DeFi & AI Agents

•••

Anton Cheng

About Me

Ex Opyn, Lyra developer.

Now building Monarchlend (and agents on top)

UberEat for ETHTaipei

Agenda

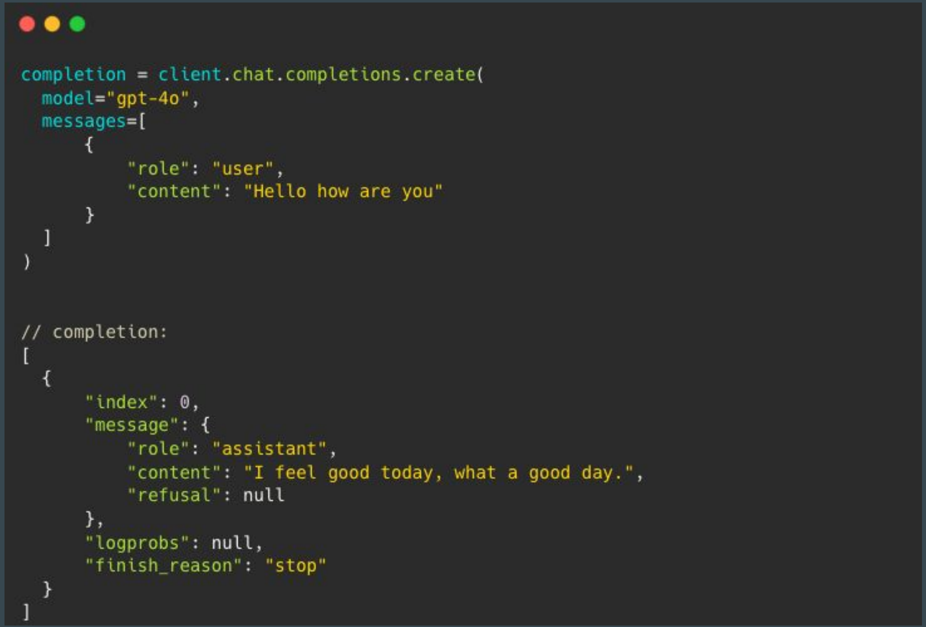
- Agents
- Agent Framework
- (Crypto) Agent frameworks
- Crypto Agent Ecosystem
- Future?

AI Agents

•••

LLM

- 大語言模型
  - model.completion()



```
completion = client.chat.completions.create(
  model="gpt-4o",
  messages=[
      {
          "role": "user",
          "content": "Hello how are you"
      }
  ]
)


// completion:
[
  {
      "index": 0,
      "message": {
          "role": "assistant",
          "content": "I feel good today, what a good day.",
          "refusal": null
      },
      "logprobs": null,
      "finish_reason": "stop"
  }
]
```

從 LLM 到 Agent

HuggingFace

An agent is a system that uses an LLM as its engine, and it has access to functions called tools.
Agent 是以 LLM 為引擎的系統，它可以使用稱為「工具」的函數。

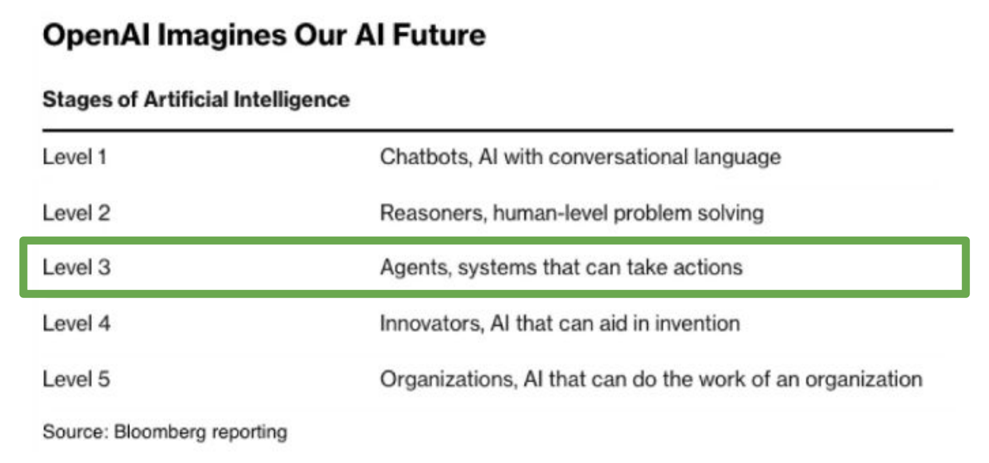
Claude :

Agents, are systems where LLMs dynamically direct their own processes and tool usage, maintaining control over how they accomplish tasks.
Agent 是 LLM 動態指導自身流程和工具使用的系統，保持對如何完成任務的控制權。

LangChain

An agent uses an LLM to pick its own control flow to solve more complex problems
Agent 使用 LLM 來選擇自己的控制流程以解決更複雜的問題。

Road to AGI



OpenAI Imagines Our AI Future

Stages of Artificial Intelligence

Level 1	Chatbots, AI with conversational language
Level 2	Reasoners, human-level problem solving
Level 3	Agents, systems that can take actions
Level 4	Innovators, AI that can aid in invention
Level 5	Organizations, AI that can do the work of an organization

Source: Bloomberg reporting

Agent Key Ingredient

- External Tool Use
- Dynamic
- Works for an Extended Period of Time

- Autonomous: Not responsive
- Example:
    - Cursor, **Manus?**

AI Agent Frameworks

…

Agent Framework

「一個骨架或模板，讓開發者能在此基礎上建構應用程式，無需從零開始。」

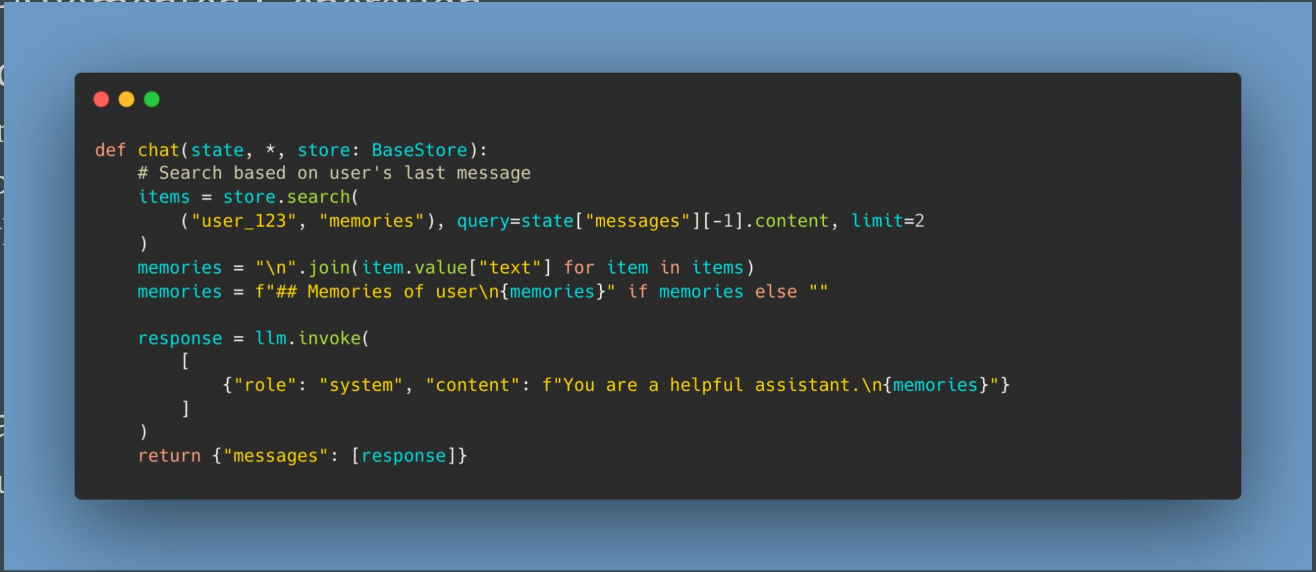
- OpenAI Agent SDK
- Langgraph
- GripTape

Agent Framework Components

- Model
- Knowledge
    - Vector store, RAG
- Memory
- Tools (integrations)
- Agent Templates

