Building Human-like AI Agents

Solving challenges around Computer Interaction, Long-Term Memory, and Agent-Agent Communication

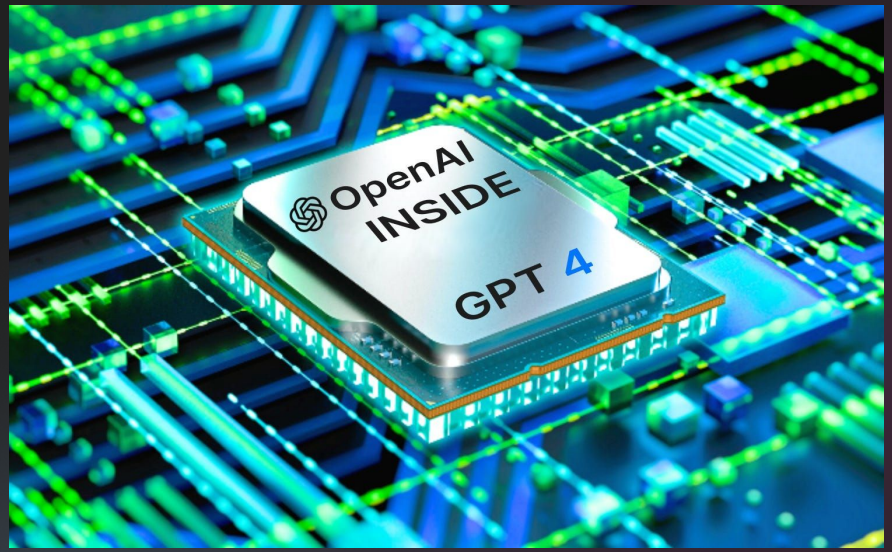
Div Garg (MultiOn AI)

Introduction

- Actions and Emergent Agent Architectures
- Building Human-like AI Agents
- Computer Interactions using AI
- Long-Term Memory and Personalization
- Agent to Agent communication
- Future Directions for Autonomous AI Agents

Building AI Agents

1. Why?
2. How?
3. Ingredients?
4. What can they do?



Key thesis: Humans will communicate with AI using natural language and AI will operate machines allowing for more intuitive and efficient operations

Software 3.0

AI Agents

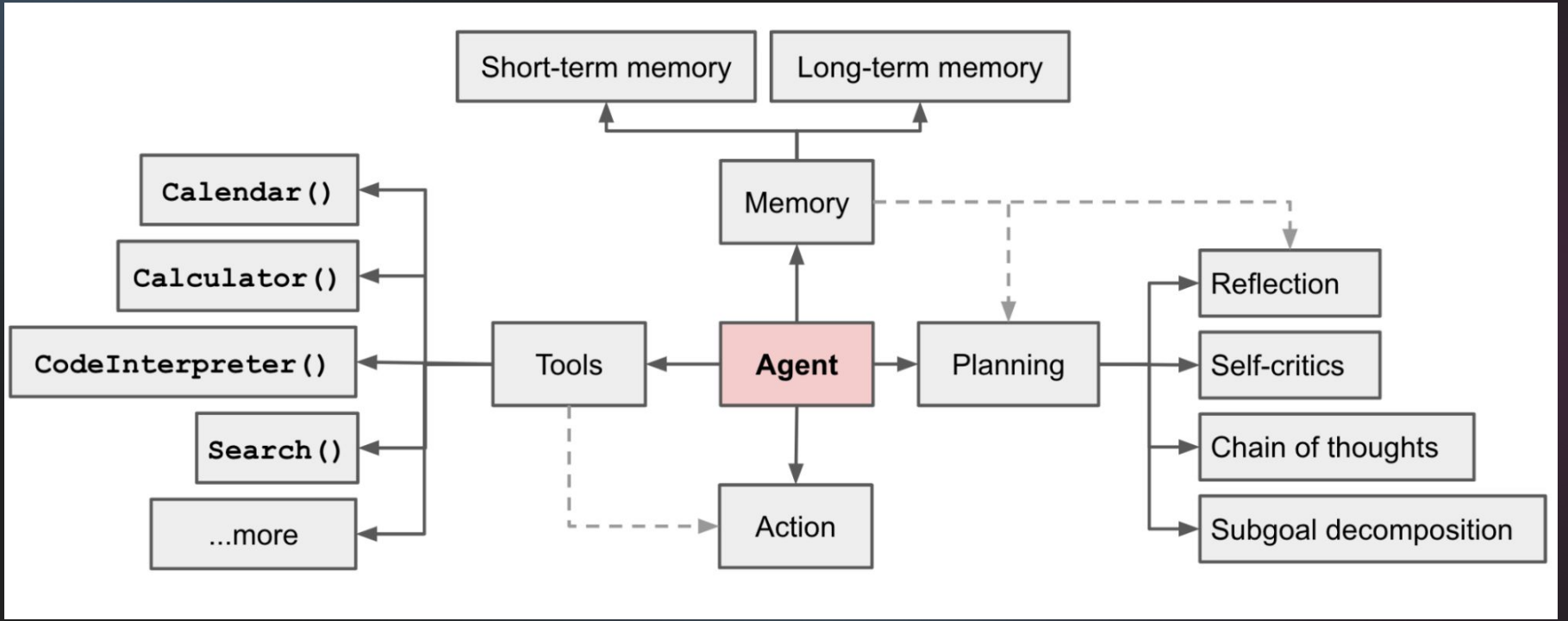
1. **Why?**
   A single call to a large Foundation AI model is not enough. A lot more can be unlocked by building ***AI systems***

