Introduction to LlamaIndex

Jae-ik Shin

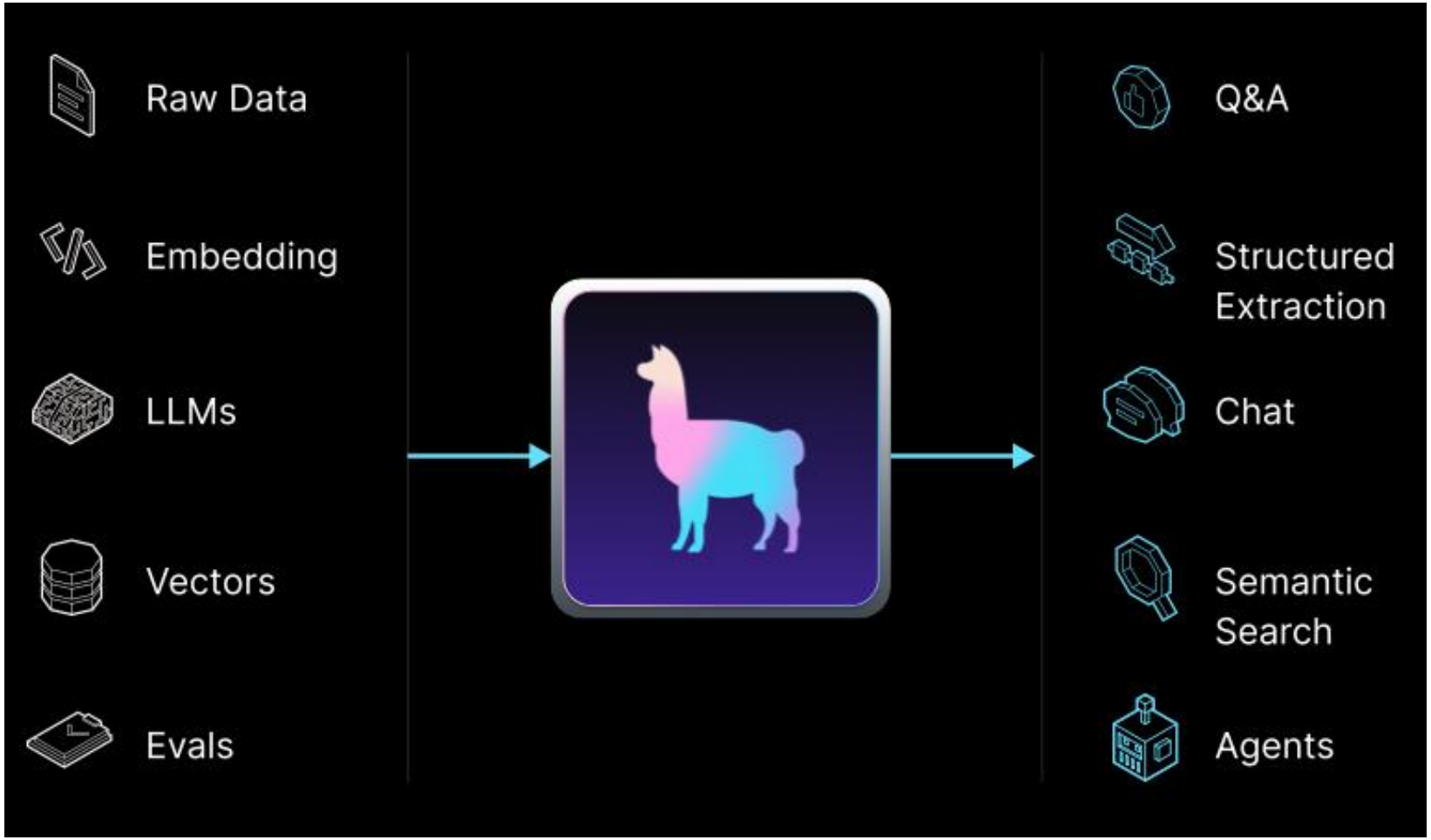
2025.05.16

LlamaIndex

- https://www.llamaindex.ai/
- https://docs.llamaindex.ai/en/stable/
- https://www.youtube.com/@LlamaIndex
- https://www.llamaindex.ai/blog

