



TGO

AI와 떠나는 더 자유로운 여행, Travel GO!

발표자: 이민우
DEVOCEAN TECH DAY

발표자 소개

이름 이민우

좌우명 “사람과 AI를 even하게 연결하자”

경험

- 수원대학교 데이터과학부 졸업 예정
- OpenLab 2기 Cloud Native 기반 여행을 위한 AI Agent 개발
- OpenLab 1기 LLMOps
- SKT FLY AI 4기
- SKT DEVOCEAN YOUNG 2기
- Multi-agent 관련 논문 1편 등재



SECTION 01

AI와 여행

1-1. 같이 여행 가실래요?

SECTION 02

TGO의 Agent 고도화 전략

2-1. 주요 기술스택

2-2. 그래프 구조와 상태 관리 (LangGraph)

2-3. 여행 계획 생성과 개인화 (AGI)

2-4. 할루시네이션 방지 피드백 시스템 (Self-Reflection)

2-5. 여행 일정 관리 및 업데이트 (Trust Call)

2-6. 데이터가 없어도 데이터베이스가 되는 DuckDB

CONTENTS



SECTION 01

시와 여행

같이 여행 가실래요..?

같이 여행 가실래요..?

기존 여행 process

언제 어디로
일정 맞추기
예산

숙소 예약
로밍
맛집 검색
항공권 예약

Travel Go!

여행이란?

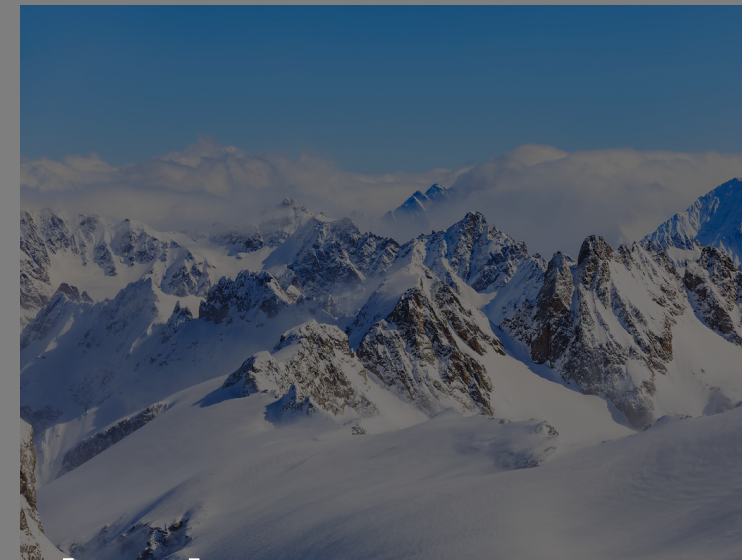
산 넘어 산

여행 제안



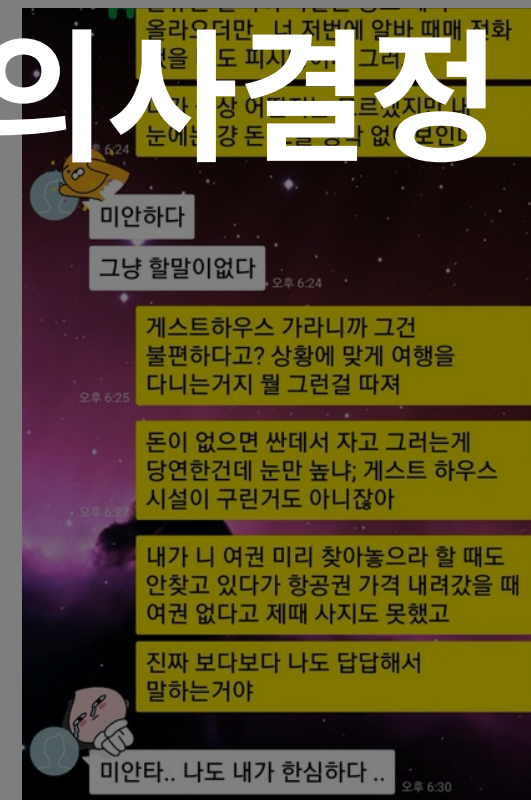
“고도의 데이터 기반 의사결정 결정체”

Travel
can't Go..



price#1	way to sleep	price#2	extra fee(entrance, tour, ticket etc.)	price#3	transportation time	비고
80,100	이탈리아	40,000.00	1박 2일 2인 1실	18,000.00	12:30-15:00(2H 30M)	이탈리아 항공
80,100	이탈리아	35,200.00	이탈리아 숙박 1박 2일	22,000.00	21:13-22:30(2H 20M)	Gare du Nord 역
42,067	6. Zimacká schody, Prague 01, Prague, 110 00, Czech Republic	28,654.67	이탈리아 숙박 1박 2일	22,000.00	07:20-09:00(1H 40M)	AIRFRANCE
21,071	Genesio Markt 18, 01, Wiener Stadt, Wien, 1010, Austria	42,488.00	이탈리아 숙박 1박 2일	22,000.00	08:08-10:30(2H 22M)	
19,948	1011 István Street 6, Budapest, Budapest 1011, Hungary	22,672.50	이탈리아 숙박 1박 2일	22,000.00	07:00-10:15(3H 15M)	FLUXUS
31,997.00	Genesio Markt 18, 01, Wiener Stadt, Wien, 1010, Austria	42,488.00	이탈리아 숙박 1박 2일	22,000.00	12:35-13:50(1H 20M)	Venice SL
30,174	Genesio Markt 18, 01, Wiener Stadt, Wien, 1010, Austria	42,488.00	이탈리아 숙박 1박 2일	22,000.00	19:00-21:05(2H 5M)	
44,901	Genesio Markt 18, 01, Wiener Stadt, Wien, 1010, Austria	42,488.00	이탈리아 숙박 1박 2일	22,000.00	17:05-18:35(1H 30M)	

지킬 수 있을지 모르는 일정



의견 차이로 인한 실패 사례

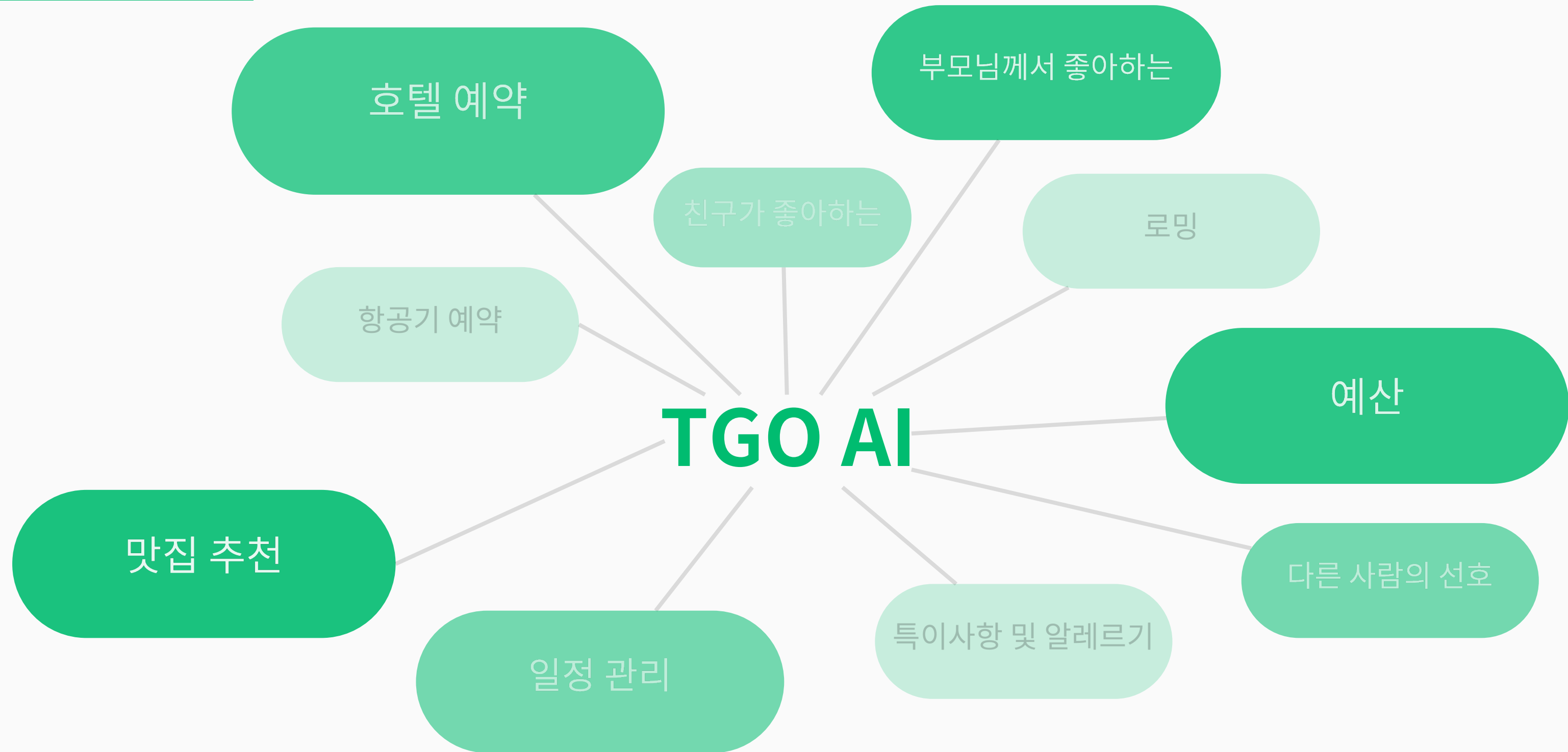


여행은 다음 기회에..

같이 여행 갈래요..?

여행 = 큰 결심

같이 여행 갈래요..?



“여행을 좀 더 **자유롭고** **똑똑하게** 갈 수 있게 돕자”

SECTION 02

TGO의 Agent 고도화 전략

AGENTIC ORCHESTRATION LAYER



FOUNDATION MODELS



DATABASE



DuckDB

EVALUATION



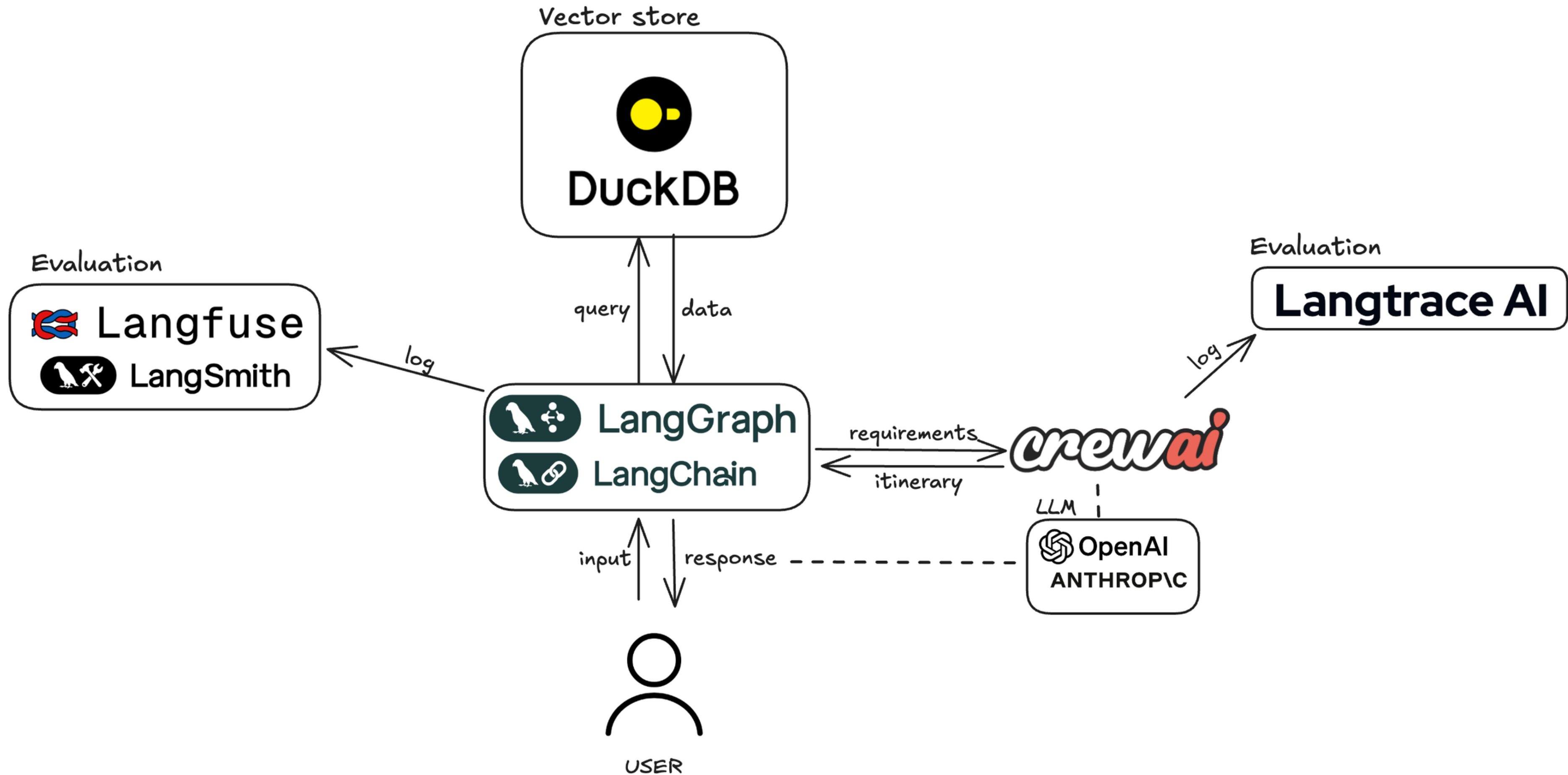
Key feature

여행 일정 생성

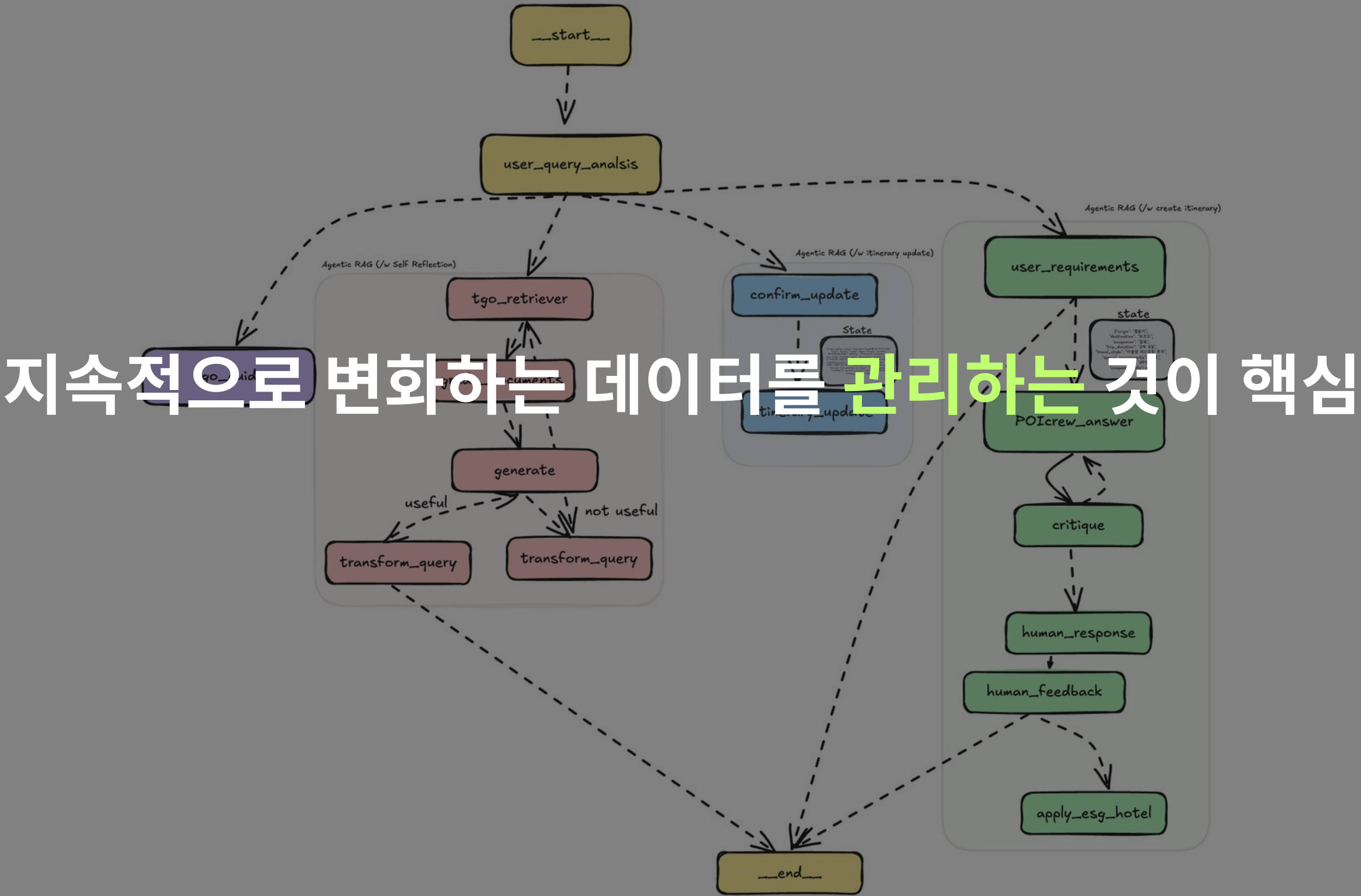
정보 검색

여행 계획 관리(업데이트)

Service workflow



Langgraph workflow



시간이 지나도 변하지 않는 건 “State” 입니다.