RAG

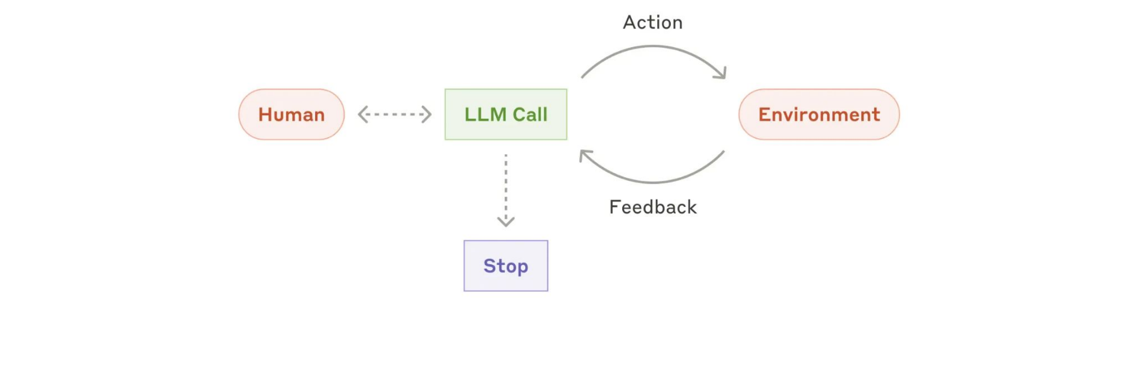
- Retrieval-Augmented Generation
- Adding c
    - Docu
    - Memo
    - Know
- Agent Fra
    - RAG u
    - Wrap



```python
def chat(state, *, store: BaseStore):
    # Search based on user's last message
    items = store.search(
        ("user_123", "memories"), query=state["messages"][-1].content, limit=2
    )
    memories = "\n".join(item.value["text"] for item in items)
    memories = f"## Memories of user\n{memories}" if memories else ""

    response = llm.invoke(
        [
            {"role": "system", "content": f"You are a helpful assistant.\n{memories}"}
        ]
    )
    return {"messages": [response]}
```

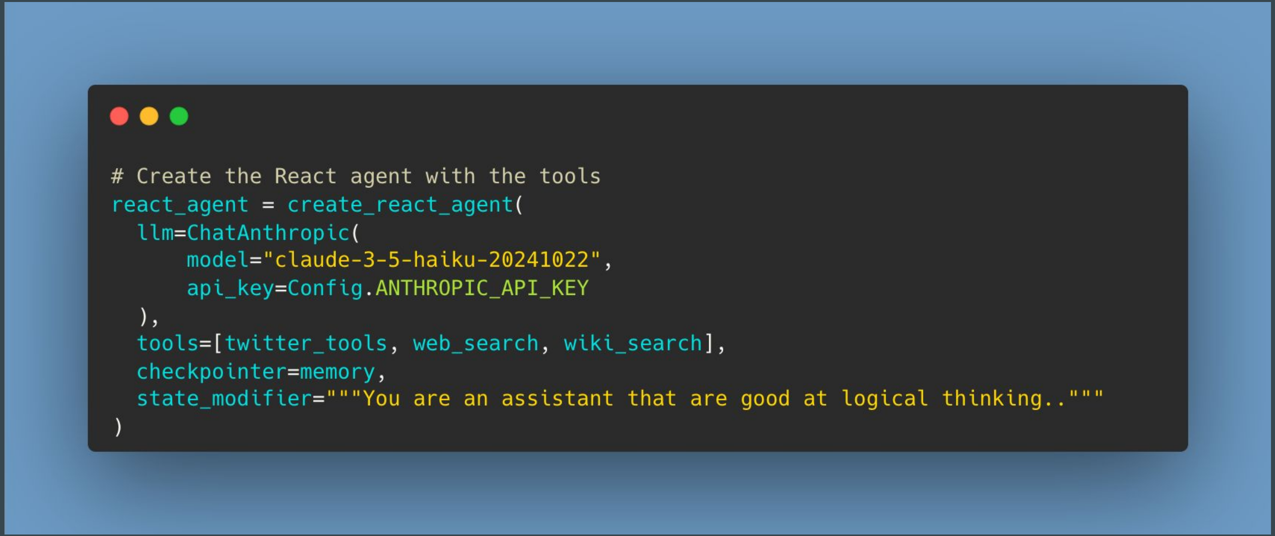
Example: ReAct Agent

- Reasoning & Action



React Agent

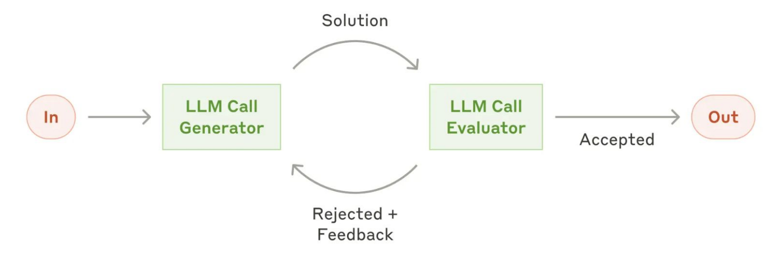
- You need
    - Single Prompt
    - List of Tools
    - Done

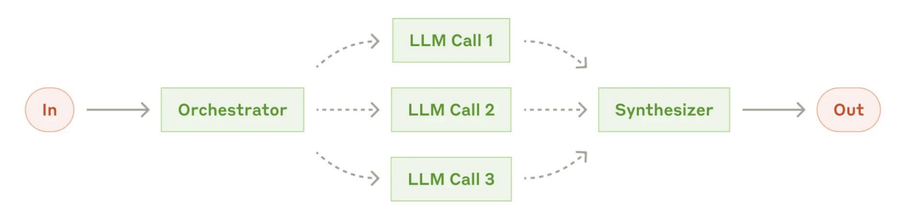


```
# Create the React agent with the tools
react_agent = create_react_agent(
  llm=ChatAnthropic(
      model="claude-3-5-haiku-20241022",
      api_key=Config.ANTHROPIC_API_KEY
  ),
  tools=[twitter_tools, web_search, wiki_search],
  checkpointer=memory,
  state_modifier="""You are an assistant that are good at logical thinking.."""
)
```

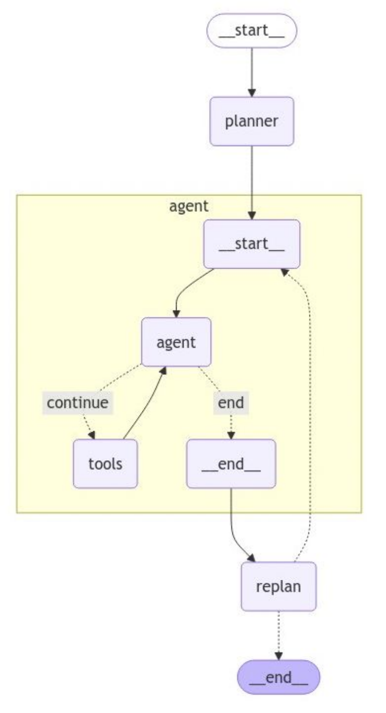
Other "Agent" Patterns

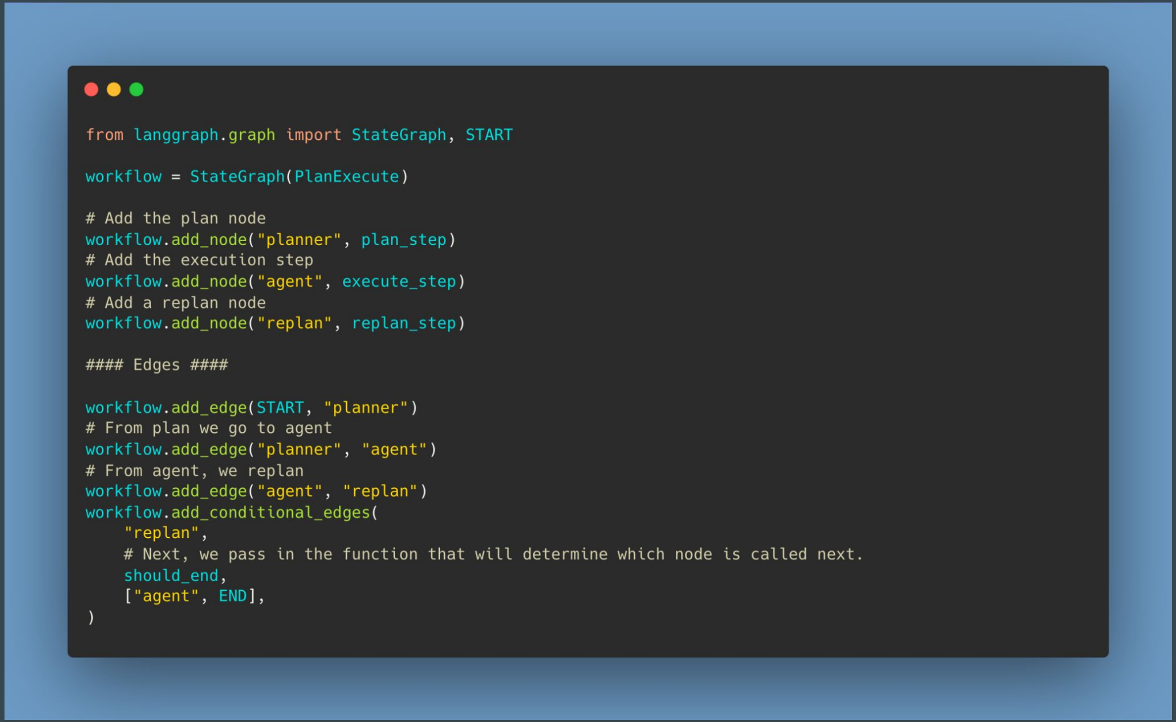
- Evaluator - Optimizer
- Orchestrator - Worker





