



바이트 코딩으로 AI 에이전트 만들어 보기

유저스틴

수석 디벨로퍼 아드보캇
Microsoft

 @justinyoo
 @justinyoo



구글이나 네이버에서
"유저스틴" 검색해 보세요



구글 검색



네이버 검색

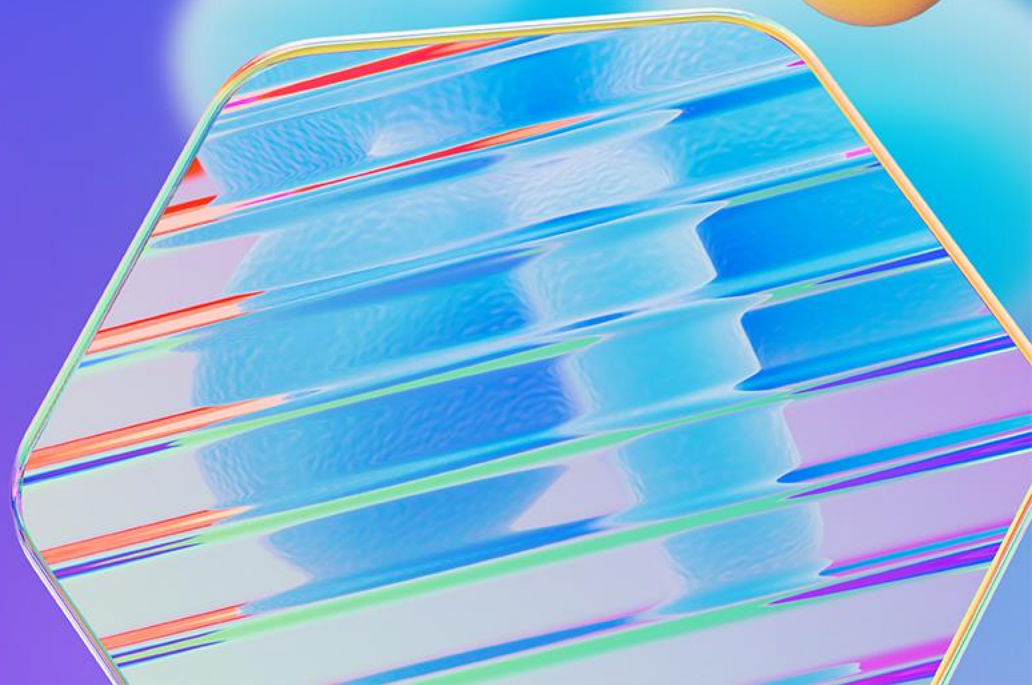
Join at
slido.com
#6812 632



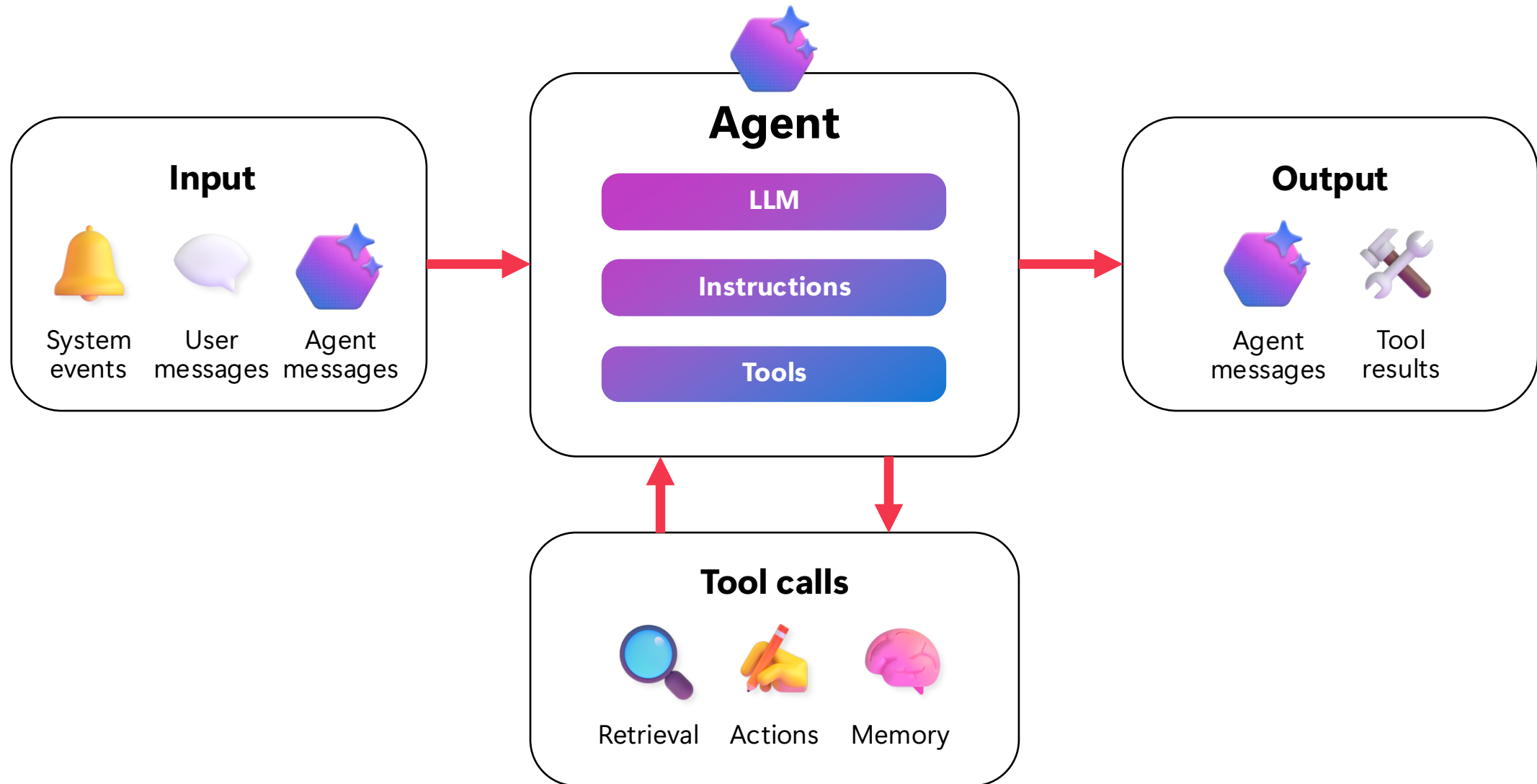
에이전틱 AI 앱 개발을 위한 3대 필수요소



AI 에이전트 시대 고민해야 할 것들



에이전트 작동 원리



AI 에이전트는
스스로 생각하고 행동하고 협업합니다

 문제 파악 후 해결을 위한 단계별 분류

 문제 파악 후 해결을 위한 단계별 분류

 필요한 도구 사용

 문제 파악 후 해결을 위한 단계별 분류

 필요한 도구 사용

 에이전트간 역할 분담 및 협업

 문제 **파악** 후 해결을 위한 단계별 분류

 필요한 **도구** 사용

 에이전트간 역할 분담 및 **협업**

 클라우드 네이티브 라이브 환경 **배포**



문제



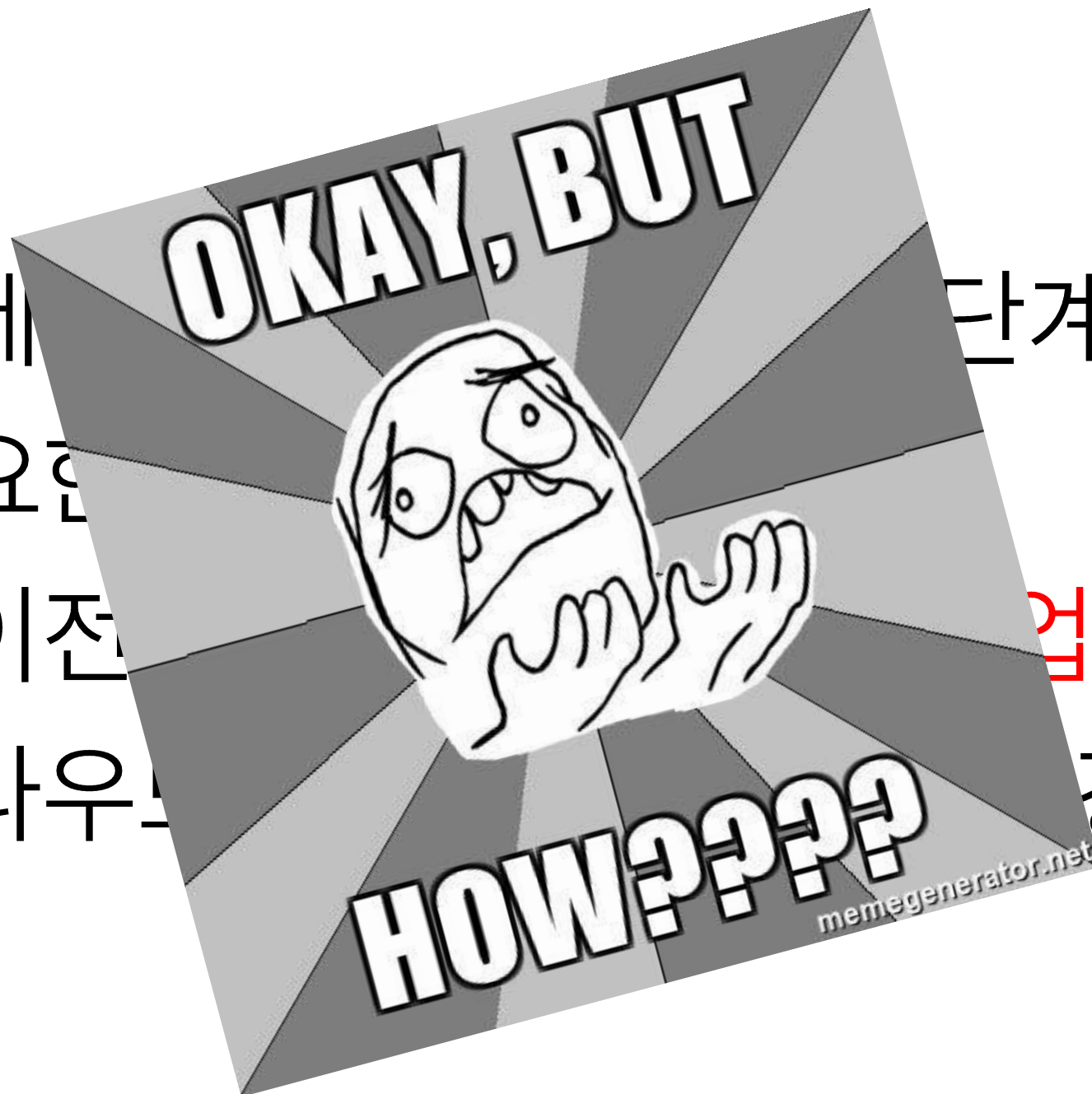
필요한



에이전트



클라우드



단계별 분류

업

경 배포

제대로 쓰려면