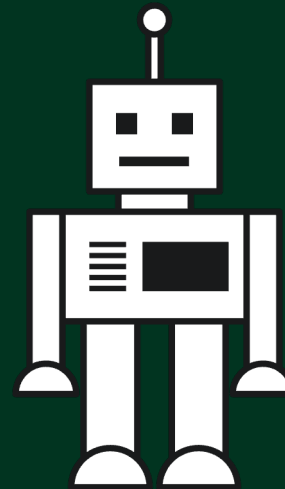


User

도착지: 없음
예산: 없음
일정: 없음

도착지: 일본
예산: 100만원
일정: None

→

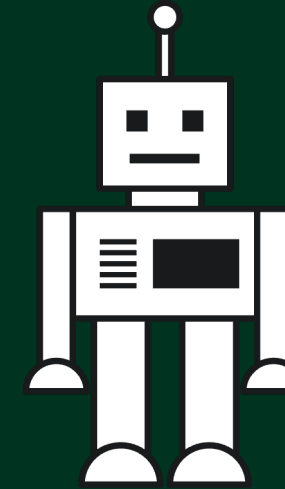


Agent

도착지: 일본
예산: 100만원
일정: 없음

여행 일정 생성 중...

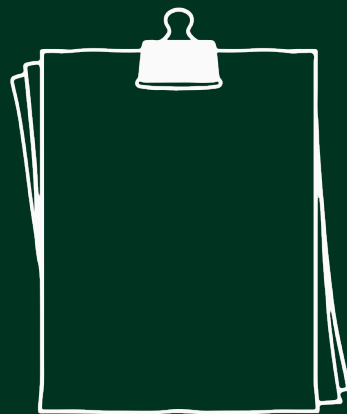
→



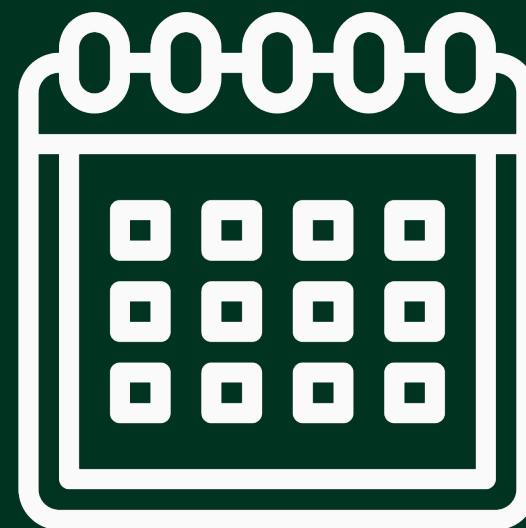
Agent

도착지: 일본
예산: 100만원
일정: 일본 오사카 자전거 여행

State 관리를 통해 할 수 있었던 것



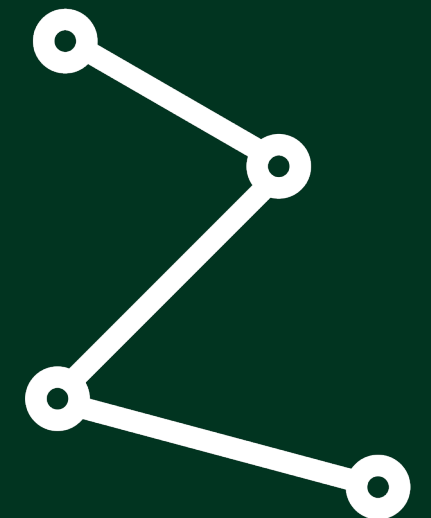
사용자 요구사항에 맞춘 일정표 생성



지속적인 일정 관리

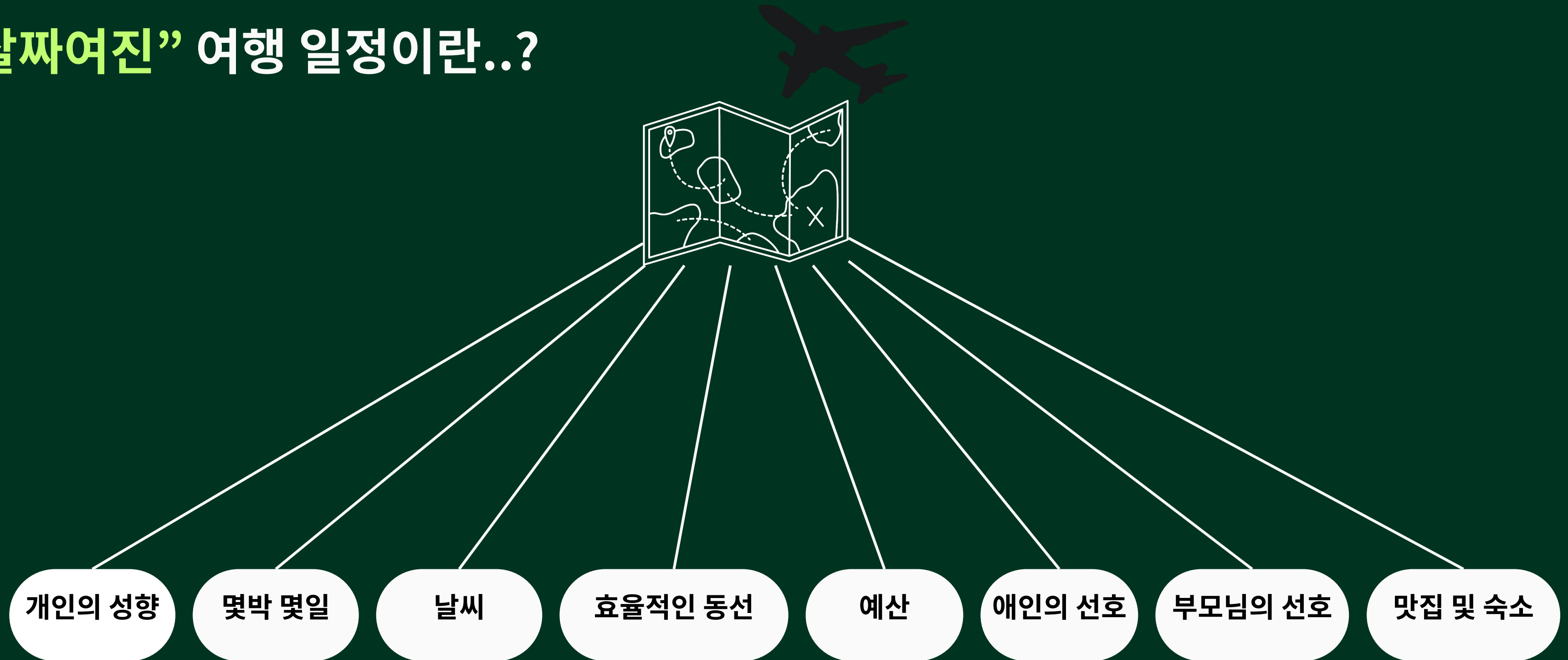


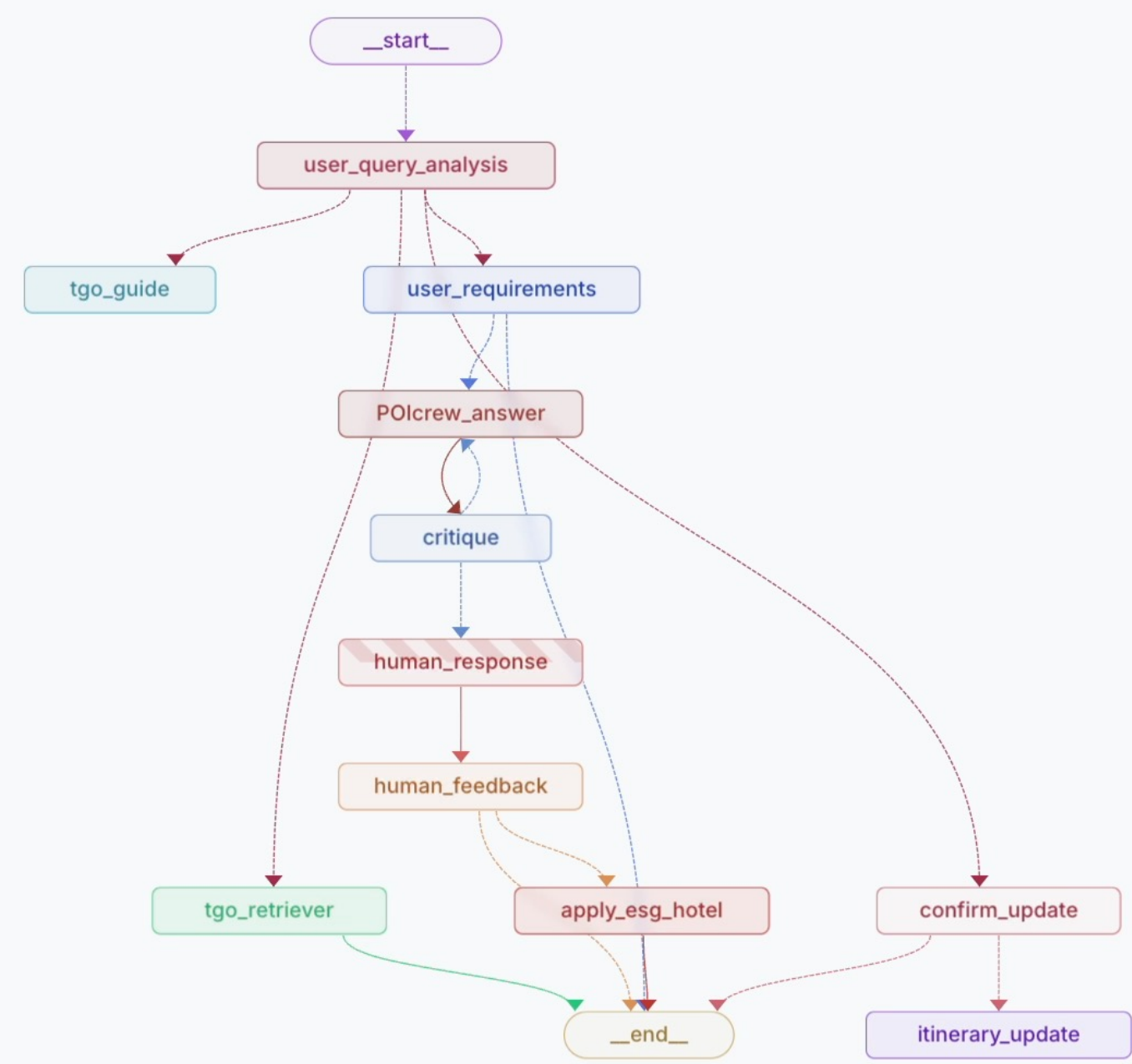
self-reflection



유저 쿼리 라우팅

“잘짜여진” 여행 일정이란..?





New Thread

Submit your input to run the assistant

Input



Messages

Required

+ Message



Configurable



Submit

OUTPUT



MBTI-LIKE 여행 성향 지표



Q: 어떻게 하면 다양한 정보를 고려해서 여행 일정을 만들 수 있을까?