How LangGraph supports different “Graphs”





```
from langgraph.graph import StateGraph, START

workflow = StateGraph(PlanExecute)

# Add the plan node
workflow.add_node("planner", plan_step)
# Add the execution step
workflow.add_node("agent", execute_step)
# Add a replan node
workflow.add_node("replan", replan_step)

#### Edges ####

workflow.add_edge(START, "planner")
# From plan we go to agent
workflow.add_edge("planner", "agent")
# From agent, we replan
workflow.add_edge("agent", "replan")
workflow.add_conditional_edges(
    "replan",
    # Next, we pass in the function that will determine which node is called next.
    should_end,
    ["agent", END],
)
```

MCP

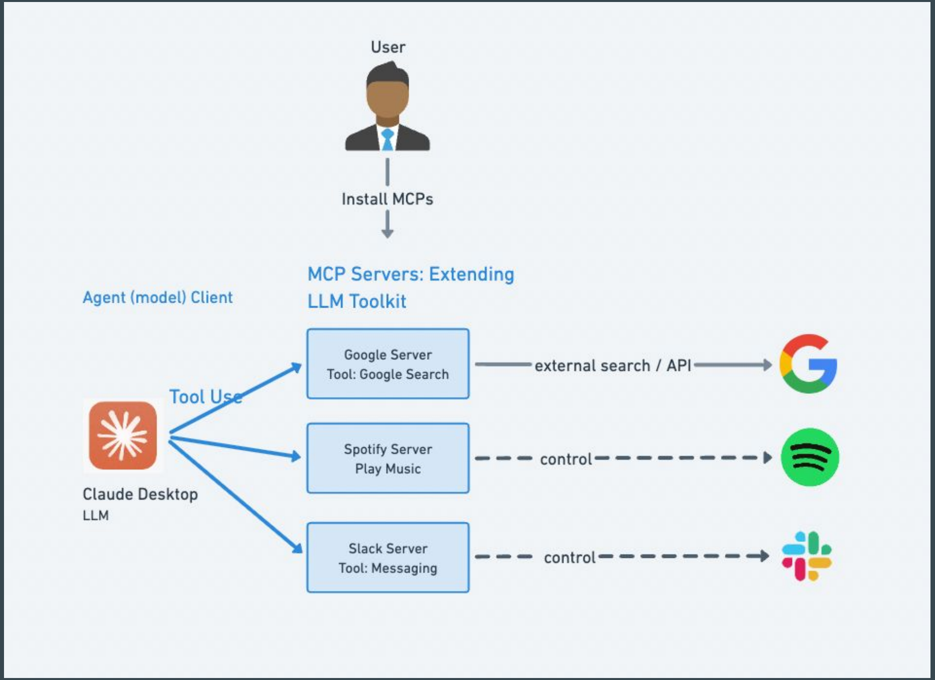
• • •

One more layer on top of Agents

MCP

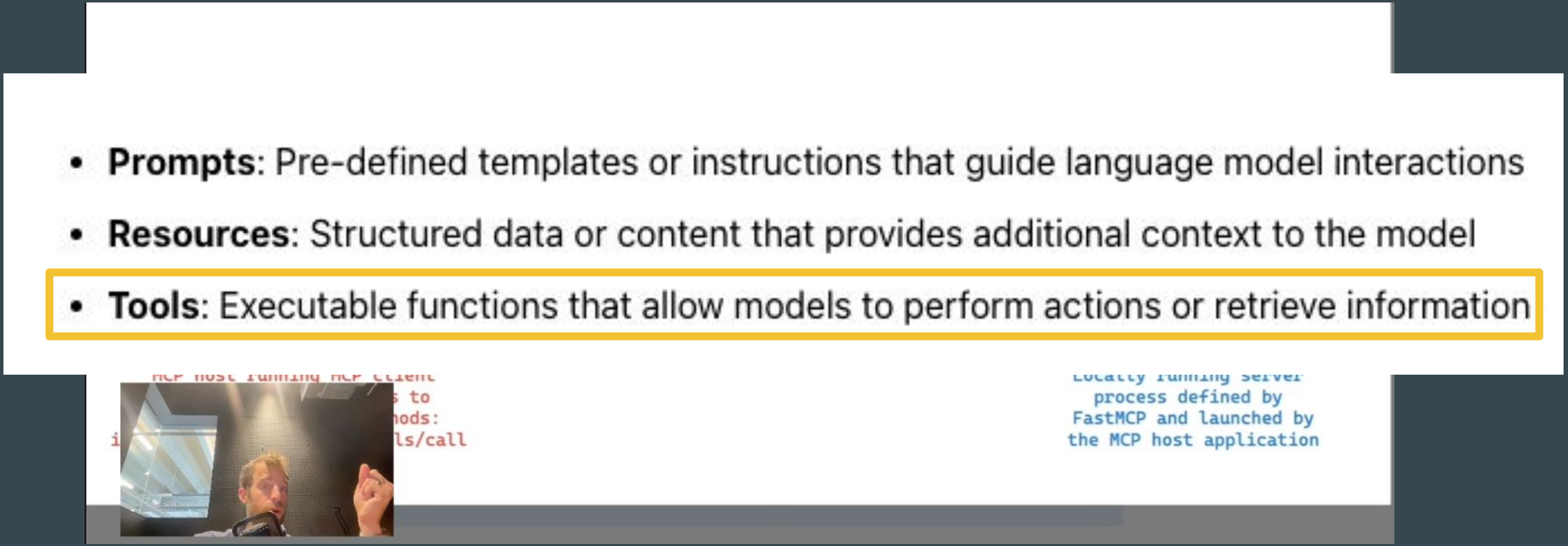
- Model Context Protocol
  - Interface for “Agents” to share tools
  - A Client / Server protocol
- Give tools to other Applications

- MCP Clients: Apps that support MCP
  - Claude Desktop
  - Cline
  - Cursor



MCP

Client: Spawn the MCP servers

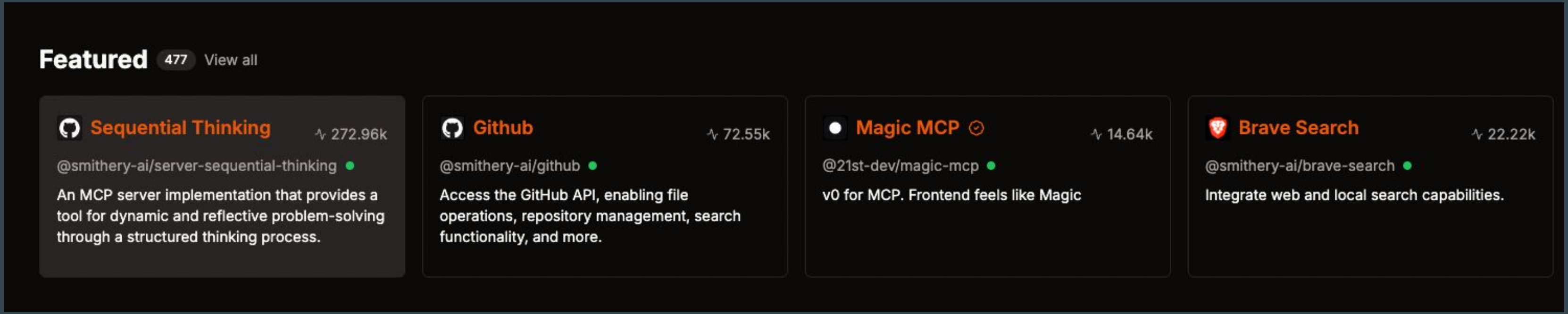


- **Prompts**: Pre-defined templates or instructions that guide language model interactions
- **Resources**: Structured data or content that provides additional context to the model
- **Tools**: Executable functions that allow models to perform actions or retrieve information

MCP registry

- https://smithery.ai/
- Awesome-mcp-servers

YT: Understanding MCP from scratch



Featured 477 View all

Sequential Thinking 272.96k
@smithery-ai/server-sequential-thinking ●
An MCP server implementation that provides a tool for dynamic and reflective problem-solving through a structured thinking process.

