

# **GraphAI/Grapys**

## **Full-Stack TypeScript Tool for AI Applications**

Isamu Arimoto

# 自己紹介

- Isamu Arimoto
  - Software engineer
    - Server/Web
- Singularity Society / TNE
  - SlashGPT/GraphAI

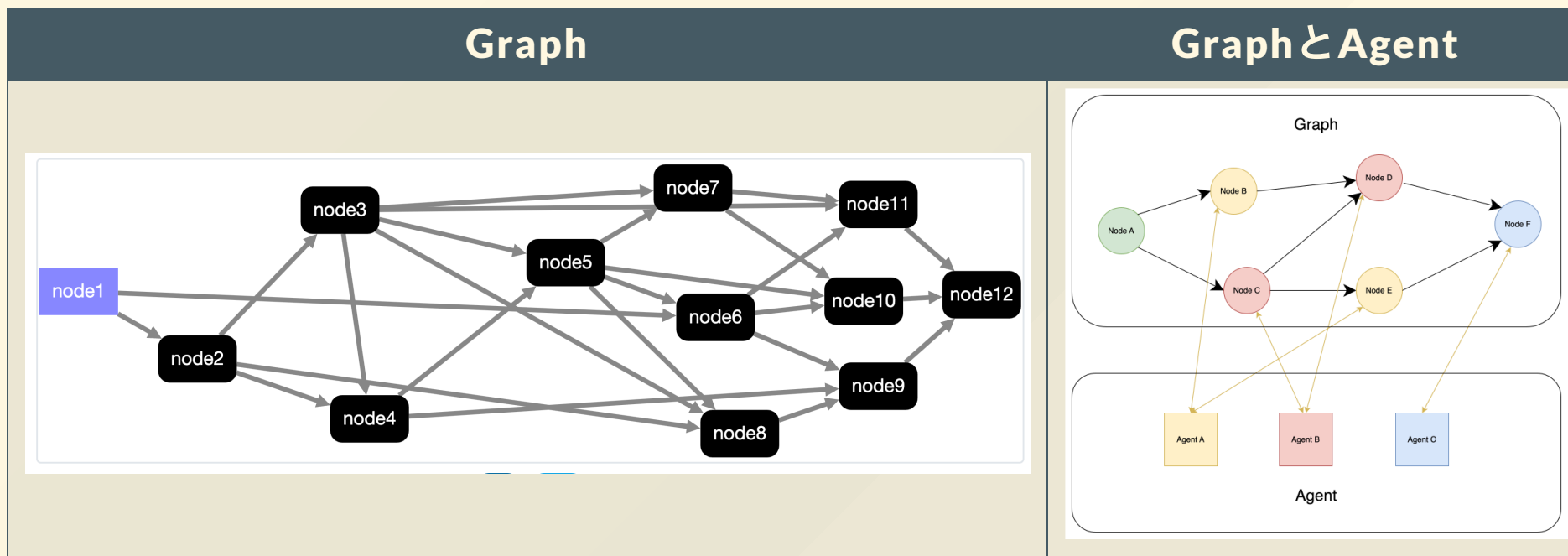
# Contents

- 内容
  - GraphAI とは
  - Grapys/GraphAIのデモ

# GraphAI とは

- TypeScriptで書かれたMulti agent フレームワーク
- サーバ/ブラウザ/それらの混在環境でも動く
  - ワークフロー(GraphData)と Agent の実装(TypeScript)が分離
  - Agentをブラウザ、サーバの任意の場所で動かせる
  - local llm(ollama)直接使える
  - ワークフローの途中でユーザとのインタクシヨン可能
  - Stream を透過的にサポート
  - Graphのネスト/Map/Loopをサポート
  - ツールが多数ある
- Agentがスキーマなどのメタ情報を持つ
- Agentサーバ
  - Agent情報一覧を返して動的に利用可能
  - SPECを合わせれば、どんなendpointでもagent化(SSEサポート)

