

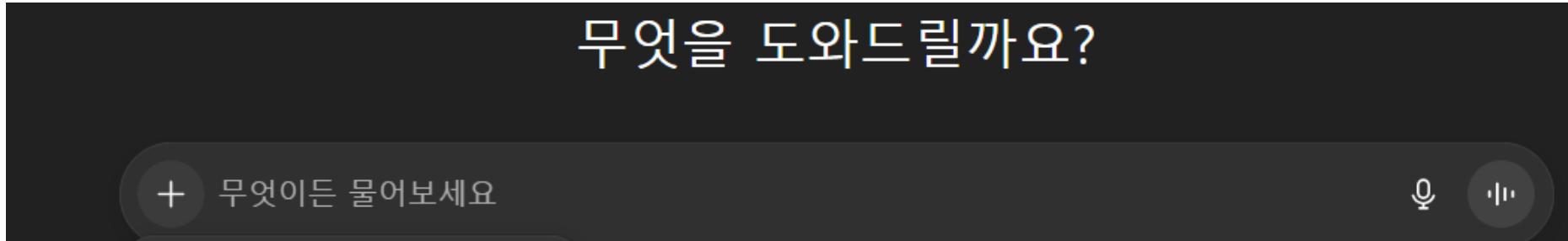
# MCP (Model Context Protocol)

추가 MCP 자료

# 1. AI, 더 잘 쓰기 위해선 어떻게 하나요?



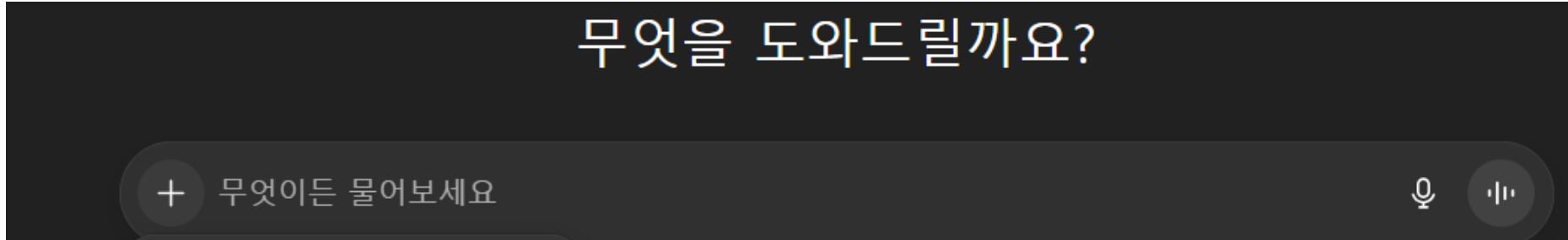
# 1. AI, 더 잘 쓰기 위해선 어떻게 하나요?



AI에 단순히 질문하는 것 만으로는 좋은 결과를 얻기 부족

AI가 **검색 엔진**을 통해 검색한 결과 바탕으로 답  
**내가 사용하는 도구**를 연결해 나의 상황에 맞게 답

# 1. AI, 더 잘 쓰기 위해선 어떻게 하나요?



AI에 단순히 질문하는 것 만으로는 좋은 결과를 얻기 부족

AI가 검색 엔진을 통해 검색한 결과 바탕으로 답  
내가 사용하는 도구를 연결해 나의 상황에 맞게 답

도구: AI가 더 답변을 잘 할 수 있게 사용하는 것

# 2. AI(Agent AI)의 구성 요소

## 지식 도구

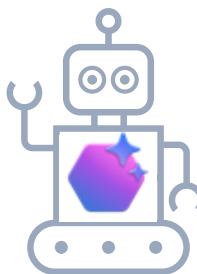
### (Knowledge tools):

AI가 필요한 정보를 외부에서  
'가져오는' 도구.  
(모델의 부족한 지식을 보충)  
(예: 웹 검색으로 죄신 날씨를  
찾아보는 것)

## 액션 도구

### (Action tools):

AI가 사용자 대신 실제 작업을  
'수행하는' 도구.  
(모델의 결정을 행동으로 옮김)  
(예: OneDrive에 요약 파일을 저장하는  
것, 이메일을 보내는 것, 캘린더에  
일정을 추가하는 것)



언어 AI 모델(Model)

# 2. AI(Agent AI)의 구성 요소

지식 도구

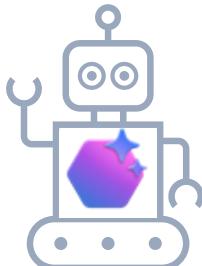
(Knowledge tools):

웹에서 날씨 검색...