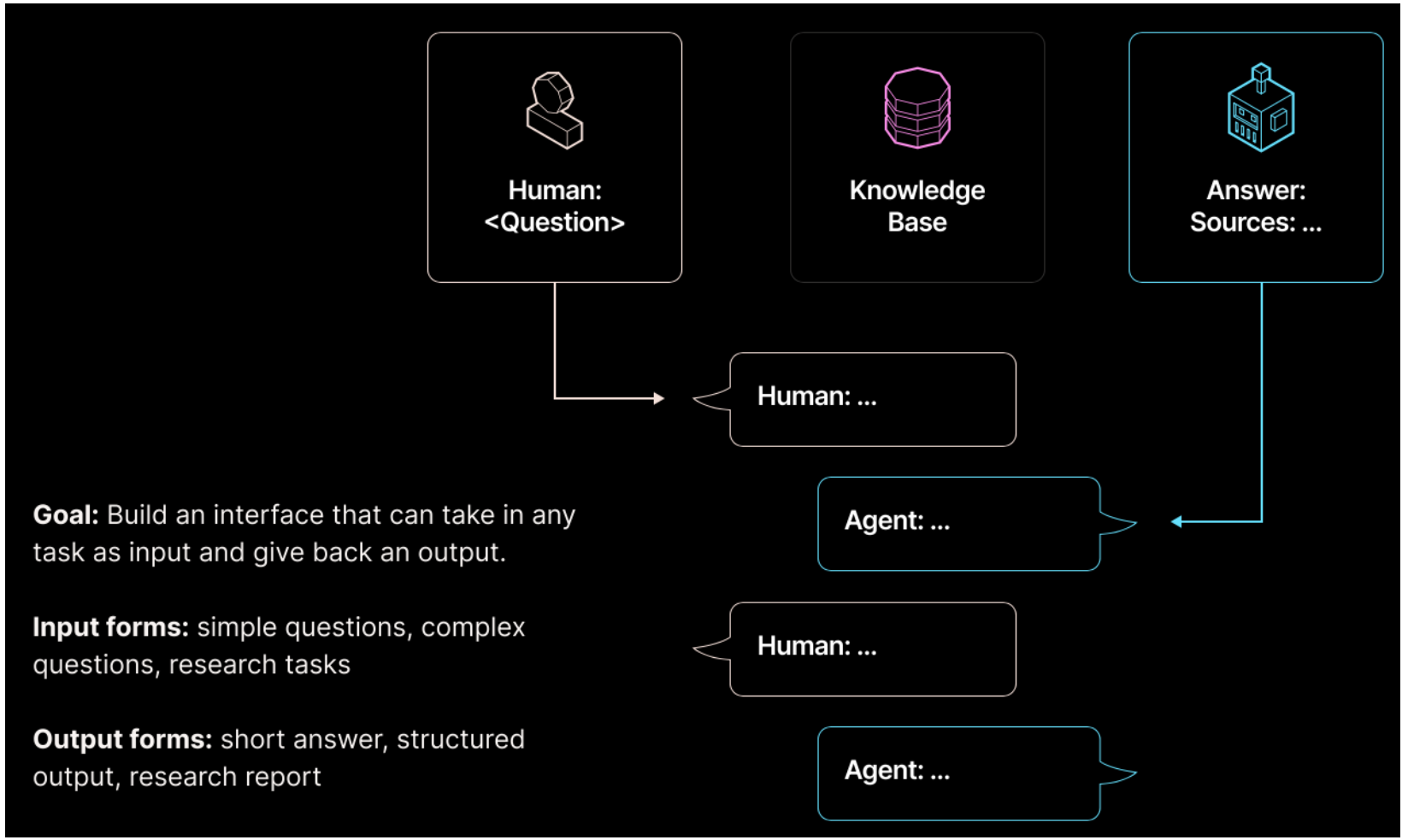
- https://github.com/run-llama/llama_index
- https://pypi.org/project/llama-index/