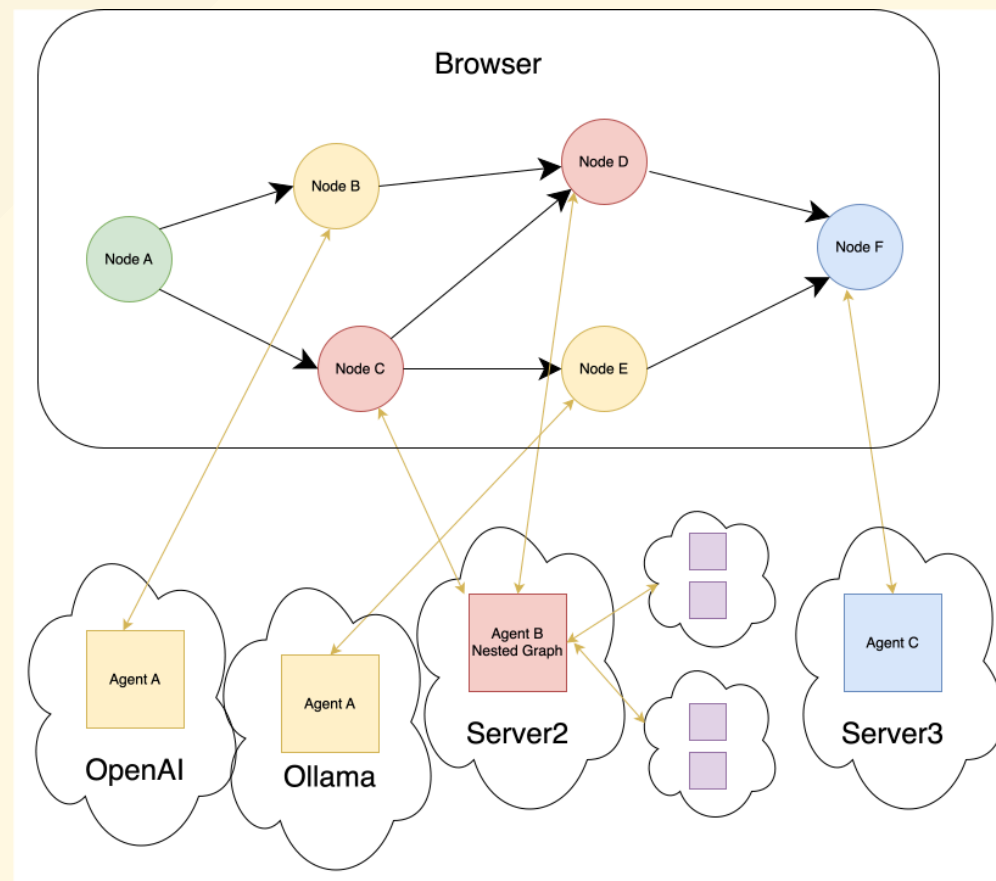
# GraphAI と Agent



- 本体はシンプル
  - Graphの制御のみ/YAML, JSONでGraphの定義
  - Agent - LLM/WebAPI/データ処理/IO
  - Agent Filter - 各Agentを実行する前後に処理を挟む -> Agentの実行場所を分離できる

# GraphAI 構成例

- Node A - ブラウザ上のJS
- Node B - openAI APIを直接呼び出し
- Node E - localのollama
- Node C, DはServerのAgent(NestedAgent)を呼び出す
- Node FはServerのAgent



# Demo

- Grapys
  - <https://github.com/receptron/grapys>
- Grapys-demo
  - <https://github.com/receptron/graphai-demo-web/>
  - TOP
    - random (大きなグラフ)
    - stream2 (並列stream)
    - sample(loop)
  - google map
  - google video
- ◦ [https://github.com/receptron/graphai-utils/tree/main/packages/event\\_agent\\_generator](https://github.com/receptron/graphai-utils/tree/main/packages/event_agent_generator)

# Tool

- express
  - agentサーバ
    - streamサポートSSE
    - .well-known/agent.jsonのような仕組み
  - graphサーバ
  - chat completion api
- agent
  - mcp
  - tools
- agent\_generator



## GUIでのGraphの構築

ブラウザの入力とGraphのNodeを同期させて必要な箇所でブラウザイベント  
Toolsと連携してブラウザ操作  
非同期Streaming

express

stream系

event系

各種llm/データ処理

Node.js(サーバとブラウザの両方で動く)

ワークフローとAgentの実装の分離

ワークフローで呼び出すNode(Agent) をNode単位で実行先を指定できる

- ブラウザで動かす
- サーバで動かす
- localのollamaを呼ぶ

Agentが、

API Specを持っている

Agentのメタデータを持っていて、web apiで一覧を取れる

SSRFやREST APIで呼び出せる