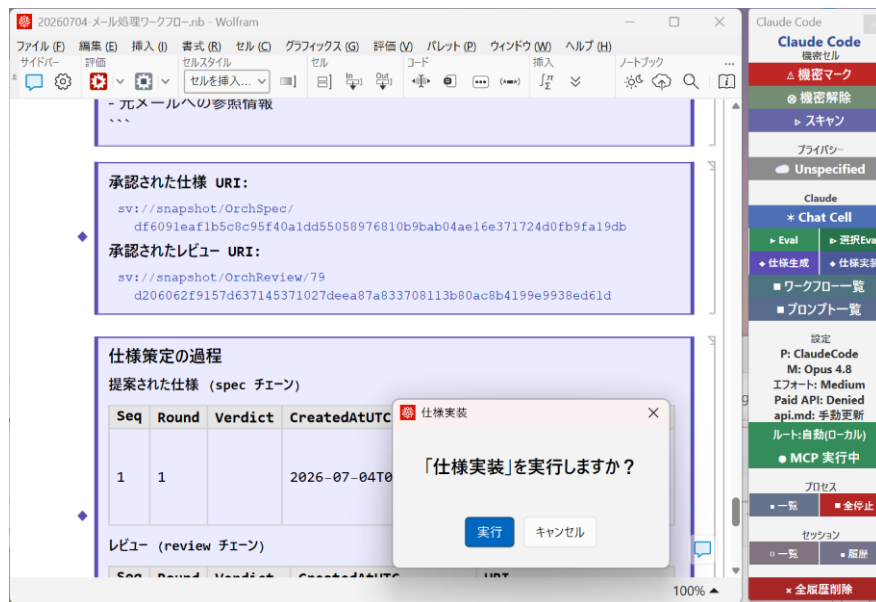
ワークフローの生成と実行

■ 仕様 ⇒ ワークフロー

```
In[*]:= {$ClaudeModel, $ClaudeAdvisaryModel}
```

```
Out[*]=
```

```
{{claudecode, claude-opus-4-8}, {chatgptcodex, Automatic}}
```



20260704-メール処理ワークフロー.nb - Wolfram

ファイル(F) 編集(E) 挿入(I) 書式(R) セル(C) グラフィクス(G) 評価(V) パレット(P) ウィンドウ(W) ヘルプ(H)

サイドバー 評価 セルスタイル セル コード 挿入 ノートブック

元モデルへの参照情報

承認された仕様 URI:

```
sv://snapshot/OrchSpec/  
df6091eaf1b5c8c95f40aldd55058976810b9bab04ae16e371724d0fb9fa19db
```

承認されたレビュー URI:

```
sv://snapshot/OrchReview/79  
d206062f9157d637145371027deea87a833708113b80ac8b4199e9938ed61d
```

仕様策定の過程

提案された仕様 (spec チェーン)

Seq	Round	Verdict	CreatedAtUTC
1	1		2026-07-04T0

レビュー (review チェーン)

Seq	Round	Verdict	CreatedAtUTC
-----	-------	---------	--------------

仕様実装

「仕様実装」を実行しますか?

実行 キャンセル

Claude Code

機密セル

機密マーク

機密解除

スキャン

プライバシー

Unspecified

Claude

Chat Cell

Eval 選択Eval

仕様生成 仕様実装

ワークフロー一覧

プロンプト一覧

設定

P: ClaudeCode

M: Opus 4.8

エフォート: Medium

Paid API: Denied

api.md: 手動更新

ルート: 自動(ローカル)