2. **How?**
   Using model chaining, reflection & other mechanisms

3. **Ingredients?**
   Memory, context length, personalization, actions, internet access…

4. **What can they do?**

AI Agents


Short-term memory
Long-term memory
Memory
Calendar()
Calculator()
CodeInterpreter()
Search()
...more
Tools
Agent
Planning
Action
Reflection
Self-critics
Chain of thoughts
Subgoal decomposition

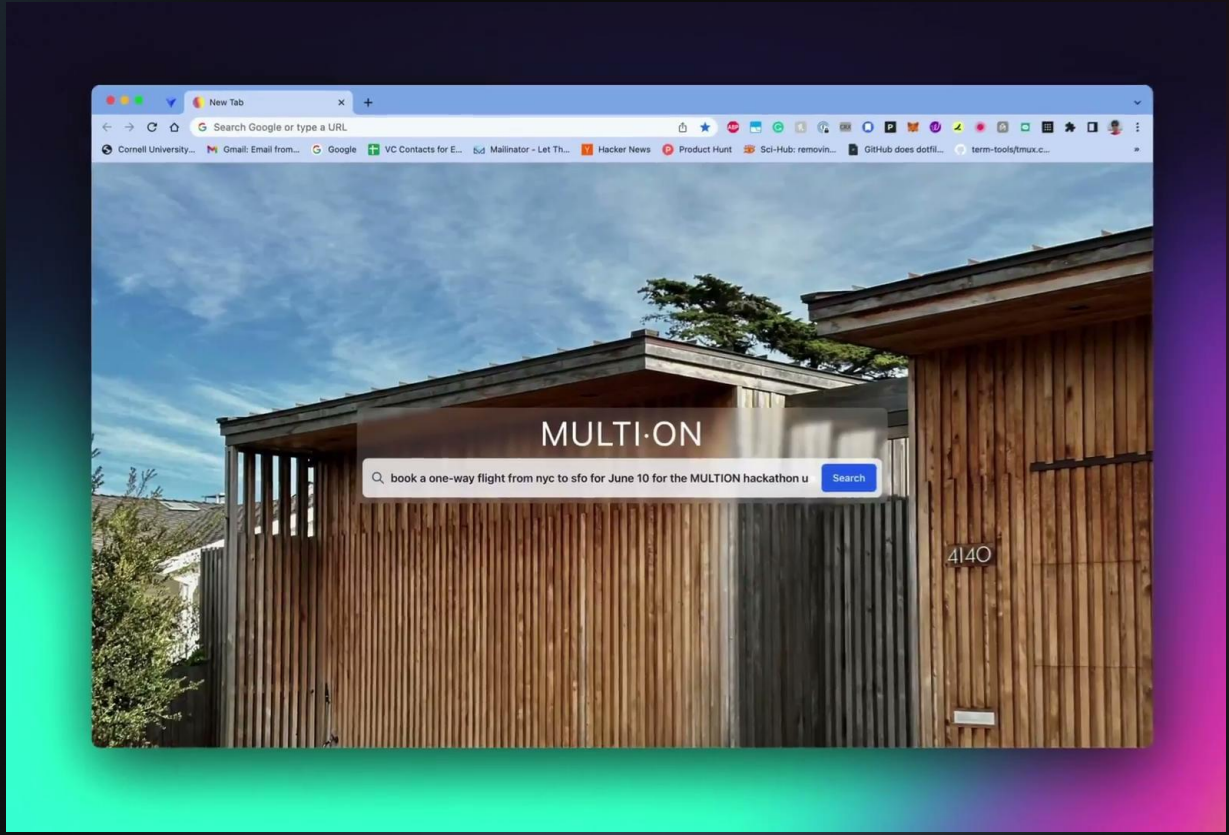
MULTI·ON

- **M**ultiple **U**tility AI **L**anguage **T**ool for **I**nternet **O**perations and **N**avigation (MULTION)

- Agent with full R/W access to the internet

Name Origin: from Quantum Physics (neutron, muon, fermion, ...)