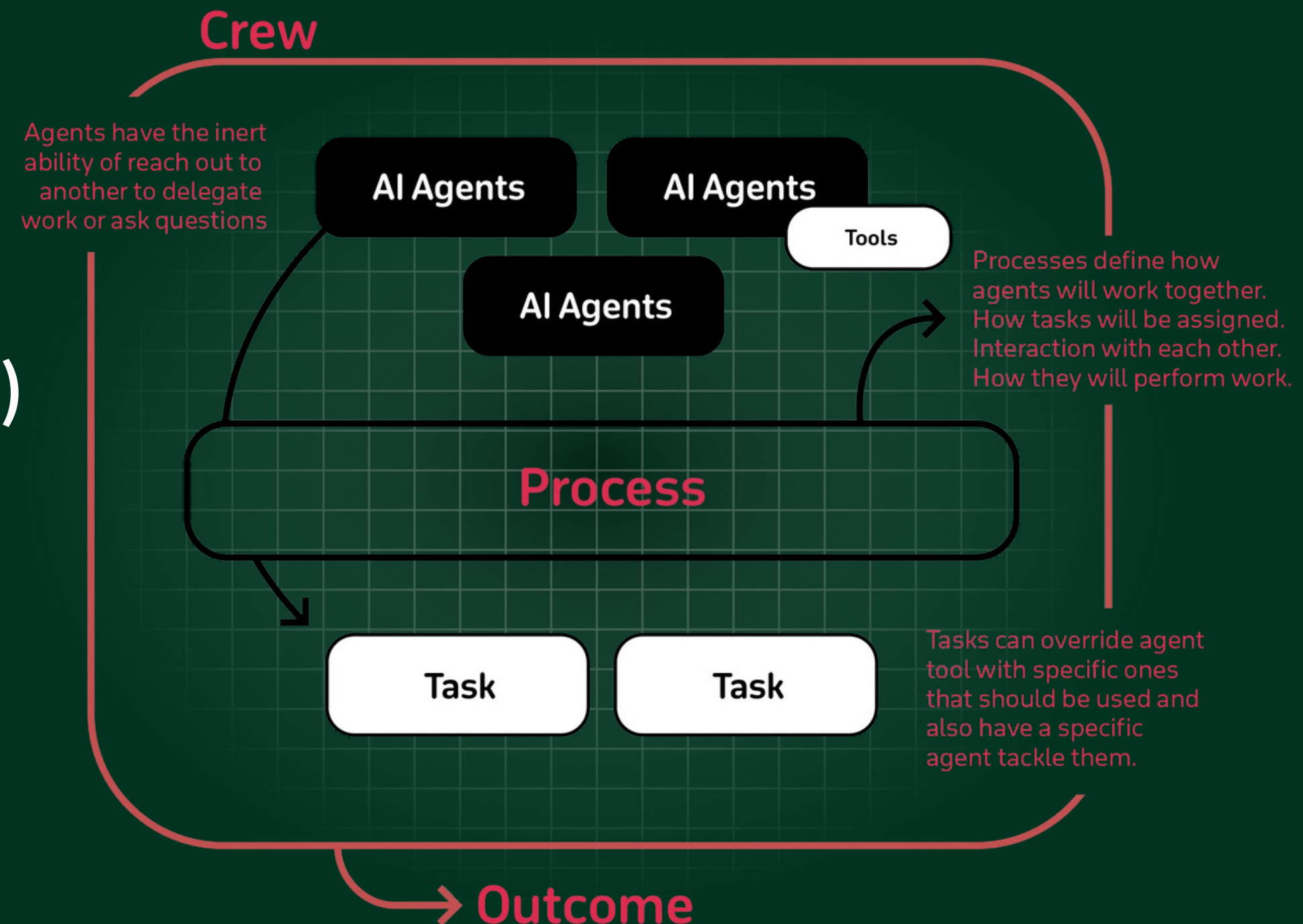
A: AGENTIC REASONING DESIGN PATTERNS을 고려

1. REFLECTION

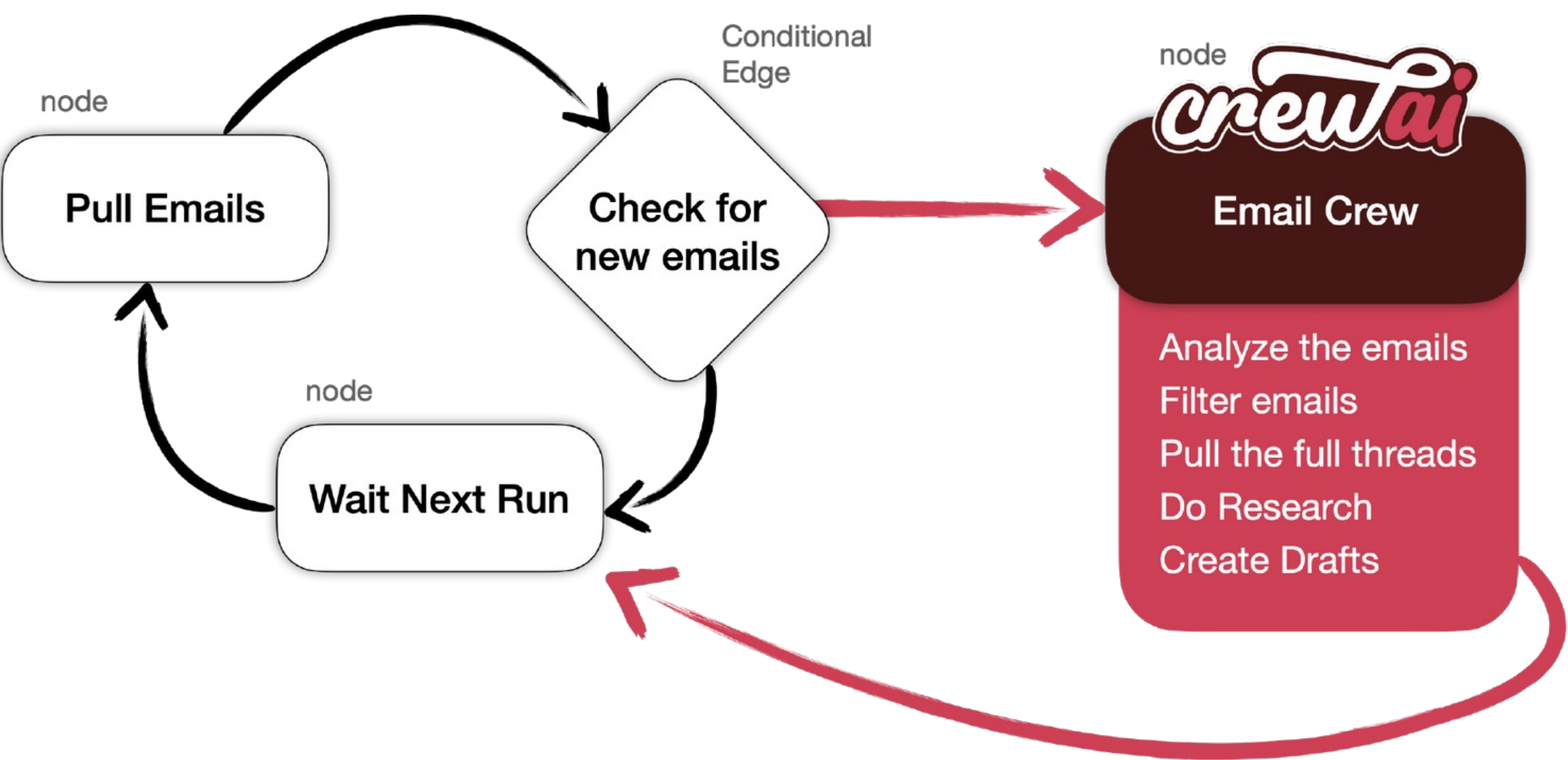
2. TOOL USE(API calls)

3. PLANNING(decide on steps for task)

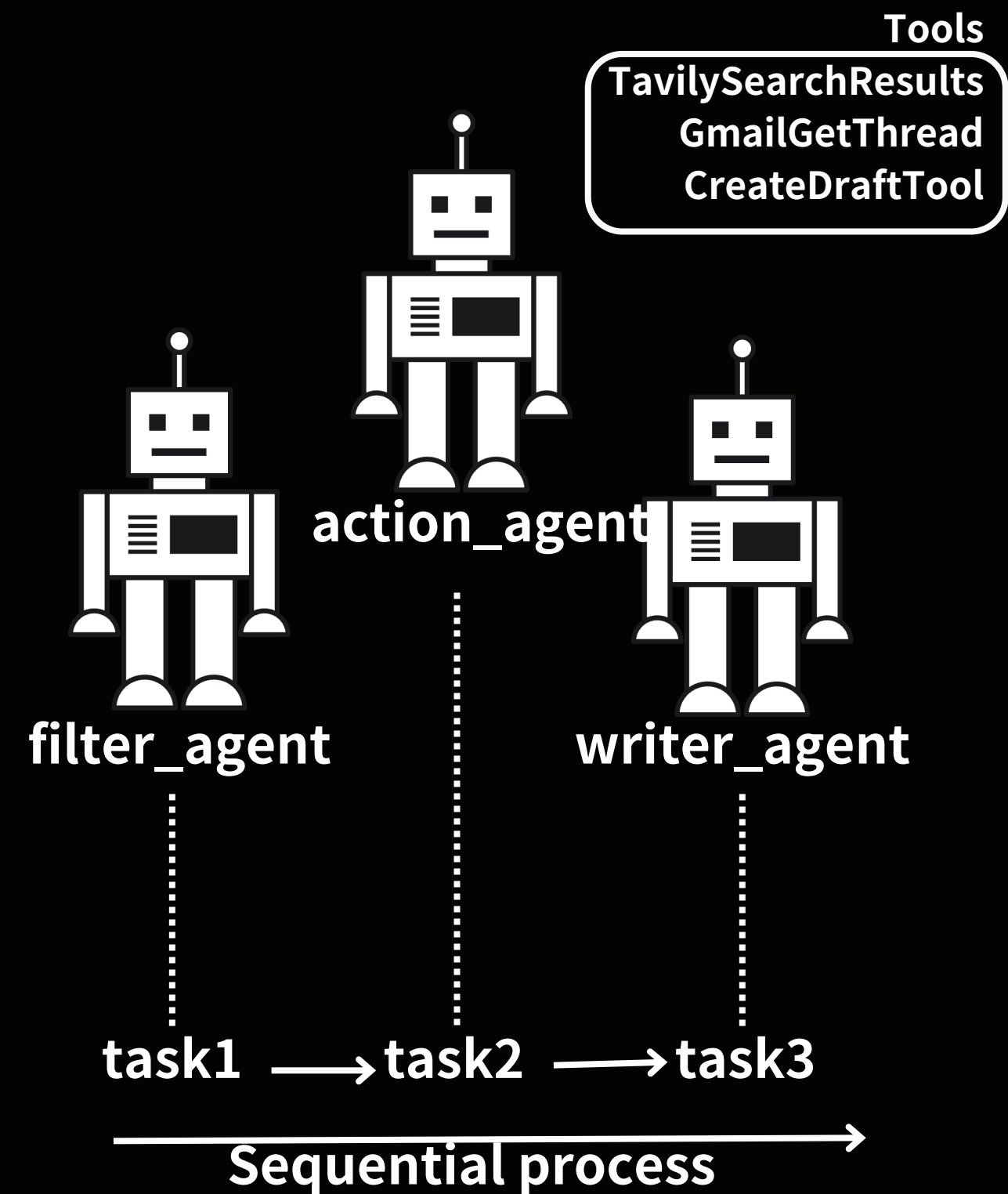
4. MULTI-AGENT COLLABORATION



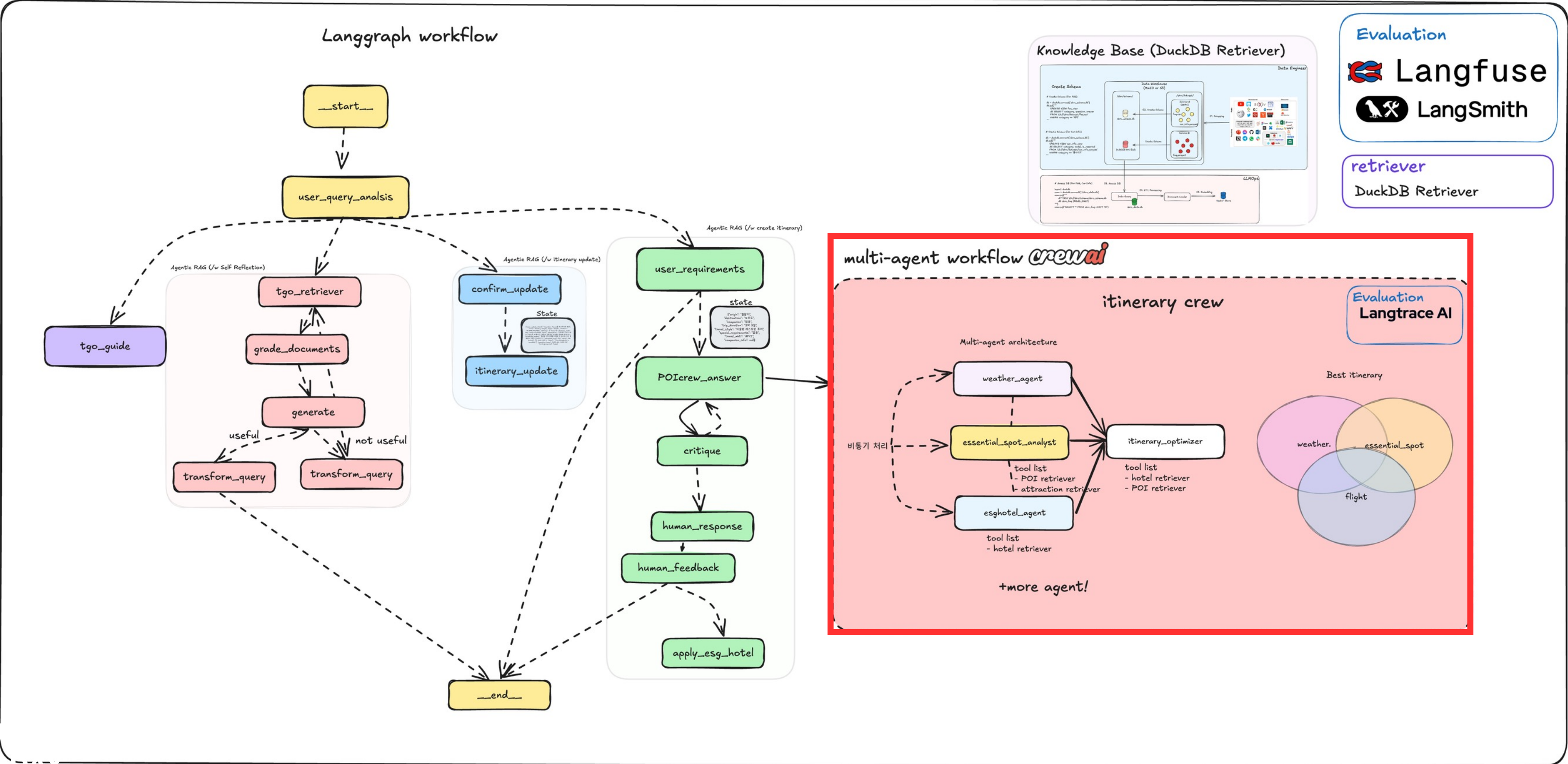
Langgraph-crewai example



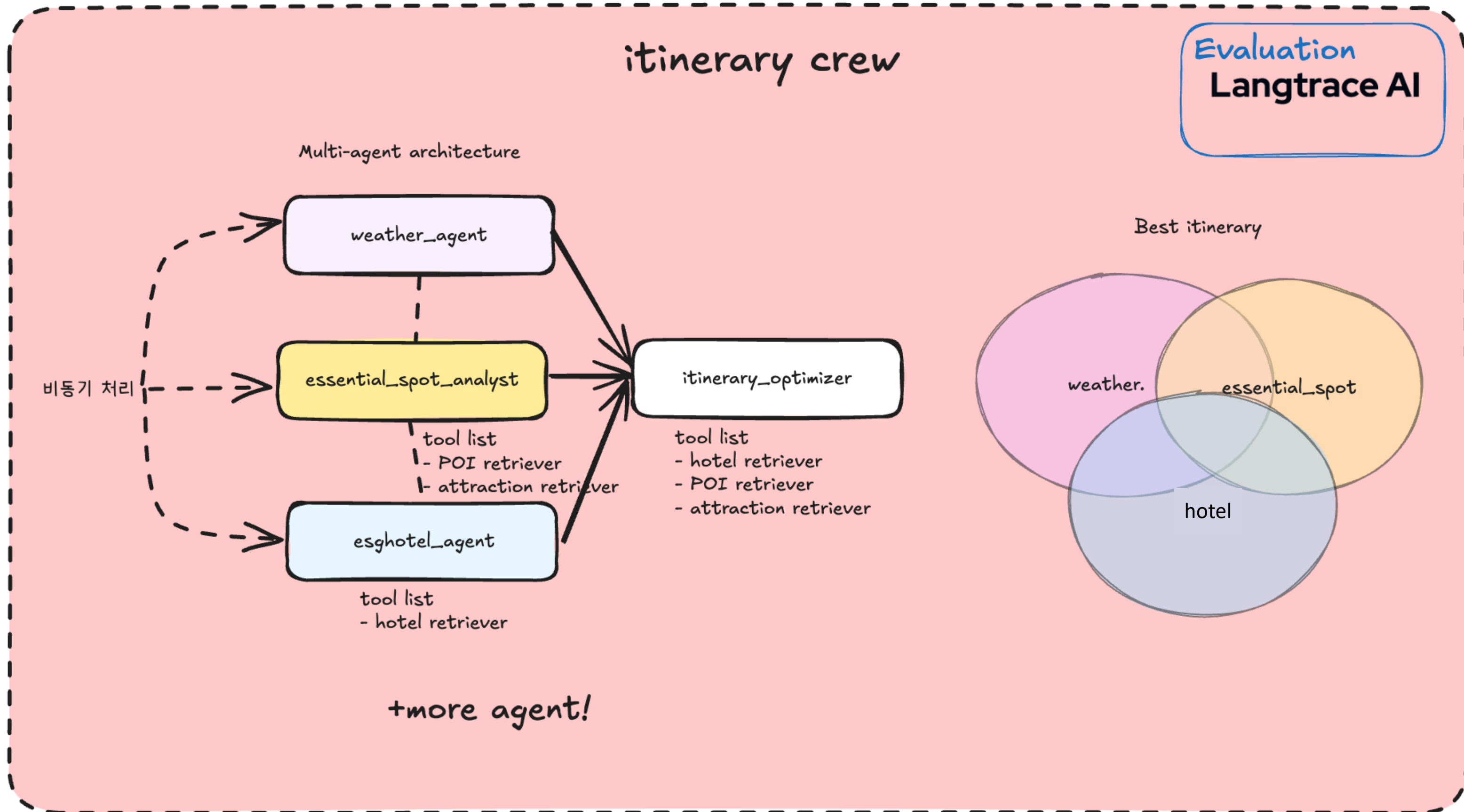
“Email crew”