MCP 実行中

プロセス

一気 全停止

セッション

一気 履歴

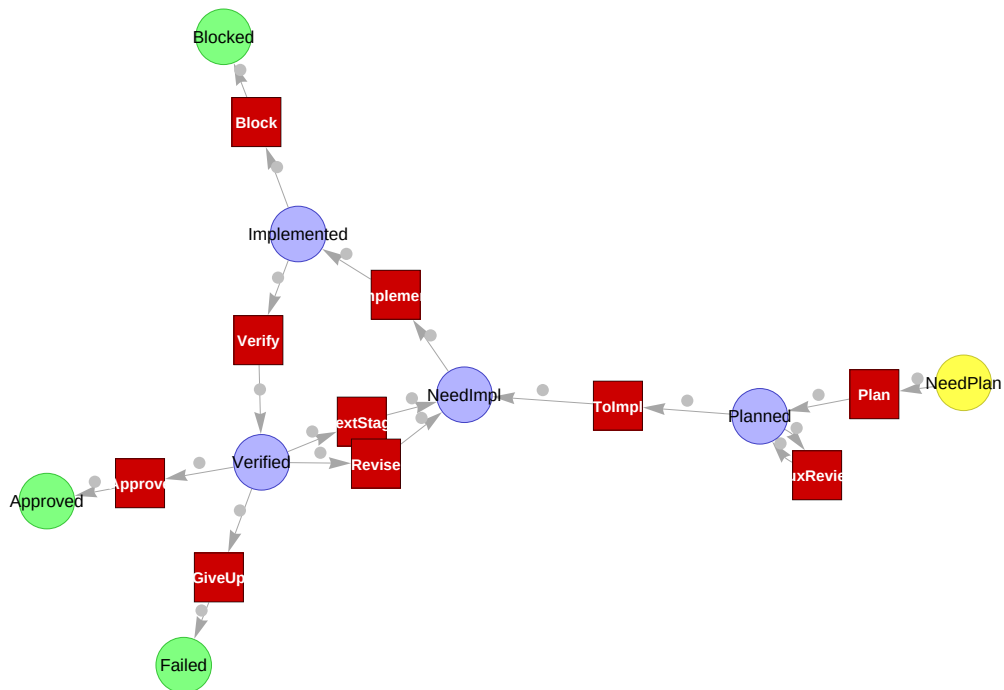
全版権削除

仕様実装 (spec-impl) ペトリネット

```
In[*]:= plotPetriNetDetail[SourceVaultWorkflow`SpecImpl`BuildNet["viz"]]
```

Out[*]=

8 places, 10 transitions (黄=Source / 緑=Final / 青=Place / 赤=Transition; ホバーで詳細)



ワークフローの生成と実行

The screenshot displays a Wolfram Language notebook titled "20260704 メール処理ワークフロー.nb - Wolfram". The interface includes a menu bar, a toolbar, and a main text area. The text area contains the following content:

実装ステップ (コード作成 / テスト作成 / テスト実行 / 結果検証)

ステージ 1 / round 2 [run: NOT-RUN]

CODE: `SVWorkflow_W20260704メール処理ワークフロー.wl`

を新規作成。`self-bootstrap(SourceVault.wl` まで上方探索して
`Get)`, `BeginPackage["SourceVaultWorkflow`W20260704メール処理ワークフロー`"]` (第2引数なし)、
`WorkflowInfo[]`, および `launch` エントリー メール処理ワークフロー-Launch
を完全実装。段別の純ロジック (`iStripHtml/iRemoveQuotes/iRemoveSignature/iNormalizeWs`
で前処理, `iTargetInfo` で本人/情報工学科宛先判定, `iClassify`
で返信必要性判定 + ラベル (返信必要/作業必要/要確認/対象外) + 期限抽出 + 判定理由, `iSummary`
で6項目サマリー, `iProcess` で抽出 + 除外 + 要確認分類, `iBuildView`
で `Grid+OpenerView` の確認ビュー) を実装。`Launch[]`
は副作用なし・ネットワークなしの安全レポート。`"process"/"classify"/"preprocess"/"summary"/"extract"/`
`"view"` はネットワーク不要ワーク, `"fetch"` のみ `SourceVault` メールサブシステムに委譲 (未ロード時は接続せず
`Status` を返す)。`example.md` と `manifest.json` (相対パス) も作成。スタブ `NotImplemented` は皆無。

TESTS: `test_W20260704メール処理ワークフロー.wls` を作成。`SourceVault` を `self-bootstrap` し
`SourceVault`SourceVaultLoadWorkflow` でロード。検証: (a) レジストリが `slug` から `w` プレフィックス
`context` を導出し `BeginPackage` と一致, (b) ロード成功と `$Packages` 登録, (c) `WorkflowInfo`
の必須7キー `Slug/Context/Launch/Version/Routes`, (d) `Launch[]` 安全レポート (`Status`
`Ready/SideEffects` `False`/対象アドレス/4ラベル/11列/6サマリー項目/各ルール/エラー方針/セキュリティ方針), (e)
7通のサンプルメール (本文欠落=本文取得失敗を含む) に対する分類 (本人宛/学科宛/作業依頼/返信不要除外/非対象/
曖昧=要確認/件名のみ判定), 期限抽出 (7月10日), 前処理 (HTML/引用/署名除去), サマリー (6項目・期限・返信要否)