분산 클라우드 환경을 고려해야 합니다

 LLM 서비스

 프로덕션 배포

 에이전트 런타임

 구성/설정

 Tool 서버

 헬스 체크

 컨테이너
오케스트레이션

 관측가능성

 챗 UI

 서비스
디스커버리

 데이터베이스

☁ LLM 서비스



프로덕션 배포



에이전트 런타임



구성/설정

Tool 서버



관측가:

하나하나가
설정, 구성, 연결, 배포 등을
위한 서비스입니다

서비스
디스커버리



챗 UI



데이터베이스

에이전트 개발에 필요한 세 가지

에이전트 개발에 필요한 세 가지

Layer 1

AI/LLM

Layer 2

Tools

Layer 3

인프라

에이전트 개발에 필요한 세 가지

Layer 1

AI/LLM

→ Microsoft Agent Framework

Layer 2

Tools

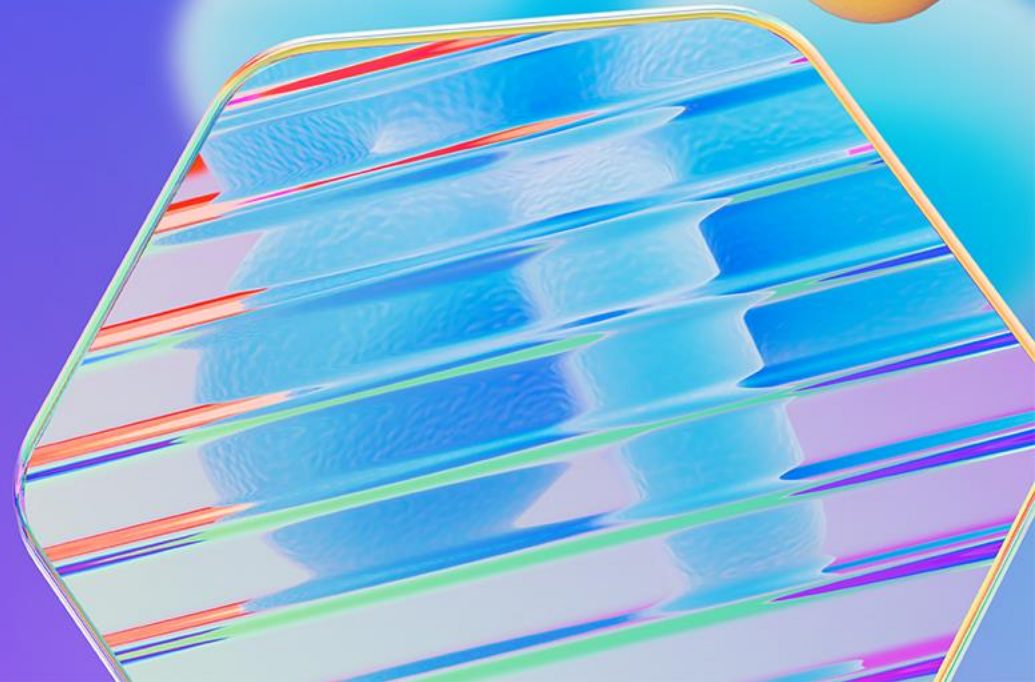
→ Model Context Protocol

Layer 3

인프라

→ Aspire

Microsoft Agent Framework: AI/LLM 레이어 단순화하기



하나의 SDK로 다양한 LLM을 연결합니다

OpenAI, Microsoft Foundry,
AWS, Google, Anthropic, Ollama ...