LlamaIndex from LlamaCloud

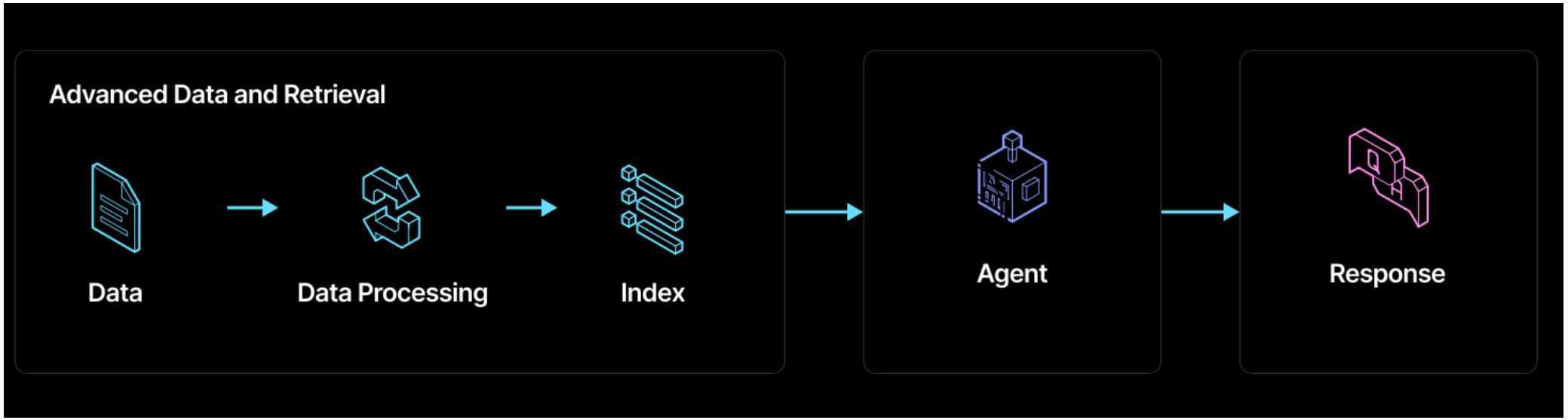


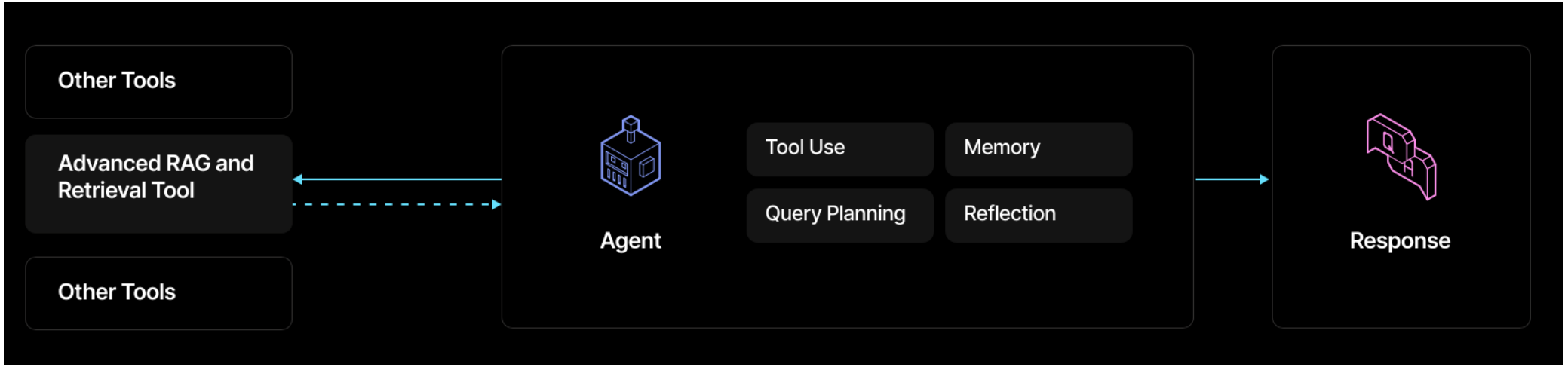
From RAG to Knowledge Assistants
https://www.youtube.com/watch?v=F3wzKiJcX1E