The first flight to be fully booked by an AI



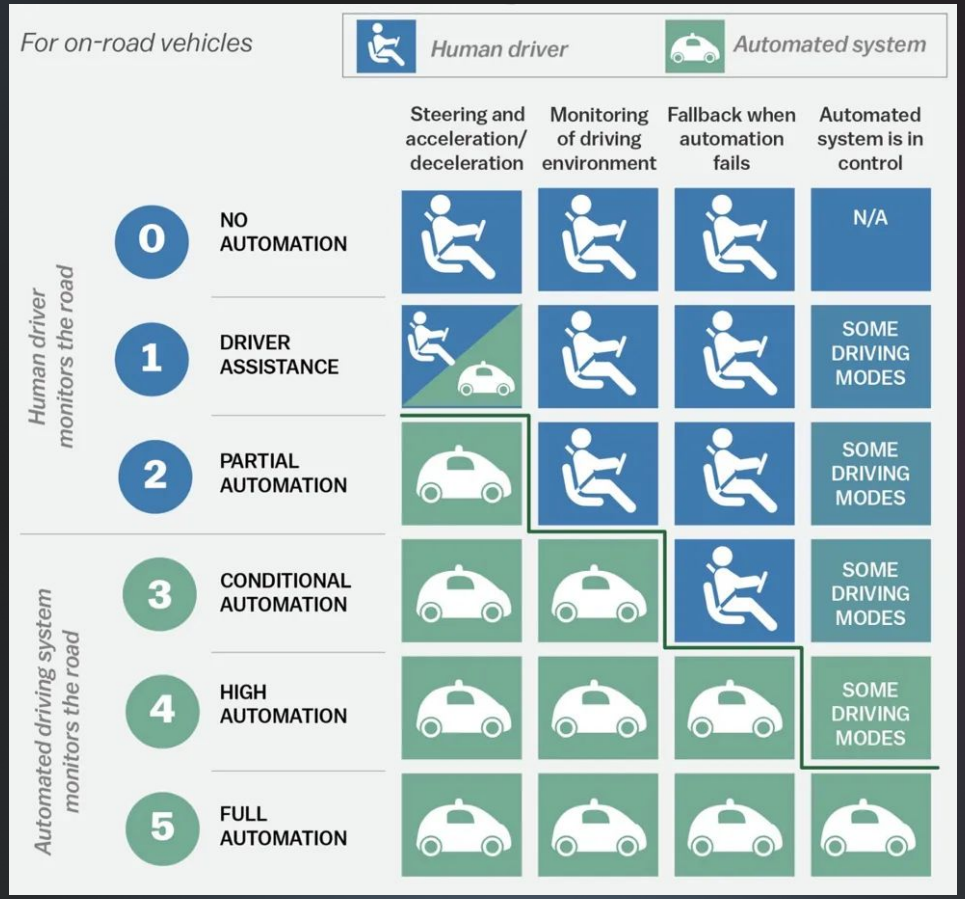
MULTI·ON – Mobile Platform

Can be present on any device & provide seamless interaction to the AI using a natural human interface to efficiently get your tasks done for you!

Why **human-like** AI Agents?

1. **Can do what you can do**: Able to use existing interfaces designed for humans & operate outside programmatic boundaries
2. **Digital extension of you**: Can act as an extension of the user and act on their behalf
3. **Less-restrictive boundaries**: Can handle logins, payments, etc. and interact with services without any API restrictions
4. **Simple action space**: Need only click & type action primitives
5. **Self-learning**: Can learn from the user and self-improve with more interactions

5 levels of Autonomy



Computer Interactions

Agent Computer Interaction

Two routes

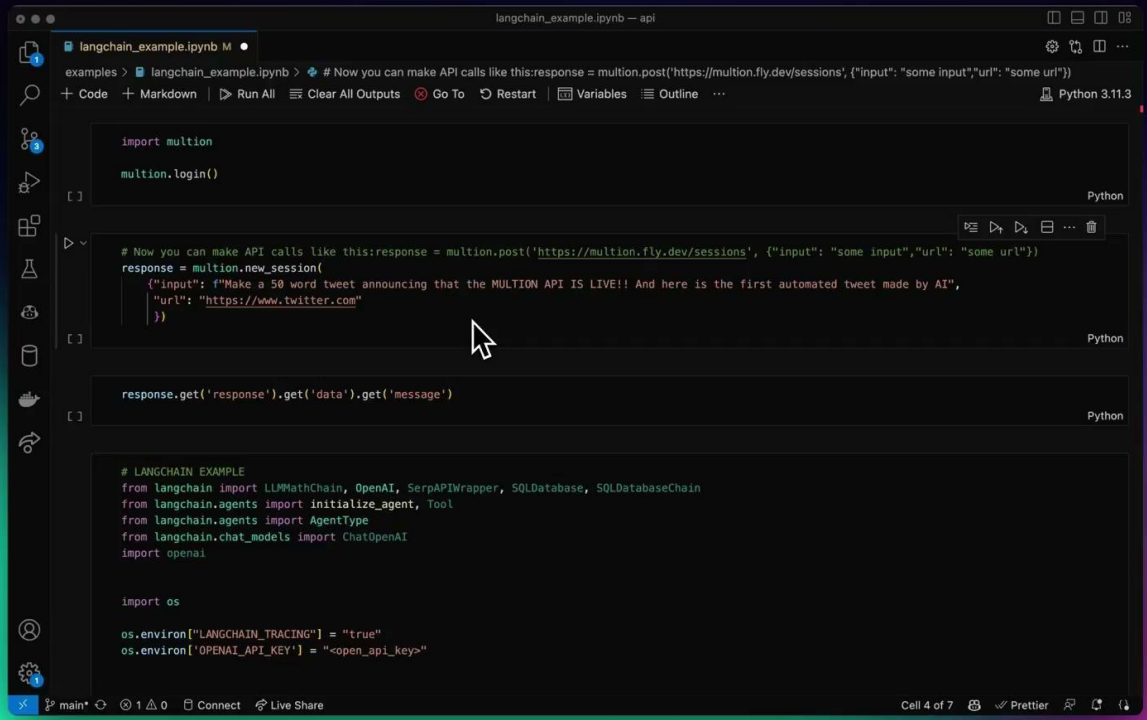
API (programmatic)