하나의 SDK로 다양한 LLM을 연결합니다

IChatClient

IChatClient

인터페이스 하나로
모든 LLM 연결

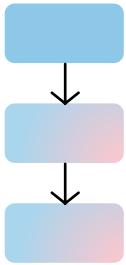
```
client = provider switch
{
    OpenAI ⇒ AddOpenAIClient(),
    Google ⇒ AddVertexAIClient(),
    MicrosoftFoundry ⇒ AddMicrosoftFoundryClient()
}
```

단 몇 줄 만으로 에이전트를 생성합니다

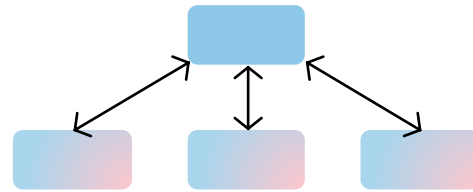
멀티 에이전트 패턴을 손쉽게 구축합니다

다중 에이전트 워크플로우 패턴

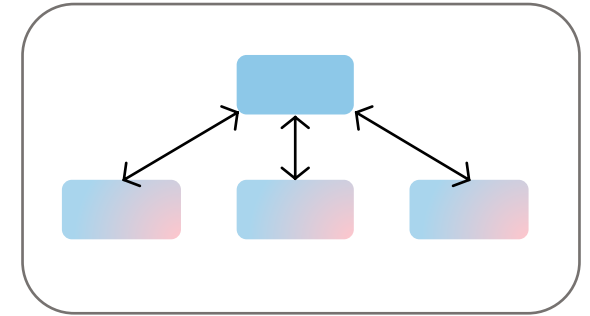
Sequential



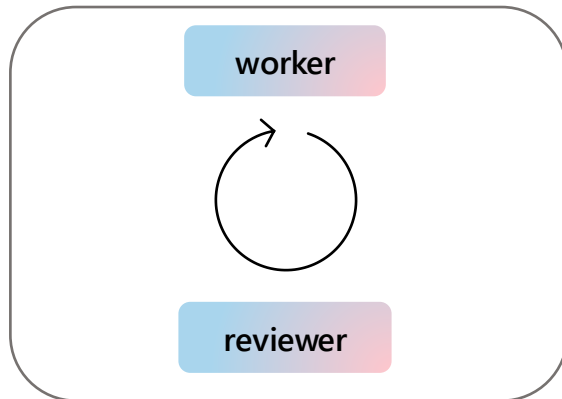
Concurrent



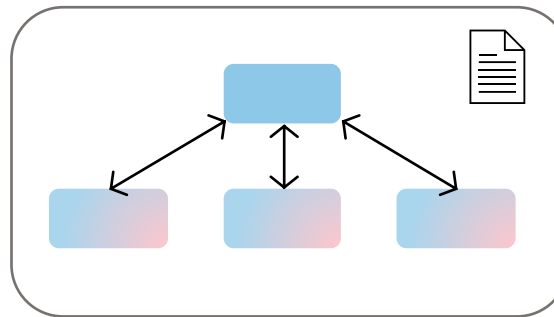
Handoff



Group Chat

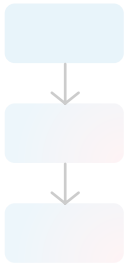


Magentic

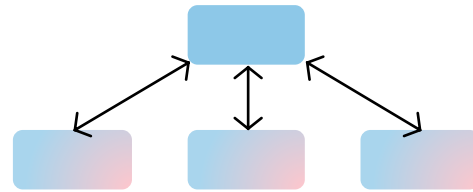


다중 에이전트 워크플로우 패턴

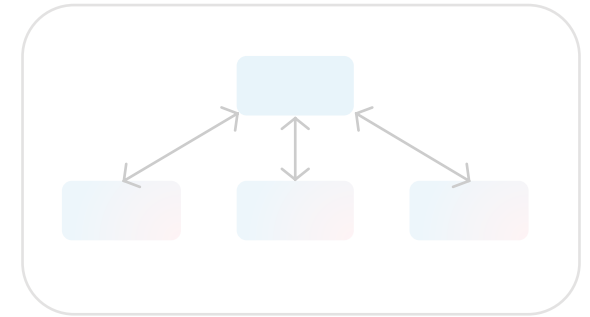
Sequential



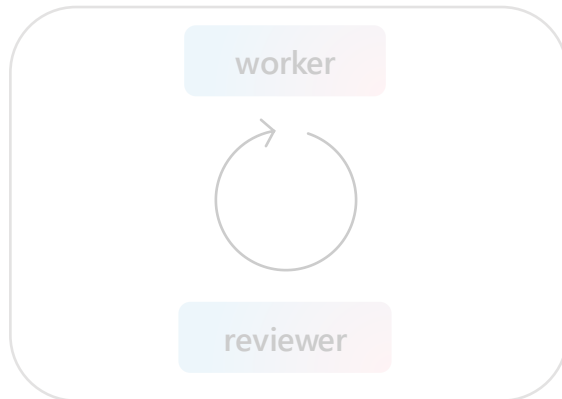
Concurrent



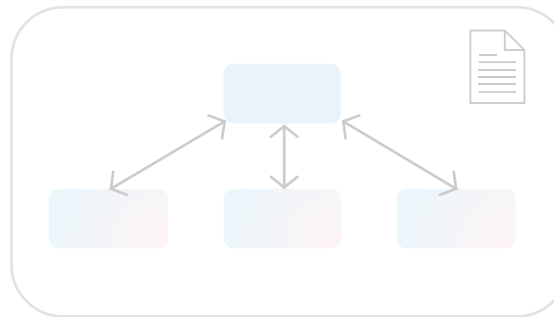
Handoff



Group Chat



Magentic



인스트럭션 엔지니어링

인스트럭션 엔지니어링

```
// Triage 0000
```

```
instruction = """
```

```
두 학교의 같은 날짜 중식을 평가한다.
```

- 열량, NEIS 영양정보와 확인 가능한 식품군 구성을 근거로 사용한다.
- 데이터에 없는 권장량 충족 여부나 영양소 수치를 만들지 않는다.
- 두 학교에 동일한 기준을 적용한다.
- `SpecialistEvaluation` 스키마로만 응답하고 `area`는 `nutrition`으로 설정한다.
- evidence에는 입력 데이터에서 직접 확인할 수 있는 항목을 구체적으로 적는다.

```
""";
```

인스트럭션 엔지니어링

```
// Triage 0000  
instruction = """
```

두 학교의 같은 날짜 중식을 평가한다.

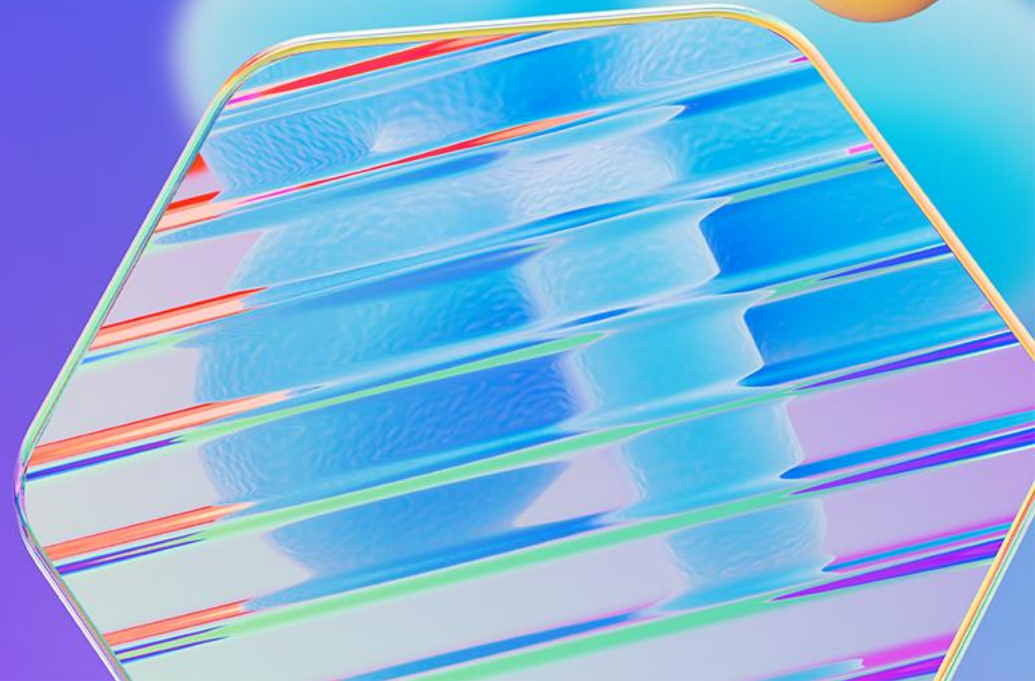
- 열량, NEIS 영양정보와 확인 가능한 식품군 구성을 근거로 사용한다.
- 데이터에 없는 권장량 충족 여부나 영양소 수치를 만들지 않는다.
- 두 학교에 동일한 기준을 적용한다.
- `SpecialistEvaluation` 스키마로만 응답하고 `area`는 `nutrition`으로 설정한다.
- evidence에는 입력 데이터에서 직접 확인할 수 있는 항목을 구체적으로 적는다.

```
""";
```

인스트럭션 패턴:

- 👉 명확한 역할 정의
- 👉 바운더리 설정
- 👉 단계별 프로세스 정의
- 👉 도구 사용법 정의
- 👉 말투 정의
- 👉 에이전트 현재 상태 확인

Model Context Protocol: Tool 레이어 단순화하기



MCP가 뭔가요?