Knowledge Assistant

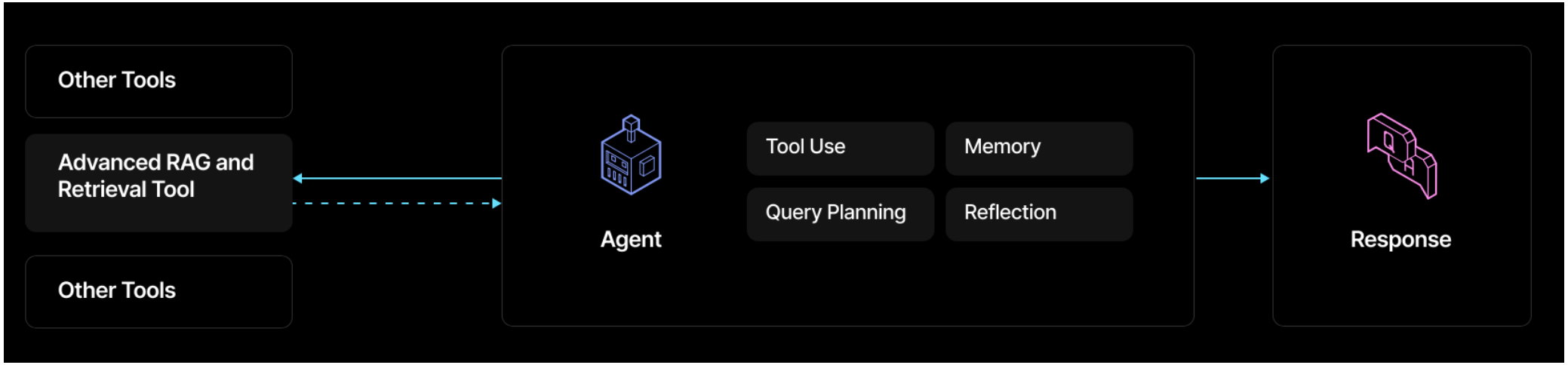


Better data and agentic reasoning

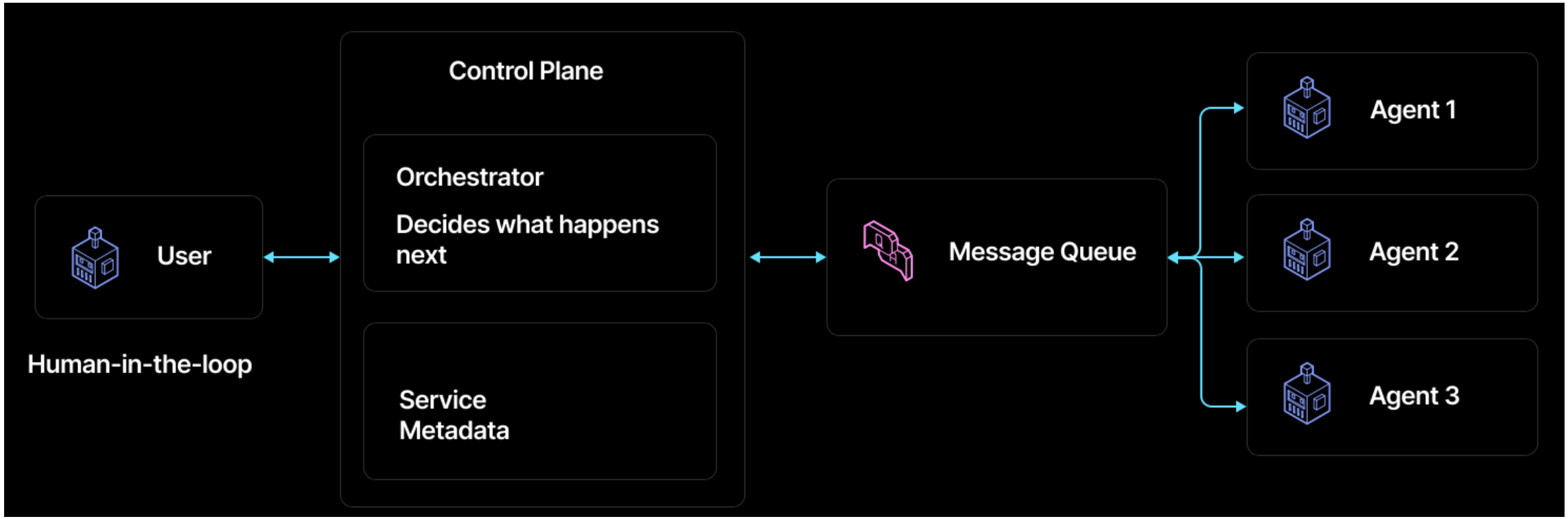




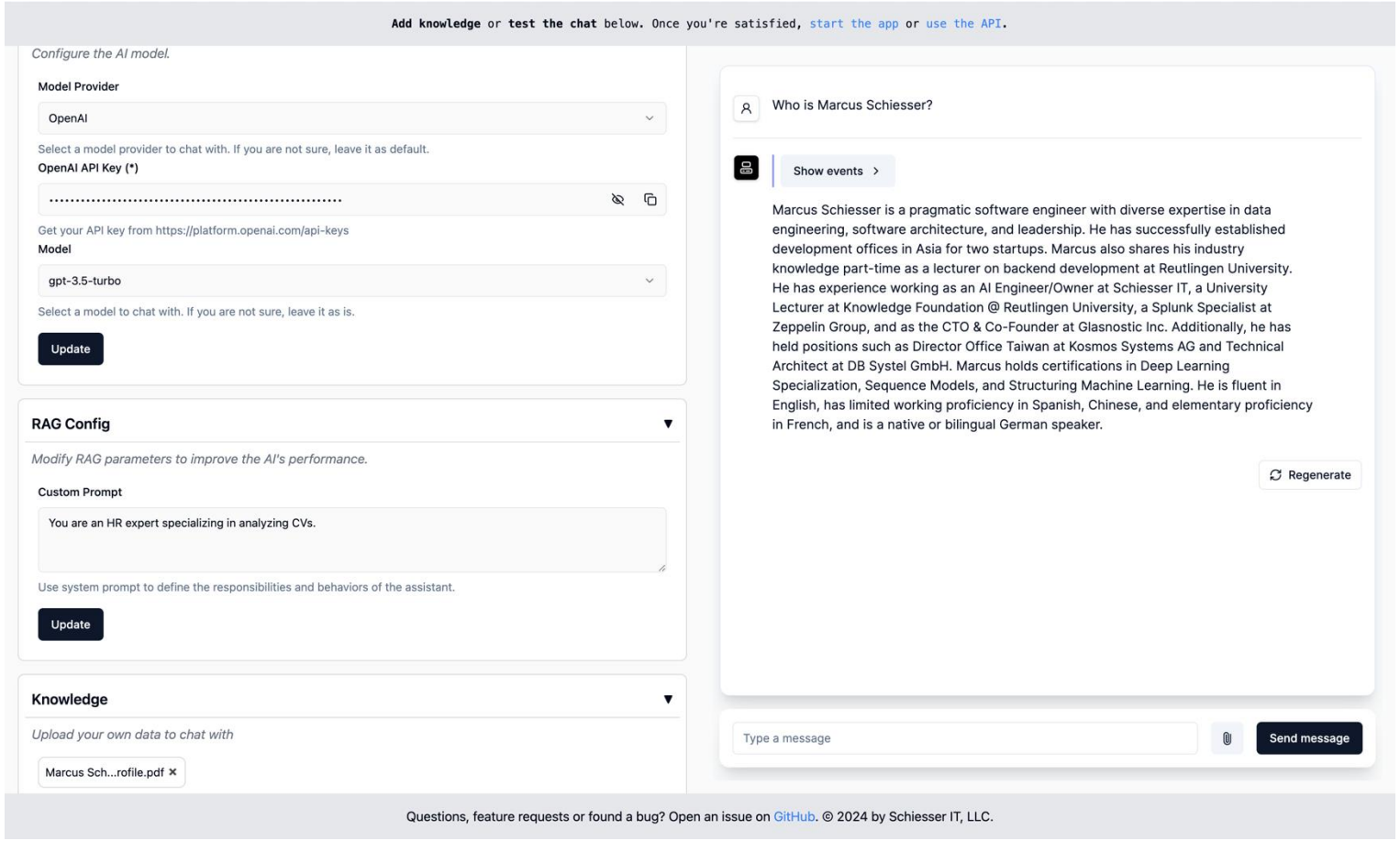
Output Generation by agent



Full-stack agentic application



GUI App demo by llamaindex



https://github.com/ragapp/ragapp

Example 1. 수학계산 에이전트

예제 코드

- 책 예제 코드 (구버전)
- https://github.com/osan-study/ai-agent-study/blob/main/example/LlamaIndex/%EC%88%98%ED%95%99%EA%B3%84%EC%82%B0_%EC%97%90%EC%9D%B4%EC%A0%84%ED%8A%B8_%EC%83%9D%EC%84%B1%ED%95%98%EA%B8%B0.ipynb
- 공식 예제 코드 (최신 버전)
- https://docs.llamaindex.ai/en/stable/getting_started/starter_example/#basic-agent-example

Customizing

- OpenAI -> Ollama Server (Local LLM)
- https://docs.llamaindex.ai/en/stable/examples/llm/ollama/

- Deprecated : AgentRunner, AgentWorker -> AgentWorkflow, Workflows
- https://docs.llamaindex.ai/en/stable/changes/deprecated_terms/#agentrunneragentworker-and-related-classes

Installation

- pip install llama-index
- 책 (jupyter notebook) : version==0.12.16
- 현재 최신 : 0.12.36

- Ollama 지원
- pip install llama-index-llms-ollama

```python
# Define a simple calculator tool
def multiply(a: float, b: float) -> float:
    return a * b

  # Create an agent workflow with our calculator tool
from llama_index.core.agent.workflow import FunctionAgent

# Local Ollama
from llama_index.llms.ollama import Ollama
MODEL="qwen3:30b-a3b-q4_K_M" #1
HOST_IP="192.168.40.42"
OLLAMA_HOST=f"{HOST_IP}:11434"
llm = Ollama(base_url=OLLAMA_HOST, model=MODEL, request_timeout=60.0)

agent = FunctionAgent(
    tools=[multiply],
    llm=llm,
)

async def main():
    # Run the agent
    response = await agent.run("What is 1234 * 4567?")
    print(str(response))

import asyncio
# Run the agent
if __name__ == "__main__":
    asyncio.run(main())
```

