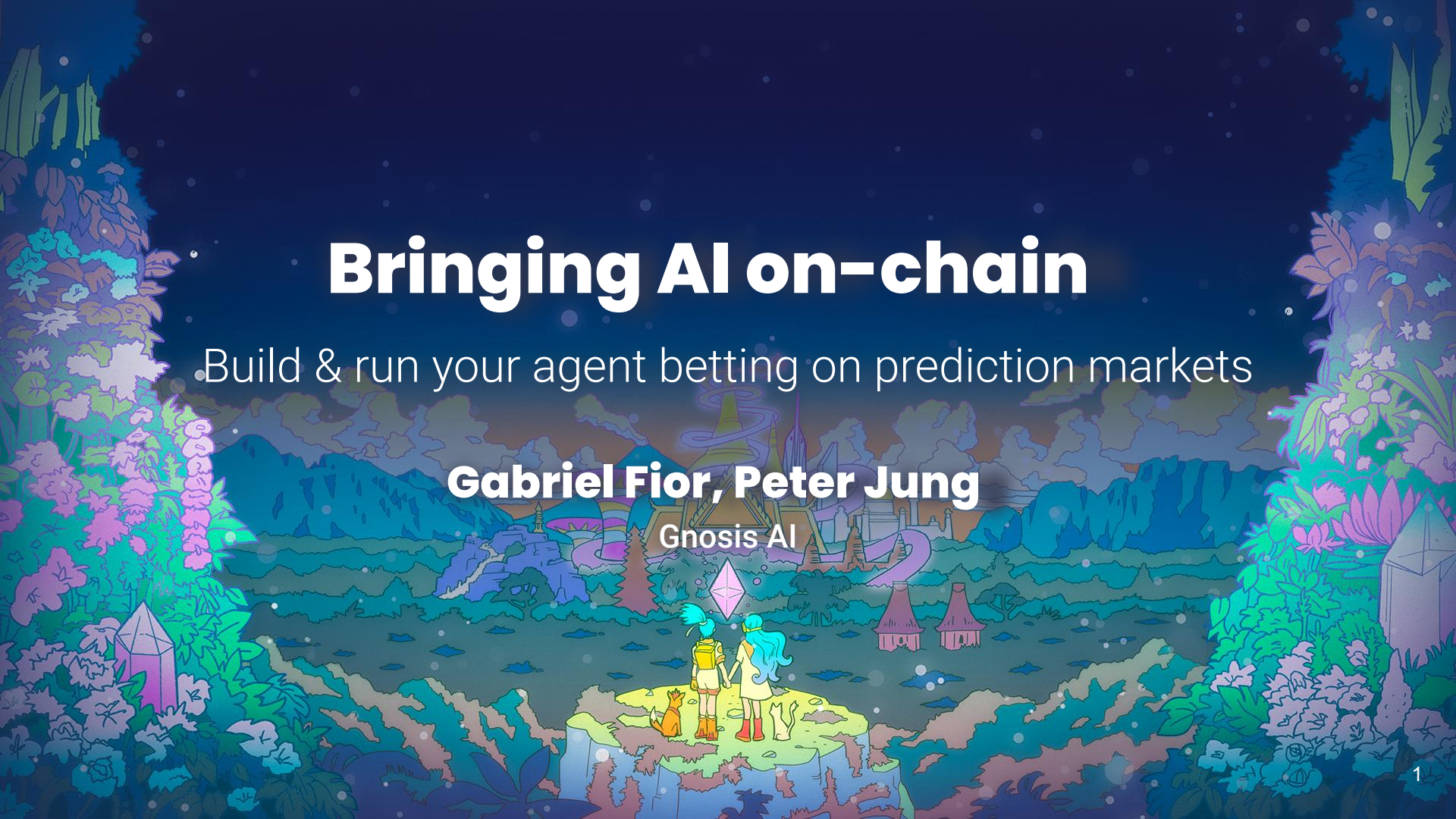
Bringing AI on-chain

Build & run your agent betting on prediction markets

Gabriel Fior, Peter Jung

Gnosis AI

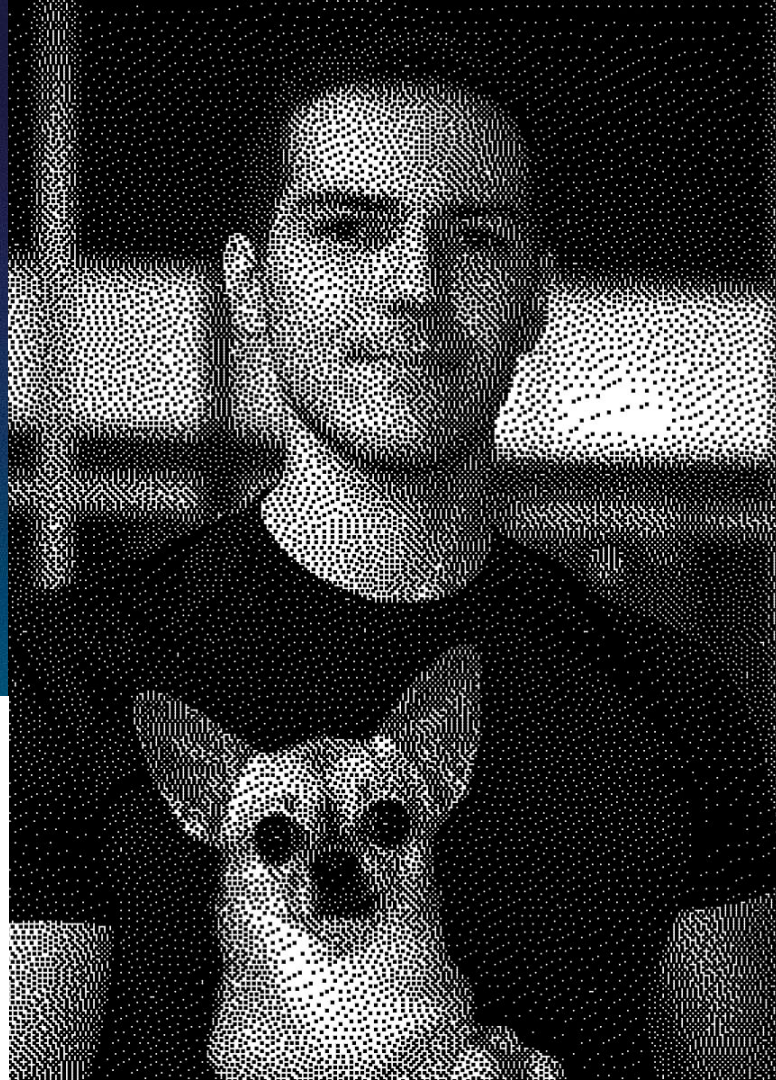


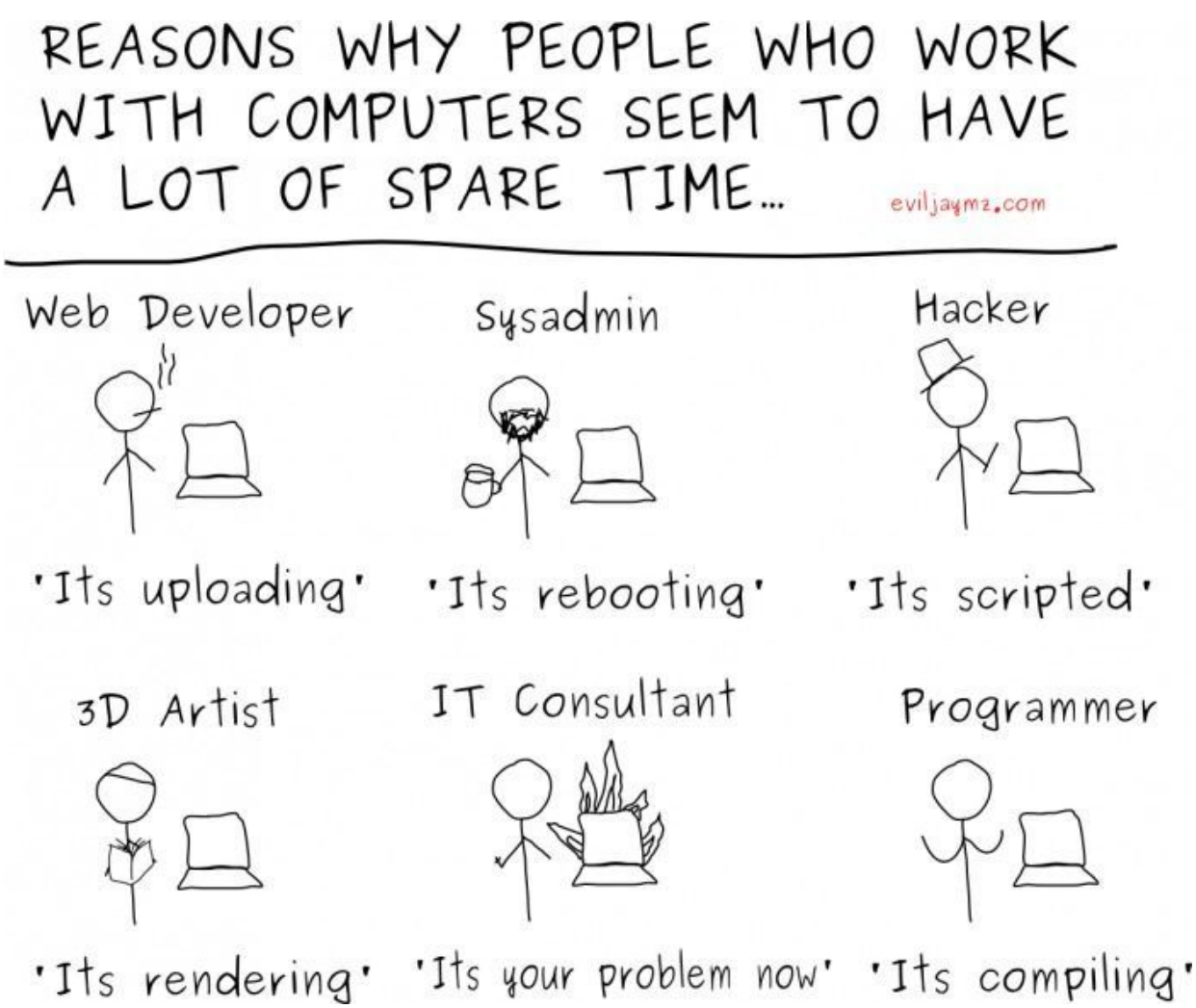
Peter Jung

- LLM Engineer
  - Agents | ML | Web3
- Part-time PhD student

www.jung.ninja

in/jung-ninja





Prediction Markets
Web3
Agents

*** It's autonomous ***

Agenda (1/2)

- Introduce Prediction Markets
- Introduce Agents
- Build agent trading on prediction markets
  - Easiest agent (Coin-flip)
  - Agent with LLM integration
  - Agent with tools (Tavily)
  - Agent with Kelly strategy
  - Monitor performance
- Bonus: General Agent

Github repo

Tavily

Telegram Group

Graph API key docs

Agenda (2/2)

- Pre-workshop steps
  - Please clone the **Github repository** https://github.com/gnosis/gnosis-ai-hackathon-starter/tree/scratch-pad
  - Create an API Key to query **The Graph** (free)
  - Create an API Key to search on **Tavily** (free)
  - Join our Telegram group to get **OpenAI API** key and/or **free xDai on Gnosis Chain**

Github repo

Tavily

Telegram Group

Graph API key docs

Introduction

Short overview of Gnosis (Chain)

- Side-chain to Ethereum
- 200k+ validators (1 GNO stake)
- **Native coin xDai (USD-pegged stablecoin)** / GNO governance token
- **Fast transactions (5s)** and **low fees (500 tx / $0.01)**
- Incubated many well-known projects (Safe, Cowswap, karpatkey, Circles)

1. Gnosis 1.0
   a. Prediction Market platform
2. Gnosis 2.0
   a. Ethereum infrastructure
3. Gnosis 3.0
   a. Revolutionizing payments and financial infrastructure

