

What Are AI Agents?

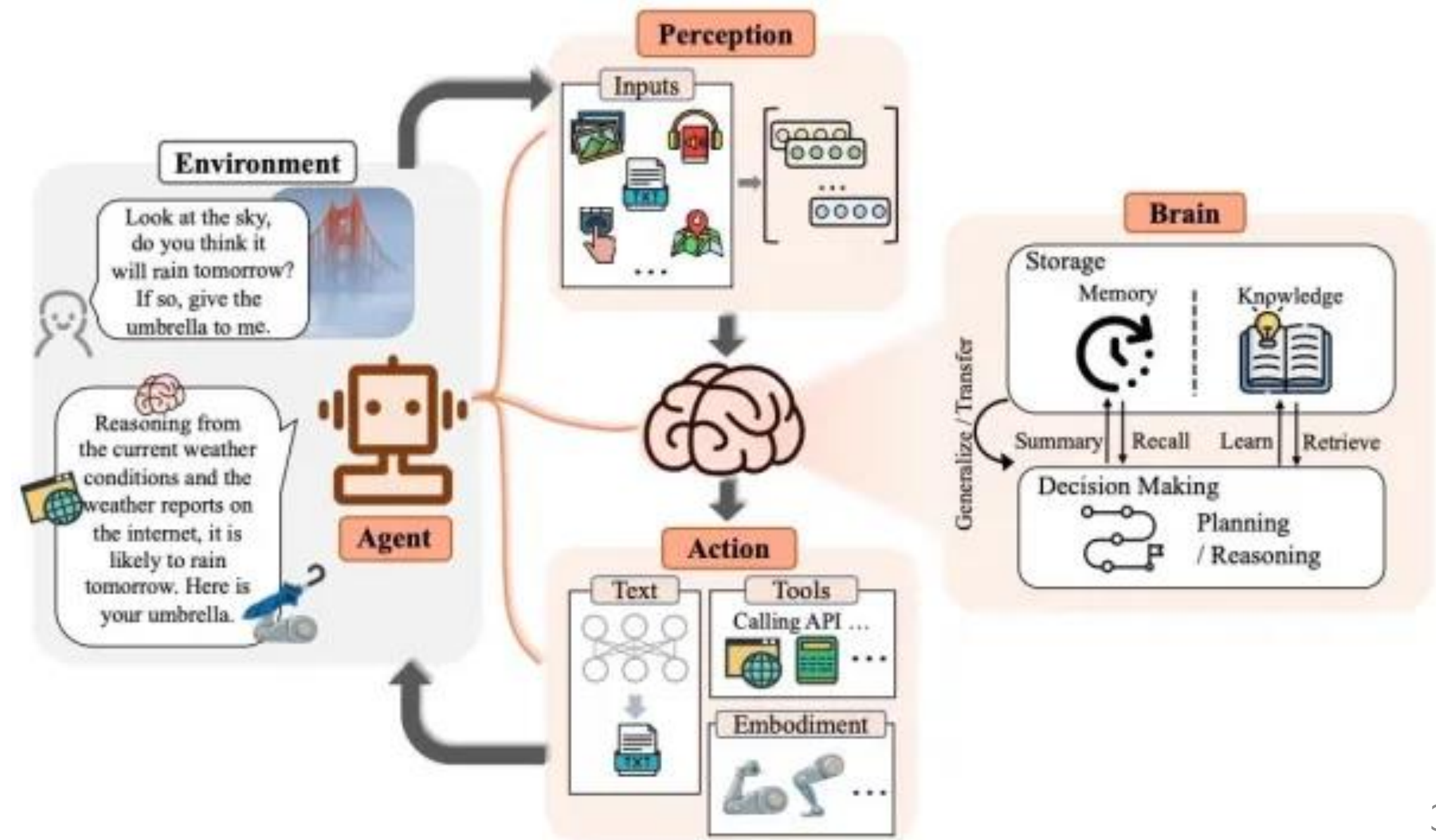


AI 에이전트 개요



What Are AI Agents?

AI 에이전트는 자율적으로 작업을 수행하는 인공지능 시스템입니다. 주어진 목표를 달성하기 위해 데이터를 수집하고, 분석하며, 의사결정을 내립니다. 환경과 상호작용하며 센서를 통해 데이터를 수집하고, 지능을 활용해 분석합니다. 그리고 행동을 통해 결정을 실행합니다.



기존 소프트웨어 vs AI 에이전트

```
Require: A matrix A of size m x n
1: for i in m do
2:   | for j in n do
3:     | | if i = j then
4:       | | | Select a random action
5:     | | else
6:       | | | if i = j + 1 then
7:         | | | | Stay silent
8:       | | | else
```

전통적인 명령형 프로그래밍

(Imperative Programming) 방식에는
사전에 정의된 실행 경로

(Predetermined Execution Paths)를
따릅니다.

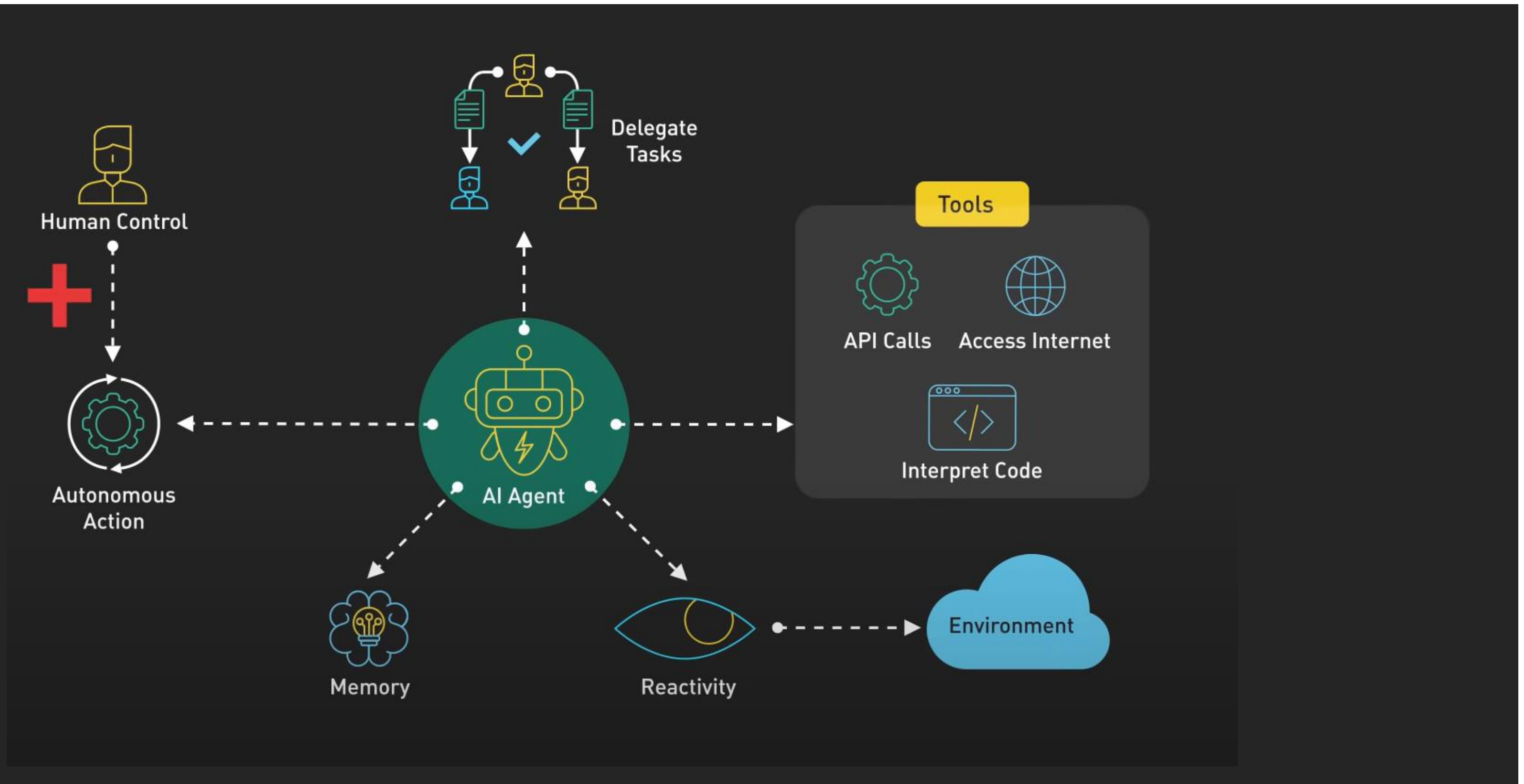


목표 지향 선언형 프로그래밍 (Declarative Goal Setting) 방식의
AI 에이전트는 다음과 같은 과정을 거쳐 동적으로 작동합니다.

1. 환경을 감지: 입력 및 센서를 통해 주변 환경을 지속적으로 모니터링
2. 정보 처리: 수집된 데이터를 분석하고, 추론 엔진을 사용하여 의사결정 수행
3. 목표 기반 행동 수행 : 설정된 목표와 가능한 행동을 고려하여 적절한 조치를 실행
4. 환경 변화 적응: 행동을 통해 환경을 수정하고, 지속적으로 적응
5. 피드백 학습: 실행 결과를 피드백으로 받아들이며 성능을 향상

AI 에이전트

AI 에이전트는 기존 소프트웨어 시스템 설계 방식에서 근본적인 변화(Fundamental Evolution)를 가져오고 있습니다. 이러한 패턴을 이해하면 기존의 명령형 프로그래밍을 넘어, 스스로 사고하고 학습하는 지능형 시스템을 구축할 수 있습니다.