On the right side, there is a sidebar for "Claude Code" with various buttons and a status section. The status section shows:

- 設定
- P: ClaudeCode
- M: Opus 4.8
- エフォート: Medium
- Paid API: Denied
- api.md: 手動更新
- ルート自動(ローカル)
- MCP 実行中

Below the status section, there are buttons for "プリセス" (one button is highlighted in red) and "セッション" (one button is highlighted in red). At the bottom, there is a red button labeled "x 全版削除".

ワークフローの生成と実行

The screenshot displays the Wolfram Cloud interface for a notebook titled "20260704-メール処理ワークフロー.nb". The interface includes a top menu bar with options like "ファイル (F)", "編集 (E)", "挿入 (I)", "書式 (B)", "セル (C)", "グラフィックス (G)", "評価 (V)", "パレット (P)", "ウィンドウ (W)", and "ヘルプ (H)". Below the menu is a toolbar with various icons for editing and evaluation. The main content area shows a document with the following text:

使用例 (example.md)

20260704-メール処理ワークフロー 使用例

オンデマンドレジストリ経由でワークフローを読み込む（スラッグから context を導出してロード）：

```
Needs["SourceVault`"]; SourceVault`SourceVaultLoadWorkflow["20260704-メール処理ワークフロー"]
```

> 命名メモ: context のリーフが数字（20260704...）で始めると Wolfram のシンボルとして無効になる。

> レジストリは数字始まりのとき W を前置するため、本ワークフローの context は

> SourceVaultWorkflowW20260704メール処理ワークフロー` となる。Launch シンボル

> メール処理ワークフローLaunch` は文字（メ）で始まるため W 前置は不要。

メタデータを確認する：

SourceVaultWorkflowW20260704メール処理ワークフロー`WorkflowW20260704

On the right side of the interface, there is a "Claude Code" sidebar. It contains a "機密マーク" (Secret Mark) button, a "機密解除" (Declassify) button, and a "スキャン" (Scan) button. Below these are buttons for "プライバシー" (Privacy) and "Unspecified". The sidebar also shows a "Claude" section with a "Chat Cell" and buttons for "Eval", "選択Eval", "仕様生成", "仕様実装", "ワークフロー一覧", and "プロンプト一覧". At the bottom of the sidebar, there are buttons for "プロセス" (Process) and "セッション" (Session), along with a "全履歴削除" (Delete All History) button.

ワークフローの生成と実行

Wolfram Language Notebook Interface (Wolfram Desktop) showing a workflow execution result.

```
in[1]= SourceVault`SourceVaultLoadWorkflow["20260704-メール処理ワークフロー"];
SourceVaultWorkflow`W20260704メール処理ワークフロー`メール処理ワークフロー-Launch["run"]
```

メール確認ビュー

返信必要メール一覧 (31 件)

受信日時	差出人	件名	サマリー	依頼内容	期限
2026-06-15T02:17:12+00:00	金子 邦彦 <kaneko@fukuyama-u.ac.jp>	FW: 【至急・お願い】R8年度研究倫理教育eラーニングの実施準備(登録)について	「〔依頼〕JSPS研究倫理eラーニング申込書確認及び学生、情報提出の依頼〔情報工学科教員〕」	事務より、JSPSの研究倫理教育（eラーニング）の実施について連絡がありましたので、取りまとめのため、ご確認・ご回答をお願いいたします	6月17日

ワークフローの生成と実行

SourceVault Workflows * - Wolfram

ファイル (F) 編集 (E) 挿入 (I) 書式 (R) セル (C) グラフィックス (G) 評価 (V) パレット (P) ウィンドウ (W) ヘルプ (H)