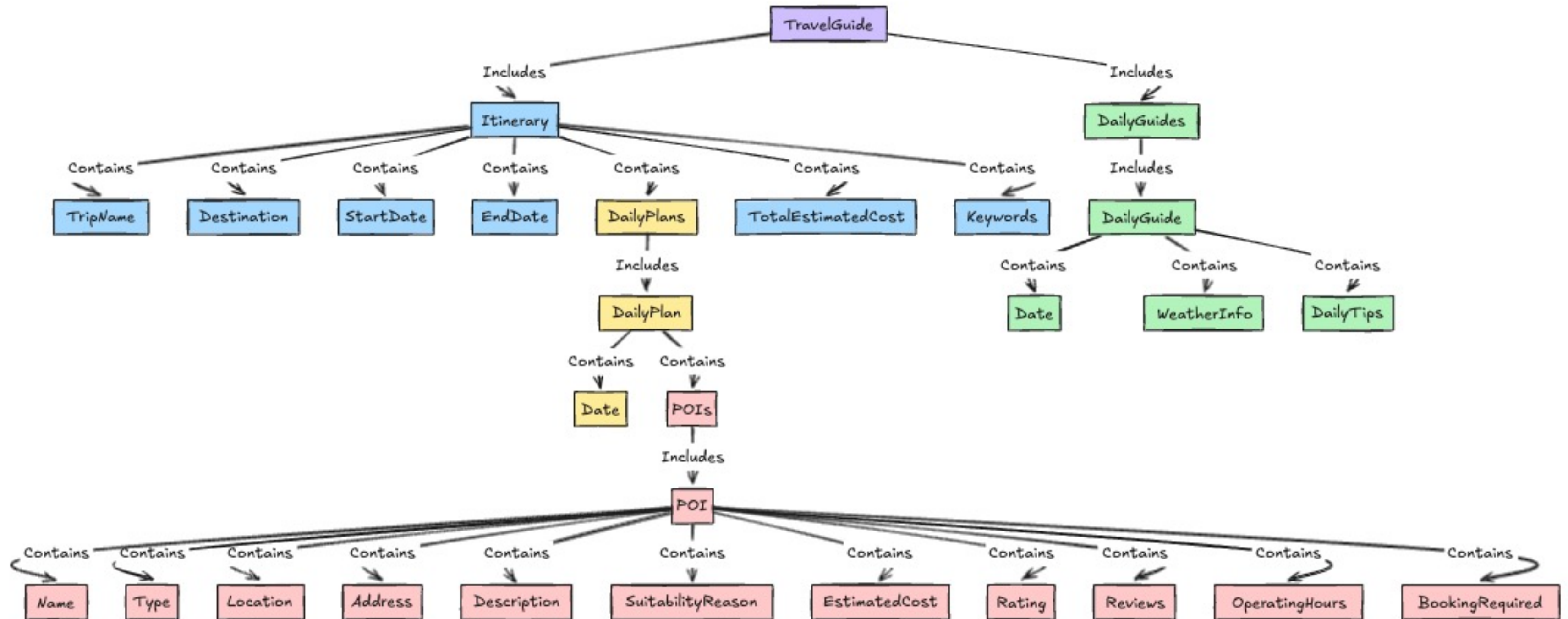
Agentic RAG workflow: LangGraph *crewai*

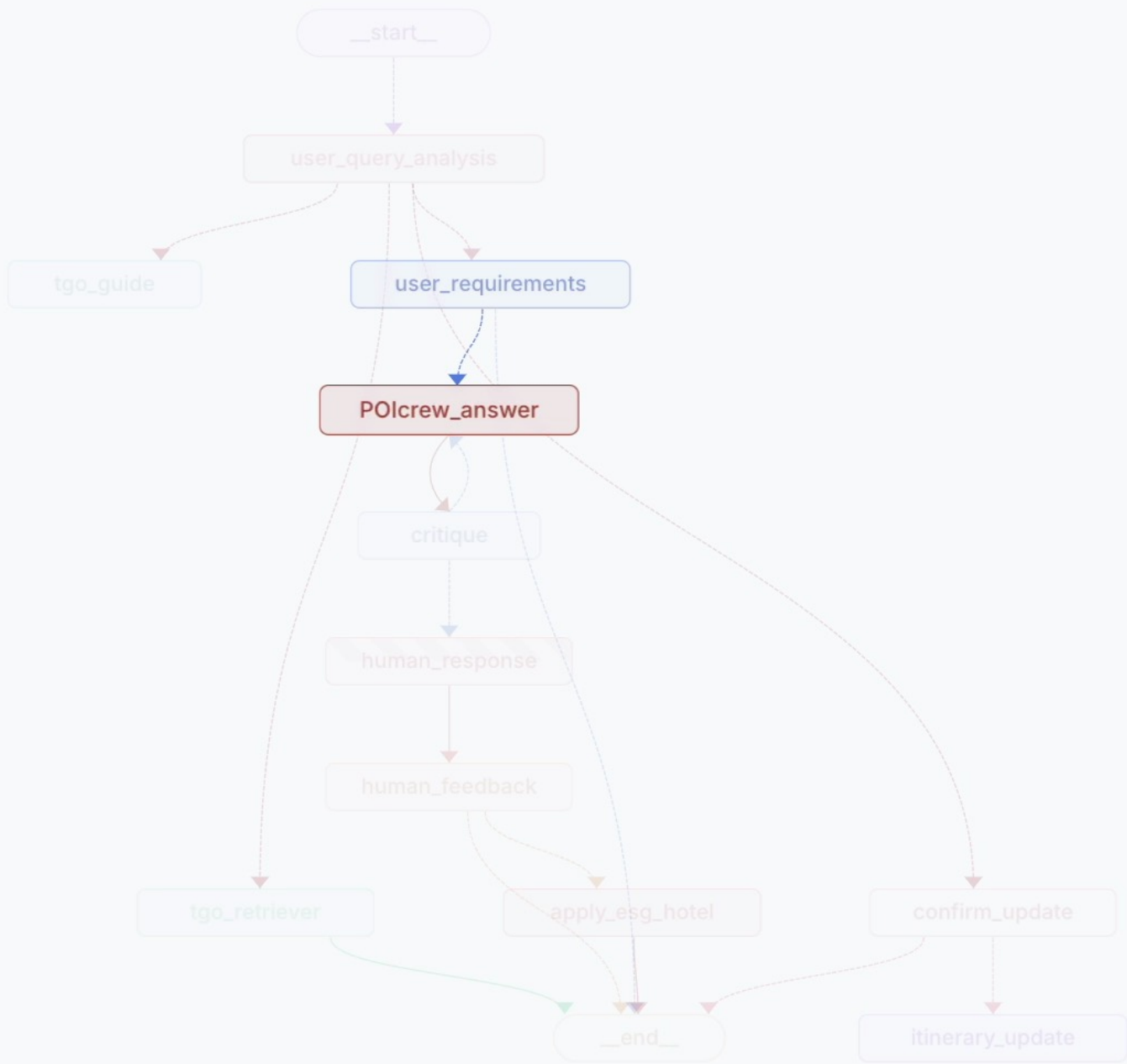


multi-agent workflow *crewai*



Itinerary data architecture





User Require ▾

true



__start__

아이들 놀이시설 포함해야하고 부모님을 위한 온천 체험을 해야해 그리고 너무 짬 음식은 피해줘 호텔은 럭셔리 호텔로

Fork 2 / 2 ⬅ ➡



user_query_analysis

Query Analysis ▾

user_requirements



user_requirements

Requirements ▾

{ "origin": "부산", "destination": "삿포로", "companion": "가족", "trip_duration": "5일", "travel_style": "가족 중심의 휴식과 체험", "special_requirements": "아이들 놀이시설 포함, 부모님을 위한 온천 체험, 짬 음식 피하기", "travel_mbti": "APCI", "companion_info": [{ "name": "소민", "relationship": "딸", "notes": ["동물원과 놀이 공원 좋아함"] }, { "name": "서준", "relationship": "남편", "notes": ["로컬 시장 탐방 즐김", "일본 전통음식에 관심 많음"] }, { "name": "정희", "relationship": "엄마", "notes": ["온천 선호", "무리 없는 일정 희망", "차분히 풍경 즐길 수 있는 장소 선호"]] }] }

Messages ▾

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Deleted Message ID

Human ID

User Require ▾

false

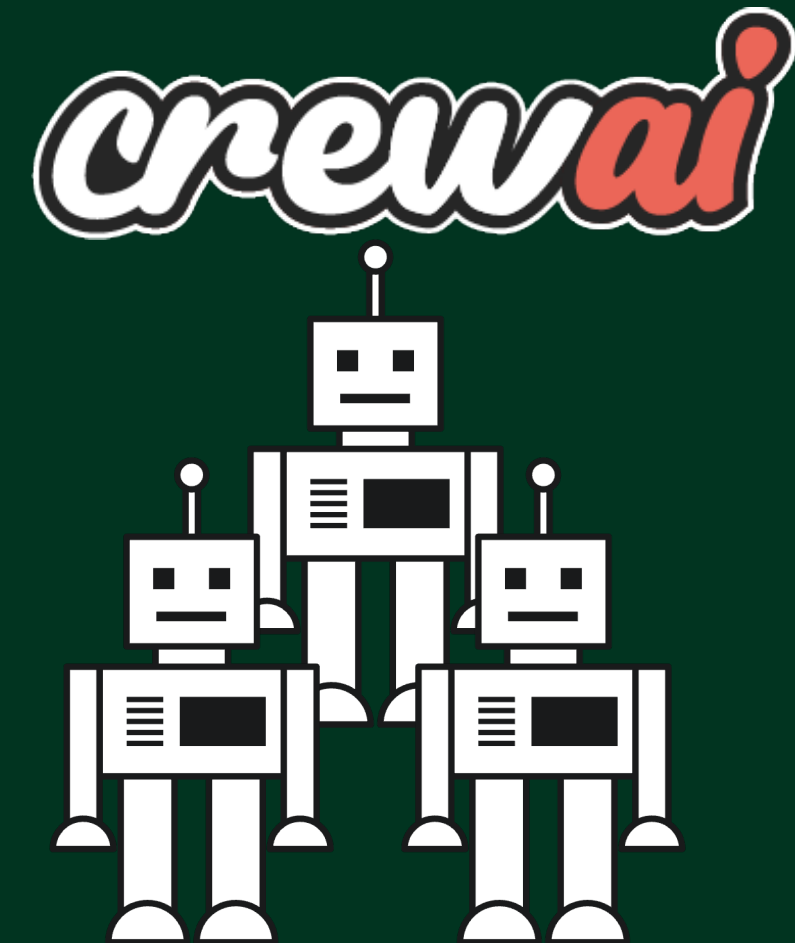
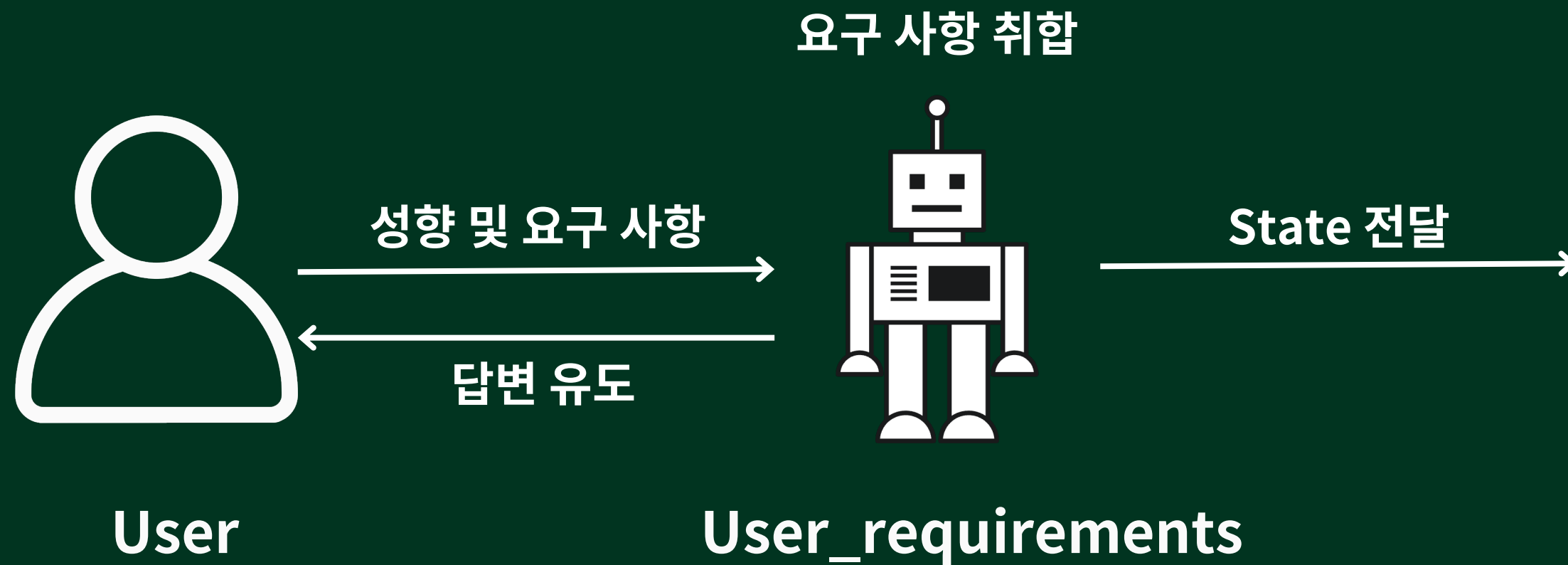
Fork 2 / 2 ⬅ ➡