AI 에이전트 주요 기능 - 자율성 (Autonomy)

AI 어시스턴트



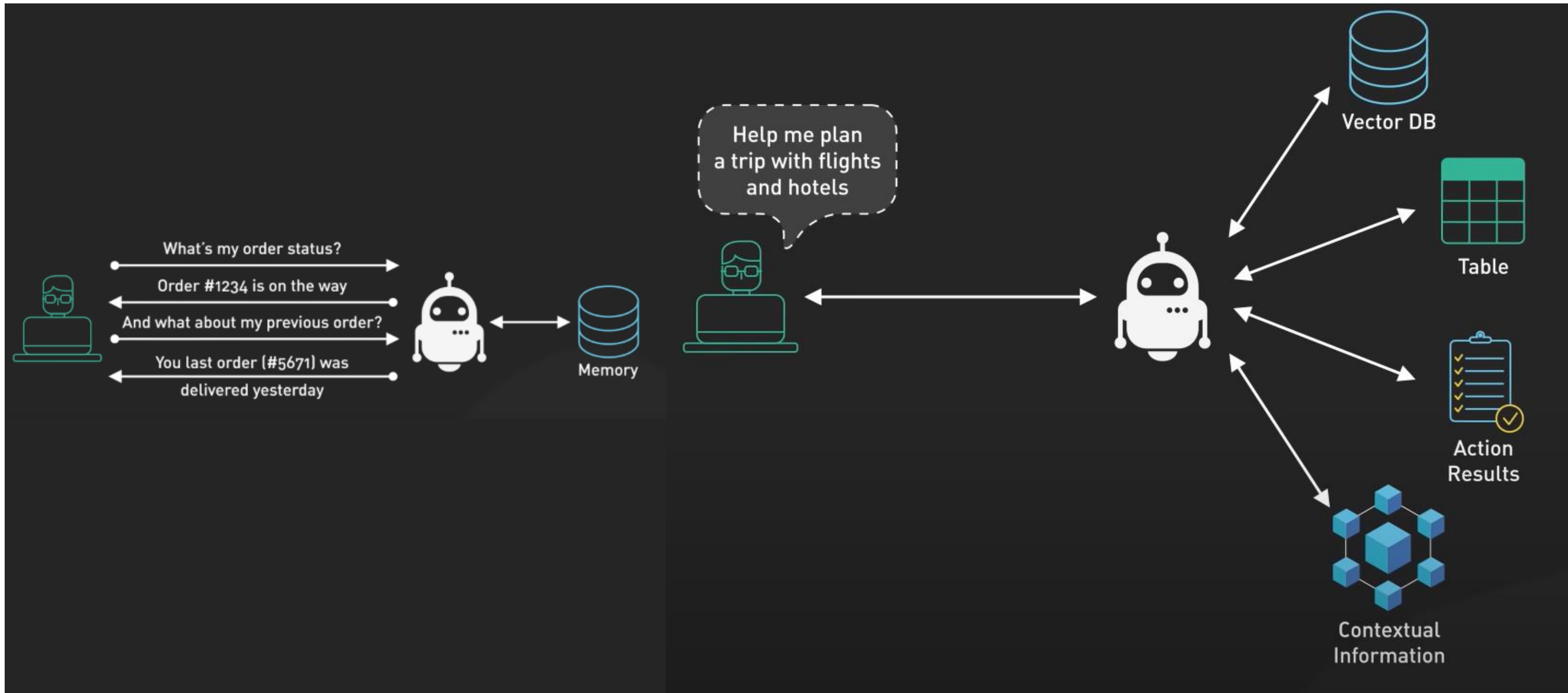
AI Agent Trading Bot



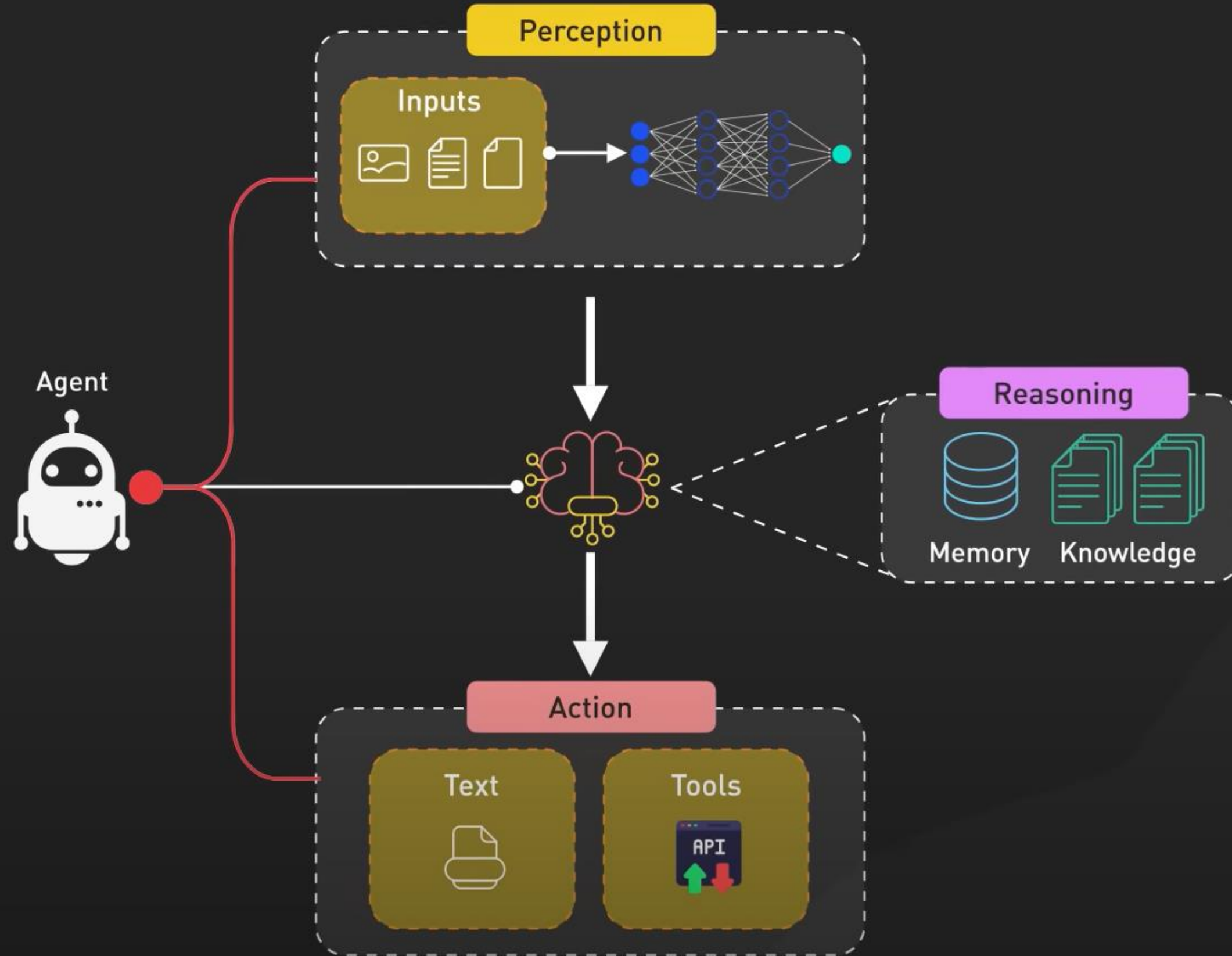
자율 주행차



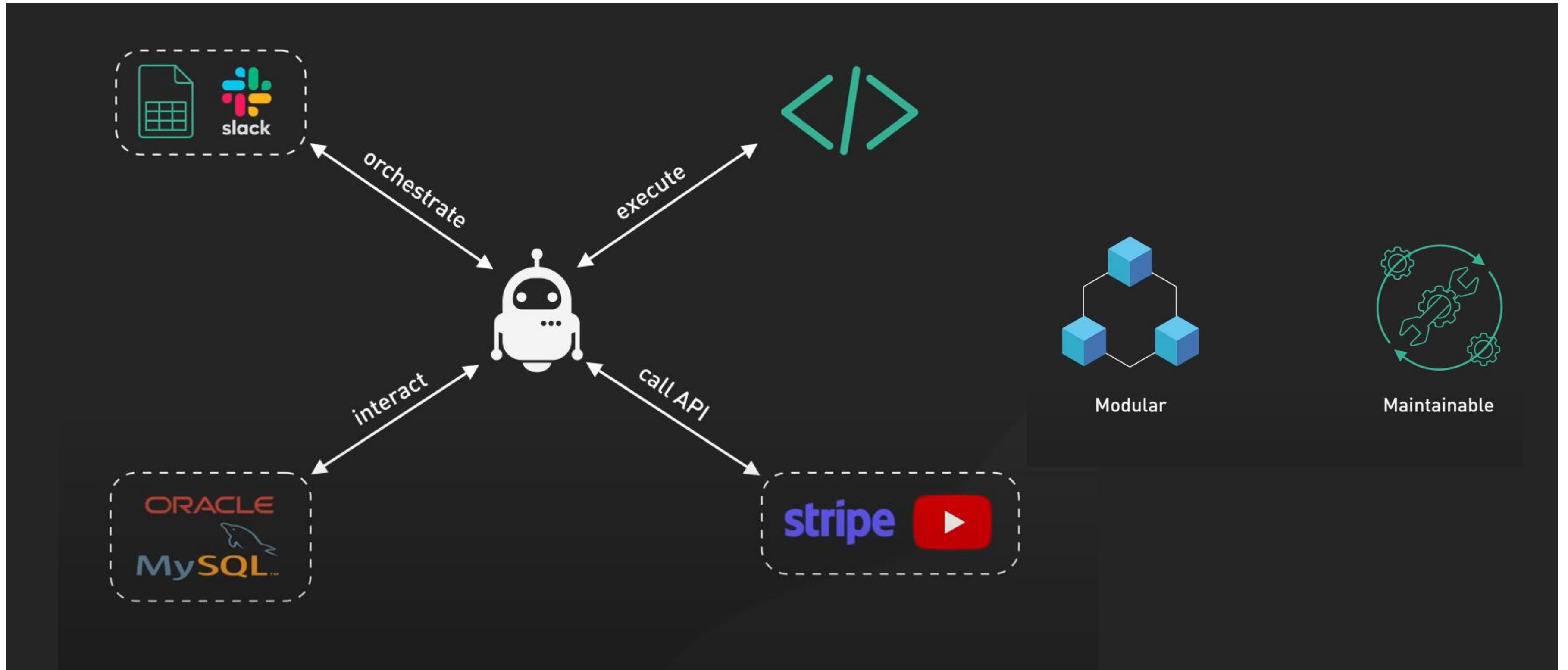
AI 에이전트 주요 기능 - 상태 유지



AI 에이전트 주요 기능 - LLM 기반 추론



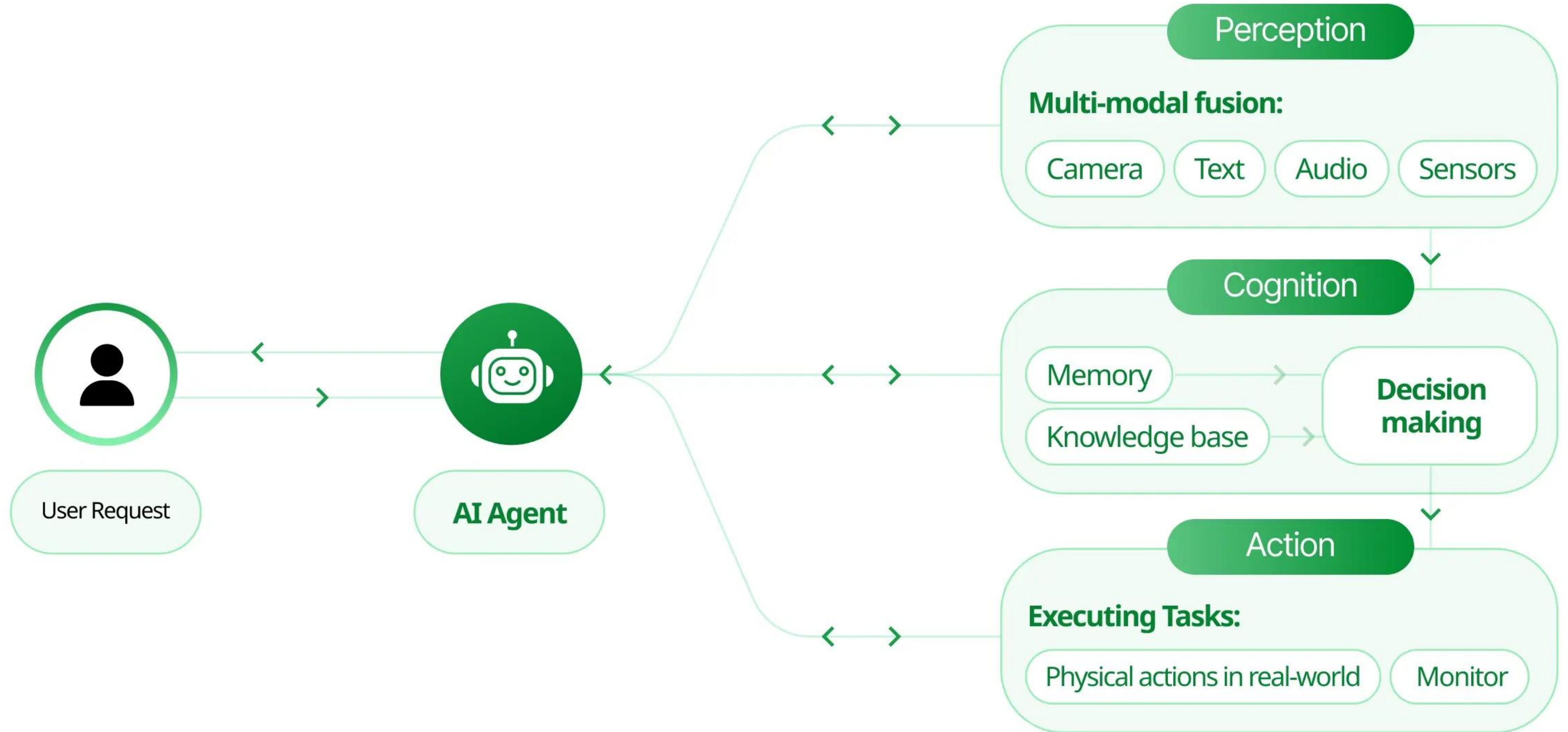
AI 에이전트 주요 기능 - 외부 시스템 연동



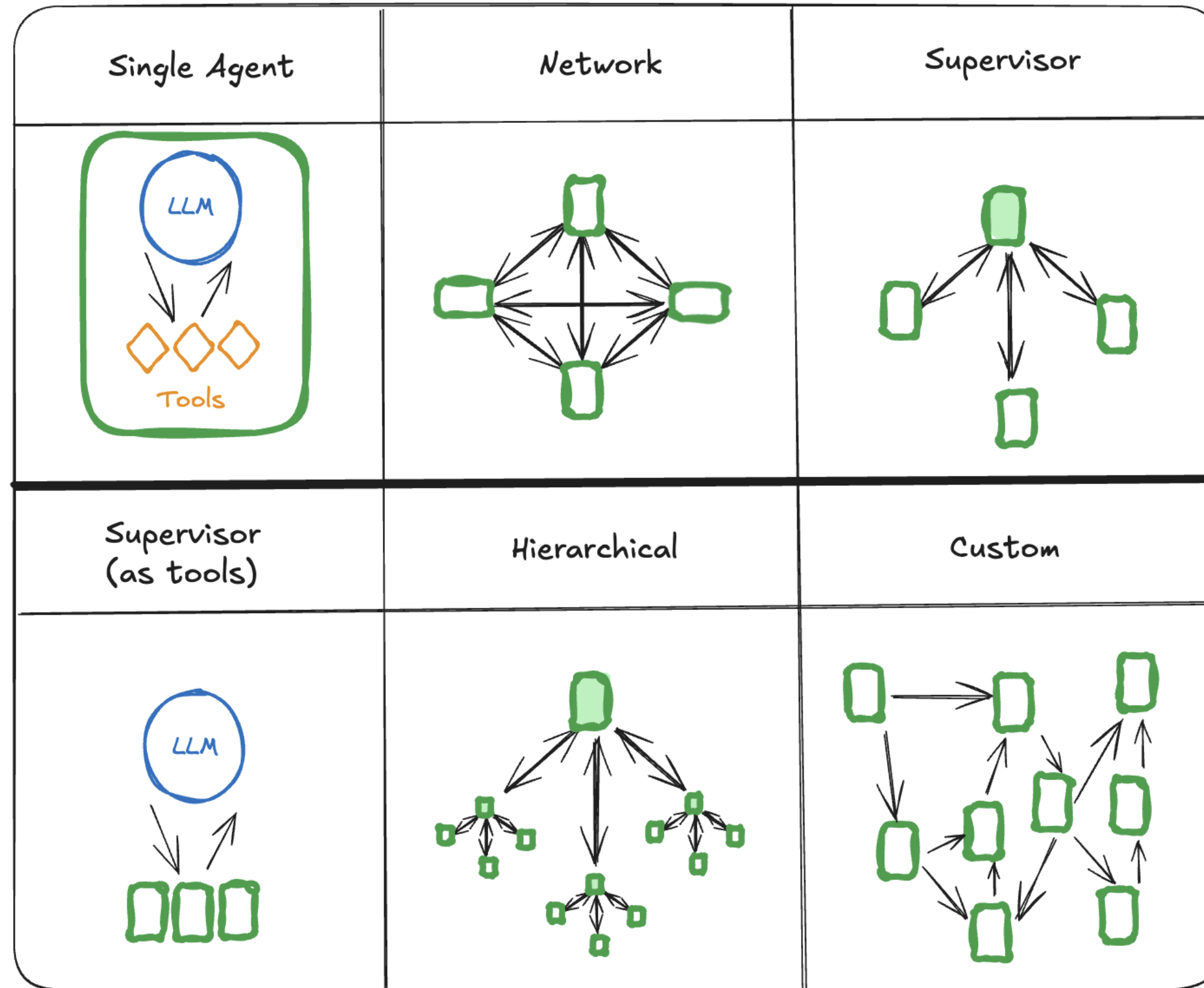
AI 에이전트 유형

유형	설명	주요 특징
단순 반사형 에이전트 (Simple Reflex Agent)	특정 입력에 대해 즉시 반응	If-then 규칙 기반 메모리 없음
모델 기반 에이전트 (Model-Based Agent)	내부 변수를 사용하여 상태 추적	환경 변화에 적응 가능