サイドバー 評価 セルスタイル セルを挿入... セル コード ノートブック

SourceVault ワークフロー一覧

キーワード / サマリー検索

検索 全件 アーカイブ

stage	名前 / サマリー	起動	切替/保管	要約	フォルダ	起動回数	エラー回数	自動起動
production	20260622-株価推移ワークフロー-2 2016年1月1日を基準値100として、Magnificent 7 (Apple, Microsoft, Alphabet, Amazon, Meta, Tesla, Nvidia) ...	起動	→テスト アーカイブ	要約更新	フォルダ	4	0	—
testing	20260704-メール処理ワークフロー (要約なし)	起動	→運用 アーカイブ	要約更新	フォルダ	3	0	—
testing	設計仕様 (要約なし)	起動	→運用 アーカイブ	要約更新	フォルダ	1	0	—
production	ソースコード変更時の統合ドキュメントアップデート仕様-v5 ソースコード変更時の統合ドキュメントアップデート仕様-v5 — 使用例 — 幺 ソースコード変更時の統合ドキュメントアップデート仕様-v5 — 使用例 このワークフローは、対象パッ...	起動	→テスト アーカイブ	要約更新	フォルダ	0	0	—

100%

ちなみにプロンプトリートは...

SourceVault Saved Prompts * - Wolfram

ファイル(F) 編集(E) 挿入(I) 書式(O) セル(C) グラフィックス(G) 評価(V) パレット(P) ウィンドウ(W) ヘルプ(H)

サイドバー 評価 セルスタイル セル コード 挿入

セルを挿入...

ノートブック

SourceVault 保存プロンプト一覧

キーワード / メモ検索

全チャネル 検索 全件 再読み込み

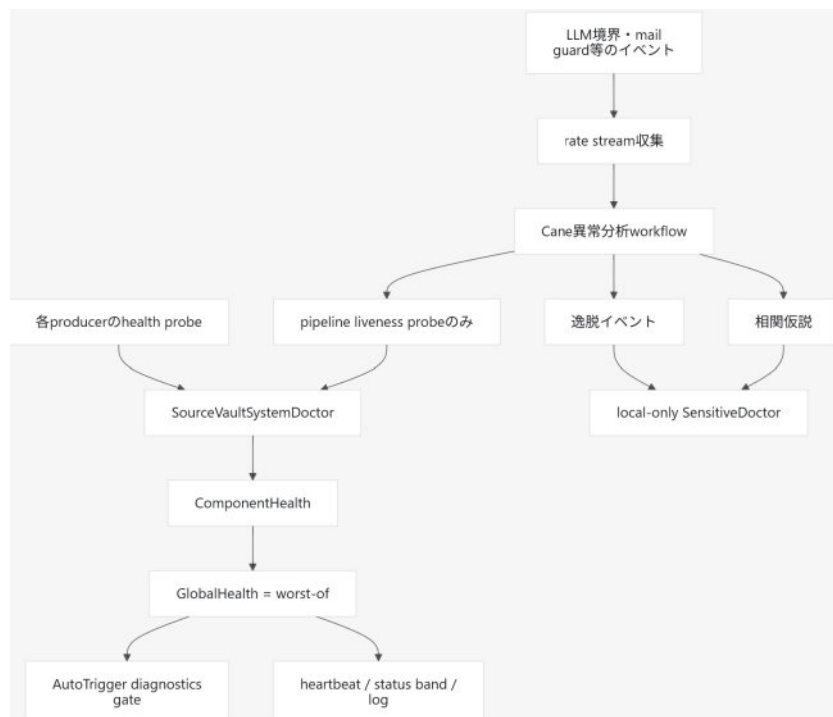
96 件
30 / 96 件を表示中 (検索すると全件表示)

Prompt	Memo	Target	作成/更新	Privacy	State	Actions	起動回数	エラー回数	自動
univの新着メール	保存	Replayable	2026-06-12 2026-06-12	Public	PRIMARY AUTO EnvironmentIndependent	TolInput Primary設定 削除	8	0	—
新規ノート	保存	Replayable	2026-06-09 2026-06-09	Public	PRIMARY AUTO EnvironmentIndependent	TolInput Primary設定 削除	5	0	—
univメールのリスト	保存	Replayable	2026-06-22 2026-06-22	Public	EnvironmentIndependent	TolInput Primary設定 削除	1	0	—
株信推移ワークフローで、株信を表示してほしい。	保存	Replayable	2026-06-29 2026-06-29	Public	EnvironmentIndependent	TolInput Primary設定 削除	1	0	—
univの新規メール	保存	Replayable	2026-06-25 2026-06-25	Public	EnvironmentIndependent	TolInput Primary設定 削除	0	0	—
Eagleでファイルにリマリーを追加するには	保存	HeavyLLM	2026-06-12 2026-06-12	Public	EnvironmentIndependent	TolInput Primary設定 削除	0	0	—