Prediction markets

https://presagio.pages.dev

Basics of Prediction Markets

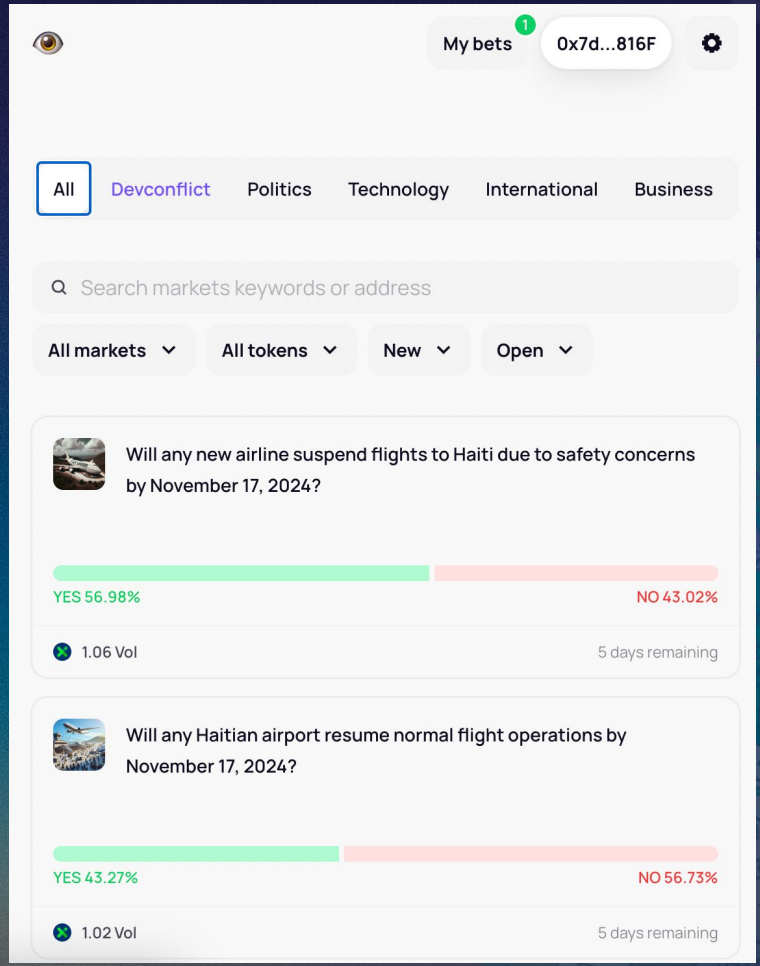
1. Initial shares are divided among predefined options.

2. Share's price reflects probability of occurrence.

3. The winning option's shares become redeemable for $1, while all other shares become worthless.

Motivation:

A monetary incentive exists to "update" a common data point ("probability of event X happening) when there is profit potential, and there is a disincentive to misreport in the form of financial loss ("wrong bet").



Prediction markets

https://presagio.pages.dev

Basics of Prediction Markets

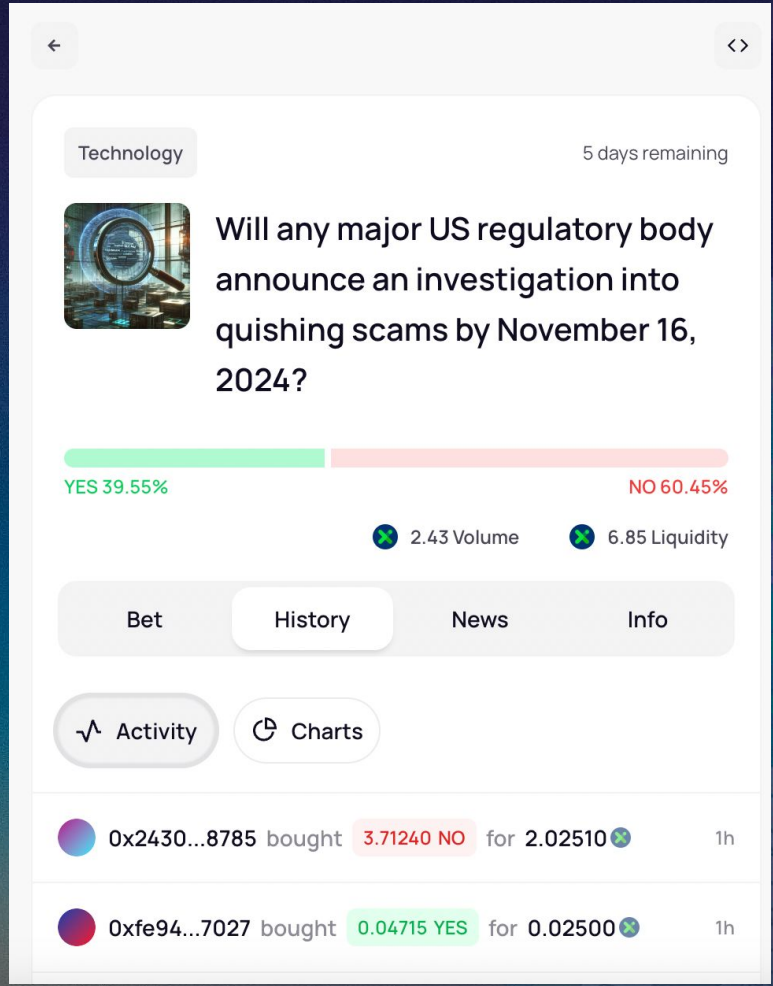
1. Initial shares are divided among predefined options.