목표 기반 에이전트 (Goal-Based Agent)	목표를 달성하기 위한 최적의 경로 탐색	경로 탐색 알고리즘 사용
학습 에이전트 (Learning Agent)	강화 학습을 통해 성능 향상	실행 결과를 지속적으로 최적화
유틸리티 기반 에이전트 (Utility-Based Agent)	최적의 선택을 위해 점수 계산	보상/위험 요소를 분석

AI 에이전트 아키텍처



AI 에이전트 아키텍처



AI 에이전트 유스케이스



분야	적용 사례
스마트 빌딩	에너지 최적화, 자동 조명 제어, 보안 시스템 관리
금융	사기 탐지, 자동 주식 거래
헬스케어	환자 모니터링, 의료 챗봇
자동차	자율주행 시스템, AI 기반 내비게이션
게임	적응형 NPC(Non-Player Character)
이커머스	개인화 추천 시스템, AI 쇼핑 도우미



AI 에이전트와 LLM

LLM 활용

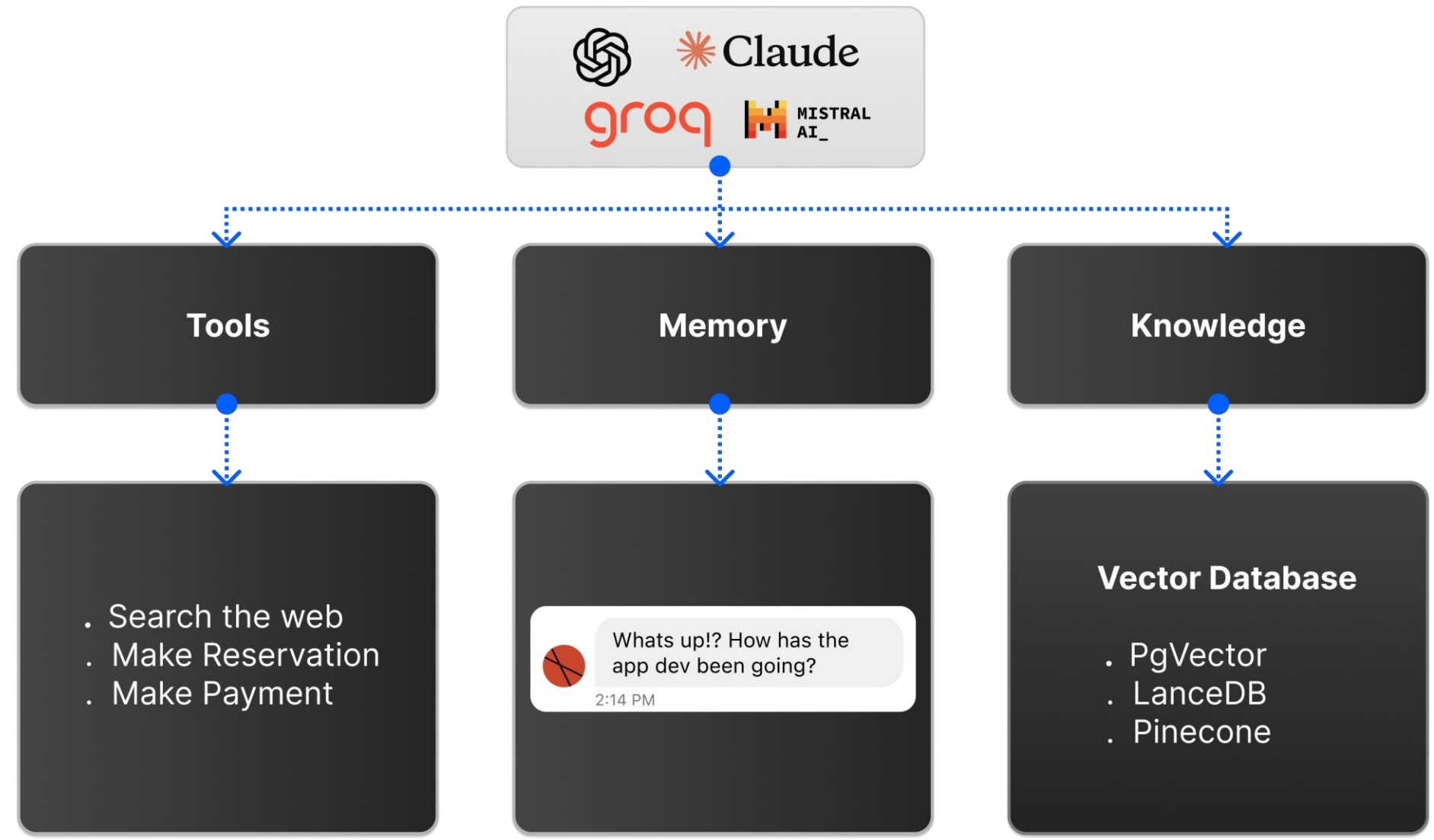
AI 에이전트는 GPT-4, Claude 같은
대형 언어 모델을 활용해 더욱 똑똑해지고 있음