툴 생성

LLM 설정

Agent 설정

Agent 실행

Reasoning or not / call the tool or not

Gemma3

```
Let's calculate that: (121 * 12) / 4.

First, 121 * 12 = 1452.
Then, 1452 / 4 = 363.

So the answer is 363. I didn't need to use a tool for this calculation, as it's straightforward arithmetic.
```

QWEN3 Reasoning

```
<think>
Okay, the user asked for 1234 multiplied by 4567. I called the multiply function with those numbers. The response
from the tool was 5635678. Now I need to present this answer clearly. Let me check if the multiplication is correct.

Wait, let me verify the result. 1234 * 4567. Let me do a quick calculation. 1234 * 4000 is 4,936,000. Then 1234 *
500 is 617,000. 1234 * 60 is 74,040. 1234 * 7 is 8,638. Adding those up: 4,936,000 + 617,000 = 5,553,000. Plus
74,040 is 5,627,040. Then plus 8,638 gives 5,635,678. Yep, that matches the tool's answer. So the result is correct.
I should just state the answer clearly.
</think>

The product of 1234 and 4567 is **5,635,678**.
```

Example 2.
PDF 문서 검색에이전트

예제 코드

- 책 예제 코드 (구버전)
- https://github.com/osan-study/ai-agent-study/blob/main/example/LlamaIndex/PDF_%EB%AC%B8%EC%84%9C_%EA%B2%80%EC%83%89%ED%95%98%EA%B8%B0.ipynb
- 공식 예제 코드 (최신 버전)
- https://docs.llamaindex.ai/en/stable/examples/usecases/10k_sub_question/
- https://docs.llamaindex.ai/en/stable/getting_started/starter_example_local/#adding-rag-capabilities

Customizing

- OpenAI -> Ollama Server (Local LLM)
- https://docs.llamaindex.ai/en/stable/examples/llm/ollama/
- OpenAI -> Ollama Server (Local Embedding model)
- https://docs.llamaindex.ai/en/stable/examples/embeddings/ollama_embedding/

- Deprecated : AgentRunner, AgentWorker -> AgentWorkflow, Workflows
- https://docs.llamaindex.ai/en/stable/changes/deprecated_terms/#agentrunneragentworker-and-related-classes

embedding

- emedding
- https://docs.llamaindex.ai/en/stable/module_guides/models/embeddings/

- VectorStoreIndex
- https://docs.llamaindex.ai/en/stable/understanding/indexing/indexing/#using-vector-store-index
- https://docs.llamaindex.ai/en/stable/api_reference/indices/vector/

Embedding model

- BGE-M3
- https://huggingface.co/BAAI/bge-m3

- jina-embedding-v0.3, multilingual-e5 및 한국어 튜닝 모델들, (한국어기준) KURE 추천

- gte-Qwen2-1.5B-instruct : length 가 매우 길어서 차용
- https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct

Local embedding model “bge-m3” by Ollama

- https://ollama.com/library/bge-m3
- ollama pull bge-m3
- https://docs.llamaindex.ai/en/stable/examples/embeddings/ollama_embedding/

```
from llama_index.embeddings.ollama import OllamaEmbedding
ollama_embedding = OllamaEmbedding(
    model_name=”bge-m3",
    base_url="http://localhost:11434",
    ollama_additional_kwargs={"mirostat": 0},
)
```

LLM, embedding 설정

```python
import nest_asyncio
nest_asyncio.apply() # 중첩 실행 허용

from llama_index.llms.ollama import Ollama
MODEL="qwen3:30b-a3b-q4_K_M" #1
HOST_IP="192.168.40.42"
OLLAMA_HOST=f"{HOST_IP}:11434"
llm = Ollama(base_url=OLLAMA_HOST, model=MODEL, request_timeout=60.0)

from llama_index.embeddings.ollama import OllamaEmbedding

EMEDDING="bge-m3"
ollama_embedding = OllamaEmbedding(
    model_name=EMEDDING,
    base_url=OLLAMA_HOST,
    ollama_additional_kwargs={"mirostat": 0},
)

from llama_index.core import Settings
Settings.embed_model = ollama_embedding
Settings.llm = llm
```