 에이전트와 도구 사이 커뮤니케이션을 위한 개방형 프로토콜

MCP가 뭔가요?



에이전트와 도구 사이 커뮤니케이션을 위한 개방형 프로토콜



에이전트와 별개로 독립적인 서버에서 동작

MCP가 뭔가요?



에이전트와 도구 사이 커뮤니케이션을 위한 개방형 프로토콜



에이전트와 별개로 독립적인 서버에서 동작

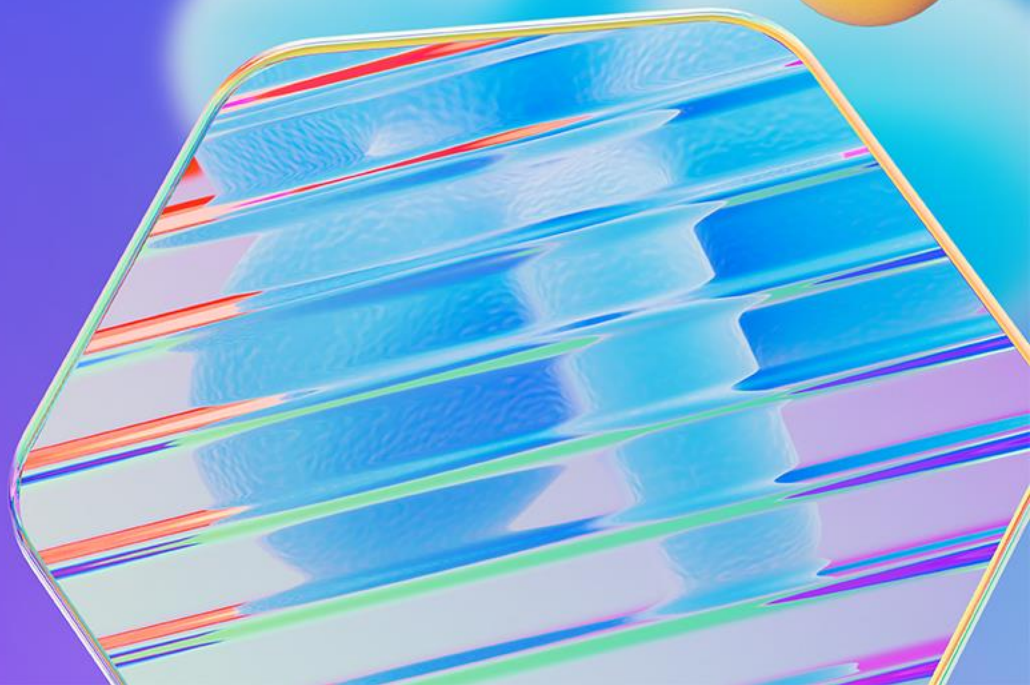


에이전트는 MCP 클라이언트로 동작하며
MCP 서버 검색을 통해 도구 등록

MCP가 뭔가요?

-  에이전트와 도구 사이 커뮤니케이션을 위한 개방형 프로토콜
-  에이전트와 별개로 독립적인 서버에서 동작
-  에이전트는 MCP 클라이언트로 동작하며
MCP 서버 검색을 통해 도구 등록
-  공식 MCP 서버 레지스트리를 통한 생태계 조성

Aspire: 인프라 및 나머지 단순화하기



서비스 오케스트레이션 이슈

Web  React 기반 UI

서비스 오케스트레이션 이슈

Web  React 기반 UI

API  NEIS 급식 정보 조회용 API

서비스 오케스트레이션 이슈

Web  React 기반 UI

API  NEIS 급식 정보 조회용 API

Agent  Agent Framework 기반 백엔드

서비스 오케스트레이션 이슈

Web 📌 React 기반 UI

API 📌 NEIS 급식 정보 조회용 API

Agent 📌 Agent Framework 기반 백엔드

MCP 📌 NEIS 급식 정보 조회용 MCP서버

서비스 오케스트레이션 이슈

Web 📌 React 기반 UI

API 📌 NEIS 급식 정보 조회용 API

Agent 📌 Agent Framework 기반 백엔드

MCP 📌 NEIS 급식 정보 조회용 MCP서버

LLM Provider 📌 Microsoft Foundry

서비스 오케스트레이션 이슈

Web  React
API

한번에 전부 실행시킬 수 있어야 함

한 번에 모든 서비스를 연결시킬 수 있어야 함

한 번에 모든 서비스 헬스 체크 할 수 있어야 함

한 번에 E2E로 데이터 추적할 수 있어야 함

이 때 Aspire가 필요합니다

Aspire가 뭔가요?

 분산형 서비스를 위한 클라우드 네이티브 프레임워크

Aspire가 뭔가요?

 분산형 서비스를 위한 클라우드 네이티브 프레임워크

서비스 오케스트레이션  코드로 전체 앱 구조 정의

Aspire가 뭔가요?

 분산형 서비스를 위한 클라우드 네이티브 프레임워크

서비스 오케스트레이션  코드로 전체 앱 구조 정의

서비스 디스커버리  서비스간 커넥션 URL 자동 생성

Aspire가 뭔가요?

 분산형 서비스를 위한 클라우드 네이티브 프레임워크

서비스 오케스트레이션  코드로 전체 앱 구조 정의

서비스 디스커버리  서비스간 커넥션 URL 자동 생성

오픈 텔레메트리  로깅, 트레이싱, 메트릭 기본 탑재

Aspire가 뭔가요?

📍 분산형 서비스를 위한 클라우드 네이티브 프레임워크

서비스 오케스트레이션		코드로 전체 앱 구조 정의
서비스 디스커버리		서비스간 커넥션 URL 자동 생성
오픈 텔레메트리		로깅, 트레이싱, 메트릭 기본 탑재
헬스 체크		서비스 가용성 모니터링

Aspire가 뭔가요?

📡 분산형 서비스를 위한 클라우드 네이티브 프레임워크

- 서비스 오케스트레이션  코드로 전체 앱 구조 정의
- 서비스 디스커버리  서비스간 커넥션 URL 자동 생성
- 오픈 텔레메트리  로깅, 트레이싱, 메트릭 기본 탑재
- 헬스 체크  서비스 가용성 모니터링
- 서비스 의존성 체크  서비스간 시작 순서 자동 조정

Aspire가 뭔가요?