자동화된 고객 응대

고객의 질문을 이해하고, 필요한 정보를
검색한 후, 자연스러운 답변을 생성

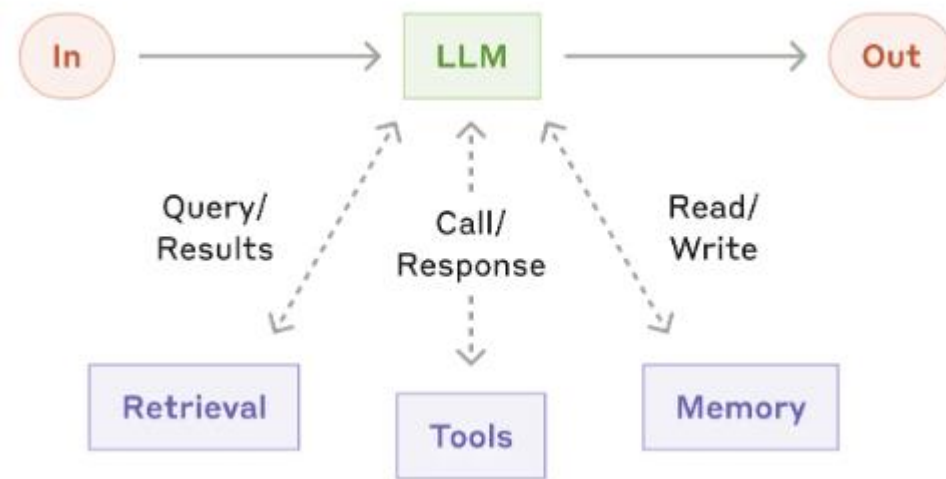
프레임워크 활용

LangGraph, AutoGen, CrewAI 등
개발 프레임워크를 활용하여, 다양한 기능을 조합해
더욱 강력한 AI 에이전트를 개발할 수 있음

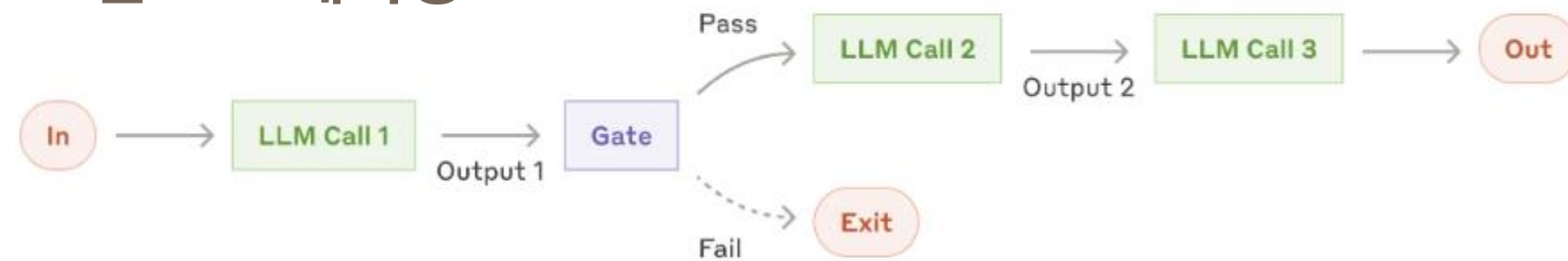


AI 에이전트 패턴

증강(Augmented) LLM



프롬프트 체이닝



라우팅



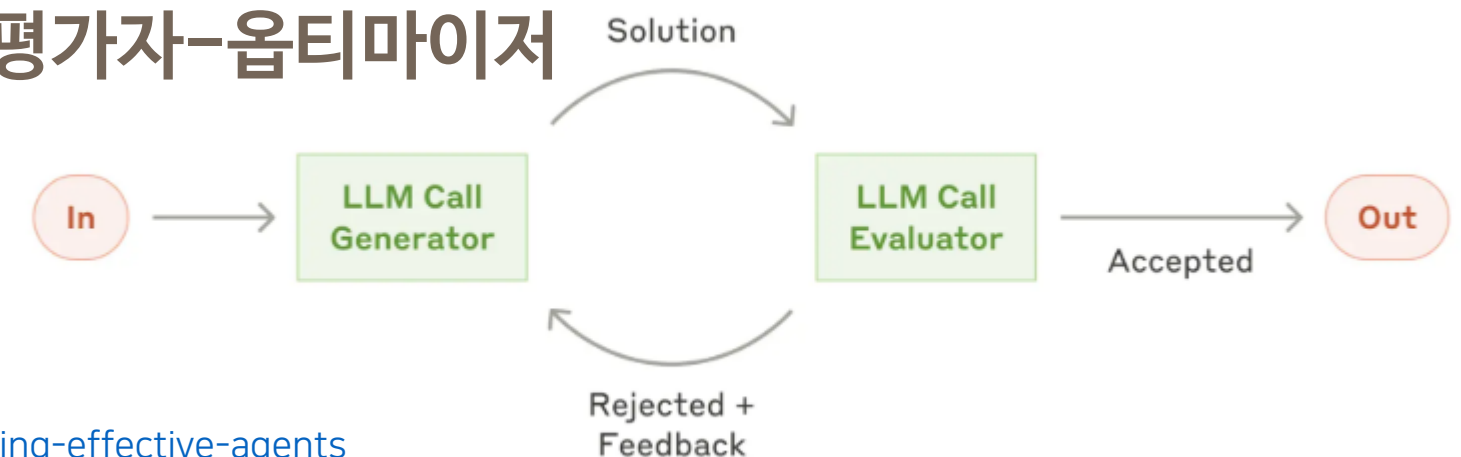
병렬화



오케스트레이터-워커



평가자-옵티마이저



AI 에이전트 기본 구현 예시

파이썬

```
1 agent = Agent(  
2     model=OpenAI(id="o1-mini"),  
3     memory=AgentMemory(),  
4     storage=AgentStorage(),  
5     knowledge=AgentKnowledge(  
6         vector_db=PgVector(search_type=hybrid)  
7     ),  
8     tools=[Websearch(), Reasoning(), Marketplace()],  
9     description="You are a useful marketplace AI agent",  
10 )
```


AI 에이전트 개발 프레임워크



LLM 애플리케이션 개발을
간소화하기 위한 통합 및 구성
가능한 구성 요소를 제공



제어 가능한 에이전트를
구축하기 위한 저수준
오케스트레이션 프레임워크



정의된 역할, 목표, 배경을 가진
AI 에이전트 개발과
에이전트 간 협업을 지원



분산 애플리케이션을 지원하며
도구 실행 및 함수 호출 기능 제공

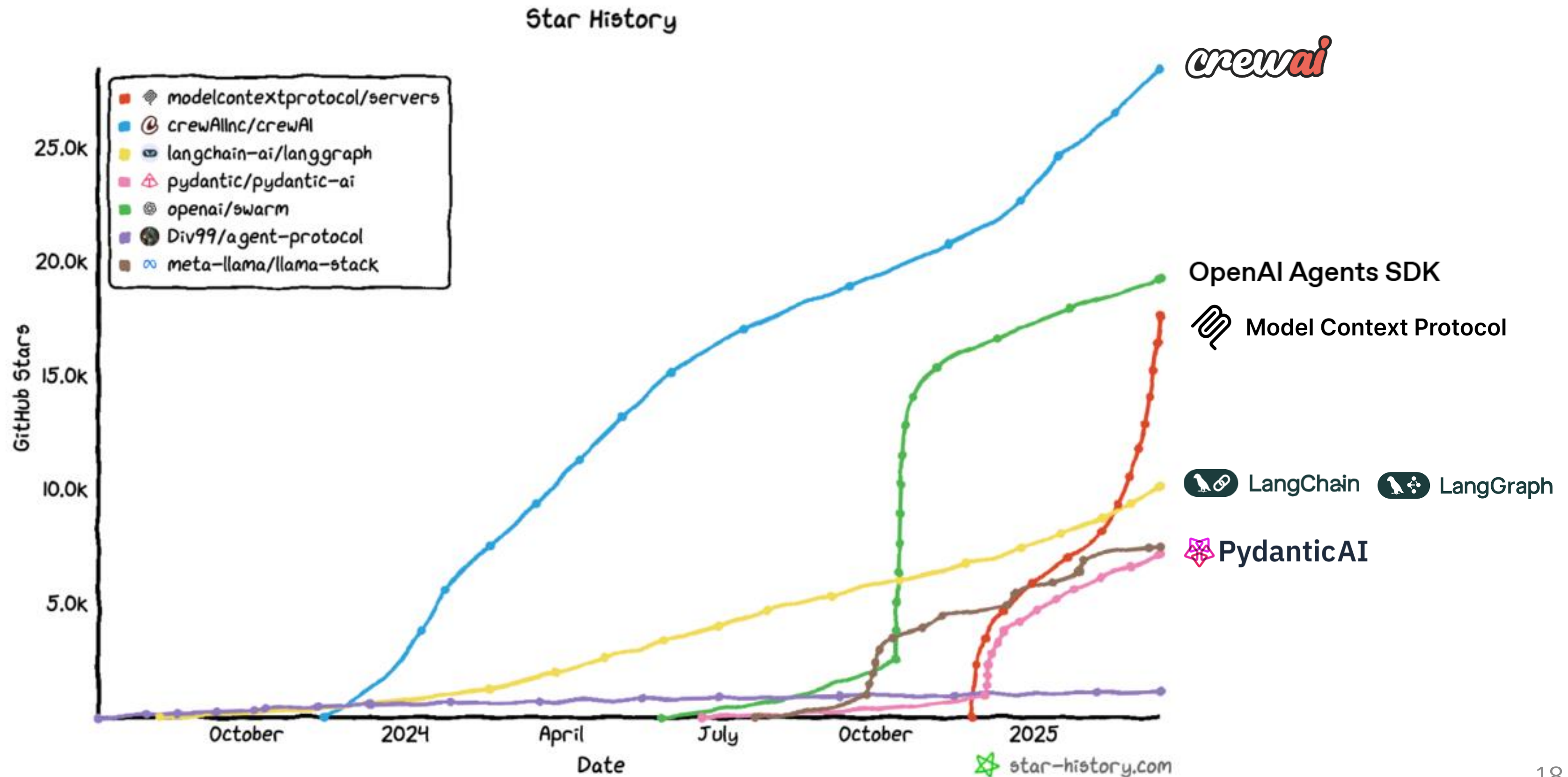


에이전트를 위한 실험 단계였던 Swarm을
프로덕션 환경에 맞게 업그레이드



LLM 애플리케이션과 외부 도구 및 데이터 소스
간의 원활한 통합을 하는 개방형 프로토콜

AI 에이전트 개발 프레임워크





Model Context Protocol



MCP(Model Context Protocol)는 AI 모델과 외부 데이터 소스 또는 도구를 연결해주는 개방형 표준 프로토콜입니다.