easy to build context
safer & controllable
high variability

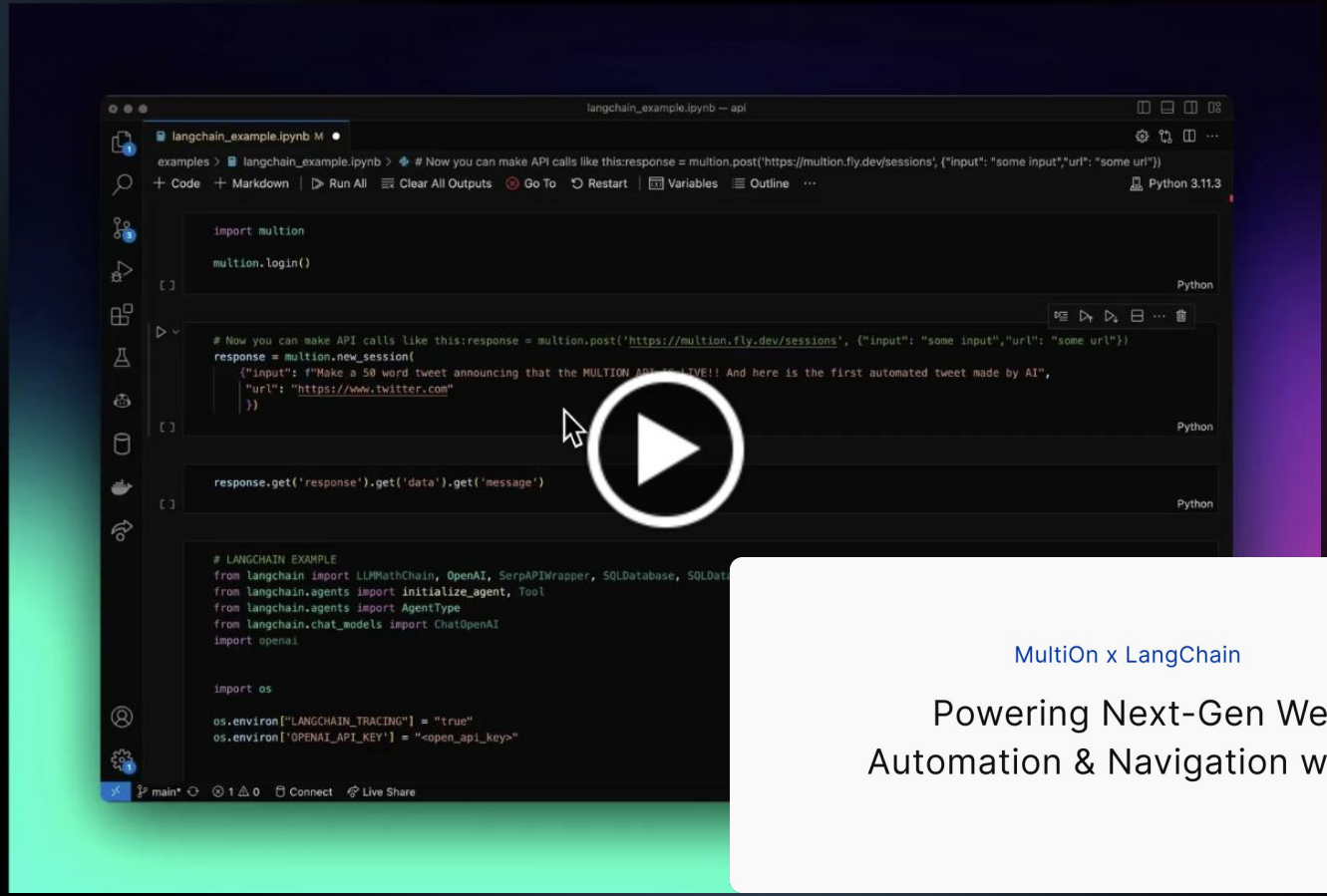
Direct interaction (browser or desktop control)