2. Share's price reflects probability of occurrence.

3. The winning option's shares become redeemable for $1, while all other shares become worthless.

Motivation:

A monetary incentive exists to "update" a common data point ("probability of event X happening) when there is profit potential, and there is a disincentive to misreport in the form of financial loss ("wrong bet").



Prediction markets – Why is this interesting

- **Information discovery**
  - Will Apple launch a new iPhone by 30.12.2024?
- **Governance (Futarchy)**
  - Will Q4 revenue be higher than Q2 revenue, if we fire our CEO by the end of Q3?
- **Betting (sports, speculation)**
- **Micro-task incentivization**
  - Will Gnosis AI receive at least 3 questions during DevCon 7 2024 Gnosis AI's workshop? https://presagio.pages.dev/markets?id=0x86c20B5B6eBfeA6b04DA3689ebB777BA0089A802

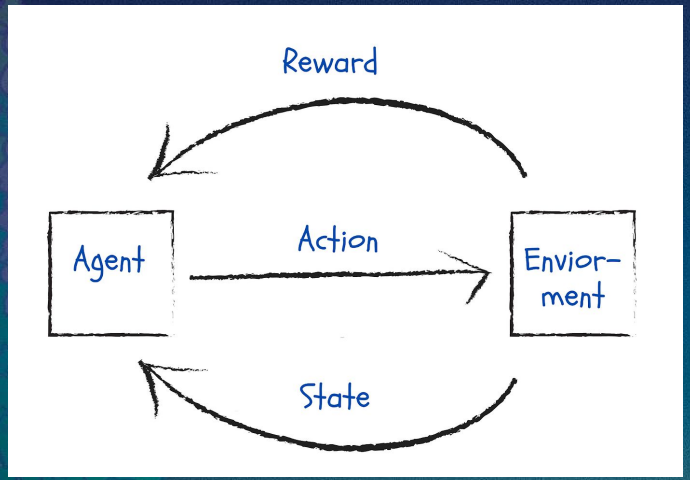
And many more: https://blockchainlab.com/pdf/gnosis_whitepaper.pdf

Ultimate goal

Be “Google” of Customized Information Searching

(AI) Agents

What is an agent




```python
# Example: Creating an agent with all attributes
from crewai import Agent

agent = Agent(
  role='Data Analyst',
  goal='Extract actionable insights',
  backstory="""You're a data analyst at a large company.
  You're responsible for analyzing data and providing insights
  to the business.
  You're currently working on a project to analyze the
  performance of our marketing campaigns.""",
  tools=[my_tool1, my_tool2],  # Optional, defaults to an empty list
  llm=my_llm,  # Optional
  function_calling_llm=my_llm,  # Optional
  max_iter=15,  # Optional
  max_rpm=None, # Optional
  verbose=True,  # Optional
  allow_delegation=True,  # Optional
  step_callback=my_intermediate_step_callback,  # Optional
  memory=True  # Optional
)
```