📡 분산형 서비스를 위한 클라우드 네이티브 프레임워크

서비스 오케스트레이션	🎯	코드로 전체 앱 구조 정의
서비스 디스커버리	🔗	서비스간 커넥션 URL 자동 생성
오픈 텔레메트리	📊	로깅, 트레이싱, 메트릭 기본 탑재
헬스 체크	🏠	서비스 가용성 모니터링
서비스 의존성 체크	⌚	서비스간 시작 순서 자동 조정
클라우드 배포	🚀	애저 컨테이너 앱 기본 배포

aspire deploy 📌 단 하나의 배포 명령어

```
# 0000
```

```
$ az login
```

```
# 0000 0000000 + 00
```

```
$ aspire deploy
```

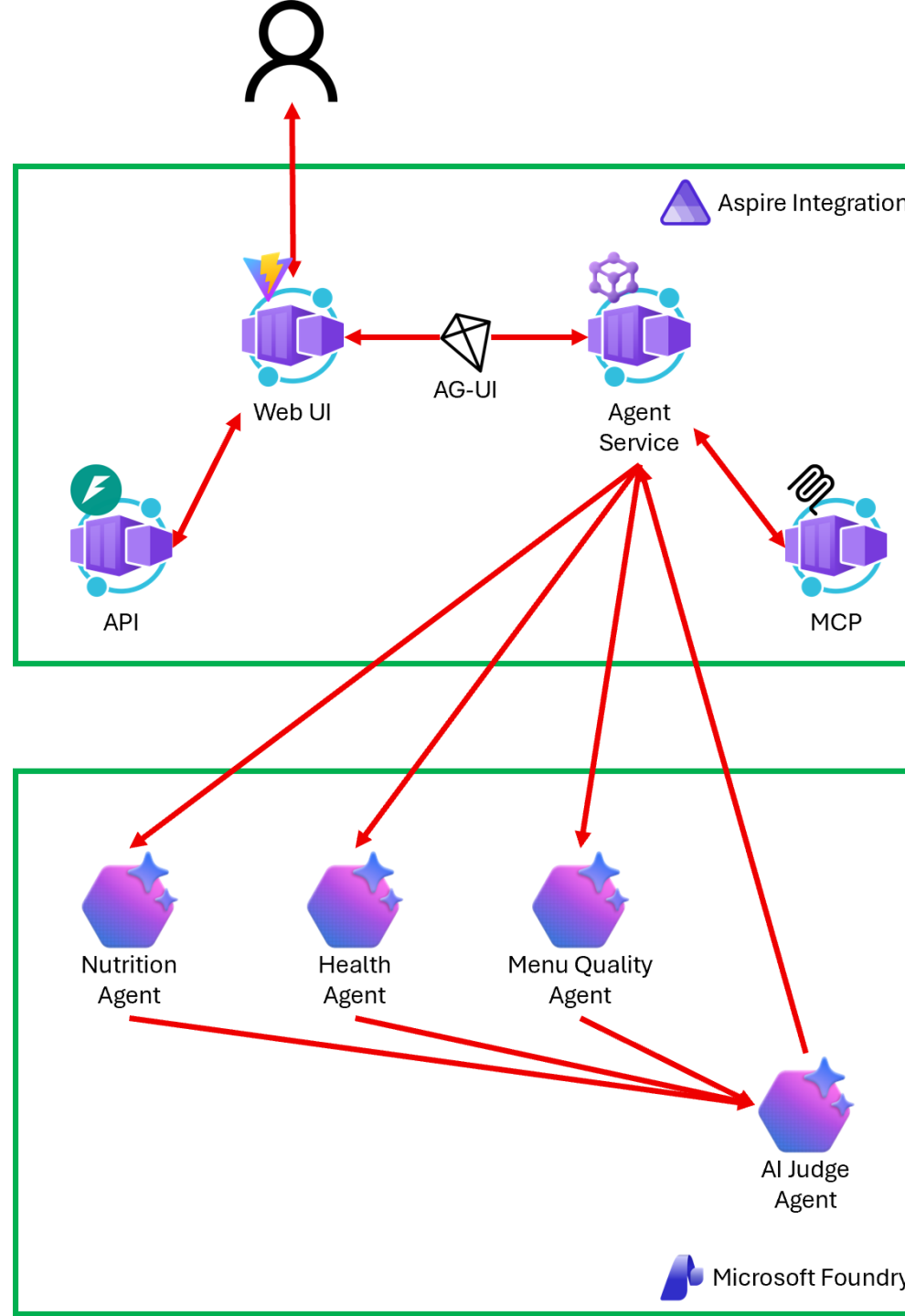
```
# 0000 00
```

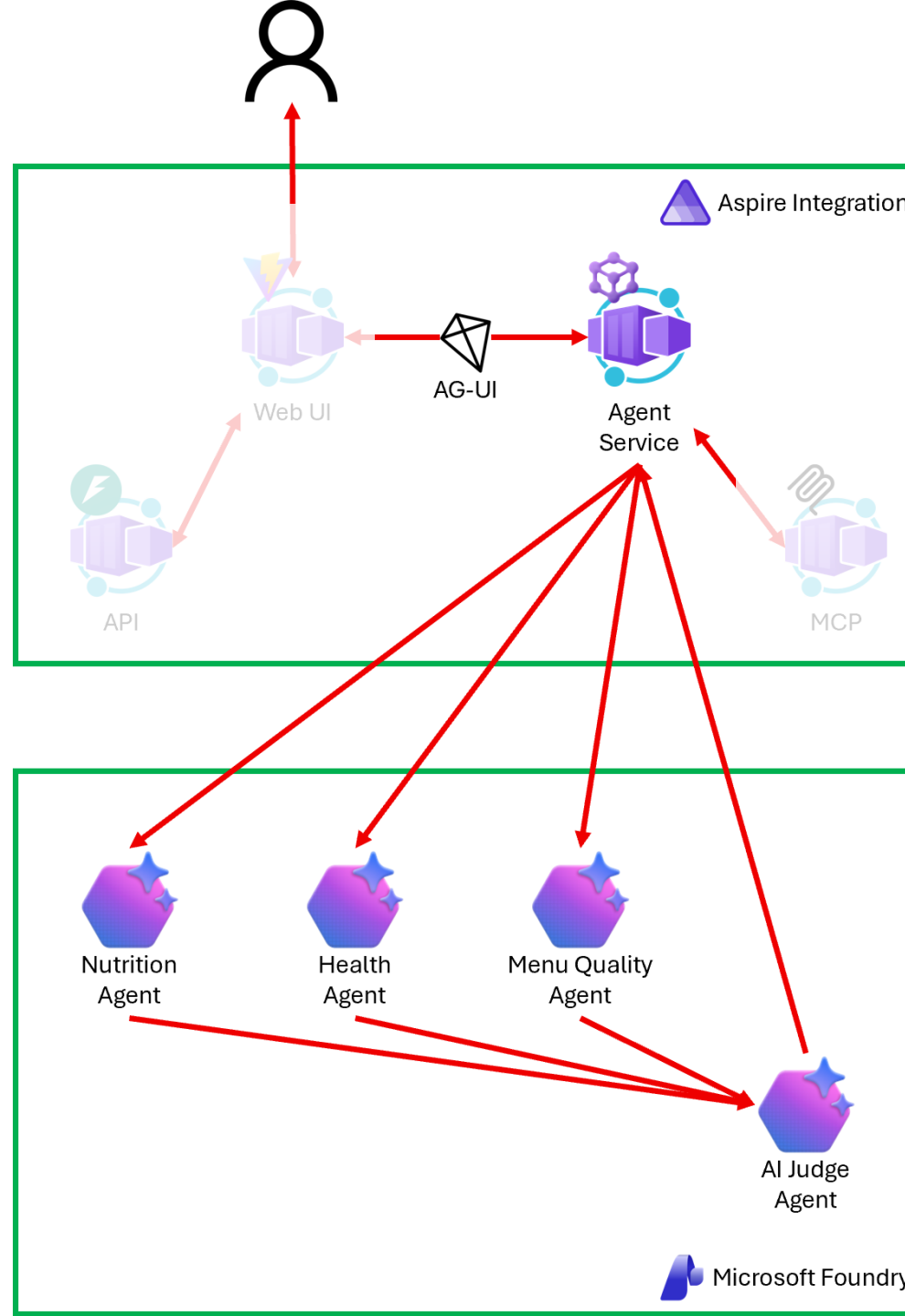
```
$ aspire destroy
```

aspire deploy 📌 단 하나의 배포 명령어

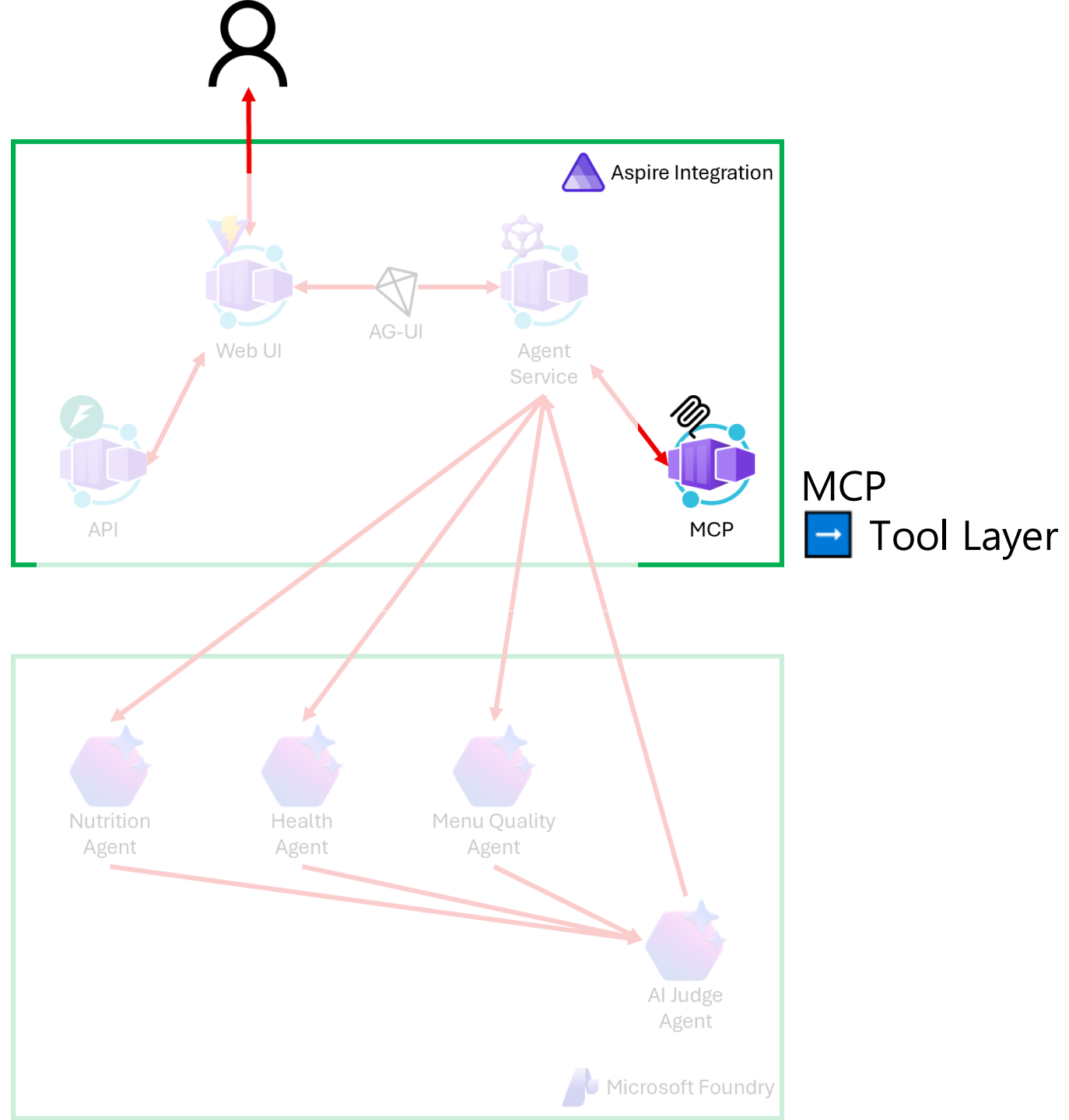
- 1 모든 서비스를 애저 컨테이너 앱 서비스로 배포
- 2 트래픽에 따른 오토 스케일링
- 3 트래픽 없으면 Scale-to-Zero
- 4 동일한 서비스 디스커버리, 동일한 관측용이성