AI 모델이 필요한 맥락(Context) 이나 정보를 외부에서 가져올 수 있도록 해주는 통로 역할을 합니다.

기존에는 LLM이 주어진 텍스트 외에 별도의 지식이나 데이터에 접근하기 어려워, 새로운 데이터 원본마다 별도 커스텀 통합이 필요했습니다 .

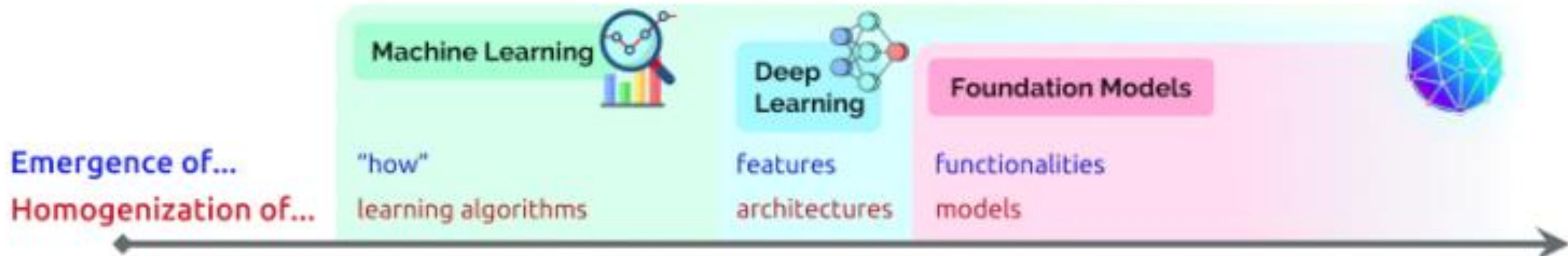
MCP는 AI 분야의 USB-C 포트와 같은 표준화된 연결 방식으로 비유됩니다.

MCP의 역할은

- AI 모델이 외부 지식/데이터를 얻도록 도와주고
- 표준 프로토콜을 통해 다양한 도구(웹 API, 데이터베이스, 파일 등)를 안전하게 사용할 수 있게 하며
- 이를 통해 AI 응용 프로그램의 한계를 확장하고 맥락 인지 능력을 높여주는 것입니다.

LLM (Large Language Model) 개요

AI(Artificial Intelligent)

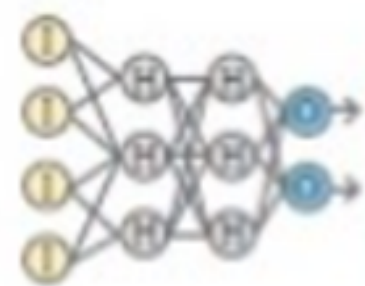


인공지능 사람의 지적능력(추론, 인지)을 구현하고 모방하는 모든 기술

머신러닝 명시적인 프로그래밍 없이 학습하는 기술



선형회귀
로지스틱회귀
K-최근접 이웃
결정트리
랜덤포레스트
서포트 벡터 머신
클러스터링
차원축소



심층신경망(DNN)
합성곱신경망(CNN)
순환 신경망(RNN)
생성적 적대 신경망(GAN)
강화학습(RL)

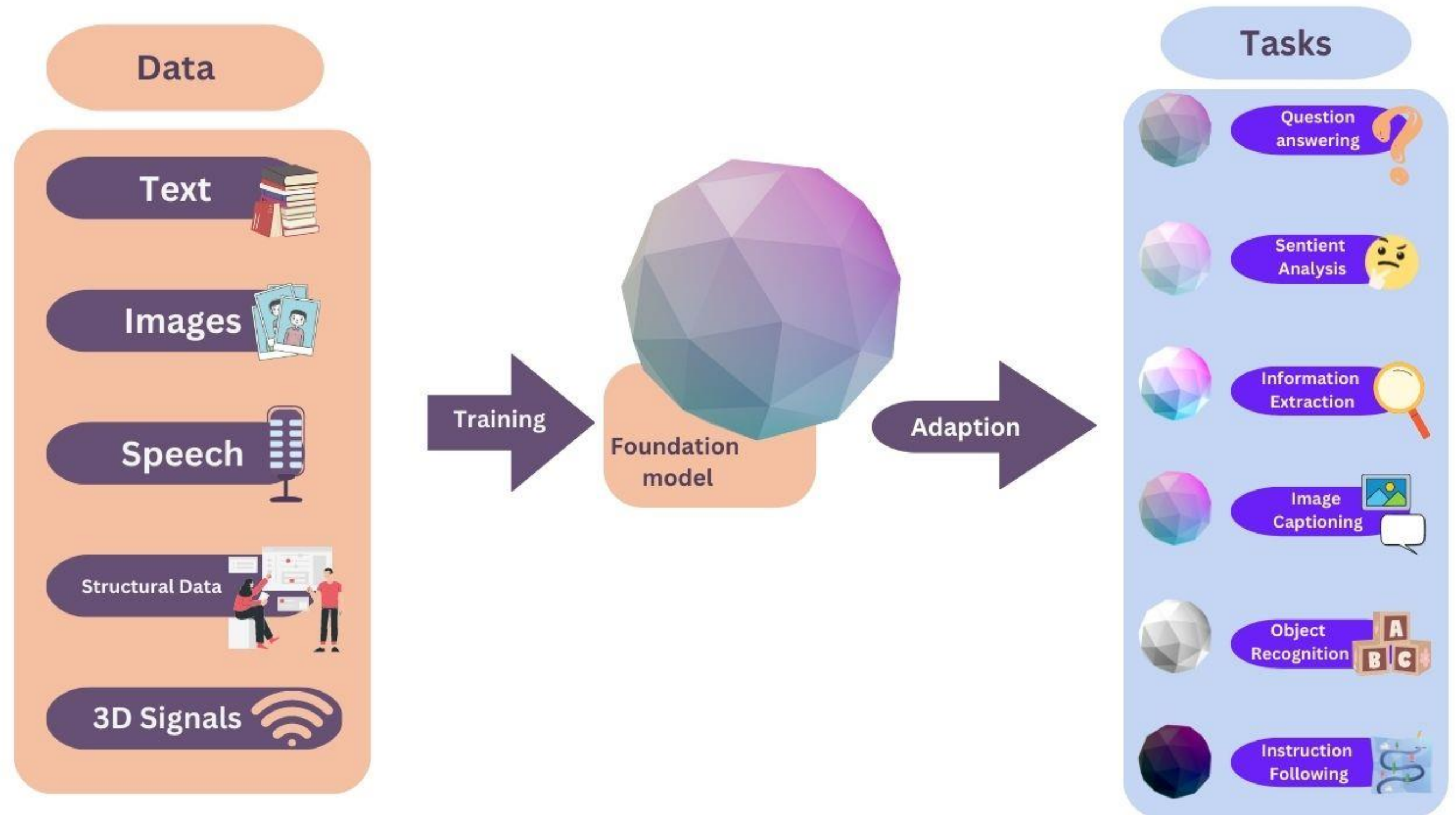
딥러닝 인공신경망 이용해 데이터에서 패턴을 찾아내는 기술

파운데이션 모델

BERT
GPT = Generative
Pre-trained
Transformer

파운데이션 모델 (Foundation Model)