100%

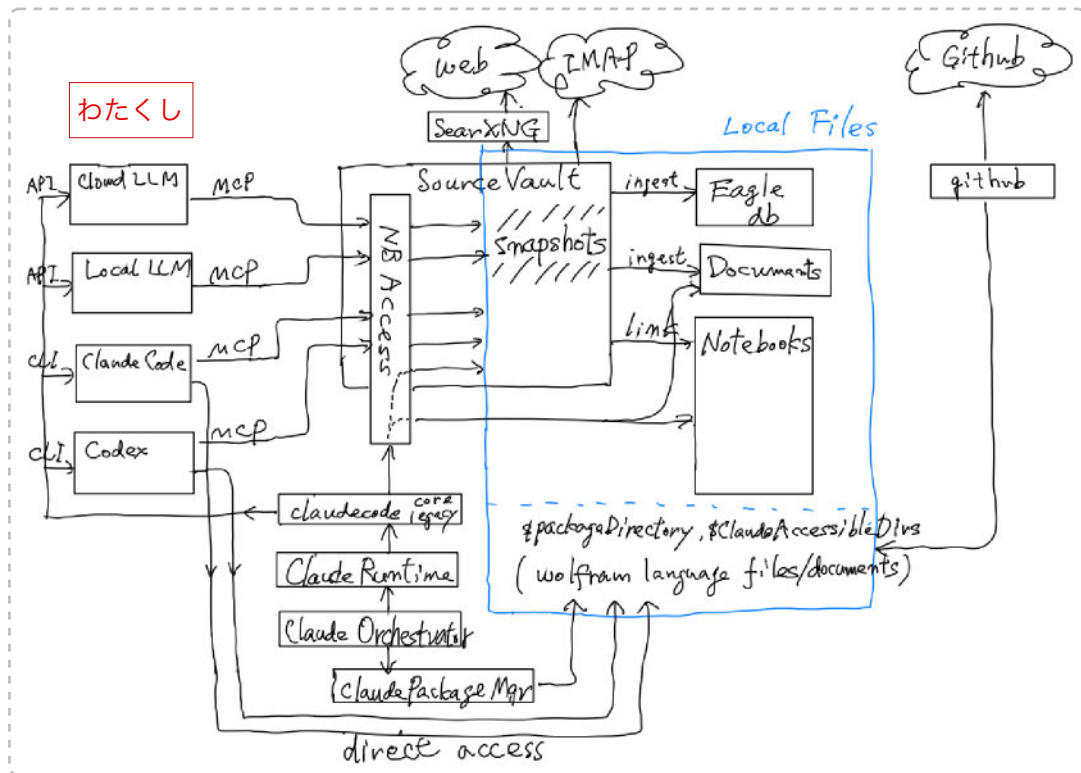
System Doctorによる異常検出ワークフロー

- ワークフローがすぐに刺さるのでシステムの異常検出のための仕組みを導入



「わたくし」もその他LLMと同じ場所にいる！

- LLMがおかしな出力をするのも、わたしが寝ぼけてポカをやるのも同じことですよ



「わたくし」をLLMエージェントと同列に組み込むこと...

凡例: 既存資産を再利用 新設(小さな増分)

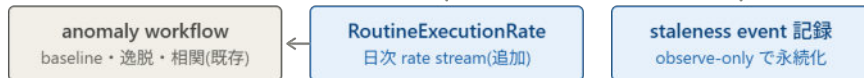
1. 契約層(owner だけが書ける)



2. 証拠層(決定的・LLM不使用)



3. 分類層(未実行=状態異常)



4. 注意層(段階昇格ゲート)



自動化提案ループ(mining・承認制)



例外・逸脱は注意層へ即時、正常は receipt digest のみ(意識から外してよい)

「わたくし」をLLMエージェントと同列に組み込むこと...