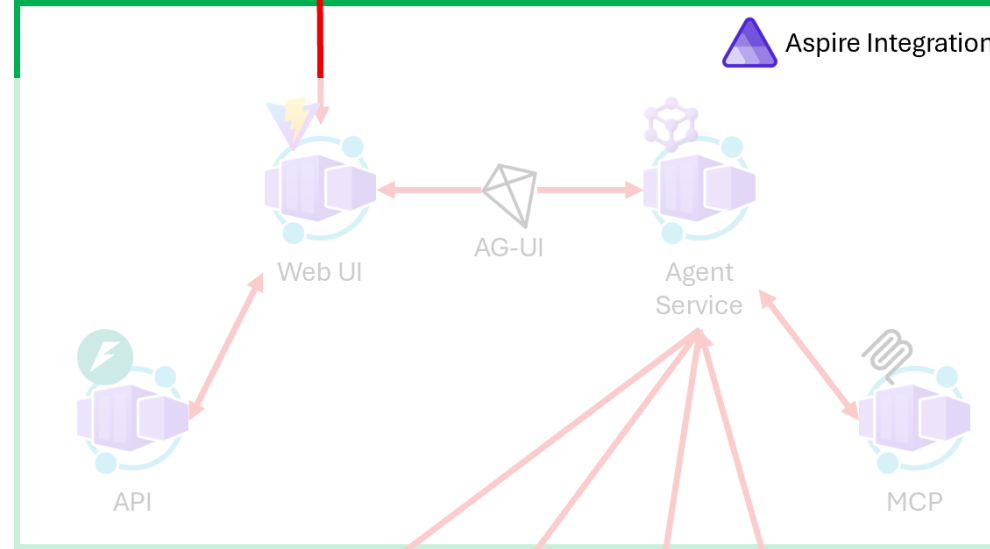
Dockerfile 필요 없음 | Docker Compose 필요 없음





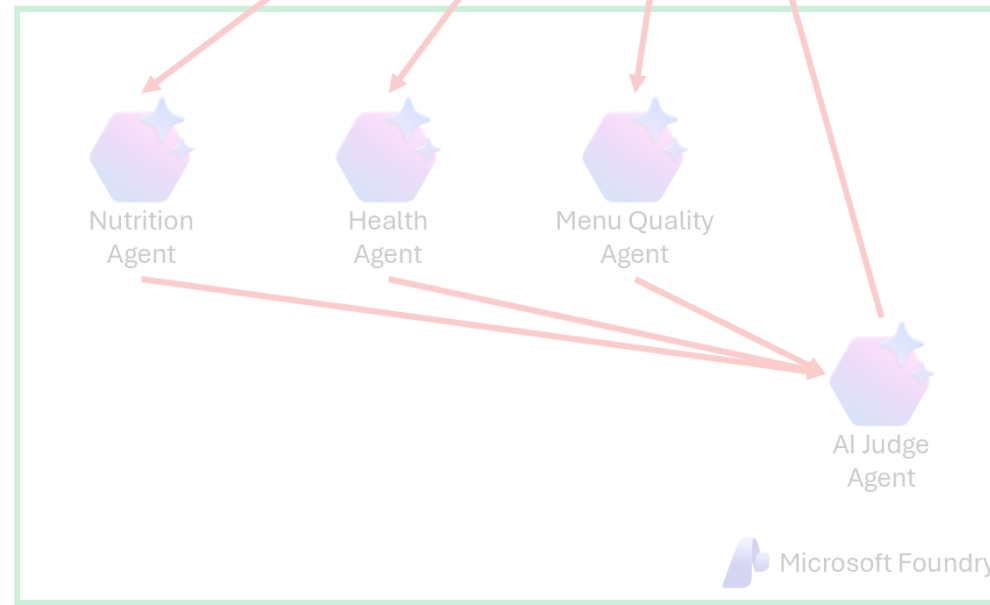
Microsoft Agent Framework
→ AI Layer





Aspire

→ Infrastructure Layer

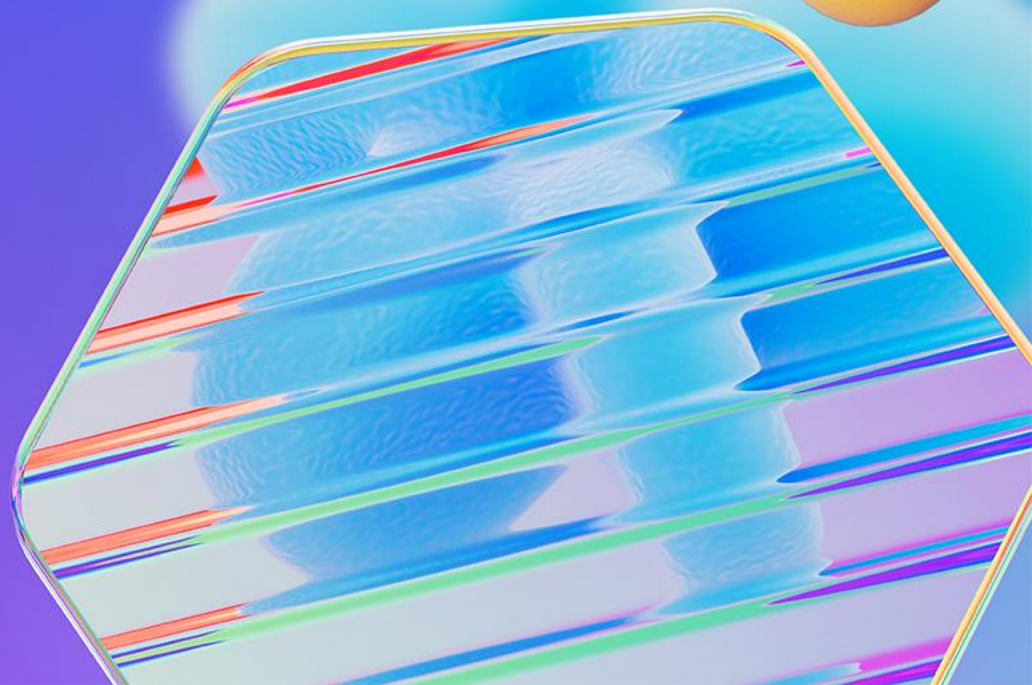




GitHub Copilot과 바이브 코딩하기



바이트 코딩 시작 전에...