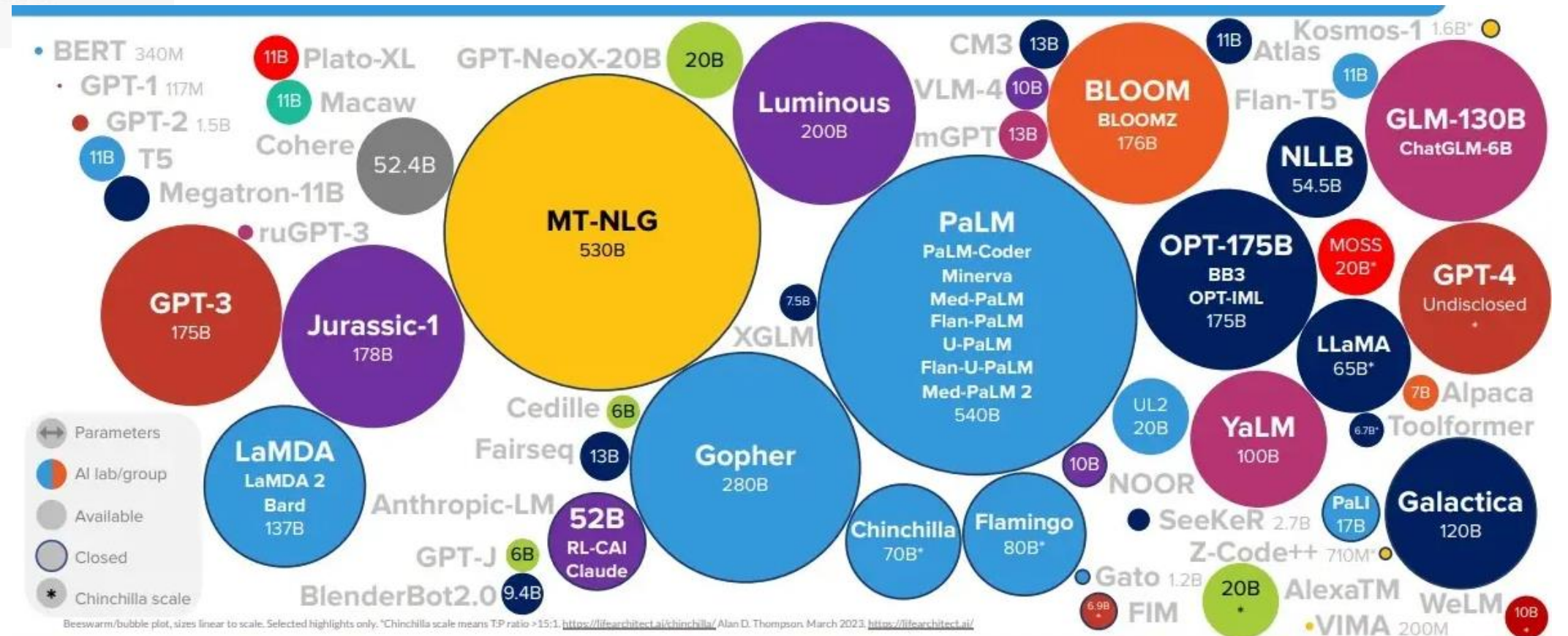
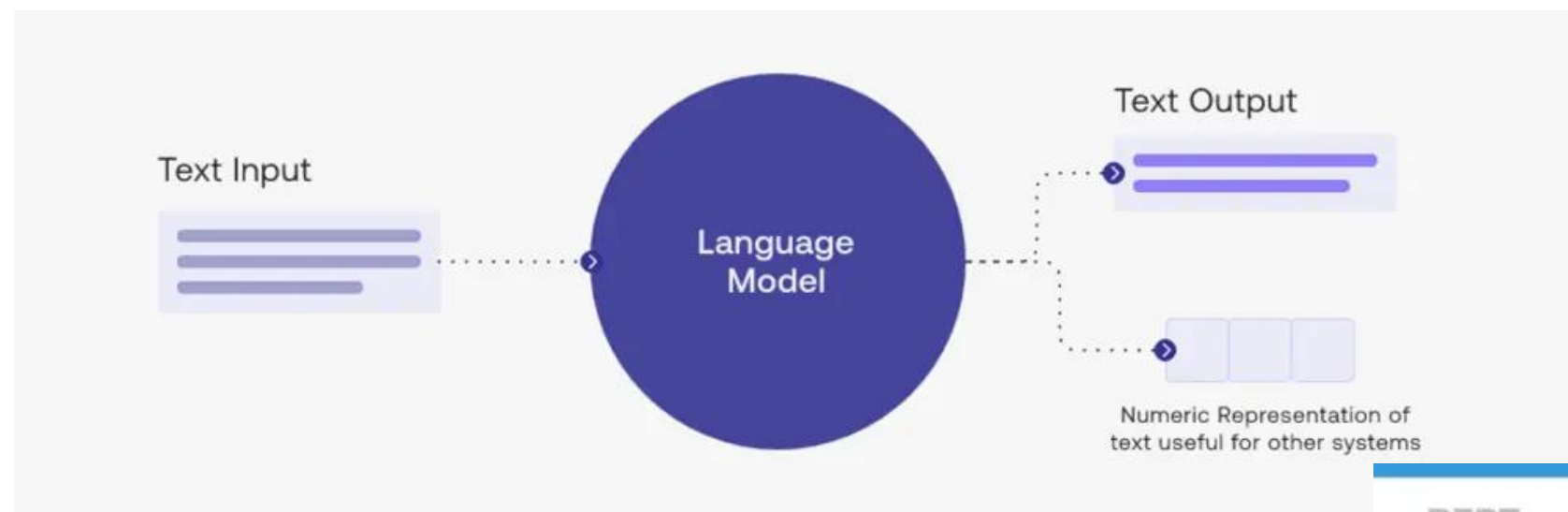
- 대용량의 폭넓은 비정형 데이터로 사전 훈련
- 복잡한 개념을 학습할 수 있는 방대한 파라미터
- 다양한 다운스트림 작업에 적용 가능
- 도메인별 데이터를 사용하여 파운데이션 모델을 사용자화



대형 언어 모델(LLM : Large Language Model)

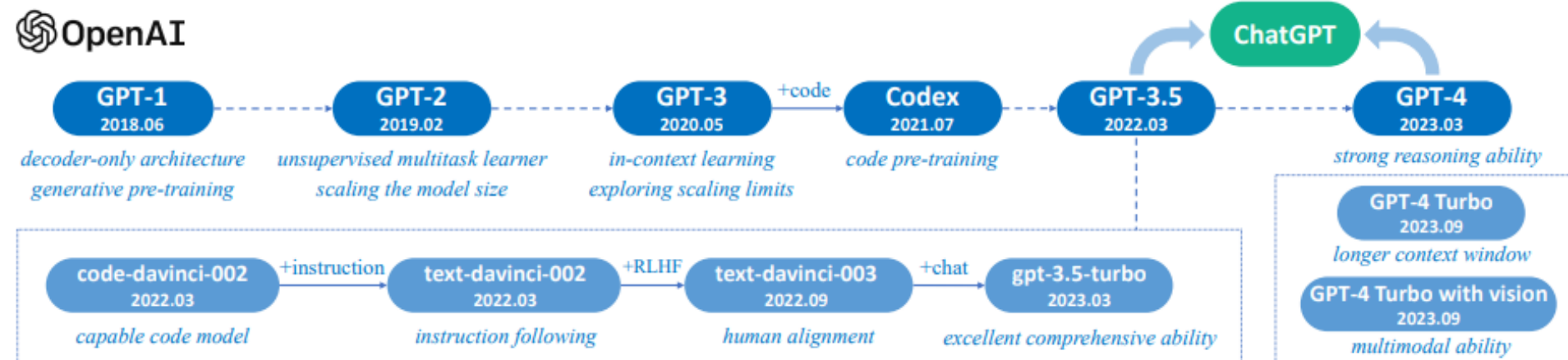
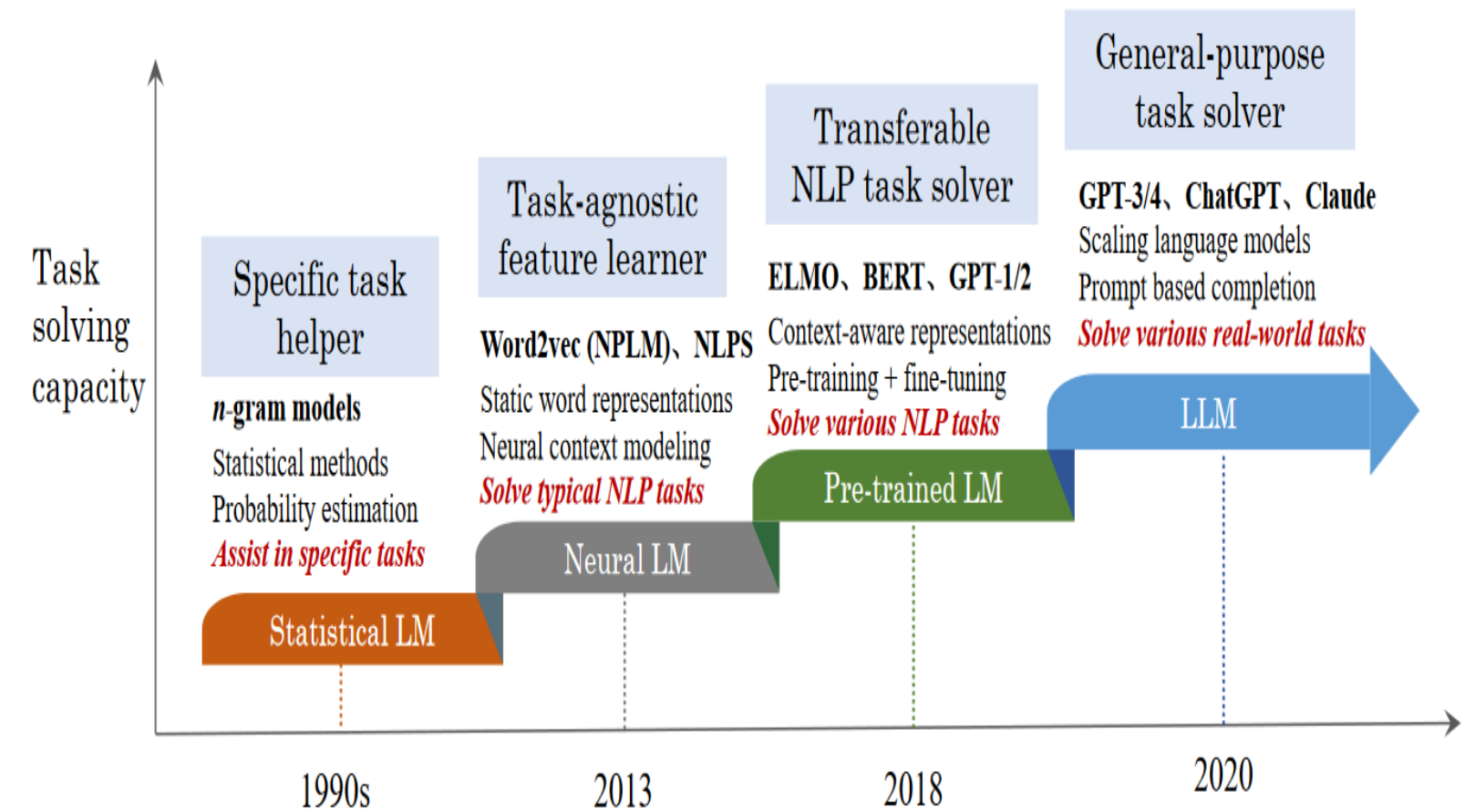
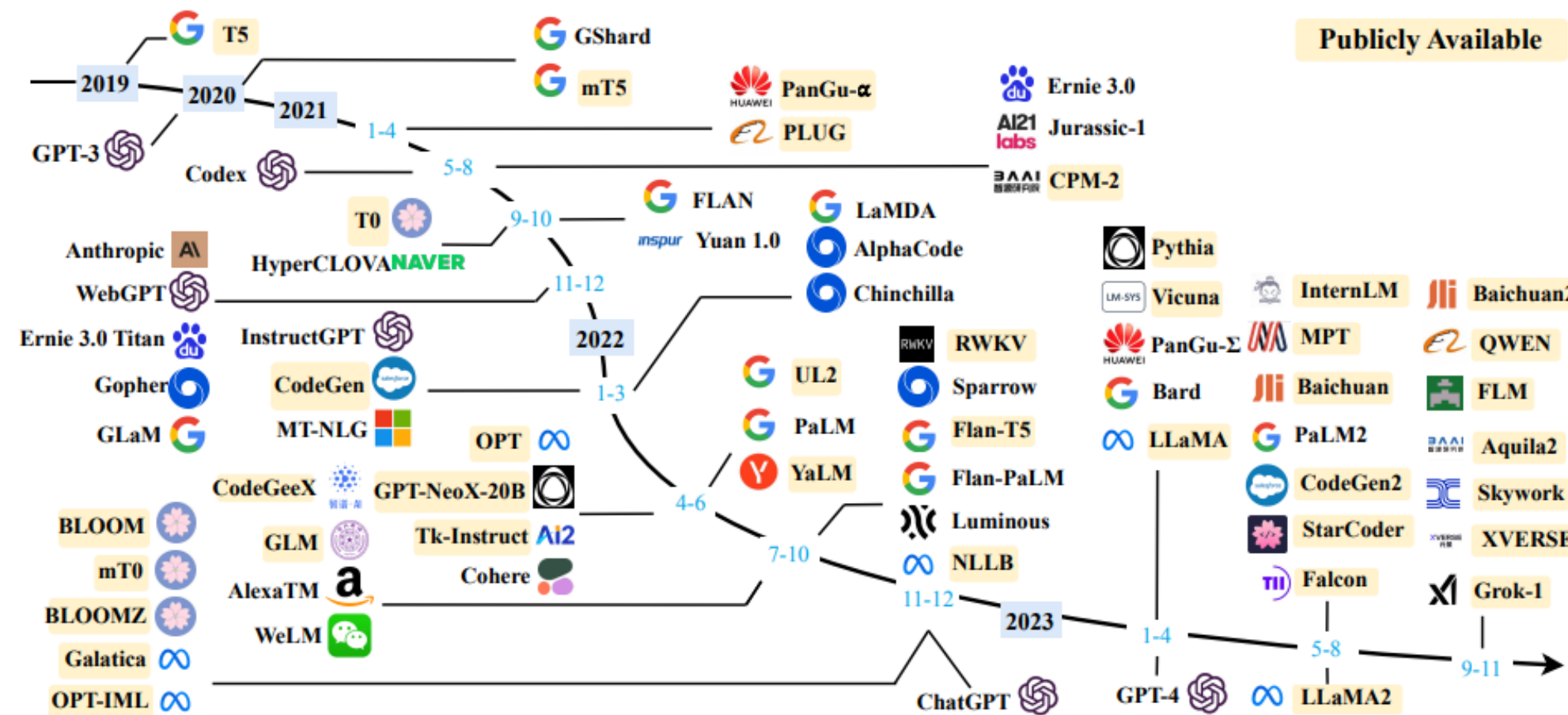
Foundation Model 중, LLM은 주어진 프롬프트에 대해 인간과 유사한 응답을 생성하기 위해 방대한 양의 텍스트 데이터로 훈련된 고급 AI 모델