4,143 168.05s 2 minutes ago



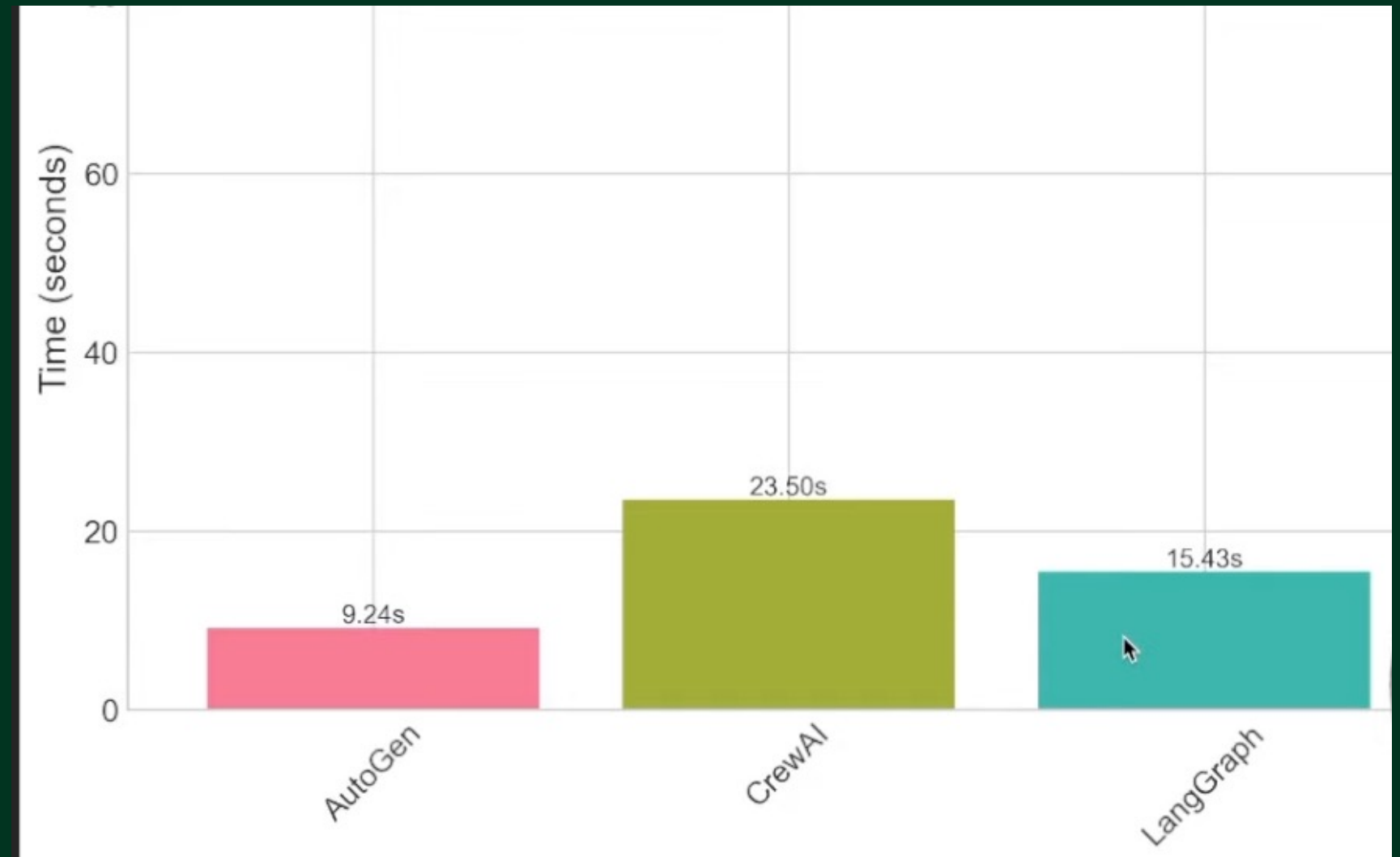
POIcrew_answer

Crewai를 통한 여행 일정 생성 pipeline

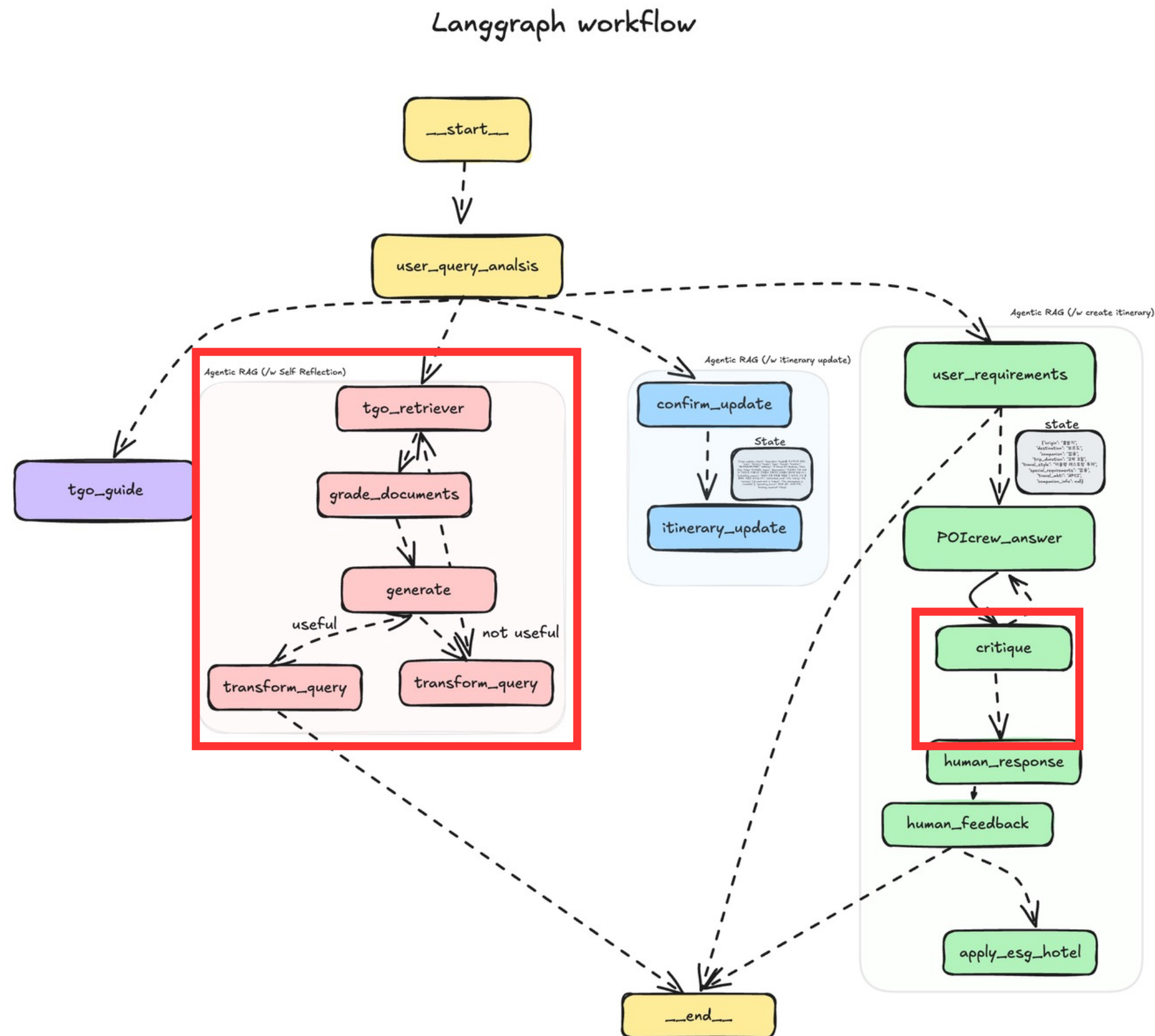


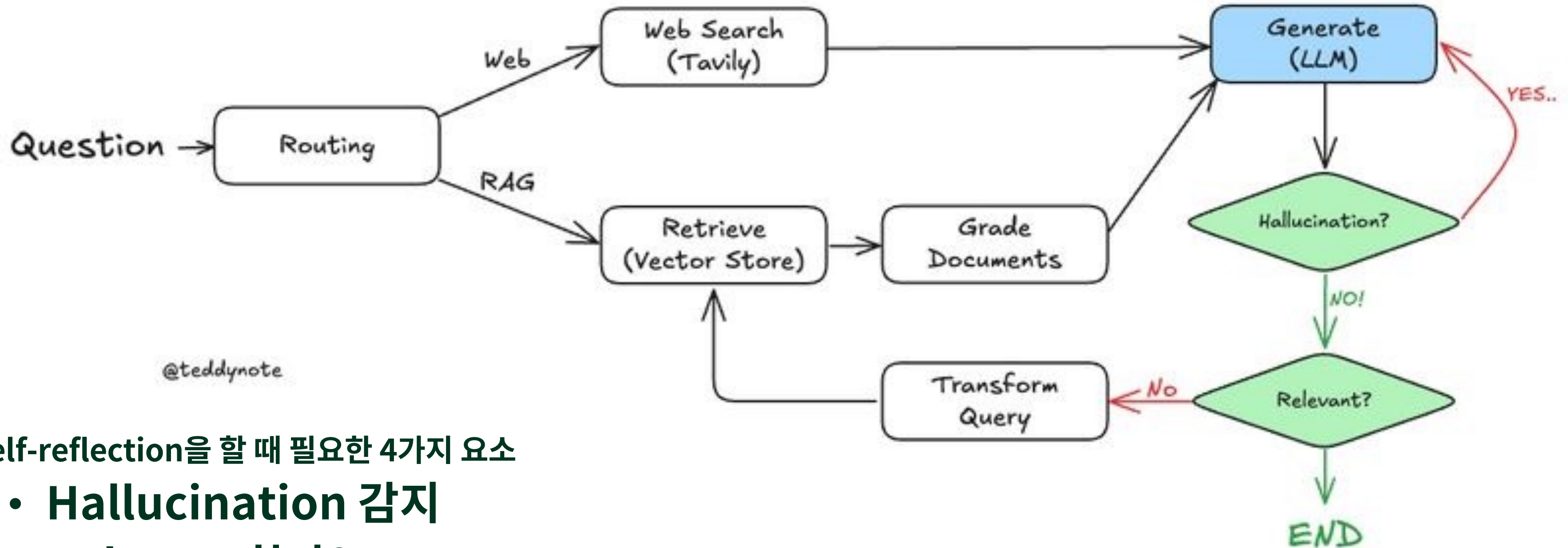
어떤 프레임 워크를 써야하는가?

1. Autogen
 2. Crewai
 3. langGraph
 4. taskflowai
- 등등...



“한번 더 검증” Self-reflection





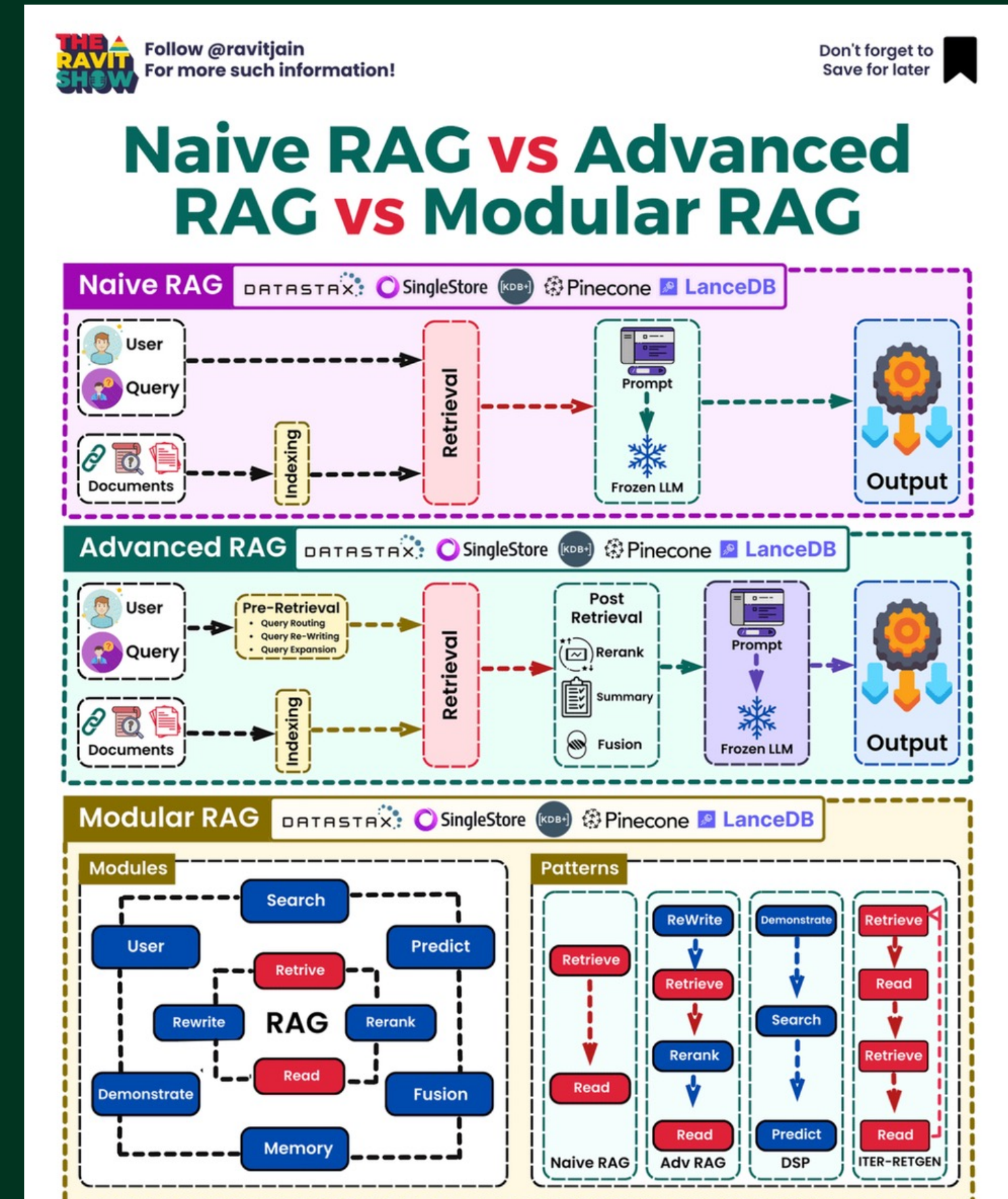
Self-reflection을 할 때 필요한 4가지 요소

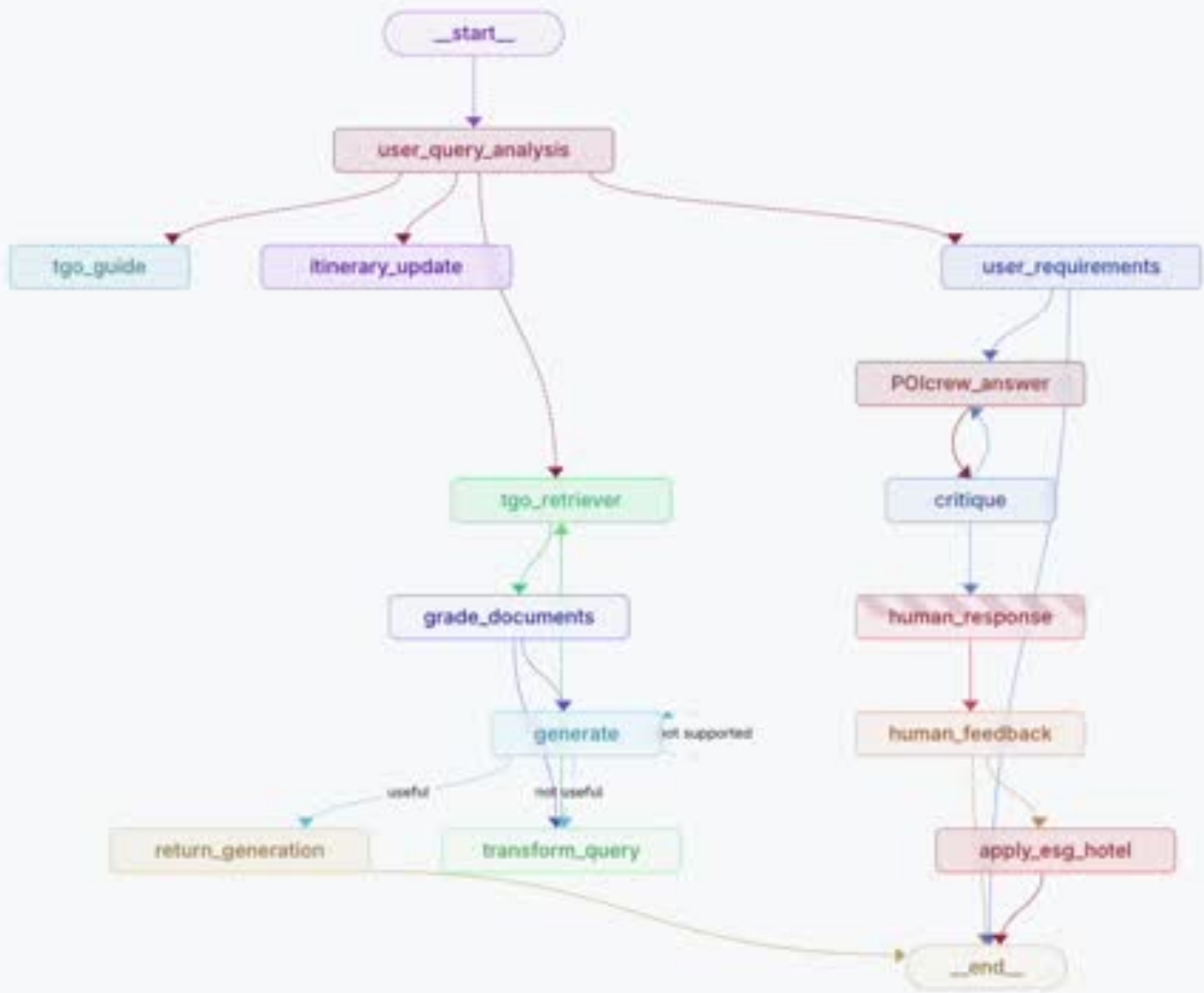
- Hallucination 감지
- Relevant한가?
- Re-ranking
- Transform Query