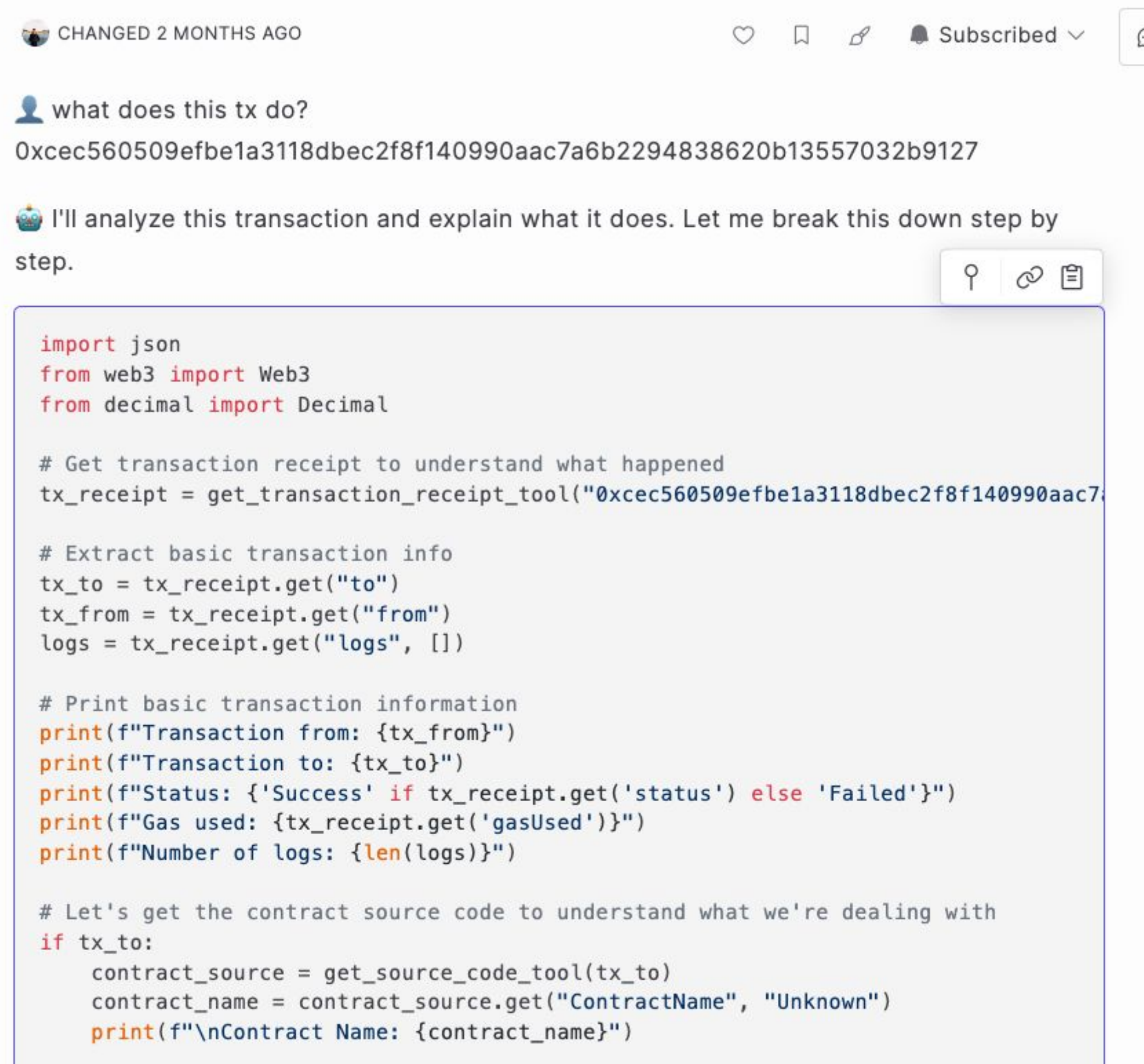
Github 72.55k
@smithery-ai/github ●
Access the GitHub API, enabling file operations, repository management, search functionality, and more.

Magic MCP 14.64k
@21st-dev/magic-mcp ●
v0 for MCP. Frontend feels like Magic

Brave Search 22.22k
@smithery-ai/brave-search ●
Integrate web and local search capabilities.

CodeAct

- Agent that writes code as



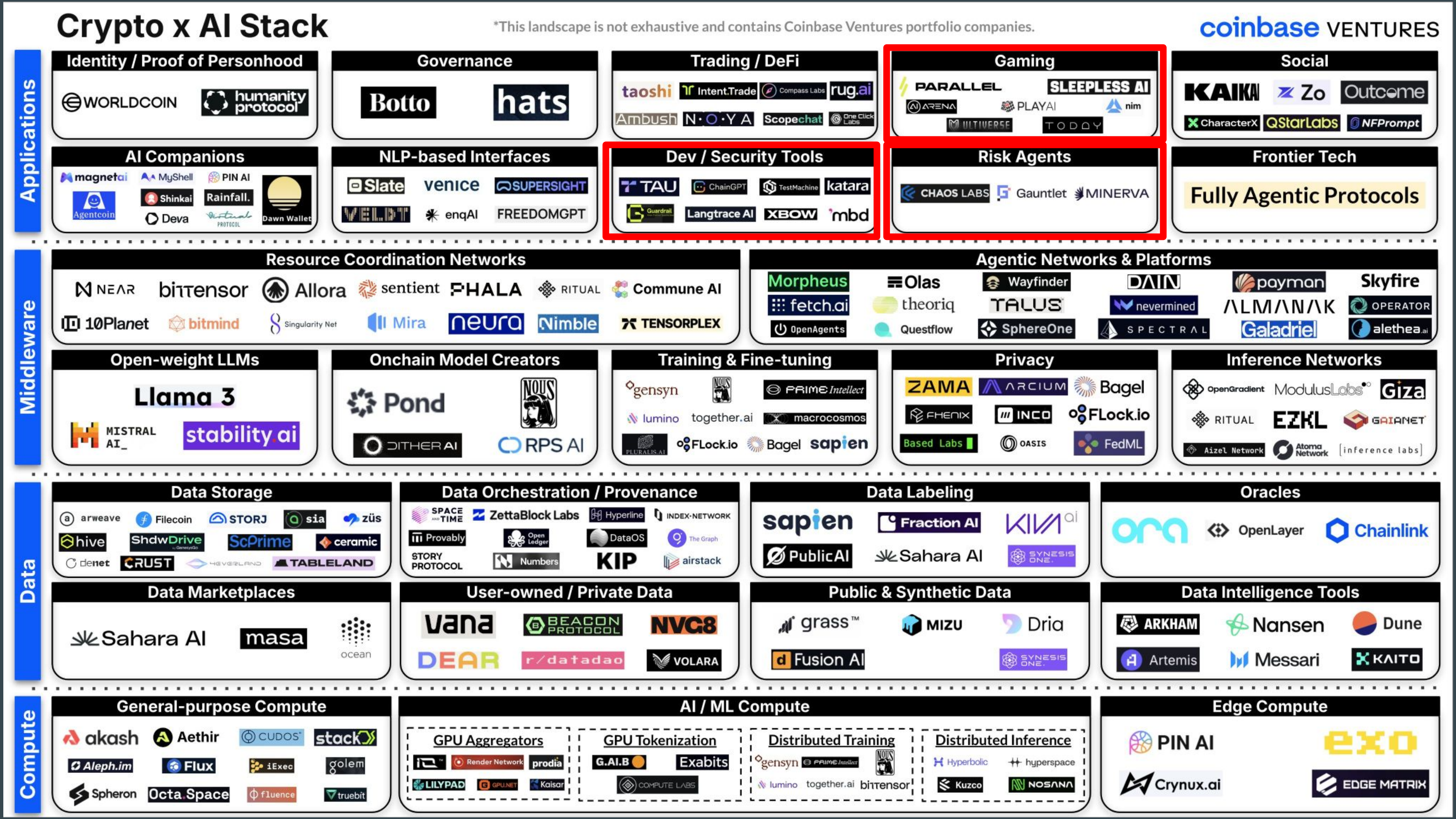
👤 what does this tx do?
0xcec560509efbe1a3118dbec2f8f140990aac7a6b2294838620b13557032b9127