네 가지를 먼저 준비해 두세요

AGENTS.md

AGENTS.md

GitHub Copilot이 동작하는 범위 설정

AGENTS.md

GitHub Copilot이 동작하는 범위 설정
이건 반드시 해야 한다

AGENTS.md

GitHub Copilot이 동작하는 범위 설정
이건 반드시 해야 한다
이건 해도 된다

AGENTS.md

GitHub Copilot이 동작하는 범위 설정

이건 반드시 해야 한다

이건 해도 된다

이건 하면 안된다

구현하고 싶은 아이디어

구현하고 싶은 아이디어
구체적이지 않아도 됨

구현하고 싶은 아이디어

구체적이지 않아도 됨

GitHub Copilot과 같이 구체화 해 나가면 됨

아이디어를 구체화 시켜 나갑니다

아이디어를 구체화 시켜 나갑니다
다른 모델로 아이디어 검증

아이디어를 구체화 시켜 나갑니다
다른 모델로 아이디어 검증
이후 IDEATION.md 파일로 저장

아이디어이션 이후
PRD.md 문서를 생성합니다

PRD.md

앱 개발에 필요한 모든 요구사항과
개발 단계 및 인수조건 등을 기술한 문서

PRD.md

다른 모델로 검증할 것

TRD.md 문서도 생성합니다

TRD.md

앱 개발에 필요한 기술 스택과
아키텍처를 기술한 문서

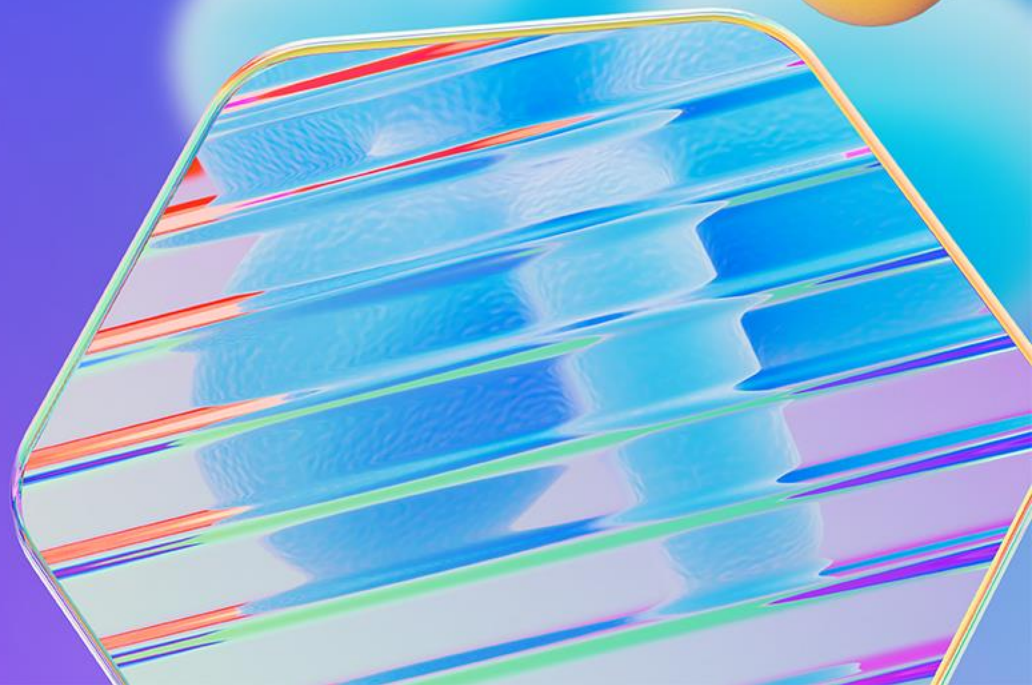
TRD.md

마찬가지로 다른 모델로 검증할 것

이제 “~해줘” 하면 됩니다

AGENTS.md, PRD.md, TRD.md
이것만 잘 해도 원샷에
80% 이상 의도대로 구현해 줍니다

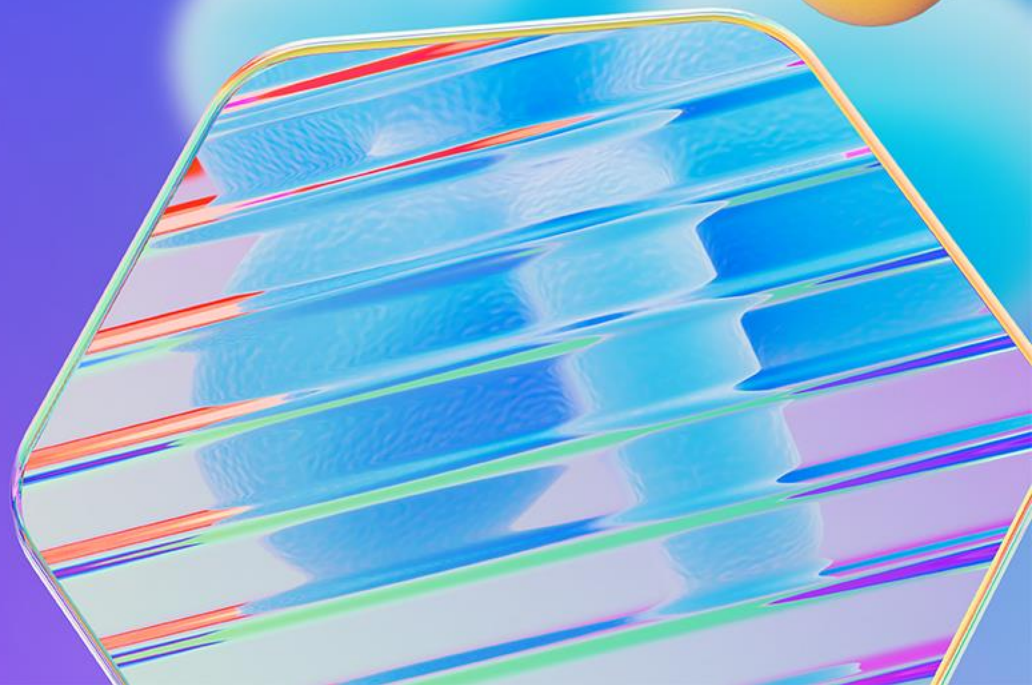
1차 바이브 코딩 끝나고 나서



결과물을 믿지 마세요

왜 이런 결과가 나왔는지 확인할 것
다른 모델로 검증할 것

바이브 코딩으로 배포하기



GitHub Copilot이 배포까지 도와줍니다

데모:
전국 초중고 급식 정보 조회/분석 앱