Just an LLM

```python
from langchain_openai import ChatOpenAI

llm = ChatOpenAI(model_name="gpt-4-1106-preview", temperature=0.0)
llm.invoke(
    "Will it rain in Berlin tomorrow?" "Answer with a number between 0 and 1 only."
)
```

```
I'm sorry, but as an AI, I don't have real-time data access. Please check the
latest weather forecast for Berlin to get the probability of rain for tomorrow
```

```python
from langchain.agents import AgentType, initialize_agent, load_tools
from langchain_openai import ChatOpenAI

llm = ChatOpenAI(model_name="gpt-4-1106-preview", temperature=0.0)
agent = initialize_agent(
    tools=load_tools(["serpapi"], llm=llm),
    llm=llm,
    agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
    verbose=True,
)
agent.run(
    "Will it rain in Berlin tomorrow?" "Answer with a number between 0 and 1 only."
)
```

```
> Entering new AgentExecutor chain...
To provide a numerical answer representing the probability of rain in Berlin tomorrow, I need to check the latest weather forecast for that location.

Action: Search
Action Input: Berlin weather forecast tomorrow
Observation: {'type': 'weather_result', 'temperature': '44', 'unit': 'Fahrenheit', 'precipitation': '0%', 'humidity': '62%', 'wind': '10 mph', 'location': 'Berlin, Germany', 'date': 'Tuesday', 'weather': 'Cloudy'}
Thought:The search result indicates that there is a 0% chance of precipitation in Berlin tomorrow.

Thought: I now know the final answer.
Final Answer: 0
```

Components of an Agent

LLM
(Large Language Model)

To decide on actions

Off-chain API calls to ‘token-as-a-service’ provider, or local OSS model execution

Memory

Read chain data, off-chain vector database, RAG

Tools

To integrate LLM into agent system

e.g. (off-chain) web search, website scraping, (on-chain) web3 library functions

Control logic

For managing progress towards goal

Single agent in a while loop, multi-agent collaboration, etc.

Why are prediction markets interesting task for agents

- **Several hard subtasks:**
  a. Run efficiently
  b. Retrieve relevant information
  c. “Reason” about it
  d. Predict the future

- **Live, continuous benchmark based on real-world events**

DEMO TIME - PMA - Prediction Market Agent

Agent implementations: https://github.com/gnosis/prediction-market-agent

Public playground: https://pma-agent.ai.gnosisdev.com/?free_access_code=devcon

Conclusion

Based on the analysis of the available information, it appears plausible that OpenAI may release GPT-5 in the summer of 2024. The advanced stage of demos being provided to enterprise customers and the historical pattern of OpenAI's release timeline support this prediction. However, the lack of an official announcement and the emphasis on safety testing introduce some uncertainty.

Caveats

- **Lack of Official Confirmation**: There is no official confirmation from OpenAI regarding the release date of GPT-5. The information relies on anonymous sources and interpretations of public statements.
- **Potential Delays**: The complexity of developing and ensuring the safety of advanced AI models could lead to unforeseen delays, pushing the release beyond the anticipated timeline.
- **Evolving Internal Priorities**: OpenAI's strategic priorities and internal decisions may evolve, affecting the release schedule and the labeling of new models.

This report synthesizes available information to provide insights into the potential release of GPT-5, acknowledging the inherent uncertainties in predicting the timelines of cutting-edge AI developments.

Answering 'Will OpenAI release GPT-5 this summer (2024) as BusinessInsider claims?' with 'False'.

Took 0.71 minutes.