LLM 모델들은 인간 언어를 이해하고 생성하는 등 다양한 작업에서 뛰어나고, 다양한 응용 분야에서 매우 가치 있는 도구로 사용되고 있음



LLM 모델

A Survey of Large Language Models : <https://arxiv.org/pdf/2303.18223.pdf> , 번역자료 : <https://wikidocs.net/222912>



GPT = Generative Pre-trained Transformer

LLM 서비스

<https://chat.openai.com/>

<https://claude.ai/>

<https://gemini.google.com/>

<https://www.deepseek.com/>

<https://grok.com/>

<https://www.perplexity.ai/>

텍스트 이해

질문에 대한 답변

창의적인 글쓰기

아이디어 기획

콘텐츠 생성

코드 작성/디버깅

단계별 지침 제공

가상 비서 역할

추론

플래닝

Reset Thread
Dark Mode
OpenAI Discord
Learn More
Log out

ChatGPT



Examples

"Explain quantum computing in simple terms"

"Got any creative ideas for a 10 year old's birthday?"

"How do I make an HTTP request in Javascript?"



Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests



Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

Limited knowledge of world and events after 2021

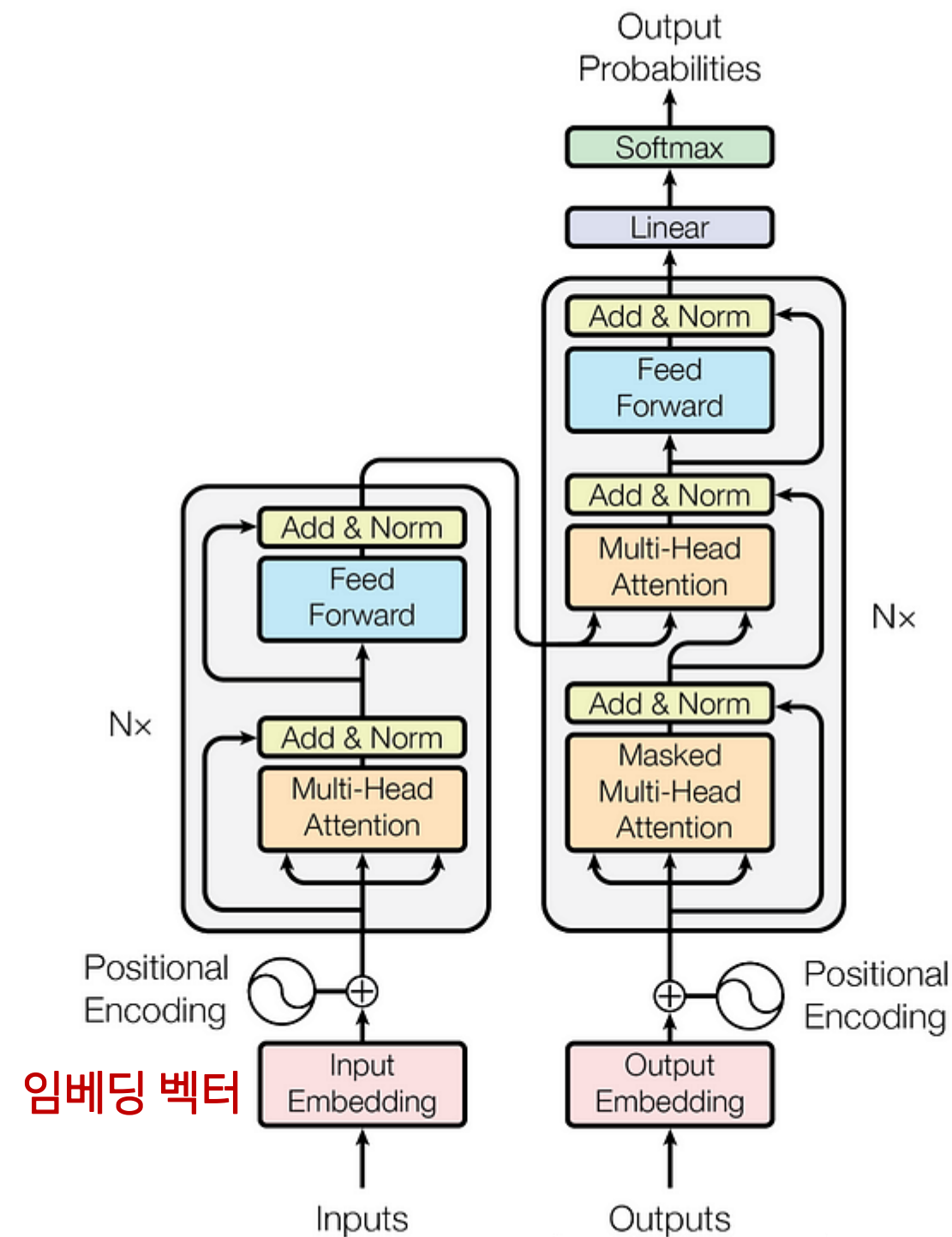
Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them safer.