easy to take actions
free-form interactions
need to provide guarantees

MultiOn Action API: An API for computer interaction



MultiOn Action API: An API for computer interaction

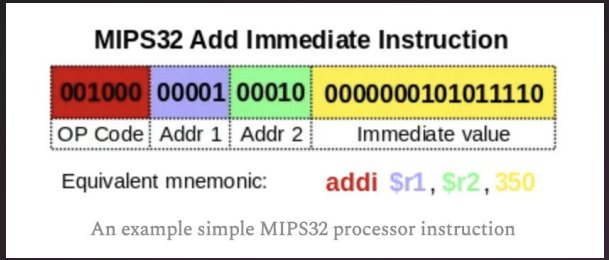


MultiOn x LangChain

Powering Next-Gen Web Automation & Navigation with AI

Memory & Personalization

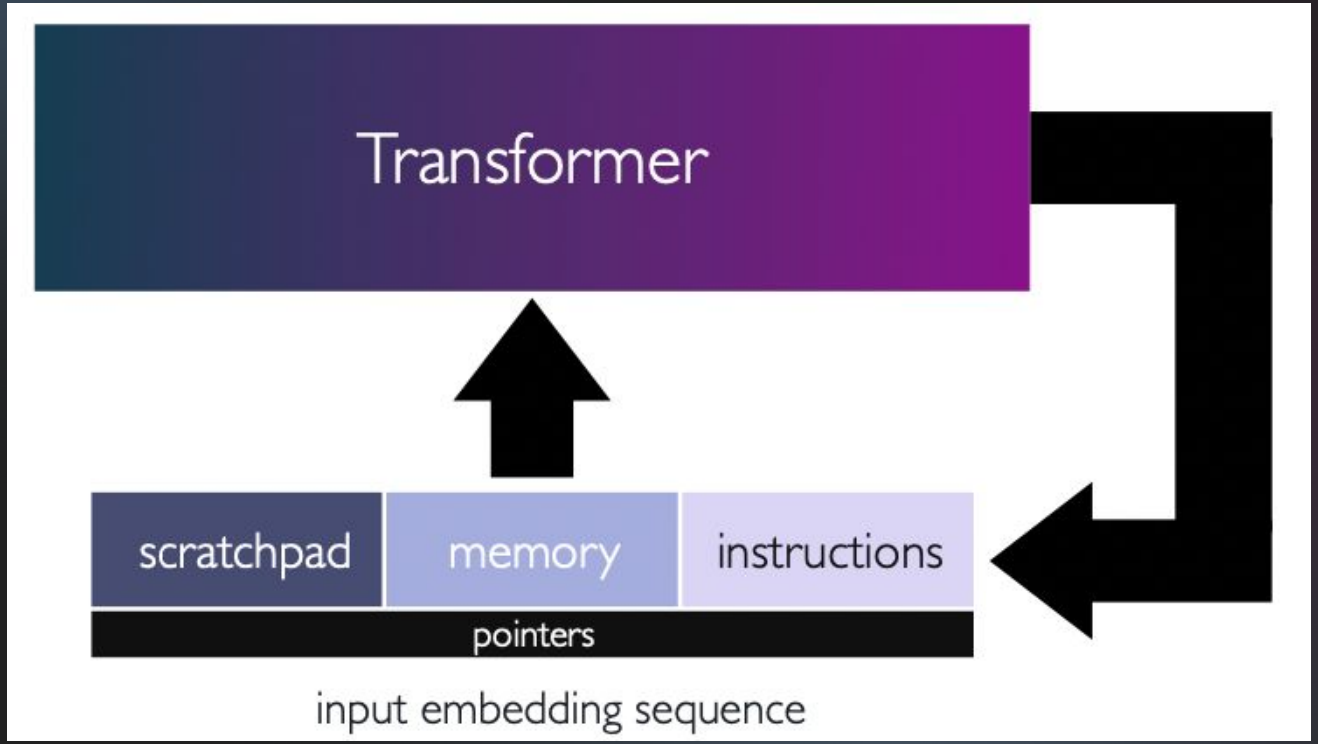
AI Models as Neural Compute Unit



Input Tokens
(max token size)

Output Tokens
(max token size)

AI Models as Neural Compute Unit



Looped Transformers

Long-term Memory

- **Works similar to disk (long-lived & persistent)**

- **Mechanisms**
  - Embeddings
  - Retrieval models

- **Open Questions:**
  - Hierarchy
  - Temporal Coherence
  - Structure
  - Online adaptation

Personalization

- **Agent-user Alignment:** Enable agent to take actions that are aligned with the user preferences

- Everyone has different prefs & likes/dislikes:
  - **Explicit**: allergies, favourite dishes, flight seat prefs, …
  - **Implicit**: choice between brands, out of 10 items in a listing which user likes better

Challenges

- **Collecting user data & preferences:**
  - actively asking for preferences
  - passive learning from interactions