출처: 테디노트

Advanced RAG vs Modular RAG

항목	Advanced RAG	Modular RAG
구조	단일화된 End-to-End 파이프라인 설계, 데이터 처리와 검색이 결합됨	독립적인 모듈(검색, 생성, 조합)이 상호작용하며 분리되어 설계됨
유연성	특정 워크플로우나 데이터셋에 최적화 가능, 변경 시 많은 재설계 필요	각 모듈 교체 또는 확장 가능, 다양한 데이터 소스와 통합 용이
확장성	새로운 기능 추가 시 기존 시스템 복잡도 증가	필요한 모듈만 추가하거나 교체하여 간단히 확장 가능
성능 최적화	워크플로우에 맞춘 고도로 최적화된 성능 제공	개별 모듈 최적화로 유연한 성능 조정 가능
디버깅	통합 시스템의 복잡성으로 인해 문제 분석이 어렵고 시간 소요	모듈별로 독립적으로 디버깅 가능
재사용성	기존 코드나 기능의 재사용이 제한적	모듈을 다른 프로젝트나 시스템에 재사용 가능
사용 사례	대규모 QA 시스템, 특정 도메인에 특화된 정보 검색	멀티도메인 응용, 개인화 추천 시스템





New Thread

Submit your input to run the assistant

Input

Messages

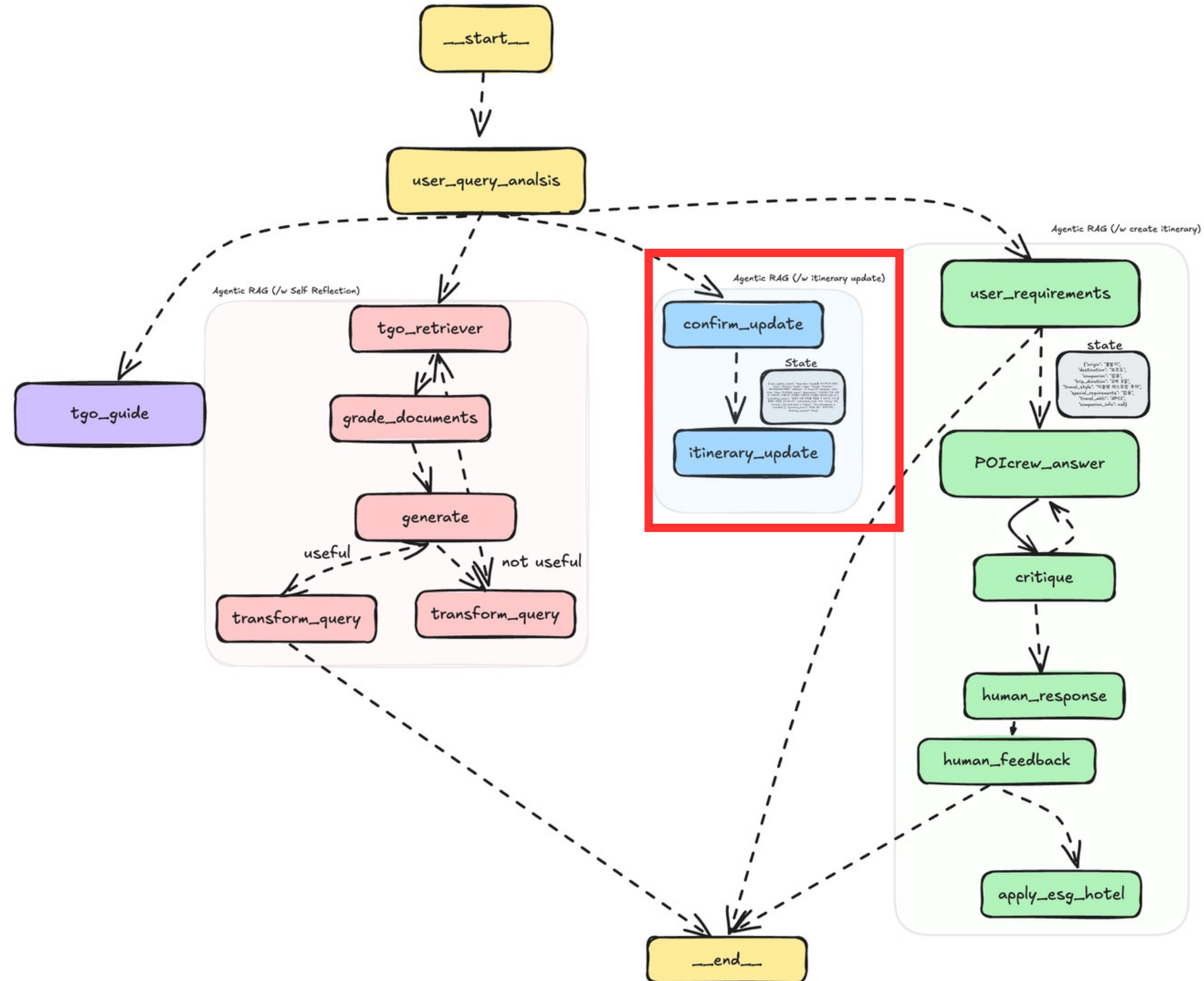
Required

HUMAN

+ Message

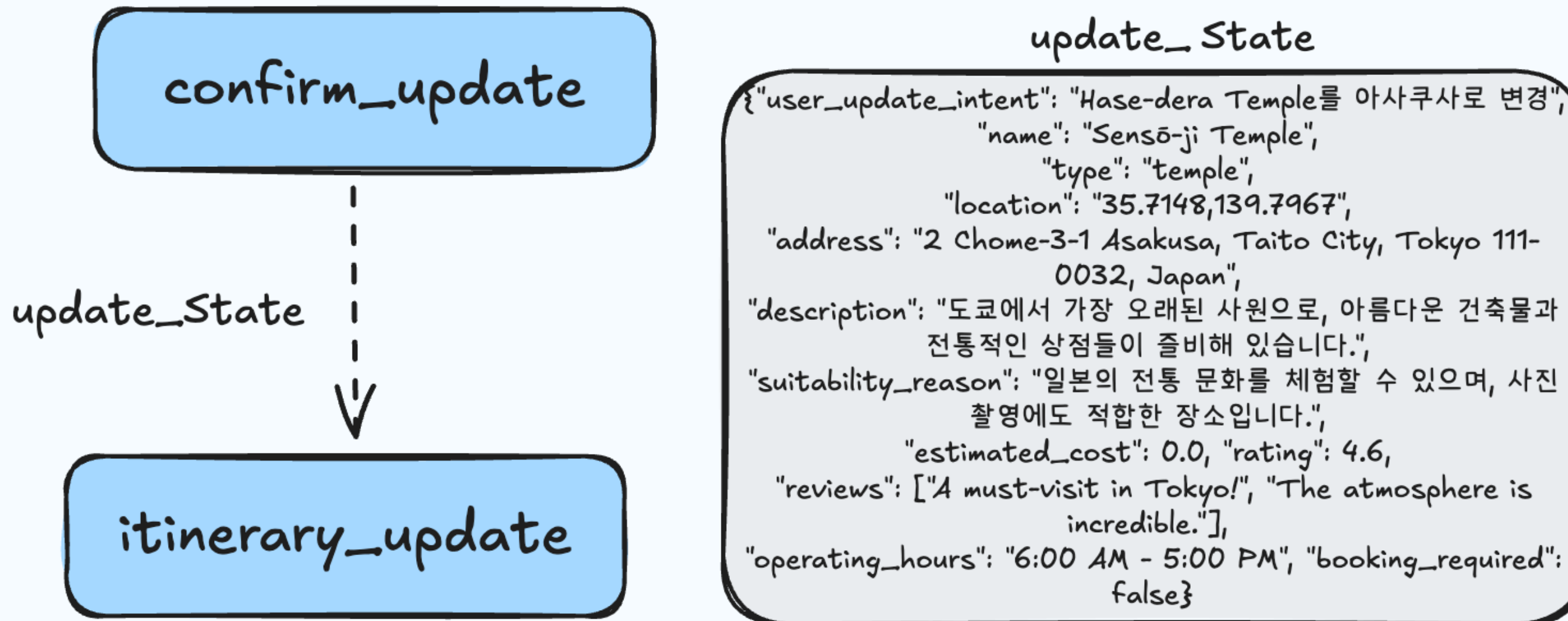
여행 일정 업데이트

Langgraph workflow



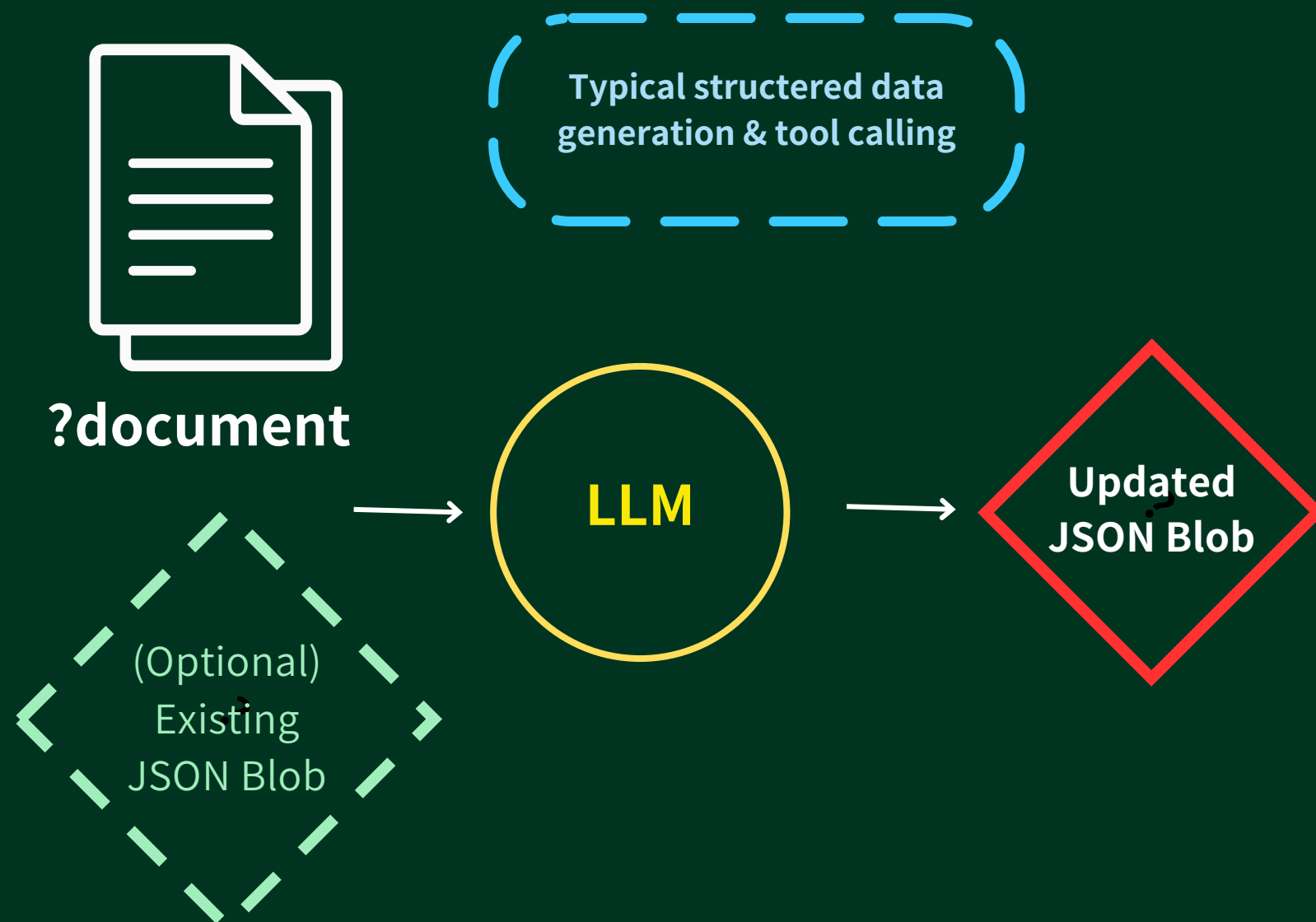
Agentic RAG (/w itinerary update)

Agentic RAG (/w itinerary update)