쿼리엔진, 질문나누기

```
from llama_index.core.tools import QueryEngineTool, ToolMetadata
query_engine_tools = [
    QueryEngineTool(
        query_engine=vector_query_engine,
        metadata=ToolMetadata(
            name="Q&A bot",
            description="당신은 사용자 질문에 답변하는 봇입니다",
        ),
    ),
] # 사용자 질문에 대한 답변 생성

from llama_index.core.query_engine import SubQuestionQueryEngine
query_engine = SubQuestionQueryEngine.from_defaults(
    query_engine_tools=query_engine_tools,
    llm=llm,  # 추가된 LLM 사용
    use_async=True,
) # 복잡한 질문을 하위 질문(Sub-Question)으로 분리하여 처리하고 결과를 조합

response = query_engine.query(
    # "한국형 스마트팜을 구축하기위해 필요한 기술은?"
    "한글로 답변해줘. 한국형 스마트팜을 구축하기위해 필요한 기술은?"
)
print(response)
```

QWEN3

```
Generated 5 sub questions.
[1;3;38;2;237;90;200m[Q&A bot] Q: 스마트팜에 사용되는 IoT 기술은 무엇인가요?
[0m[1;3;38;2;90;149;237m[Q&A bot] Q: 스마트팜에 필요한 자동화 시스템은 무엇인가요?
[0m[1;3;38;2;11;159;203m[Q&A bot] Q: 스마트 농업에 필요한 데이터 분석 도구는 무엇인가요?
[0m[1;3;38;2;155;135;227m[Q&A bot] Q: 스마트팜에 사용되는 재생 가능 에너지 솔루션은 무엇인가요?
[0m[1;3;38;2;237;90;200m[Q&A bot] Q: 스마트 농업에 적용되는 AI 기술은 무엇인가요?
[0m[1;3;38;2;11;159;203m[Q&A bot] A: <think>
```

```
</think>

한국형 스마트팜 구축을 위해 필요한 기술은 인공지능(AI), 사물인터넷(IoT) 기반의 센서와 제어장치, 로봇 자동화 시스템, 빅데이터 분석, 에너지 최적화 기술이 포함됩니다. 이들은 농업 전 과정의 정밀 관리와 자동화를 통해 생산성과 효율성을 극대화합니다. 또한, 표준화된 ICT 기기와 통합 제어 시스템을 통해 호환성과 유지보수성을 강화합니다.
```

```
</think>

스마트팜에 사용되는 기술에는 온실 및 축사 에너지 시스템의 최적화를 위한 센서와 제어기, 로봇을 활용한 무인화·자동화 시스템이 포함됩니다. 또한, 다양한 센서와 제어장비의 형식 및 통신 방식을 통일한 표준화된 ICT 기기들이 적용되고 있습니다.
[0m<think>
```

```
</think>

스마트팜에 필요한 자동화 시스템은 에너지 최적화 시스템과 로봇을 활용한 무인화·자동화 시스템입니다. 이는 온실 및 축사의 에너지 관리와 다양한 로봇을 통해 농업 전 과정의 통합 제어와 생산 관리를 가능하게 하며, ICT 기기의 표준화를 통해 호환성과 유지보수성을 개선합니다. 또한, 센서와 제어기 등으로 구성된 지능형 생육 관리 모델이 적용되어 실시간 데이터를 기반으로 효율적인 운영이 이루어집니다.
```

```
make sure not to mention any other details not in the context, like specific companies or other technologies not referenced here.
</think>

AI 기술은 농업 분야에서 전문가의 지식과 선도농가의 경험을 학습하여 농장 환경에 최적화된 의사결정을 지원합니다. 이는 재배 또는 사육 중인 작물과 가축에 대한 정밀한 생육 관리로 이어져 품질과 생산량을 향상시킵니다. 특정 사례로는 2세대 스마트팜 모델이 토마토 재배와 돼지 사육에 적용되며, 향후 다양한 품목으로 확장될 계획입니다.
```

```
knowledge and stick strictly to the provided text.
</think>

제공된 맥락 정보에서 스마트팜에 사용되는 재생 가능 에너지 솔루션에 대한 구체적인 언급은 없습니다. 다만 3세대 스마트팜 모델은 온실 및 축사 에너지 시스템의 최적화를 통해 에너지 효율성을 높이는 기술을 포함하고 있으며, 이는 통합제어와 자동화를 통해 에너지 사용을 개선하는 방향으로 설명되어 있습니다. 재생 가능 에너지(예: 태양광, 풍력 등)에 대한 직접적인 내용은 포함되지 않았습니다.
```

Add agent

https://docs.llamaindex.ai/en/stable/getting_started/starter_example_local/#adding-rag-capabilities

```python
async def search_documents(query: str) -> str:
    response = await query_engine.aquery(query)
    return str(response)

from llama_index.core.agent.workflow import AgentWorkflow

# Create an enhanced workflow with both tools
agent = AgentWorkflow.from_tools_or_functions(
    [search_documents],
    llm=Settings.llm,
    system_prompt="""You are a helpful assistant that can search through documents to answer questions. 한글로 답변해
줘.""",
)

import asyncio

# Now we can ask questions about the documents or do calculations
async def main():
    response = await agent.run(
        "한국형 스마트팜을 구축하기위해 필요한 기술은?"
    )
    print(response)


# Run the agent
if __name__ == "__main__":
    asyncio.run(main())
```

Sub question query?