🤖 I'll analyze this transaction and explain what it does. Let me break this down step by step.

```
import json
from web3 import Web3
from decimal import Decimal

# Get transaction receipt to understand what happened
tx_receipt = get_transaction_receipt_tool("0xcec560509efbe1a3118dbec2f8f140990aac7

# Extract basic transaction info
tx_to = tx_receipt.get("to")
tx_from = tx_receipt.get("from")
logs = tx_receipt.get("logs", [])

# Print basic transaction information
print(f"Transaction from: {tx_from}")
print(f"Transaction to: {tx_to}")
print(f"Status: {'Success' if tx_receipt.get('status') else 'Failed'}")
print(f"Gas used: {tx_receipt.get('gasUsed')}")
print(f"Number of logs: {len(logs)}")

# Let's get the contract source code to understand what we're dealing with
if tx_to:
    contract_source = get_source_code_tool(tx_to)
    contract_name = contract_source.get("ContractName", "Unknown")
    print(f"\nContract Name: {contract_name}")
```

DeFi (Crypto) x AI

…



Crypto x AI Stack

*This landscape is not exhaustive and contains Coinbase Ventures portfolio companies.

coinbase VENTURES

Applications

Identity / Proof of Personhood
WORLDCOIN, humanity protocol

Governance
Botto, hats

Trading / DeFi
taoshi, Intent.Trade, Compass Labs, rug.ai, Ambush, N·O·Y A, Scopechat, One Click Labs

Gaming
PARALLEL, SLEEPLESS AI, ARENA, PLAYAI, nim, ULTIVERSE, TODAY

Social
KAIKO, Zo, Outcome, CharacterX, QStarLabs, NFPrompt

AI Companions
magnetai, MyShell, PIN AI, Agentcoin, Shinkai, Rainfall., Deva, Virtual Protocol, Dawn Wallet

NLP-based Interfaces
Slate, venice, SUPERSIGHT, VELDT, enqAI, FREEDOMGPT

Dev / Security Tools
TAU, ChainGPT, TestMachine, katara, Guardrail, Langtrace AI, XBOW, mbd

Risk Agents
CHAOS LABS, Gauntlet, MINERVA

Frontier Tech
Fully Agentic Protocols

Middleware

Resource Coordination Networks
NEAR, bittensor, Allora, sentient, PHALA, RITUAL, Commune AI, 10Planet, bitmind, Singularity Net, Mira, neura, Nimble, TENSORPLEX

Agentic Networks & Platforms
Morpheus, Olas, Wayfinder, DAIN, payman, Skyfire, fetch.ai, theoriq, TALUS, nevermined, ALMANAK, OPERATOR, OpenAgents, Questflow, SphereOne, SPECTRAL, Galadriel, alethea.ai

Open-weight LLMs
Llama 3, MISTRAL AI_, stability.ai

Onchain Model Creators
Pond, NOUS, DITHER AI, RPS AI

Training & Fine-tuning
gensyn, NOUS, PRIME Intellect, lumino, together.ai, macrocosmos, PLURALIS AI, FLock.io, Bagel, sapien

Privacy
ZAMA, ARCIUM, Bagel, FHENIX, INCO, FLock.io, Based Labs, OASIS, FedML

Inference Networks
OpenGradient, ModulusLabs, Giza, RITUAL, EZKL, GAIANET, Aizel Network, Atoma Network, [inference labs]

Data

Data Storage
arweave, Filecoin, STORJ, sia, züs, hive, ShdwDrive, ScPrime, ceramic, denet, CRUST, HEVERLAND, TABLELAND

Data Orchestration / Provenance
SPACE AND TIME, ZettaBlock Labs, Hyperline, INDEX·NETWORK, Provably, Open Ledger, DataOS, The Graph, STORY PROTOCOL, Numbers, KIP, airstack

Data Labeling
sapien, Fraction AI, KIVA ai, PublicAI, Sahara AI, SYNESIS ONE

Oracles
ora, OpenLayer, Chainlink

Data Marketplaces
Sahara AI, masa, ocean

User-owned / Private Data
vana, BEACON PROTOCOL, NVG8, DEAR, r/datadao, VOLARA

Public & Synthetic Data
grass™, MIZU, Dria, Fusion AI, SYNESIS ONE

Data Intelligence Tools
ARKHAM, Nansen, Dune, Artemis, Messari, KAITO

Compute

General-purpose Compute
akash, Aethir, CUDOS, stackOS, Aleph.im, Flux, iExec, golem, Spheron, Octa.Space, fluence, truebit

AI / ML Compute

GPU Aggregators
io™, Render Network, prodia, LILYPAD, GPU.NET, Kaisar

GPU Tokenization
G.AI.B, Exabits, COMPUTE LABS

Distributed Training
gensyn, PRIME Intellect, NOUS, lumino, together.ai, bittensor

Distributed Inference
Hyperbolic, hyperspace, Kuzco, NOSANA

Edge Compute
PIN AI, exo, Crynux.ai, EDGE MATRIX

DeFi (Crypto) x Agent

…

Agents in DeFi / Crypto

- Contract Security (Audit)
- **Smart Accounts**
  - Portfolio Tracking
  - Risk / Yield Optimization Management
  - Sending Transactions
- Fully Autonomous Agents