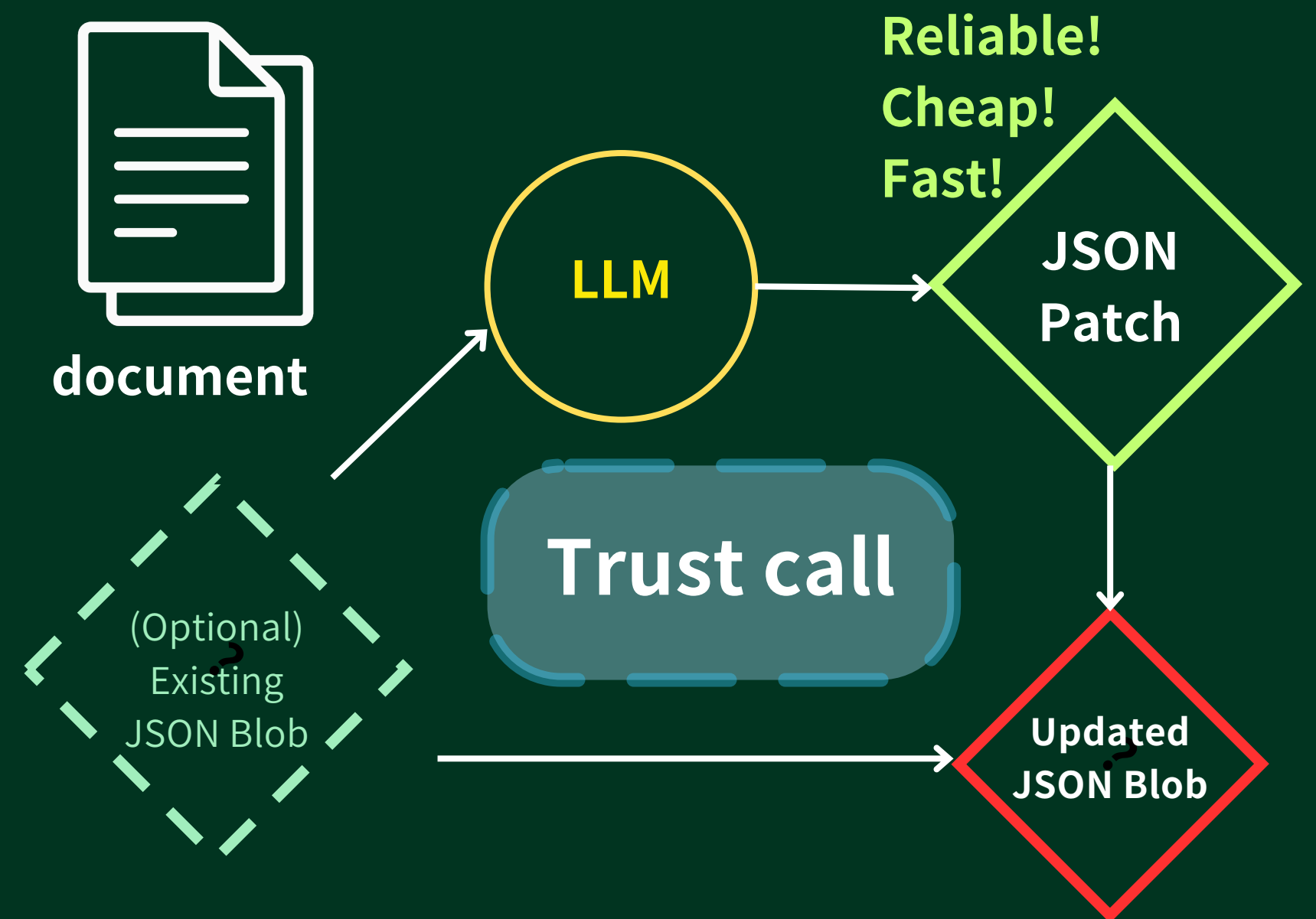


Trust call을 이용한 여행 일정 업데이트

기존 JSON 업데이트



Trust call을 이용한 JSON 업데이트



수정을 위한 JSON Patch

```
[  
  { "op": "replace", "path": "/age", "value": 26 },  
  { "op": "add", "path": "/city", "value": "Seoul" }  
]
```

LLM



Input JSON data

```
{ "name": "Alice", "age": 25 }
```

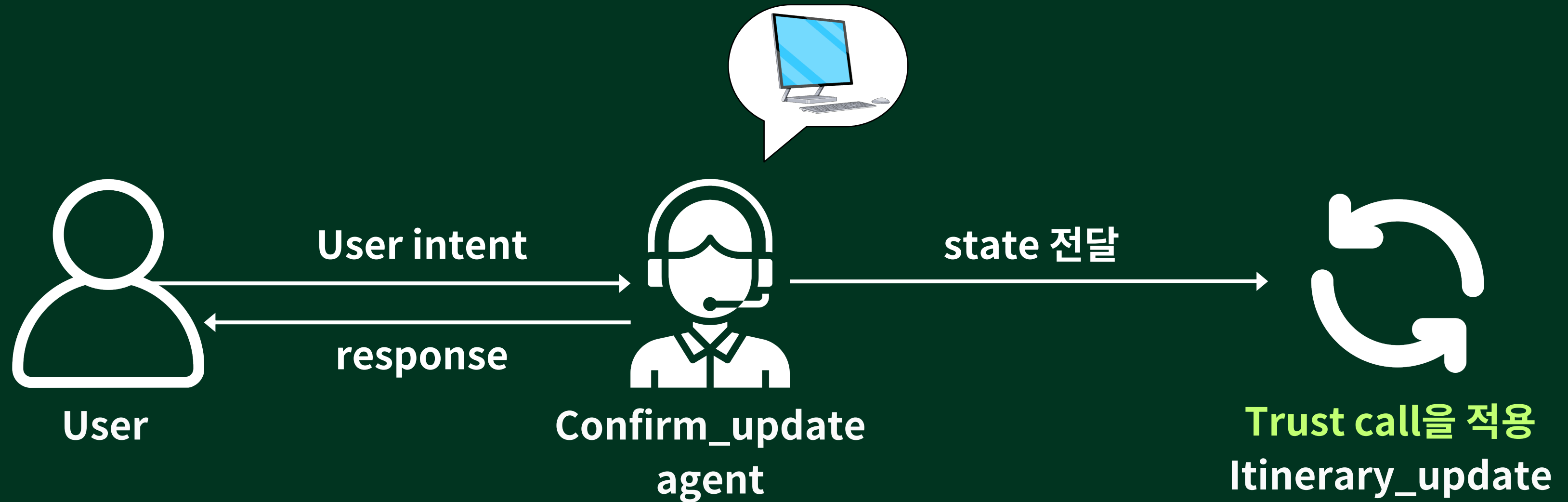
APPLY

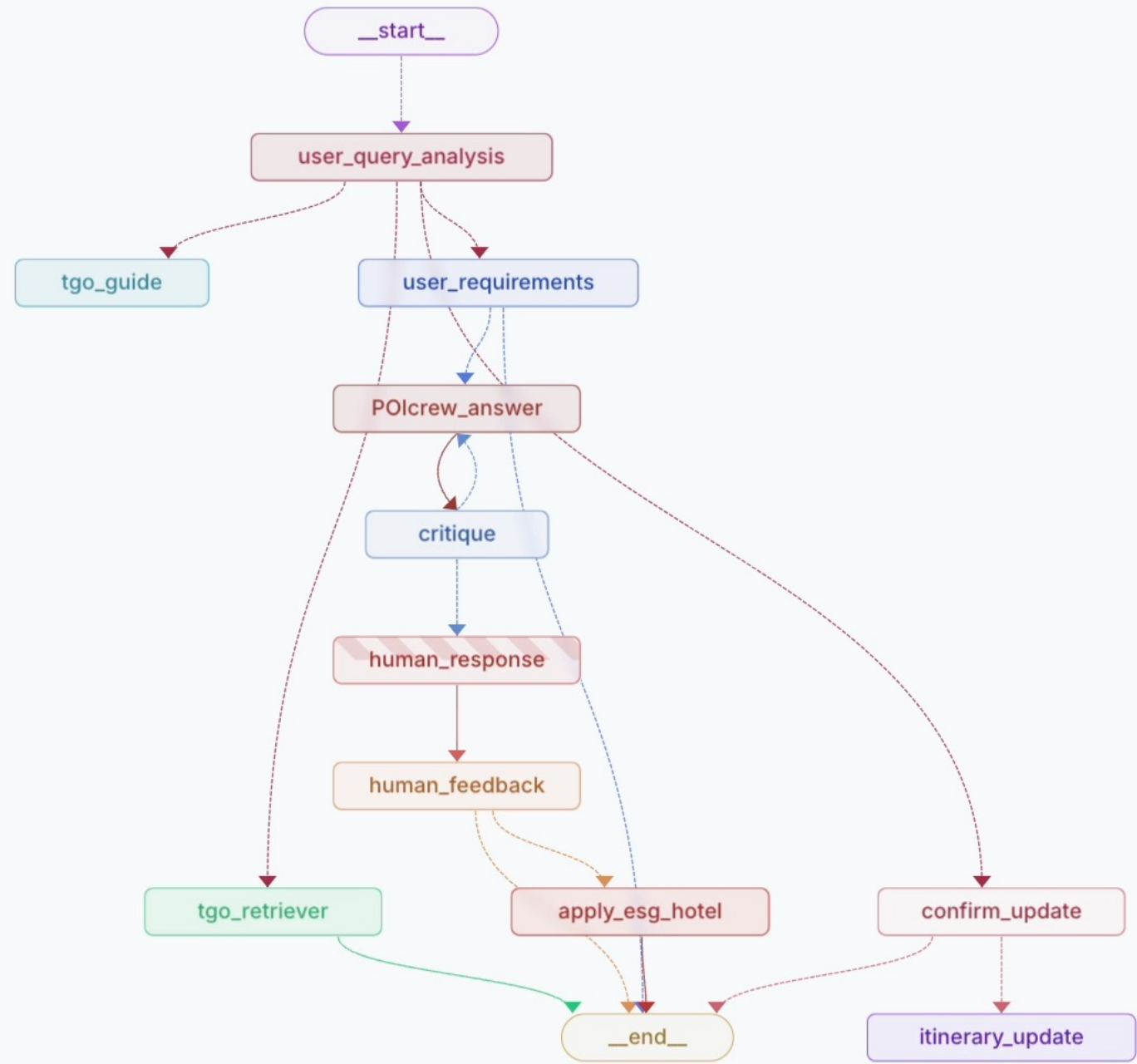


Output

```
{ "name": "Alice", "age": 26, "city": "Seoul" }
```

Trust call을 이용한 여행 일정 업데이트하기





Input

Messages

HUMAN

+ Message

Configurable

Submit

C

critique

Accepted

true

Apply Esg

true

Messages

Human

ID

이 여행 일정은 가족 중심의 휴식과 체험을 잘 반영하고 있습니다. 추천된 POI는 아이들이 좋아할 만한 테마파크와 놀이 시설, 부모님을 위한 온천 체험이 포함되어 있어 가족 모두의 관심사와 선호도를 충족합니다. 또한, 저염식 옵션을 제공하는 레스토랑을 포함하여 특별 요구사항인 짬 음식 피하기를 고려했습니다. 일정은 5일 동안 균형 있게 구성되어 있으며, 각 날마다 다양한 활동이 포함되어 있어 지루하지 않게 진행될 것입니다. 추천 이유도 명확하고 설득력 있으며, POI의 다양성과 균형이 잘 유지되고 있습니다. 전반적으로 이 여행 일정은 적절하다고 판단됩니다.

AI

ID

```
{
  "name": "후지 사파리 파크",
  "type": "사파리",
  "location": "35.3064,138.9361",
  "address": "Suyama, Susono, Shizuoka 410-1231, Japan",
  "description": "다양한 동물들과의 사파리 체험을 즐길 수 있는 장소.",
  "suitability_reason": "동물과 모험을 좋아하는 가족을 위한 방문 장소.",
  "estimated_cost": 28694.0,
  "rating": 4.8,
  "reviews": [
    "An amazing experience with wildlife.",
    "Must visit for animal lovers."
  ],
  "operating_hours": "10:00 - 16:00",
  "booking_required": true
}
```

Human

ID

ESG 실천하는 호텔로 업데이트 해드릴까요? (예/아니오)

H

human_response

아니요

Fork 2 / 2

H

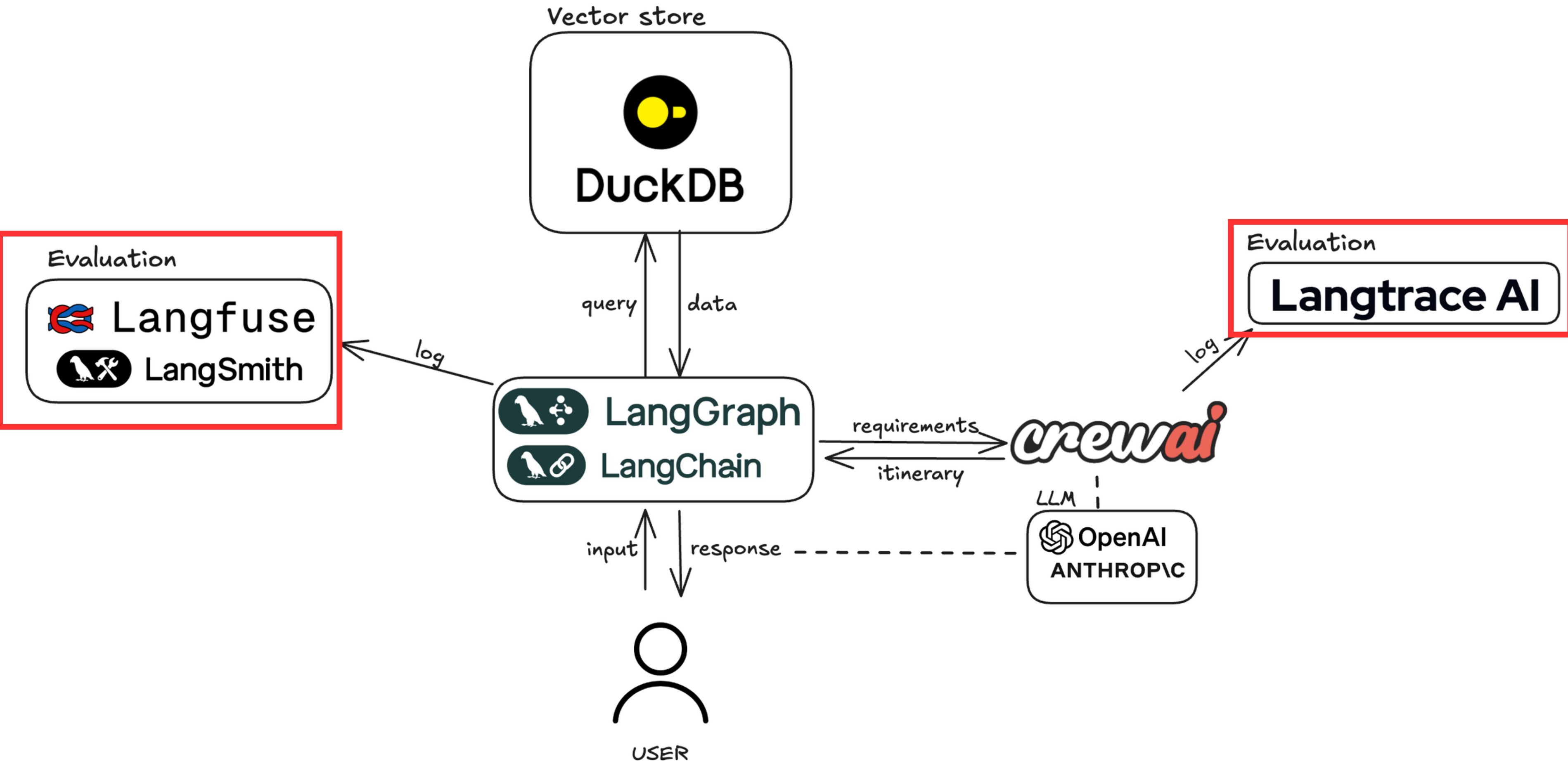
human_feedback

넵 알겠습니다. 계획표는 맘에 드셨나요? 궁금한 점이나 변경사항 있으시면 언제든지 말해주세요!