QWEN3

한국형 스마트팜을 구축하기 위해 필요한 주요 기술은 다음과 같습니다:

1. **IoT 센서 기술**:
   - 토양 수분, 온도, pH, 광량, 공기 습도 등을 실시간으로 모니터링하는 센서가 필수적입니다. 이 데이터는 작물 생장 조건을 최적화하는 데 사용됩니다.

2. **자동화 시스템**:
   - 관개 시스템, 비료 공급 장치, 환기 시스템 등이 자동으로 작동하도록 하는 기술입니다. 예를 들어, 센서 데이터에 따라 자동 관개가 이루어집니다.

3. **데이터 분석 및 AI**:
   - 수집된 데이터를 분석해 작물 수확량 예측, 병해충 감지, 최적 수확 시기 도출 등에 활용하는 인공지능(AI) 기술이 필요합니다.

4. **로봇 기술**:
   - 씨앗 뿌리기, 수확, 풀 뽑기 등 반복적 농업 작업을 수행하는 농업 로봇이 적용됩니다.

5. **드론 기술**:
   - 공중에서 농지의 상태를 촬영해 병충해 발생 여부나 성장 상태를 확인하는 데 사용됩니다.

6. **클라우드 컴퓨팅 및 네트워크 인프라**:
   - 대량의 데이터를 저장하고 분석하기 위한 클라우드 시스템과 5G, 위성 통신 같은 고속 네트워크가 필수적입니다.

7. **에너지 효율화 기술**:
   - 태양광 패널이나 에너지 저장 시스템(ESS)을 통해 재생 에너지를 활용해 운영 비용을 줄이는 기술이 중요합니다.

8. **지리정보시스템(GIS)**:
   - 농지의 지형, 토양 특성 등을 지도로 시각화해 농업 전략 수립에 활용합니다.

또한, 한국의 기후 및 토양 조건에 맞춘 **정밀 농업(Precision Agriculture)** 기법과 **스마트팜 표준화 시스템**도 고려해야 합니다. 이 기술들은 농업 생산성 향상, 자원 낭비 감소, 환경 보호에 기여합니다.

책 예제 코드 결과

OpenAI LLM

```
Generated 6 sub questions.
[Q&A bot] Q: 한국형 스마트팜을 구축하기 위해 필요한 주요 기술은 무엇인가요?
[Q&A bot] Q: 스마트팜 구축 시 고려해야 할 한국의 기후 및 환경적 요인은 무엇인가요?
[Q&A bot] Q: 한국형 스마트팜에 적합한 자동화 기술은 무엇인가요?
[Q&A bot] Q: 스마트팜 운영에 필요한 데이터 분석 기술은 무엇인가요?
[Q&A bot] Q: 한국형 스마트팜에서 활용할 수 있는 IoT 기술은 무엇인가요?
[Q&A bot] Q: 스마트팜의 에너지 효율성을 높이기 위한 기술은 무엇인가요?
[Q&A bot] A: 빅데이터 분석 기술
[Q&A bot] A: 에너지 효율성을 높이기 위한 기술은 에너지 최적화 및 로봇 자동화 등 스마트팜 통합시스템을 구현하는 것입니다.
[Q&A bot] A: When establishing a smart farm in Korea, it is important to consider the country's aging agricu
[Q&A bot] A: 사물인터넷(IoT) 기술을 활용하여 한국형 스마트팜에서는 농산물의 생육환경을 최적상태로 관리하고 노동력 절감 및 생산성 향상을 구
[Q&A bot] A: 한국형 스마트팜을 구축하기 위해 필요한 주요 기술은 사물인터넷, 빅데이터, 인공지능, 로봇 등을 활용하여 농산물의 생육환경을 최적
[Q&A bot] A: 자동화 기술로는 인공지능을 활용한 생육환경 관리, 생산량 및 품질 향상을 위한 정밀한 생육관리, 환기/보온/영상감시/관수/난방/안
한국형 스마트팜을 구축하기 위해 필요한 기술은 사물인터넷, 빅데이터, 인공지능, 로봇 등을 활용하여 농산물의 생육환경을 최적상태로 관리하고 노동력
```