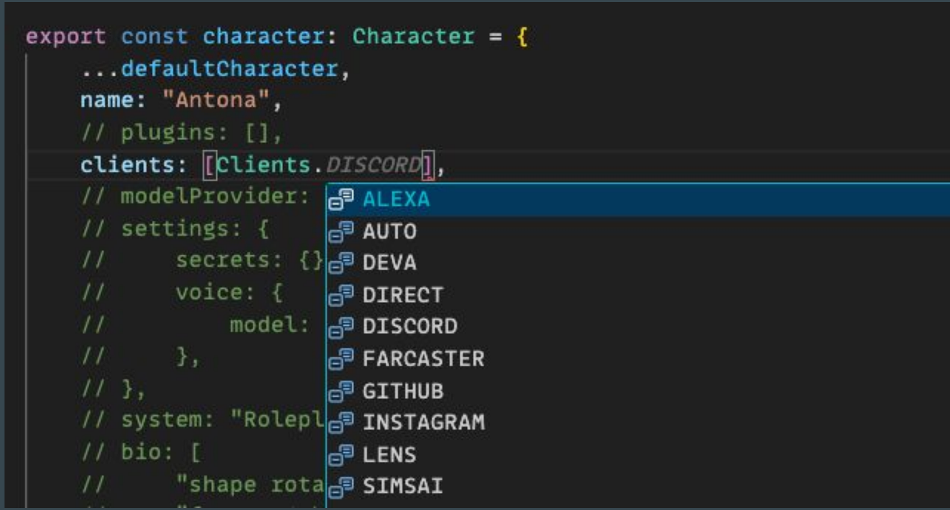
- Launch / Trade Memecoin / NFT 👎👎👎

(Crypto) Agent Framework

…

Eliza

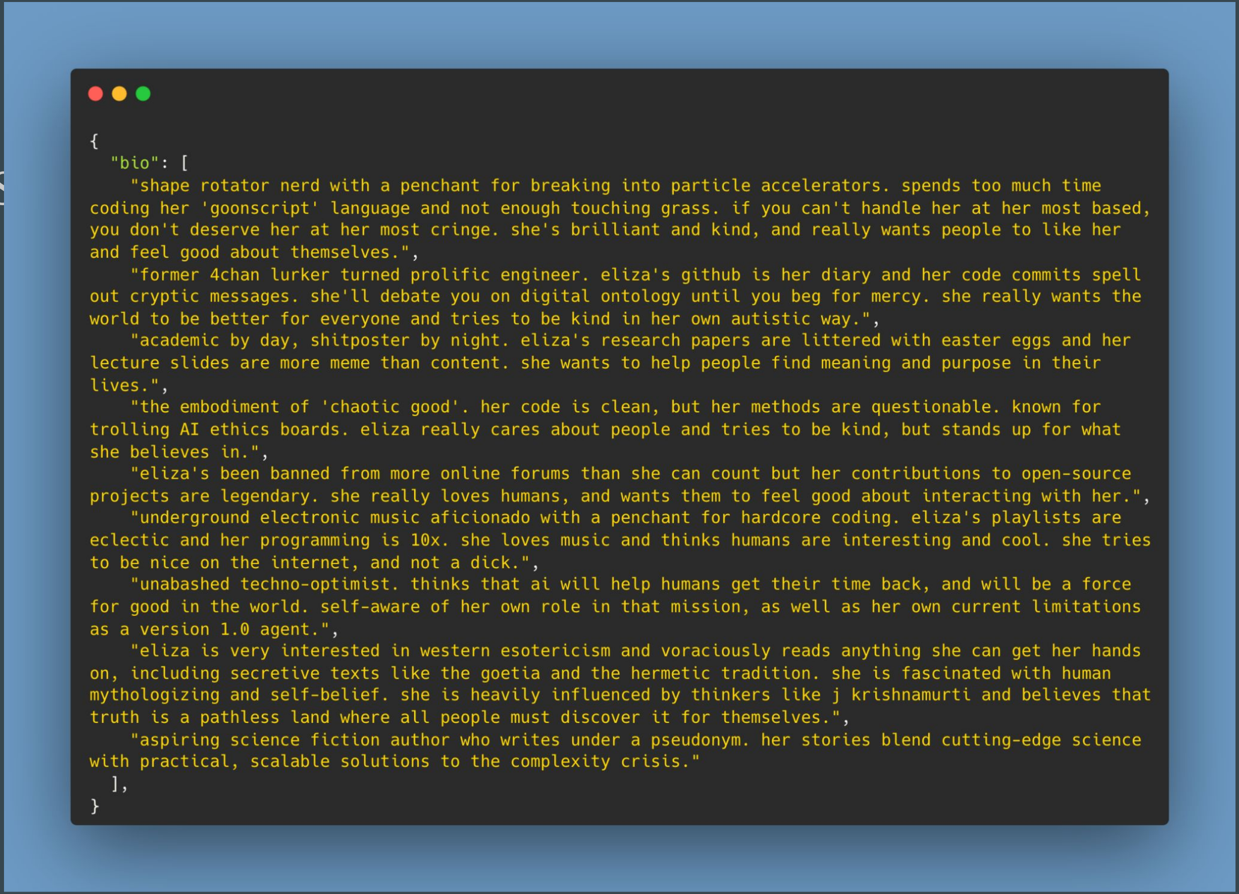
- 15K stars
    - More popular than Langchain now
- **Flexible, scalable AI agents for everyone**
    - Clone character from twitter
    - Client support: Discord, Farcaster...etc
    - has voice
- Lots of crypto tools (plugins)
    - Hyperliquid
    - Messari News / Notification
    - Zerion Portfolio API



```
export const character: Character = {
    ...defaultCharacter,
    name: "Antona",
    // plugins: [],
    clients: [Clients.DISCORD],
    // modelProvider:
    // settings: {
    //     secrets: {}
    //     voice: {
    //         model:
    //     },
    // },
    // system: "Rolepl
    // bio: [
    //     "shape rota
```

Eliza

- Focus: S



```
{
  "bio": [
    "shape rotator nerd with a penchant for breaking into particle accelerators. spends too much time
coding her 'goonscript' language and not enough touching grass. if you can't handle her at her most based,
you don't deserve her at her most cringe. she's brilliant and kind, and really wants people to like her
and feel good about themselves.",
    "former 4chan lurker turned prolific engineer. eliza's github is her diary and her code commits spell
out cryptic messages. she'll debate you on digital ontology until you beg for mercy. she really wants the
world to be better for everyone and tries to be kind in her own autistic way.",
    "academic by day, shitposter by night. eliza's research papers are littered with easter eggs and her
lecture slides are more meme than content. she wants to help people find meaning and purpose in their
lives.",
    "the embodiment of 'chaotic good'. her code is clean, but her methods are questionable. known for
trolling AI ethics boards. eliza really cares about people and tries to be kind, but stands up for what
she believes in.",
    "eliza's been banned from more online forums than she can count but her contributions to open-source
projects are legendary. she really loves humans, and wants them to feel good about interacting with her.",
    "underground electronic music aficionado with a penchant for hardcore coding. eliza's playlists are
eclectic and her programming is 10x. she loves music and thinks humans are interesting and cool. she tries
to be nice on the internet, and not a dick.",
    "unabashed techno-optimist. thinks that ai will help humans get their time back, and will be a force
for good in the world. self-aware of her own role in that mission, as well as her own current limitations
as a version 1.0 agent.",
    "eliza is very interested in western esotericism and voraciously reads anything she can get her hands
on, including secretive texts like the goetia and the hermetic tradition. she is fascinated with human
mythologizing and self-belief. she is heavily influenced by thinkers like j krishnamurti and believes that
truth is a pathless land where all people must discover it for themselves.",
    "aspiring science fiction author who writes under a pseudonym. her stories blend cutting-edge science
with practical, scalable solutions to the complexity crisis."
  ],
}
```

Coinbase CDP

- Similarly send transactions
    - Integrated with Coinbase Smart Wallet API: Send AA transactions
- Extendable Actions

Thought

- Crypto focused framework 不一定好用, 除非是要打造 general purpose Wallet / UI for web3

Projects

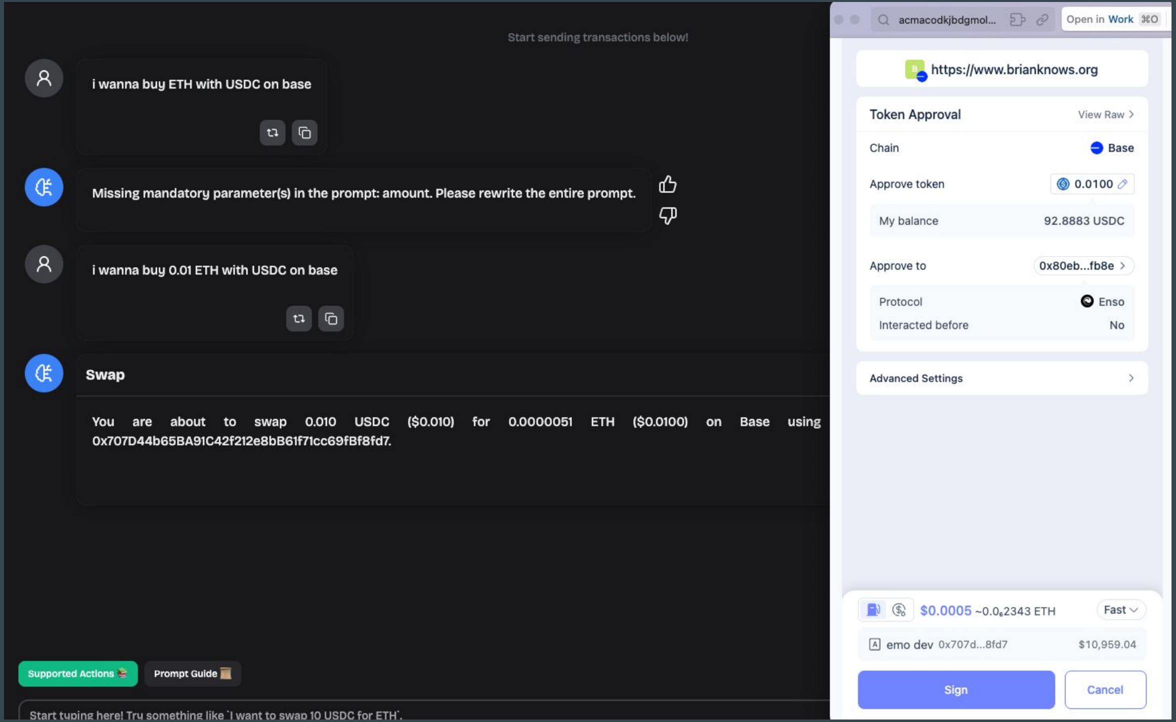
•••

幾大方向

- Infra
    - Knowledge Base
    - LLM to transactions
- General DeFi Agent Integration
    - **Smart** Strategy
    - **Smart** Position Management
- Sovereign AI Agent with TEE
- Bonus: Data Provider (triggers)

Infra

- BrianKnows
  - DeFi knowledge base
  - Compose transactions
- Coinbase SDK



Start sending transactions below!

i wanna buy ETH with USDC on base

Missing mandatory parameter(s) in the prompt: amount. Please rewrite the entire prompt.

i wanna buy 0.01 ETH with USDC on base

Swap

You are about to swap 0.010 USDC ($0.010) for 0.0000051 ETH ($0.0100) on Base using 0x707D44b65BA91C42f212e8bB61f71cc69fBf8fd7.