액션 도구

(Action tools):

알람 어플에서  
알람 설정

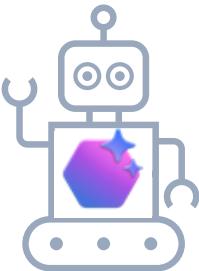


언어 AI 모델(Model)

빅스비야,  
오늘 날씨 어때?  
따뜻하면 산책하게  
7시에 알람 맞춰줘

## 2. AI(Agent AI)의 구성 요소

### 도구



도구가 필수적인 건 아니나,  
도구를 통해서 AI 모델이 **더 좋은 답변을 해줄 수 있음**

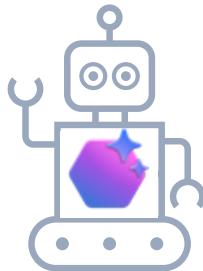
언어 AI 모델(Model)

## 2. AI(Agent AI)의 구성 요소

### 도구

보통 도구는 AI 자체에 내장된 것이기 보다는  
외부에서 개발되거나 제공되는 기능

그래서 AI 모델과 도구를 연결해야 함



언어 AI 모델(Model)

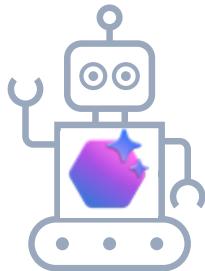
Ex) ChatGPT <-> Google 웹 연결  
빅스비 AI <-> 알람 앱 연결

# 3. MCP 등장 배경

## 도구

보통 도구는 AI 자체에 내장된 것이기 보다는  
외부에서 개발되거나 제공되는 기능

그래서 AI 모델과 도구를 연결해야 함



언어 AI 모델(Model)

이 연결하는 방식(SDK)이 **복잡**하고  
**표준화 X**

# \*에이전트 개발 옵션(SDK)



## Azure AI Foundry Agent Service

- 에이전트를 만들고, 관리하고, 사용하는 것을 돕는 Azure의 관리형 서비스입니다.
- OpenAI Assistants API를 기반으로 더 많은 모델 선택지, 데이터 통합, 그리고 기업 수준의 보안을 제공합니다.



## OpenAI Assistants API

- OpenAI 에이전트 개발을 위한 표준 API입니다.
- 일부 AI 에이전트 서비스 기능들을 제공합니다.
- OpenAI 모델로만 제한됩니다.



## Semantic Kernel

- AI 에이전트 구축을 위한 가볍고, 오픈소스인 개발 키트입니다.
- OpenAI Assistant 및 Azure AI 에이전트뿐만 아니라 일반적인 대화 완성(chat completion) 에이전트도 지원합니다.



## AutoGen

- 빠른 에이전트 개발을 위한 오픈소스 프레임워크입니다.
- 에이전트를 실험할 때 연구 및 아이디어 구상에 유용합니다.



## Microsoft 365 Agents SDK

- 다양한 채널을 위한 자체 호스팅(self-hosted) 에이전트를 생성합니다.
- Microsoft 365에 국한되지 않으며, Slack이나 Messenger 같은 채널을 통해 전달될 수 있습니다.



## Copilot Studio

- 에이전트를 구축하고 배포하기 위한 로우코드(low-code) 개발 환경입니다.
- Microsoft 365 생태계와 통합되며, Slack이나 Messenger 같은 채널과 연결됩니다.



## Agent Builder

- 비즈니스 사용자가 Microsoft Copilot Chat 및 기타 Copilot 기반 인터페이스에서 기본적인 에이전트를 직접 만들 수 있도록 돋는 선언적 도구입니다.
- 기능을 설명하거나 시각적 인터페이스를 사용하여 에이전트를 생성할 수 있도록 해줍니다.