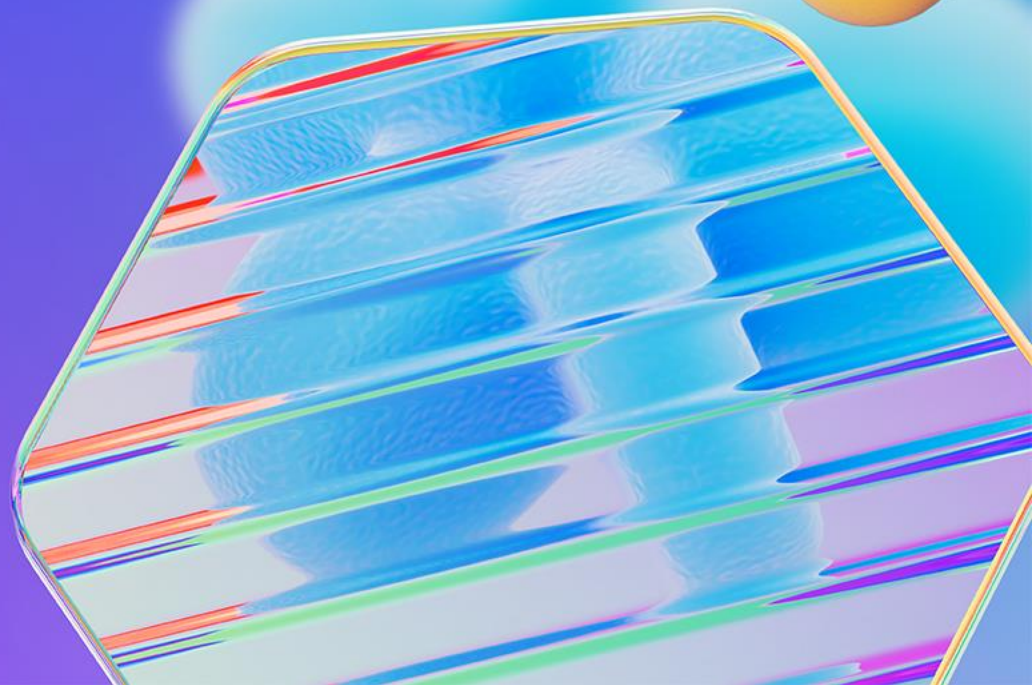
NEIS에서 제공하는 공개 API 활용

github.com/devkimchi/battle-school-lunch

정리



에이전틱 AI 앱 개발을 위해 고려해야 할 것들



Agent Framework으로 AI 계층 추상화

Agent Framework으로 AI 계층 추상화
MCP 서버로 도구 추가

Agent Framework으로 AI 계층 추상화

MCP 서버로 도구 추가

Aspire로 리소스 오케스트레이션

Agent Framework으로 AI 계층 추상화

MCP 서버로 도구 추가

Aspire로 리소스 오케스트레이션

aspire deploy로 클론  실행  배포 한방에

Agent Framework으로 AI 계층 추상화

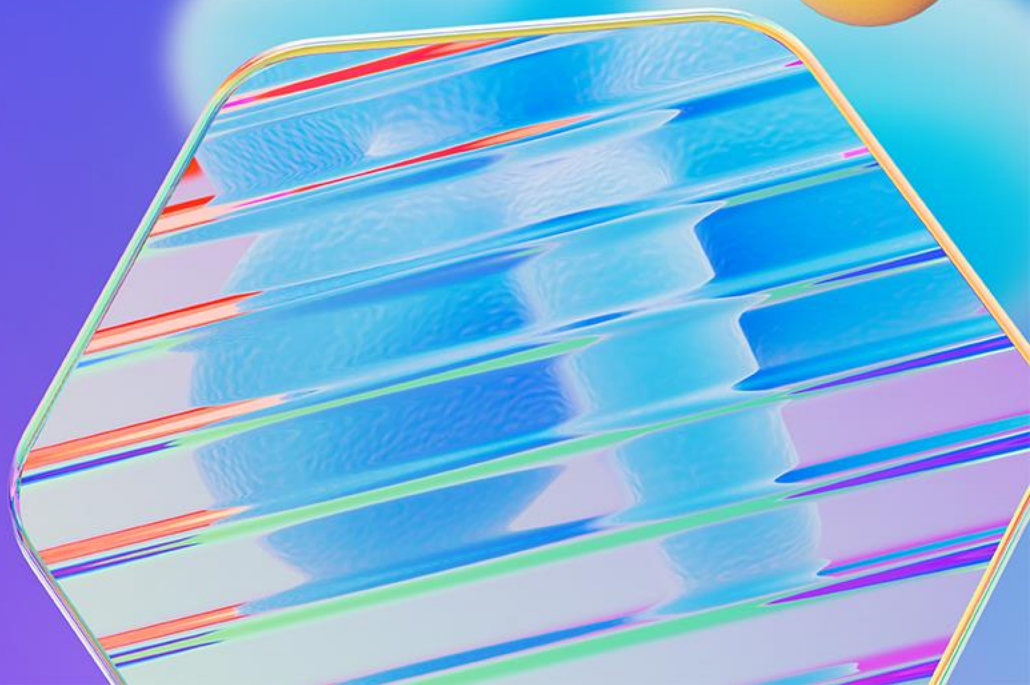
MCP 서버로 도구 추가

Aspire로 리소스 오케스트레이션

aspire deploy로 클론  실행  배포 한방에

모든 에이전트 앱에 동일 패턴 적용 가능

GitHub Copilot과 바이브 코딩하기



AGENTS.md 🖱️ 필요한 경우 계속 수정

AGENTS.md 🖐️ 필요한 경우 계속 수정

IDEATION.md 🖐️ 다양한 모델로 검증

AGENTS.md 🖱️ 필요한 경우 계속 수정

IDEATION.md 🖱️ 다양한 모델로 검증

PRD.md 🖱️ 다양한 모델로 검증

AGENTS.md 📌 필요한 경우 계속 수정

IDEATION.md 📌 다양한 모델로 검증

PRD.md 📌 다양한 모델로 검증

TRD.md 📌 다양한 모델로 검증

AGENTS.md 👉 필요한 경우 계속 수정

IDEATION.md 👉 다양한 모델로 검증

PRD.md 👉 다양한 모델로 검증

TRD.md 👉 다양한 모델로 검증

해줘! 👉 다양한 모델로 검증