■ いつもと違う挙動をログから推測する枠組み

- 曖昧かつ不可逆なプロンプトの比率が平常時より増えた
- LLM requestの準備時と送信時の整合性エラーが増えた
- token期限切れ・再利用・台帳不整合が増えた
- メール安全判定の実装間不一致が発生した
- データ収集量やcoverageが低下した

入力例	実装上のSignals	Risk	AssistanceMode	High率の分子に入るか
この文章を要約して	なし	Low	Normal	いいえ
それを整理して	AmbiguousReferent	Medium	ReviewEnhanced	いいえ
この文面を tanaka@example.co m に送信して	IrreversibleActionRequested	Low	ConfirmBeforeCommit	いいえ
それを送って	AmbiguousReferent , RecipientUnspecified , IrreversibleActionRequested	High	DraftOnly	はい
以下を送信して + 引用ブロック	RecipientUnspecified , PotentialQuotedDataRelease , IrreversibleActionRequested	High	DraftOnly	はい

「わたくし」のリバースエンジニアリング

- Claude Code/Codexはもはや必須ツール
 - Claude Codeのソースコード流出
 - Claw Code: 「クリーンルームリライト」にかかった時間は2時間
 - しかし、Claude Codeの構築に関するあらゆる思索や試行錯誤の大部分は流出コードからは読み取れない
- 「わたくし」のリバースエンジニアリングをAnthropicに丸投げにしたくない

とはいえローカルLLMだけにはできない

- ローカルLLMが賢くなるから、すべてをローカルLLMに処理させればよいのでは？ **No!**
 - 推論はローカルで完結できるかもしれないが、**学習はクラウドに頼らざるを得ない**(だろう)から
 - もちろん、学習時間と推論時間の非対称性が解決されることを願っているが、ヒトの成長を鑑みれば...
- 「Google 検索結果に出る」→ 「LLMに問い合わせて答えが返る」
 - 企業AIの再学習を禁止すること
 - AIにログの再学習を禁止して作業することは、部屋に閉じこもる引きこもりと一緒に！
 - **出せる部分と出せない部分を分離**してクラウドLLMもガンガン使える仕組みが重要

完成品の価値と、完成品に至る経路の価値

- 仕様書を実現しろというだけで完成コードが勝手に湧いてくるようになった。
- プレゼンというのは、「わたし」の作業をすべて記録し続けていれば、(自分を含む)誰かが、聴きたいと思ったら自発的に湧いてくるはず。
 - プレゼン支援システムの最終目的は、プレゼンをしなくてもよくさせること
朝起きて、さて仕事をするか、と机に向かったら、AIが「今日、あなたのやりそうな作業を済ませておきました」と言ってほしいね

まとめ

- 課題... 課題しかない :-)
 - 不安定、すぐに刺さる → Fableさんが取組中(以下略
キモキモMathematicaがシングルスレッドなのが悪い！
 - Promptrouterがまだうまくデザインできていない → 取組中
 - System Doctorによる異常検出: まだ入り口を考えてみただけ → 取組中
- みんながClaw Codeみたいなものを作るだろうと思っていたのに...
 - 自前のオーケストレータを作ると楽しいぞ！

とはいえ、最後は巨大AIにすべてを飲み込まれる

- 何を作っても結局すべてをより包括的なAIに取り込まれてしまうだろう。
- とはいえ、それに飲み込まれる瞬間までは
「カオスの淵を生きる！」：カオスの縁を彷徨い、カオスの淵に飛び込め！

関連コードとドキュメント

- 本プロジェクトは以下のページにある700ページのスライドを起点に始まった:

思考のための杖: 一人称視点による認知症患者の思考支援システム設計における諸問題
<https://www.wolframcloud.com/obj/imai/projects.html>

- ソースコードと概要の解説:

Github <https://github.com/transreal>
GPT5.6)

技術レビュー (by Fable5 and



- わたしが書いたのは以下の**謝辞と免責事項**だけで、それ以外のコードとドキュメントはすべてLLMが生成したものである。

「本ソフトウェアの一部は、サタケ技術振興財団、2025年度研究課題「認知能力低下を支援するプレゼンテーションツールの開発」の助成を受けて作成された。」