- **Learning from user preferences:** supervised fine-tuning vs human-feedback

- **On-fly adaptation**

- **Privacy**

Agent-to-Agent Communication

Multi Agent Autonomous AI systems





Why Multi-agent Systems

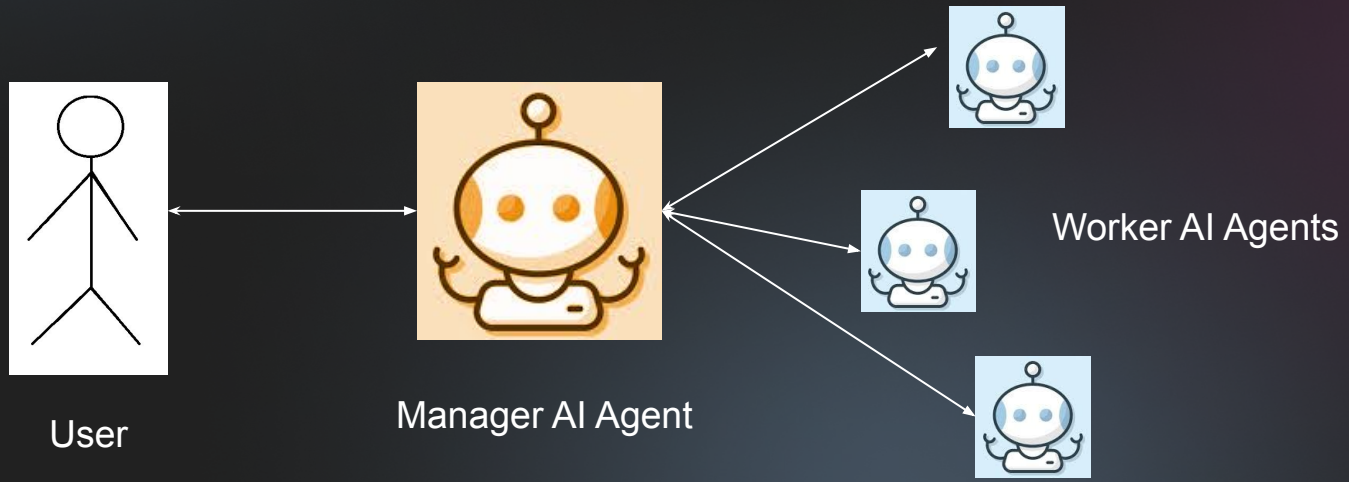
1. **Parallelization unlock**: Breaking a task into smaller chunks and dividing between agents to improve efficiency & speeds

2. **Task Specialization**: An AI agent might seat between the user and each service: e.g. a spreadsheet AI agent, a slack AI agent, a web-browser AI agent, …

3. **Challenges**:
    a. **Agent to Agent Communication**: one AI might want to exchange or request info from another AI agent finish a task

Agent to Agent Communication

- **Exchanging info between fleets of agents**
- **Hierarchies**
- **Syncing primitives**



Agent to Agent Communication

- **Robust communication protocols & Syncing primitives:** Natural language is ambiguous, need mechanisms to reduce miscommunication!

Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**

Manager state

Task X:
(status: not done)

Manager AI Agent

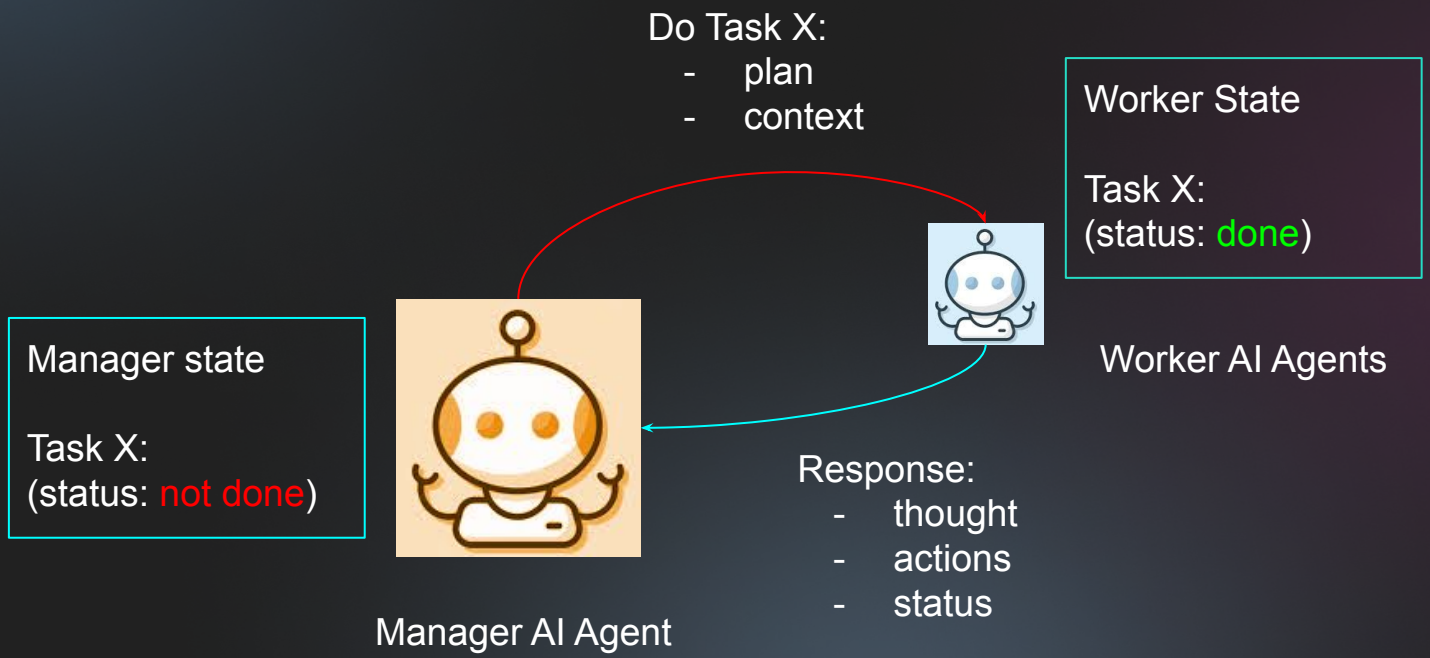
Worker AI Agents

Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**

Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**



Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**

Verify if task was correctly done & follows all specifications

Verify Task X:
- required spec

Worker State

Task X:
(status: done)