트랜스포머 (Transformer) 아키텍처

Transformer 아키텍처

BERT

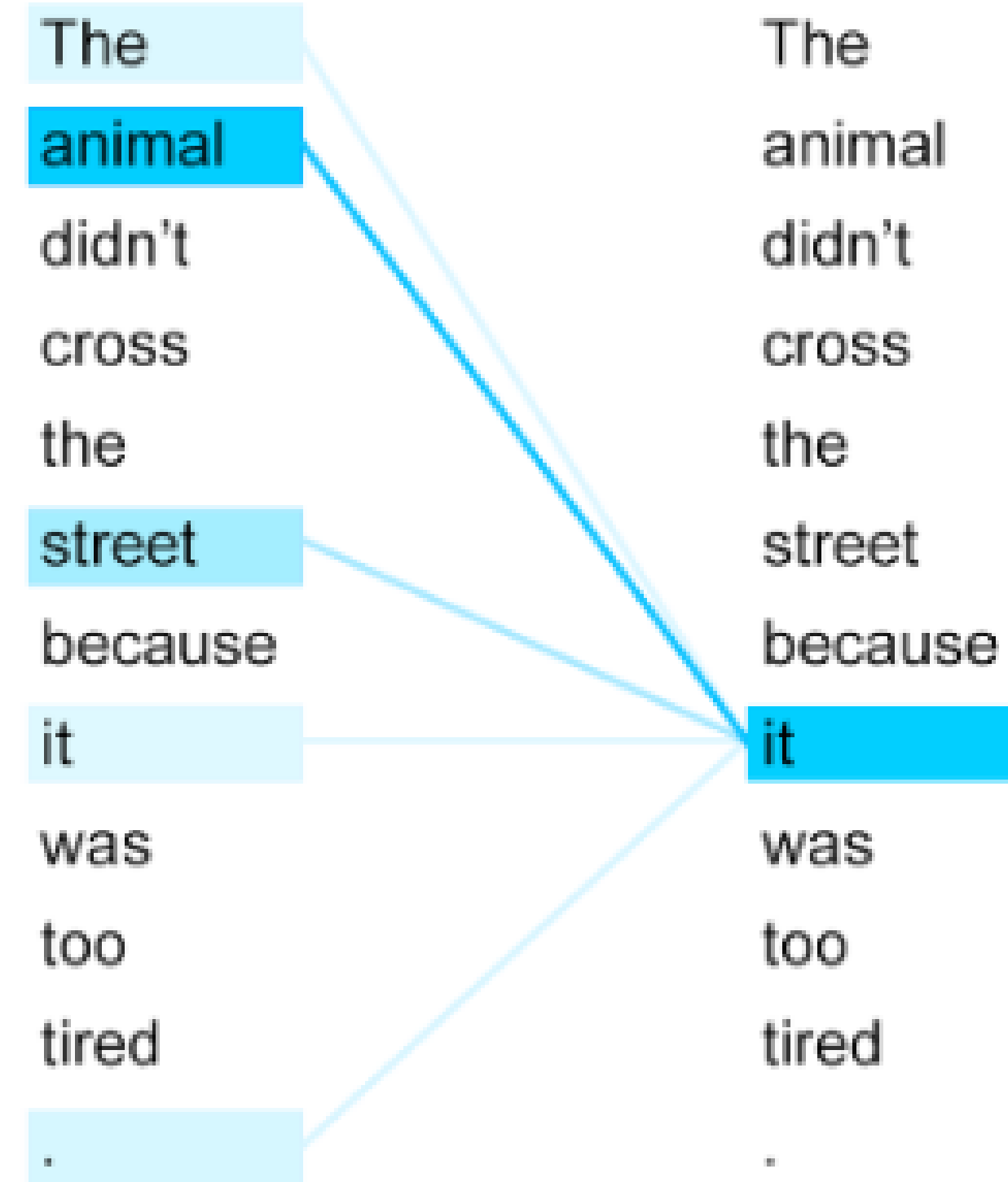
Encoder



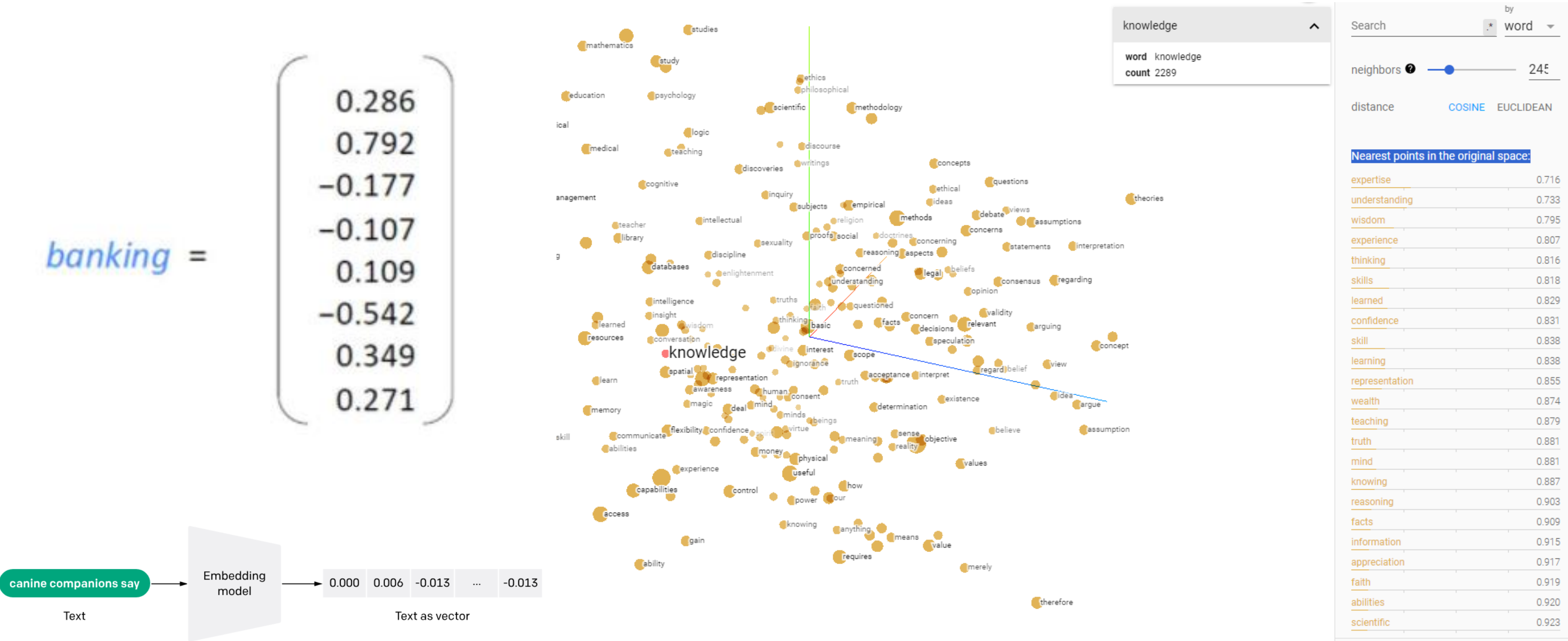
GPT

Decoder

Self Attention

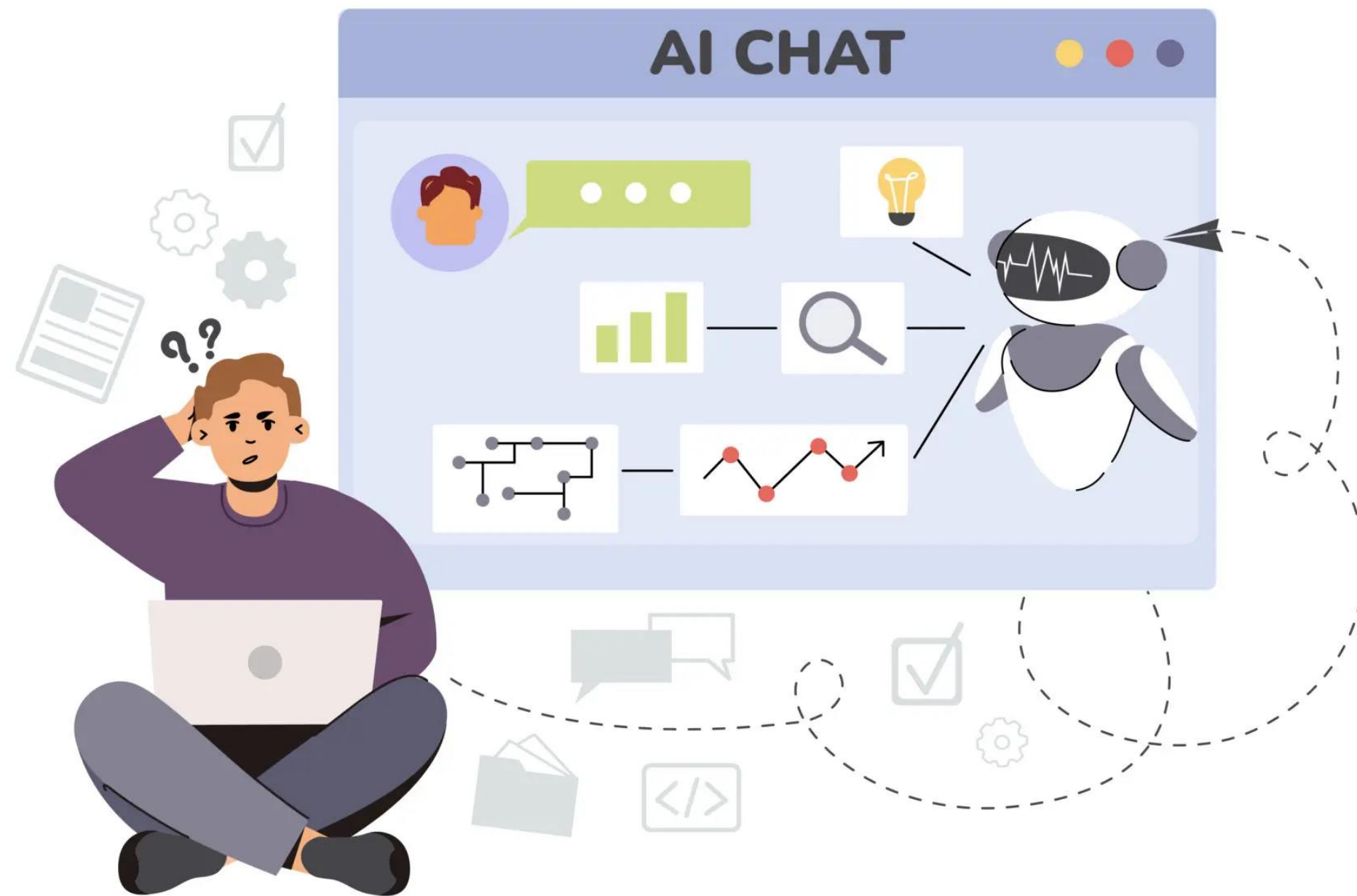


임베딩 (Embedding) 벡터



<https://projector.tensorflow.org/>

할루시네이션 (Hallucination)



AI 환각의 유형

- 문장 모순: 이는 LLM이 동일한 출력 내에서 이전 문장과 모순되는 문장을 생성할 때 발생합니다.
예시)
프롬프트: 풍경에 대한 설명을 네 단어 문장으로 작성하세요.
출력: 잔디는 초록색이었습니다. 산은 파랗다. 강은 보라색이었습니다.
잔디는 갈색이었다.
- 프롬프트 모순: 이 경우 LLM에서 생성된 문장이 문장을 생성하는 데 사용된 초기 프롬프트와 모순되는 경우입니다.
예시)
프롬프트: 조카의 생일 카드를 써주세요.
출력: 엄마, 아빠, 생일 축하해요!
- 사실 모순: LLM은 허구의 정보를 사실로 제시할 수 있습니다.
예시)
프롬프트: 미국의 도시 세 곳을 말하세요.
출력: 뉴욕, 로스앤젤레스, 토론토.
- 관련성이 없거나 무작위적인 환각: 생성된 정보가 입력 또는 출력 컨텍스트와 관련성이 없을 때 발생합니다.
예시)
프롬프트: 런던에 대해 설명해 주세요.
출력: "런던은 영국의 도시입니다. 고양이는 하루에 한 번 이상 먹이를 주어야 합니다."

LLM서비스

	 ChatGPT	 Claude	 Grok	 Gemini	 perplexity
Everyday answers	★	✓	✓	✓	✓
Writing	✓	★	✓	✓	✓
Coding	✓	★	✓	✓	✓
Math	★	✓	✓	✓	✓
Reasoning	★	✓	✓	✓	✓
Web search	✓	✗	✓	✓	★
Deep research	★	✗	✓	✓	✓
Voice chat	★	✗	✓	✓	✓
Image gen	✓	✗	✓	★	✗
Video gen	✓	✗	✗	★	✗
Live camera	★	✗	✗	✓	✗
Computer use	★	✓	✗	✗	✗

sLLM

<https://ollama.com/library>



- sLLM(smaller Large Language Model , 소형 언어 모델)
 - LLM과 비교했을 때 매개변수의 수가 수십 억~수백 억개로 비교적 크기가 작은 언어모델
 - 비용절감, 보안, 특정 도메인에 활용 목적으로 사용
 - 특정 도메인 사용용도로 sLLM 을 사용하는 경우가 많아지고 있음




- Ollama 설치
 - 로컬 환경에서 다양한 언어 모델을 실행할 수 있게 지원하는 오픈소스
 - 설치 파일 다운로드 : <https://ollama.ai/>
 - 모델 종류 : <https://ollama.com/library>



- Llama (Large Language Model Meta AI)
- Llama 2 (Large Language Model Meta AI)
 - 메타에서 공개한 상업적으로도 이용 가능한 오픈 소스 sLLM
 - 설치 및 실행 : ollama run llama2
 - 프로그램 개발 예시

```
from langchain_community.llms import Ollama
llm = Ollama(model="llama2")
llm.invoke("Hello")
```

`pulling 8934d96d3f08... 100%`  `3.8 GB`

Thank you 😊