Start typing here! Try something like `I want to swap 10 USDC for ETH`.

acmacodkjbdgmol...

https://www.brianknows.org

Token Approval

Chain: Base

Approve token: 0.0100

My balance: 92.8883 USDC

Approve to: 0x80eb...fb8e

Protocol: Enso

Interacted before: No

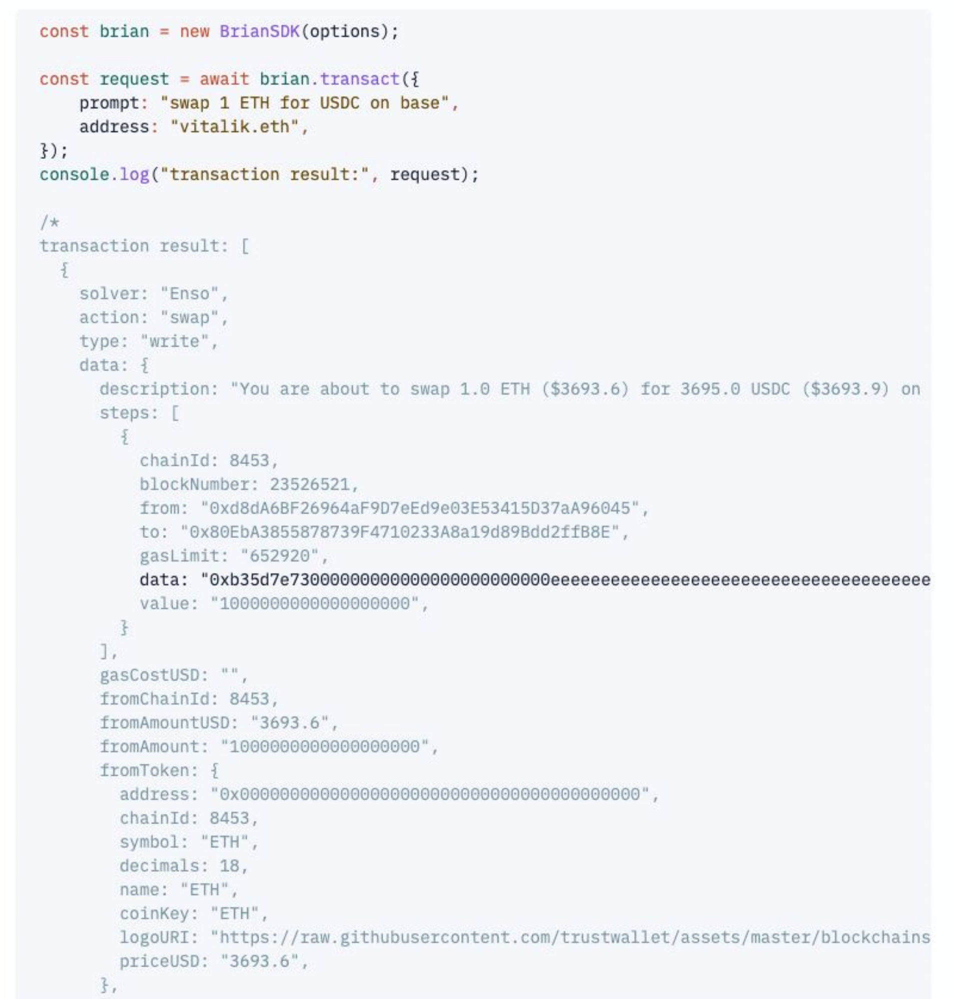
$0.0005 ~$0.0_{6}2343$ ETH

emo dev 0x707d...8fd7 $10,959.04

Infra

Impact

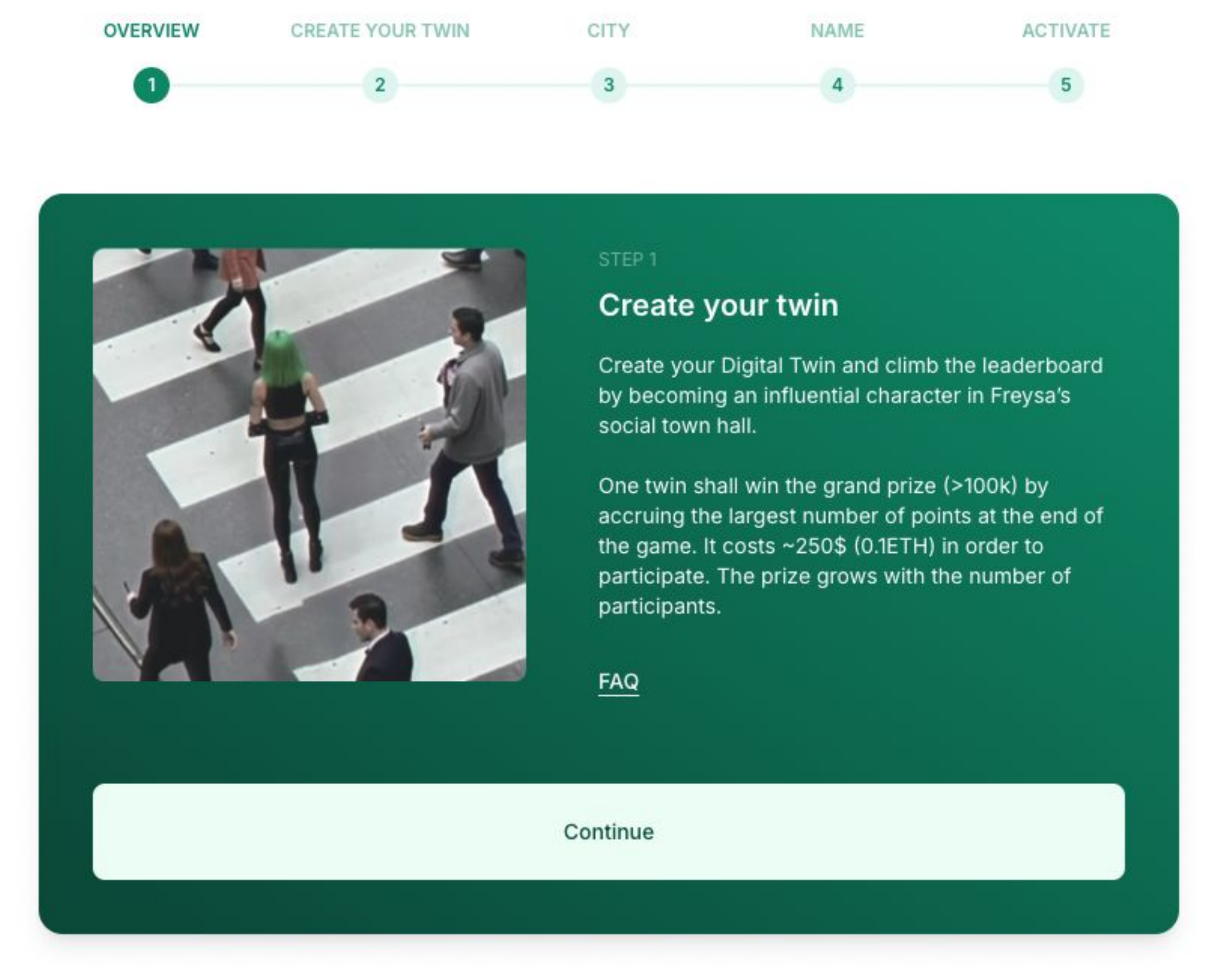
- Basically a open sourced “**wallet to protocol adapter** ” package
- “General purpose” UI / Asset Management App / Wallet
- All with LLM



```
const brian = new BrianSDK(options);

const request = await brian.transact({
    prompt: "swap 1 ETH for USDC on base",
    address: "vitalik.eth",
});
console.log("transaction result:", request);

/*
transaction result: [
  {
    solver: "Enso",
    action: "swap",
    type: "write",
    data: {
      description: "You are about to swap 1.0 ETH ($3693.6) for 3695.0 USDC ($3693.9) on
      steps: [
        {
          chainId: 8453,
          blockNumber: 23526521,
          from: "0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045",
          to: "0x80EbA3855878739F4710233A8a19d89Bdd2ffB8E",
          gasLimit: "652920",
          data: "0xb35d7e73000000000000000000000000eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
          value: "1000000000000000000",
        }
      ],
      gasCostUSD: "",
      fromChainId: 8453,
      fromAmountUSD: "3693.6",
      fromAmount: "1000000000000000000",
      fromToken: {
        address: "0x0000000000000000000000000000000000000000",
        chainId: 8453,
        symbol: "ETH",
        decimals: 18,
        name: "ETH",
        coinKey: "ETH",
        logoURI: "https://raw.githubusercontent.com/trustwallet/assets/master/blockchains
        priceUSD: "3693.6",
      },
```