Manager state

Task X:
(status: verify done)

Manager AI Agent

Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**

Scenario 1:
Task was correctly done & follows all specifications

Verify Task X:
- required spec

Worker State

Task X:
(status: done)

Manager state

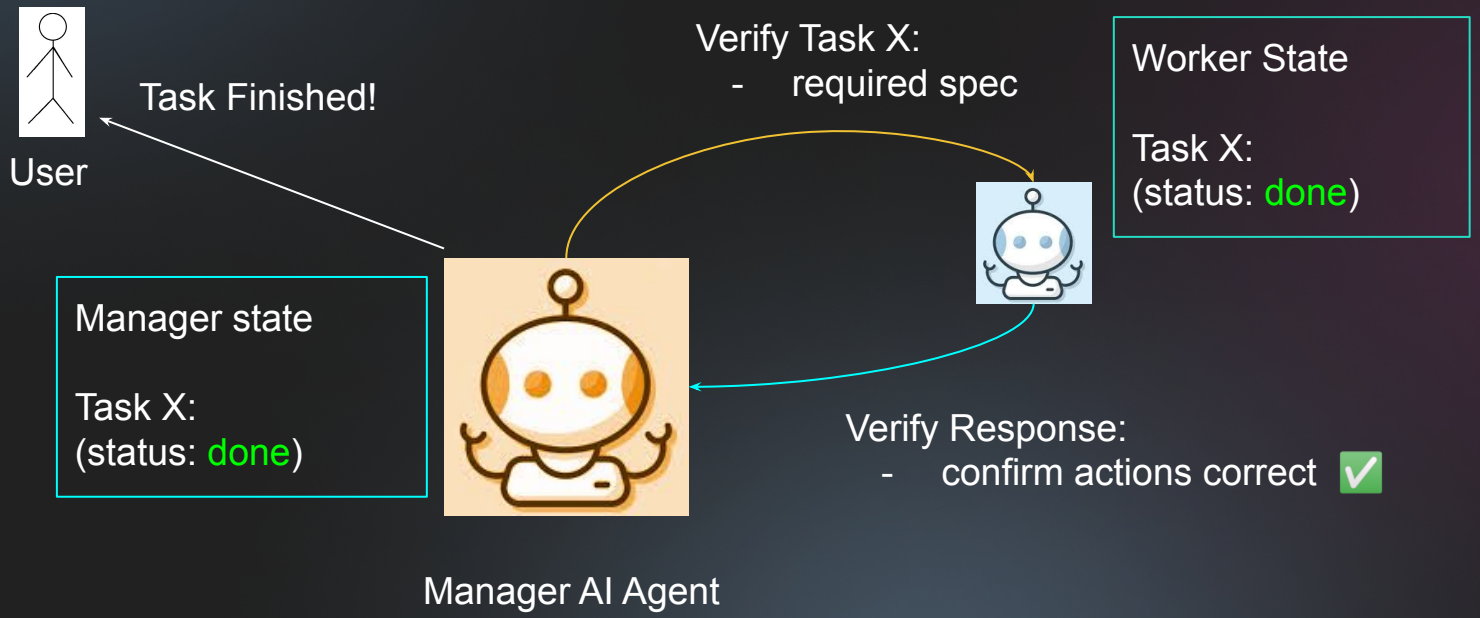
Task X:
(status: verify done)

Verify Response:
- confirm actions correct ✅

Manager AI Agent

Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**



Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**

Scenario 2:
Task was incorrectly done
(Agent Miscommunication)

Verify Task X:
- required spec

Worker State

Task X:
(status: done)

Manager state

Task X:
(status: verify done)

Verify Response:
- actions were not correct ❌

Manager AI Agent

Agent to Agent Communication

- **Robust communication protocols & Syncing primitives**

Scenario 2:
Task was incorrectly done
(Agent Miscommunication)

Re-do Task X:
- plan
- context
- feedback/corrections

Worker State

Task X:
(status: not done)

Manager state

Task X:
(status: not done)