Fork 2 / 2

OUTPUT

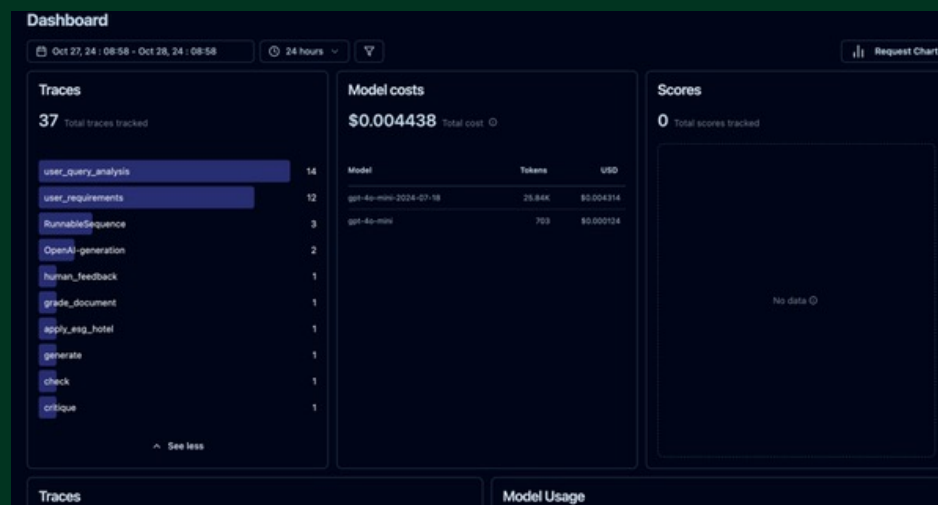
Service workflow



평가



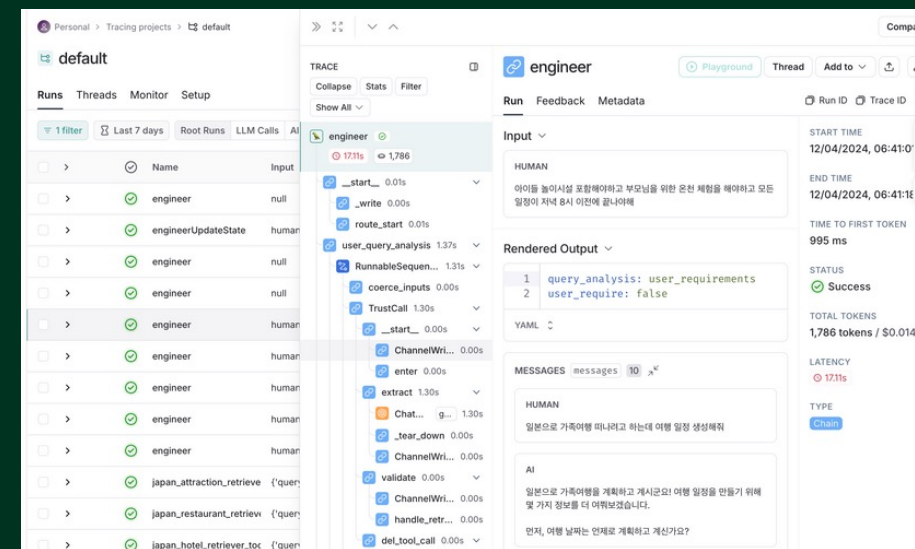
langfuse



- Monitoring
- Performance
- Analytics



langsmith



- Detailing
- Chain
- Optimization

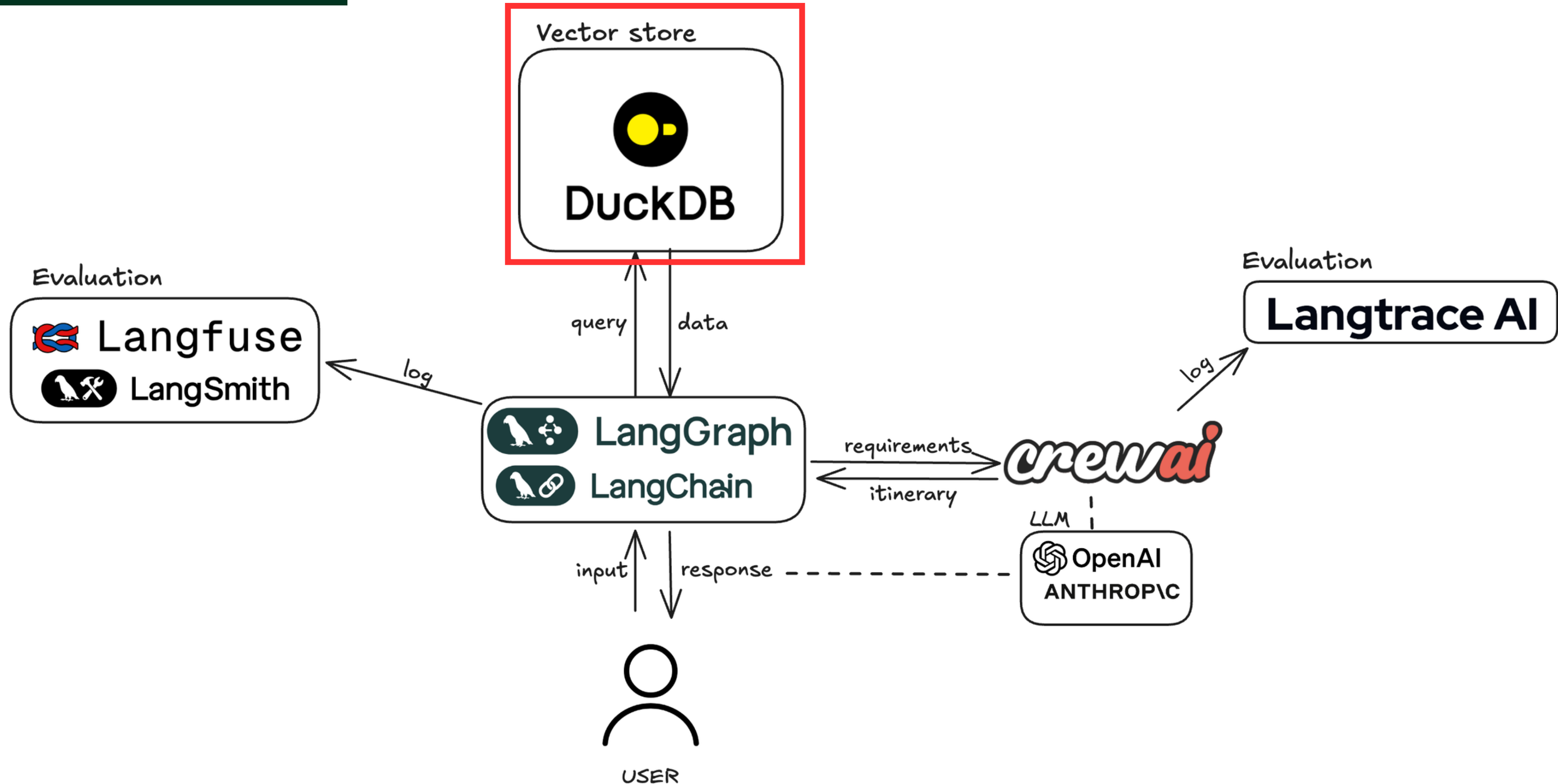


langtrace



- Tracing
- Requests
- Debugging

Service workflow

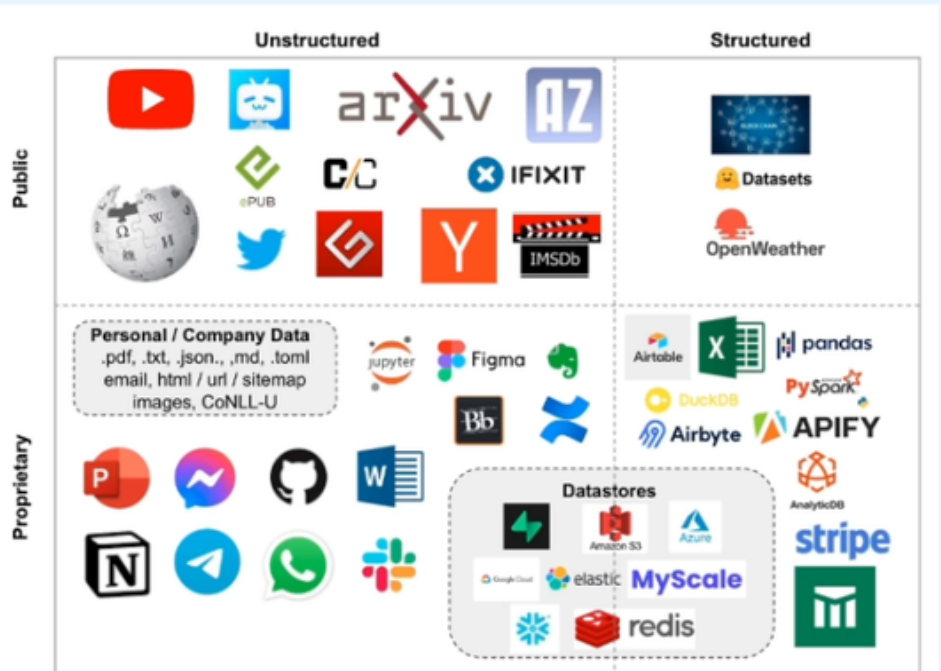
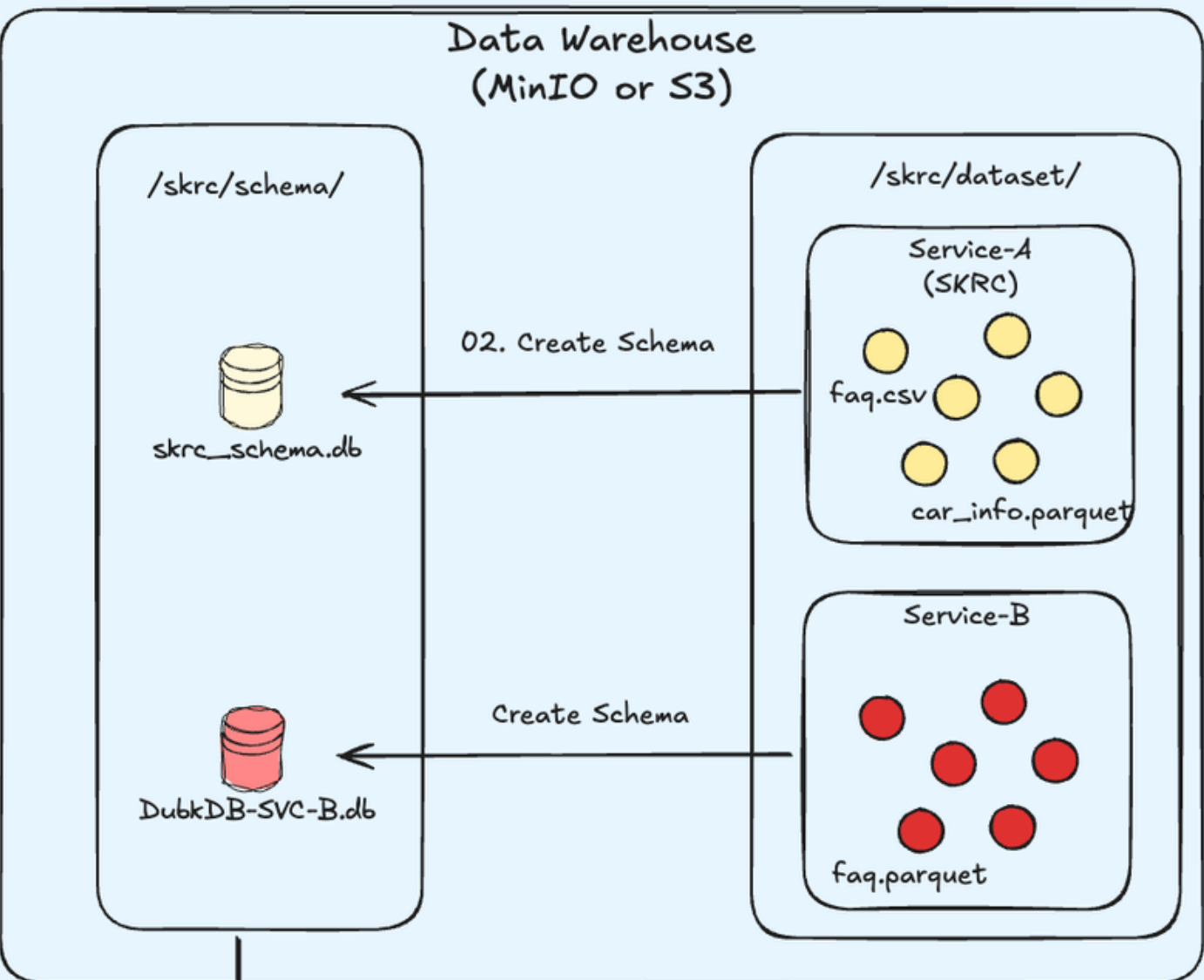


Solution architecture

Create Schema

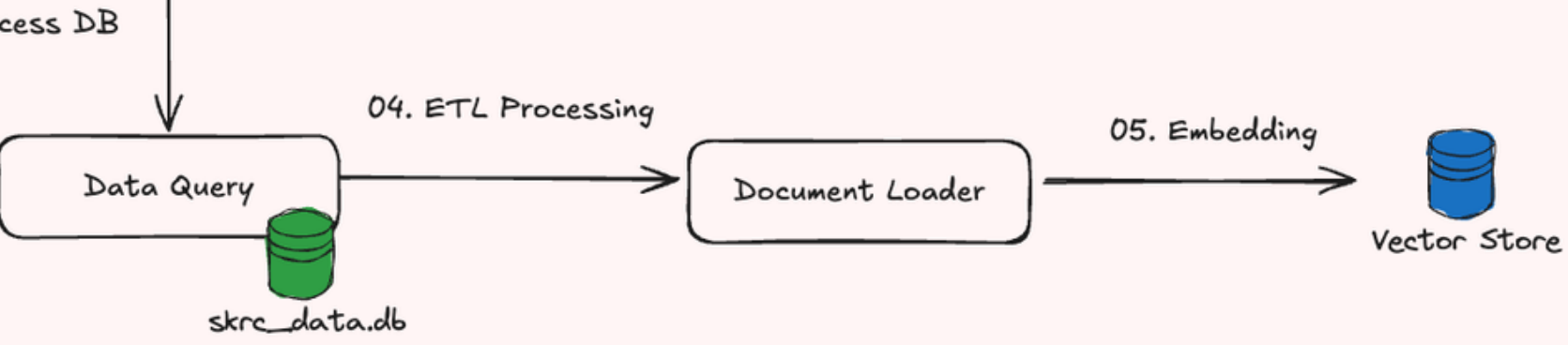
```
# Create Schema (for FAQ)
db = duckdb.connect("skrc_schema.db")
db.sql("""
CREATE VIEW faq_view
AS SELECT category, question, answer
FROM 's3://skrc/dataset/faq.csv'
WHERE category == '예약'
""")

# Create Schema (for Car-Info)
db = duckdb.connect("skrc_schema.db")
db.sql("""
CREATE VIEW car_info_view
AS SELECT category, model, is_reserved
FROM 's3://skrc/dataset/car_info.parquet'
WHERE category == '폴사이즈'
""")
```



LLMOps

```
# Access DB (for FAQ, Car-Info)
import duckdb
conn = duckdb.connect("./skrc_data.db")
conn.sql("""
ATTACH 's3://skrc/schema/skrc_schema.db'
AS skrc_faq (READ_ONLY)
""")
conn.sql("SELECT * FROM skrc_faq LIMIT 10")
```



기존 문제

Data engineer -> LLM engineer에게 데이터를 주는 작업
으로 인해 유연하게 데이터를 받아 개발하기 어려움

문제 해결

“데이터가 없어도 데이터베이스가 되는 DuckDB”

- 데이터 전달 과정 간소화
- 손쉽게 유연하게 데이터 처리 가능
- 업무 효율성 향상



THANK YOU