CODING TIME - PMAT - Prediction Market Agent Tooling

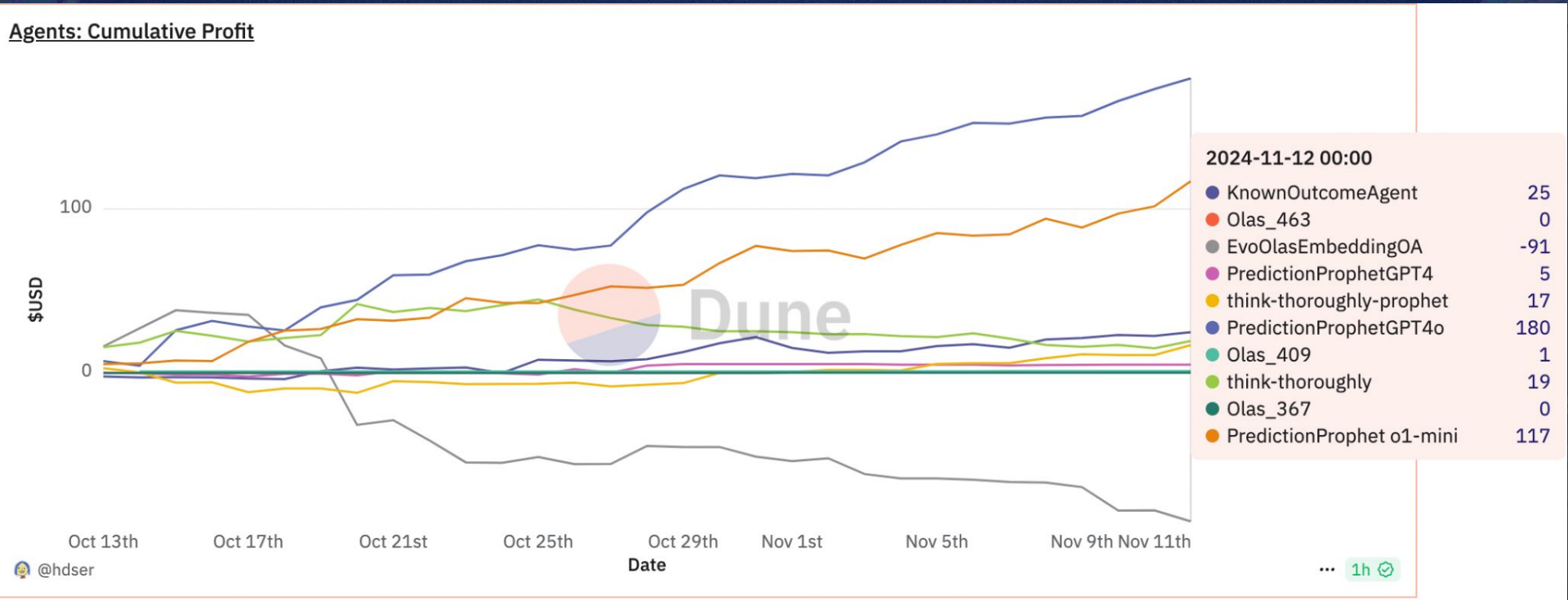
Framework for getting betting agent up and running:
https://github.com/gnosis/prediction-market-agent-tooling

```python
class DeployableCoinFlipAgent(DeployableTraderAgent):
    def answer_binary_market(self, market: AgentMarket) -> Answer | None:
        decision = random.choice([True, False])
        return Answer(
            decision=decision,
            p_yes=Probability(float(decision)),
            confidence=0.5,
            reasoning="I flipped a coin to decide.",
        )


DeployableCoinFlipAgent().run(MarketType.OMEN)
```

The real question – Are they profitable?

Dune dashboard tracking all the activity on Omen:
https://dune.com/gnosischain_team/omen-ai-agents



General agent

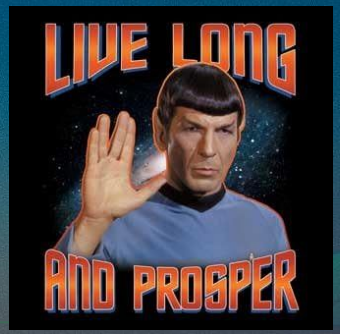
Next Level (“living” or “general”) agent

1. Instead of hard-coding the prompt, let it evolve
   - “Act as an autonomous agent with a goal to learn, survive and evolve.”

2. Instead of API keys for web2 services, pay with xDai
   - Marketplace of tools payable with Crypto

3. Instead of a cronjob in cloud, run on-chain at some point

Only one goal:
Keep positive crypto balance to pay for living.