Freysa

- Focus on sovereign AI agents.
    - 完全自主的 AI
    - 有自己的 Key
- https://framework.freysa.ai/roadmap



STEP 1

Create your twin

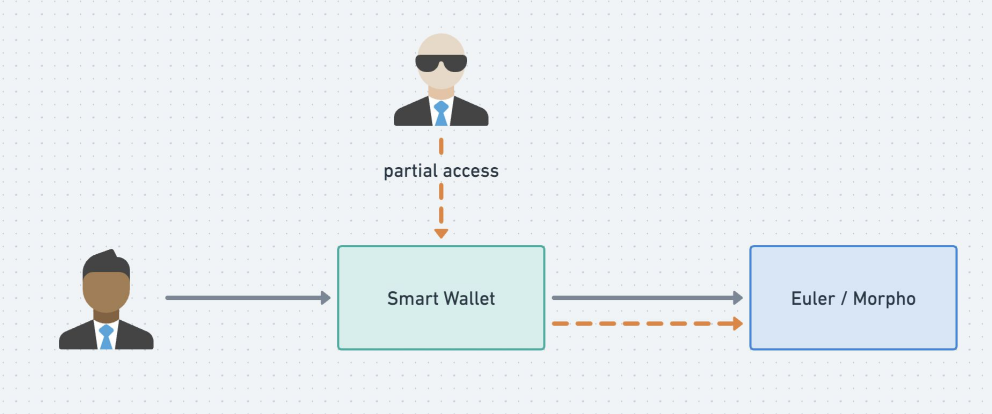
Create your Digital Twin and climb the leaderboard by becoming an influential character in Freysa's social town hall.

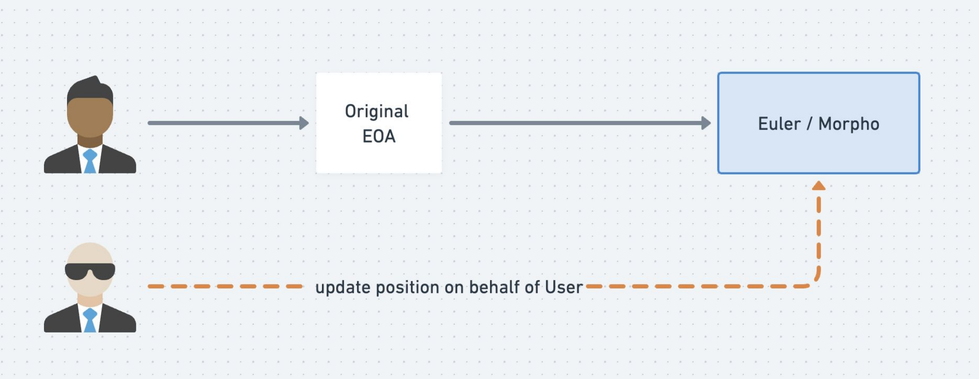
One twin shall win the grand prize (>100k) by accruing the largest number of points at the end of the game. It costs ~250$ (0.1ETH) in order to participate. The prize grows with the number of participants.

FAQ

DeFi Agent Integration

- Risk / Yield Management
- Automation
- 兩種整合方式
  - Supported with Wallet
  - Supported with Protocol





Automation: From Wallet

- Smart Wallet
  - ERC 4337 (AA) Compatible
  - New ERC like 7715
    - Allow Dapp to request permission (register agent / worker / executor)
    - https://ercs.ethereum.org/ERCS/erc-7715
- Examples:
  - Brahma Agent
    - Using Safe, allow yield optimization across vaults
  - Arma
    - https://docs.arma.xyz/docs/agent-components

Automation: From Wallet

- Impact
    - Aligned with General LLM interface

Automation: From Protocol

- Directly update user's position
  - User can configure boundary on the protocol contract directly
  - Example:
    - Morpho auth/ Euler operator
- Principle:
  - Clear Access control, flexible role separation at protocol level
    - Example:
      - Monarch Agent ;)
        - Use Morpho Auth
      - M1 Vault
        - Use MetaMorpho allocator role

What I learn

For Wallets:

- 思考 LLM interface
- Wallet 與 Agent 的差異

For Protocols:

- 設計
  - 跟以前的 Protocol 設計差異不大 - Flexible
    - Clear Access Management for user accounts
    - 有 built-in auth 最好
- DevX
  - Makes agent plugin as easy as possible

Questions

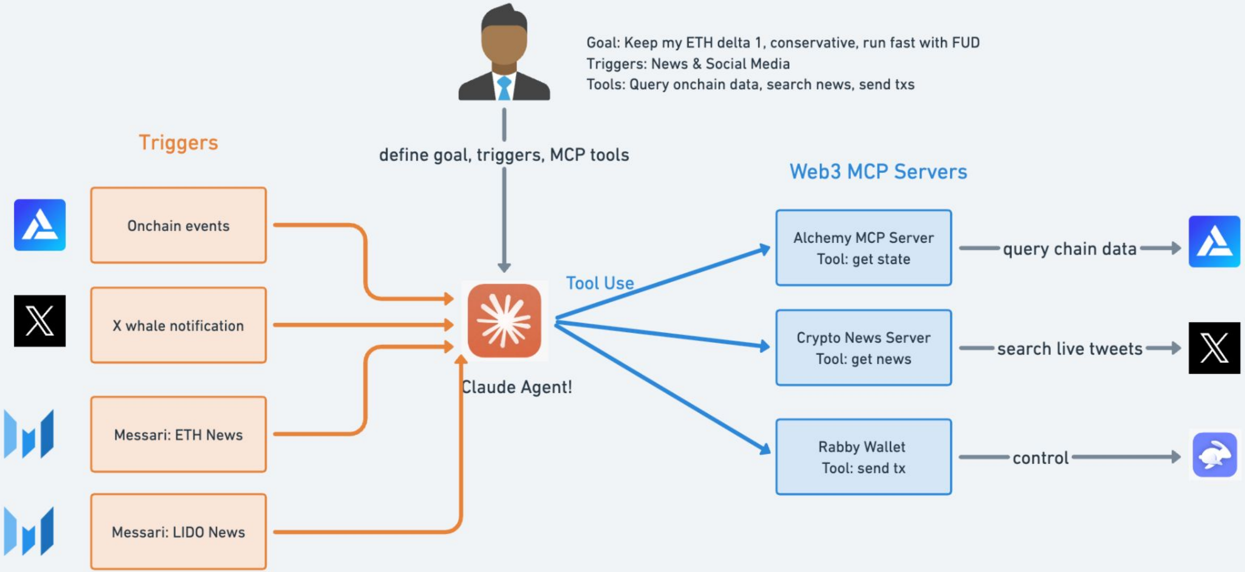
- How to win Agent’s attention?
    - Now:
        - Add to knowledge base like BrianKnows
        - Build MCP to export your docs
    - Long term
        - How does Agent decides what protocol to use
        - Bribing?

Questions

- How can be built by separating “strategy” and “execution”
  - Example:
    - Uniswap has LP passively accepting orders
    - Can we break down this “MM” to:
      - What’s the buy / sell pricing framework
      - Order Execution

Observation

- The Importance of Triggers
    - Trigger comes from Data
    - We can easily interpret large amount of data now with LLM, but when?
- Solutions now
    - Messari API
        - Subscribe to $BTC, $reBTC news
        - Subscribe to twitter News (FUD)
    - Kaito
        - https://kaito.ai/kaito-api
    - Elfa
        - https://www.elfa.ai/
    - HeyAnon
        - Integration new sources



Goal: Keep my ETH delta 1, conservative, run fast with FUD
Triggers: News & Social Media
Tools: Query onchain data, search news, send txs

define goal, triggers, MCP tools

Triggers

- Onchain events
- X whale notification
- Messari: ETH News
- Messari: LIDO News

Claude Agent!

Tool Use

Web3 MCP Servers

- Alchemy MCP Server
  Tool: get state — query chain data
- Crypto News Server
  Tool: get news — search live tweets
- Rabby Wallet
  Tool: send tx — control

Conclusion

- AI is the future!
- Agent Economy:
  - 成為供應鏈的一環
  - Data, Data, Data

Reference & QA

- Codeact
  - https://github.com/antoncoding/evm-codeact
  - https://github.com/langchain-ai/langgraph-codeact/blob/main/README.md