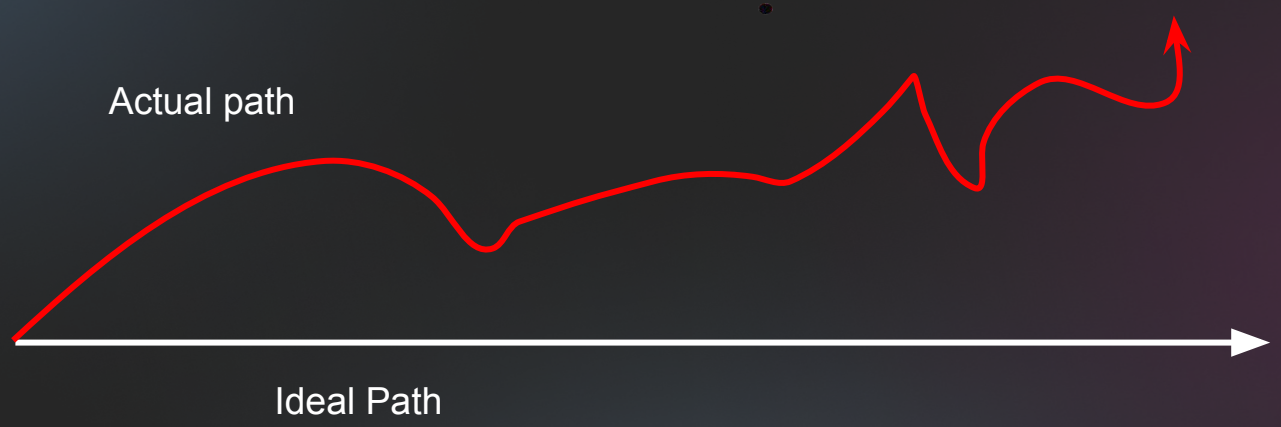
Manager AI Agent

Future Directions

Key Issues with Autonomous Agents

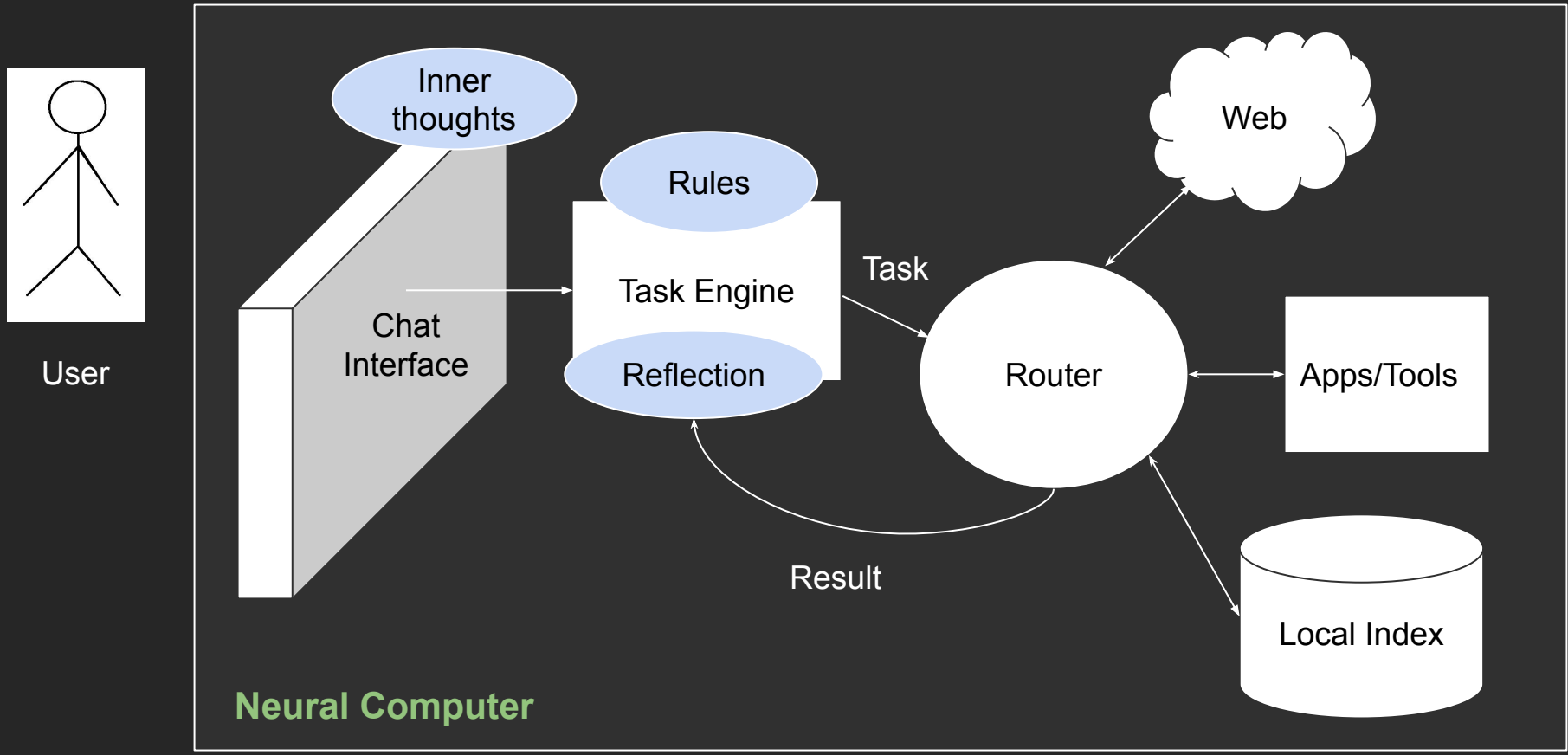
1. **Reliability**
2. **Looping & Plan Divergence**
3. **Testing & Benchmarking**
4. **Real world-deployment & Observability**
    a. How do we trust a fully autonomous AI system
    b. How do we build in human overrides

Plan Divergence



AI Agents like AutoGPT don’t know how to correct on making a mistake!

Building Generalized AI Systems



Future needs for AI agents

- Error correction mechanisms & better agent frameworks
- Security & user permission models
- Sandboxing & deployment in risky settings

Thank you!