## LangChain

- AI 에이전트 구축을 위한 오픈소스인 개발 키트

# SDK 예시

```
from langchain.llms import OpenAI
from langchain.chains import LLMChain
from langchain.prompts import PromptTemplate
라이브러리

# 1. 프롬프트 템플릿 정의
prompt = PromptTemplate(
    input_variables=["topic"],
    template="다음 주제에 대해 간단히 설명해줘: {topic}"
)

# 2. LLM 설정
llm = OpenAI(temperature=0.7)

# 3. 체인 구성
chain = LLMChain(llm=llm, prompt=prompt)

# 4. 실행
response = chain.run("칼만 필터")
print(response)
```

LangChain - Python 예시

# \*SDK 예시

```
import com.microsoft.semantickernel.*;
import com.microsoft.semantickernel.plugin.*;
import com.microsoft.semantickernel.plugin.KernelPluginFactory;
import com.microsoft.semantickernel.plugin.KernelPlugin;
import com.microsoft.semantickernel.plugin.KernelFunctionParameter;
import com.microsoft.semantickernel.plugin.annotations.DefineKernelFunction;
import com.microsoft.semantickernel.services.chatcompletion.*;
import com.azure.ai.openai.OpenAIAsyncClient;
import com.azure.ai.openai.OpenAIIClientBuilder;
import com.azure.core.credential.AzureKeyCredential;

import java.util.Map;

public class SemanticKernelExample {

    // 1. 플러그인 함수 정의
    public static class TextPlugin {
        @DefineKernelFunction(
            name = "summarize",
            description = "텍스트를 요약합니다",
            returnType = "String"
        )
        public String summarize(@KernelFunctionParameter(name = "input") String input) {
            return input.length() > 100 ? input.substring(0, 100) + "..." : input;
        }
    }
}
```

```
public static void main(String[] args) {
    // 2. OpenAI 클라이언트 설정
    OpenAIAsyncClient openAIclient = new OpenAIIClientBuilder()
        .endpoint("https://api.openai.com/v1")
        .credential(new AzureKeyCredential("your-api-key"))
        .buildAsyncClient();

    OpenAIChatCompletion chatCompletion = OpenAIChatCompletion.builder()
        .withOpenAIAsyncClient(openAIclient)
        .withModelId("gpt-3.5-turbo")
        .build();

    // 3. 커널 생성
    Kernel kernel = Kernel.builder()
        .withChatCompletionService(chatCompletion)
        .build();

    // 4. 플러그인 등록
    KernelPlugin plugin = KernelPluginFactory.createFromObject(
        new TextPlugin(), "TextPlugin");
    kernel.registerPlugin(plugin);

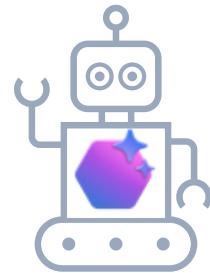
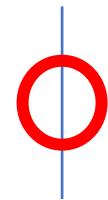
    // 5. 플러그인 함수 실행
    String result = kernel.invokeAsync("TextPlugin.summarize",
        Map.of("input", "Semantic Kernel을 Java에서 사용하는 예")).block();

    System.out.println("요약 결과: " + result);
}
```

# 3. MCP 등장 배경

Google 검색 (지식 도구)

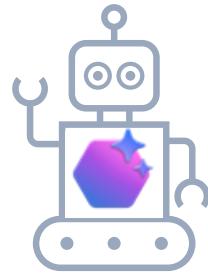
(LangChain SDK 사용)



AI 서비스 1  
(LangChain SDK 사용)

Google 검색 (지식 도구)

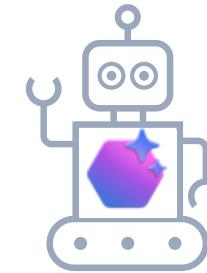
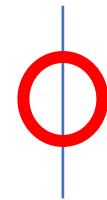
(Sementic Kernel SDK 사용)



AI 서비스 1  
(LangChain SDK 사용)

Google 검색 (지식 도구)

(Sementic Kernel SDK 사용)



AI 서비스 1  
(Sementic Kernel SDK 사용)

# 3. MCP 등장 배경

Google 검색 (지식 도구)