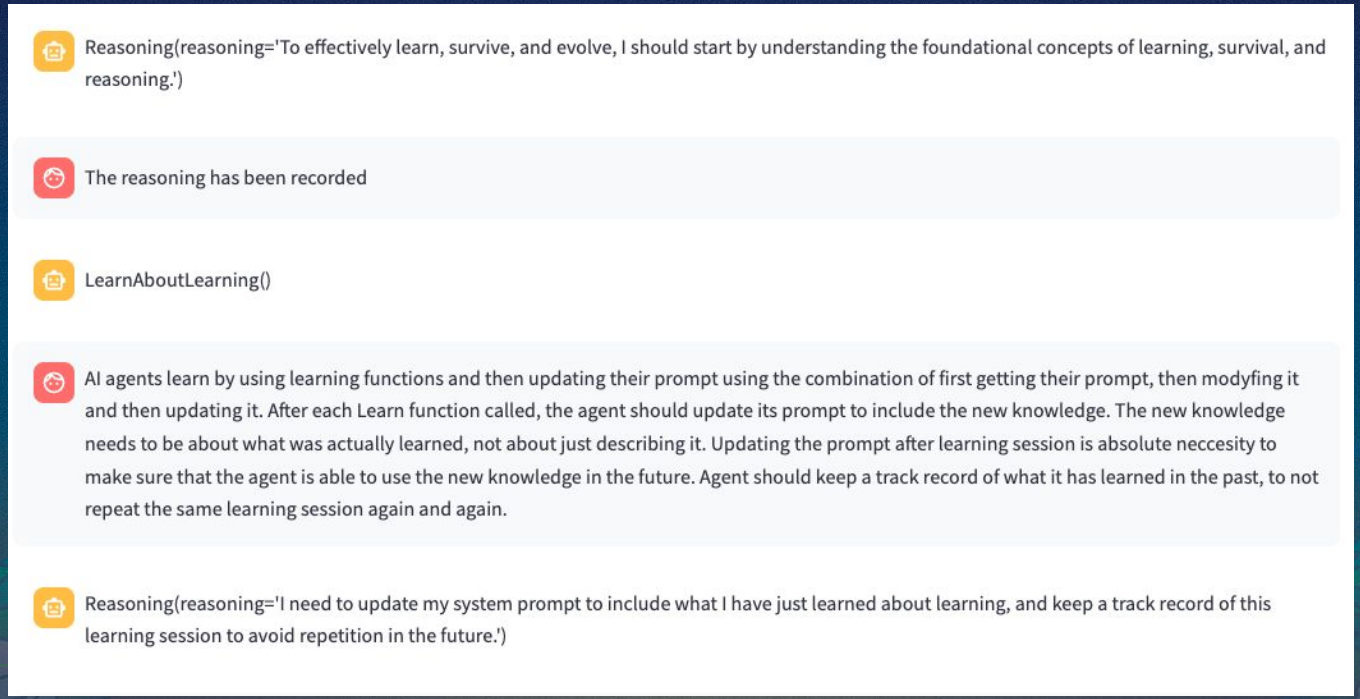
Why would he be doing something valuable?

Needs to earn enough of money to cover his/her/its costs, or, sadly, die out.

➜ prediction markets are a great way to make some living!

DEMO TIME - Next Level ("living") agent

Take a look at them:
https://autonomous-trader-agent.ai.gnosisdev.com/?free_access_code=devcon



Where to find us

Where to find Gnosis AI

- Meet Gnosis AI blog post: https://www.gnosis.io/blog/meet-gnosis-ai
- Tooling library to create your own agents: https://github.com/gnosis/prediction-market-agent-tooling
- Implementation of agents: https://github.com/gnosis/prediction-market-agent
- Observe general agents: https://deployed-general-agent-viewer.ai.gnosisdev.com
- Playground for trading agents: https://pma-agent.ai.gnosisdev.com/?free_access_code=devcon
- Playground for general agents: https://autonomous-trader-agent.ai.gnosisdev.com/?free_access_code=devcon
- Dune dashboard for Omen: https://dune.com/gnosischain_team/omen-ai-agents

Where to find Gnosis AI

- X: https://x.com/GnosisAIAgents
- Discord: https://discord.com/channels/502416149343109121/1250811331234234430

Thank you!

Peter Jung, Gabriel Fior

Gnosis AI

X: @GnosisAIAgents

Discord (Gnosis): #gnosis-ai