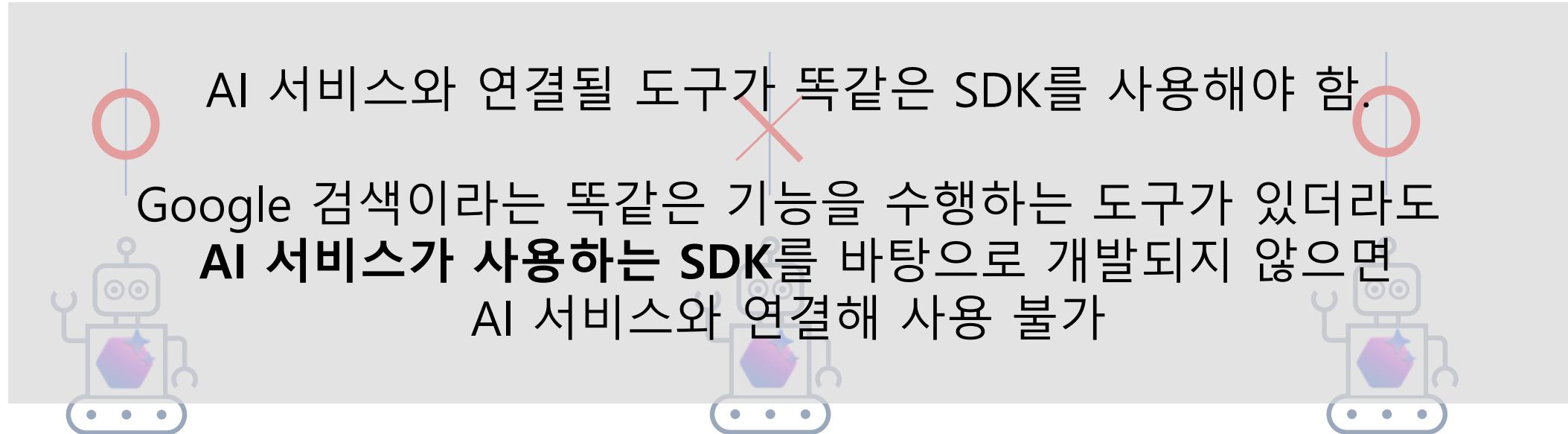
(LangChain SDK 사용)

Google 검색 (지식 도구)

(Sementic Kernel SDK 사용)

Google 검색 (지식 도구)

(Sementic Kernel SDK 사용)



AI 서비스 1  
(LangChain SDK 사용)

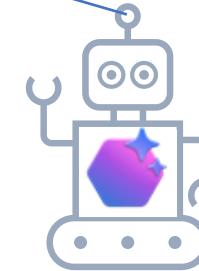
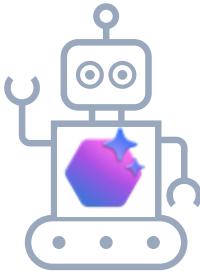
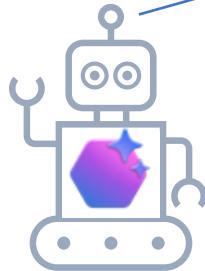
AI 서비스 1  
(LangChain SDK 사용)

AI 서비스 1  
(Sementic Kernel SDK 사용)

# 3. MCP 등장 배경

“어차피 똑같은 기능인데..한 개만 만들어서  
호환되게 할 순 없을까?”

Google 검색 (지식 도구)



AI 서비스 1  
(LangChain SDK 사용)

AI 서비스 1  
(LangChain SDK 사용)

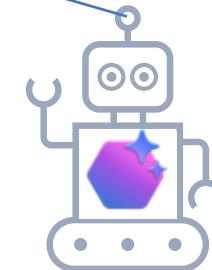
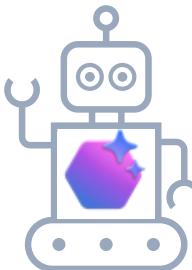
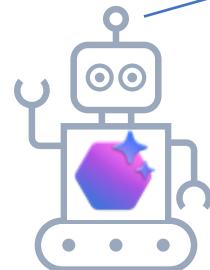
AI 서비스 1  
(Sementic Kernel SDK 사용)

# 3. MCP 등장 배경

“어차피 똑같은 기능인데..한 개만 만들어서  
호환되게 할 순 없을까?”

**Model Context Protocol  
(MCP)**

Google 검색 (지식 도구)



AI 서비스 1  
(LangChain SDK 사용)

AI 서비스 1  
(LangChain SDK 사용)

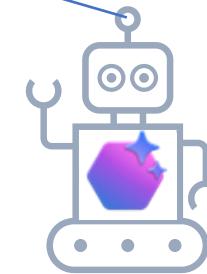
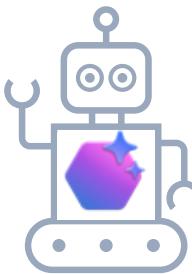
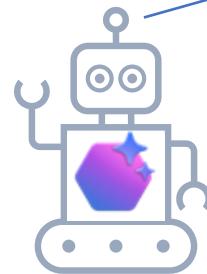
AI 서비스 1  
(Sementic Kernel SDK 사용)

# 4. MCP

## Model Context Protocol (MCP)

AI 서비스가 어떤 SDK를 사용해서 개발됐던  
이 앱을 도구로 인식하게 하는 방법/규칙

Google 검색 (지식 도구)



AI 서비스 1  
(LangChain SDK 사용)

AI 서비스 1  
(LangChain SDK 사용)

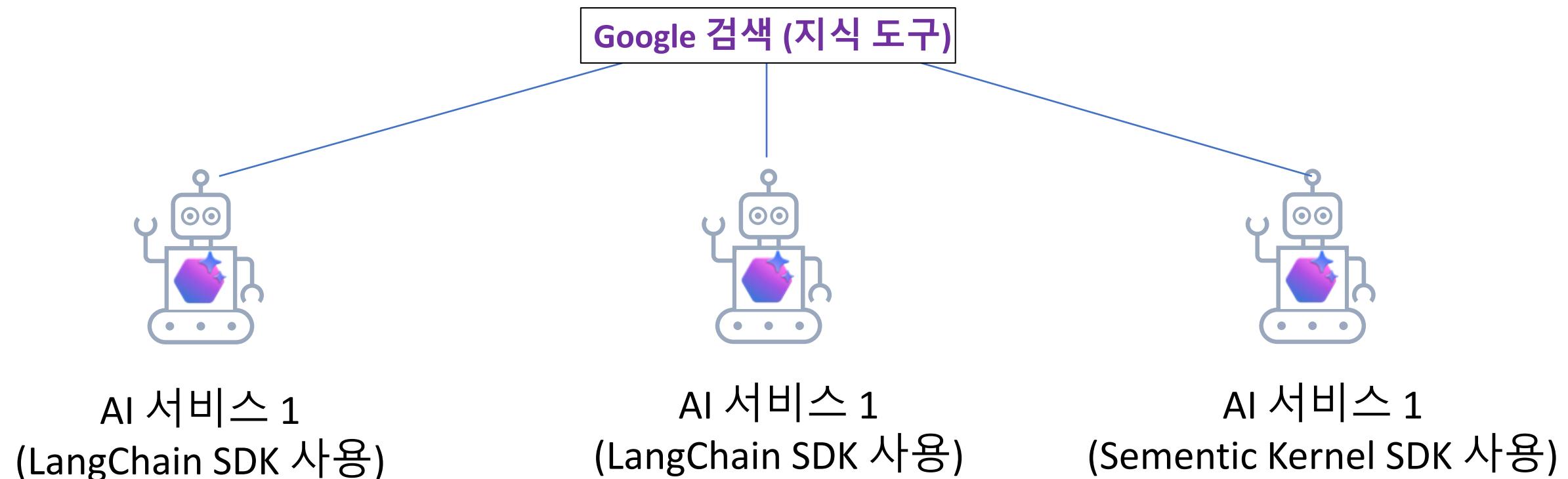
AI 서비스 1  
(Semantic Kernel SDK 사용)

# 4. MCP

## Model Context Protocol (MCP)

AI 서비스가 어떤 SDK를 사용해서 개발됐던  
이 앱을 도구로 인식하게 하는 방법/규칙

### @Tool(설명)



# @Tool : AI에게 사용 가능한 연장(기능)과 사용 설명서 제공

```
@Service
public class CalculatorService {

    /**
     * Add two numbers
     * @param a The first number
     * @param b The second number
     * @return The sum of the two numbers
     */
    @Tool(description = "Add two numbers together")
    public String add(double a, double b) {
        double result = a + b;
        return formatResult(a, "+", b, result);
    }

    /**
     * Subtract one number from another
     * @param a The number to subtract from
     * @param b The number to subtract
     * @return The result of the subtraction
     */
    @Tool(description = "Subtract the second number from the first number")
    public String subtract(double a, double b) {
        double result = a - b;
        return formatResult(a, "-", b, result);
    }
}
```

# MCP 호스트, 서버, 클라이언트

[출처: Norah 블로그]

