

AI 도입 여정

실무에서 마주한 문제와 해결의 길

.NET AI DAY 2025

㈜아이엠지
장아린 선임연구원/ 최영환 주임
E. arinee@iamz.co.kr
E. choeyh@iamz.co.kr

odinbox.co.kr



아이엠지의 서비스는 단순한 연결이 아닙니다.
우리의 가치는 연결, 그 이상의 특별함입니다.

※ 자동차 부품사 전반의 솔루션 보유

QMS, PMS, DMS, MMS, CMS

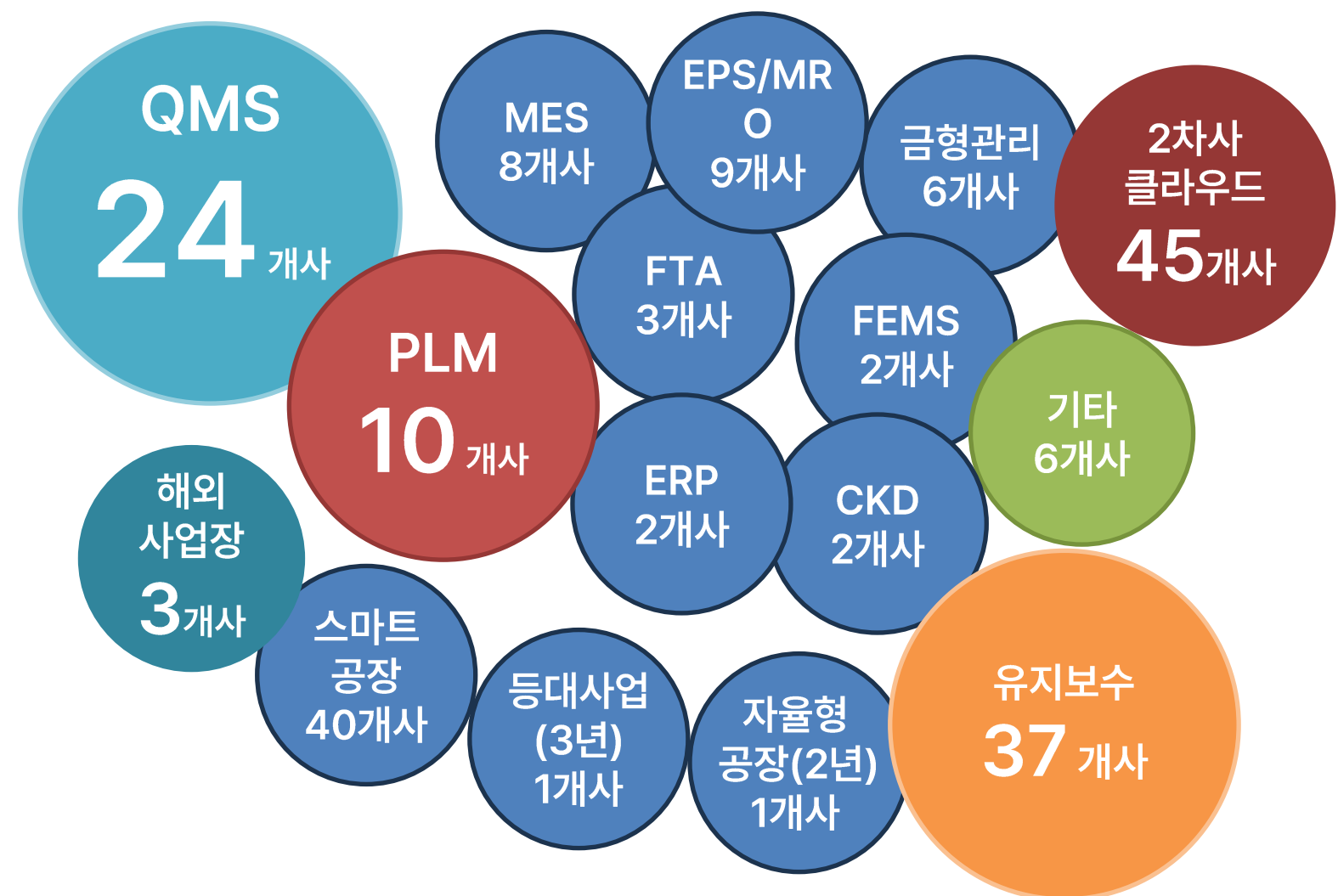
※ IAMZ PLM (based Aras Innovator PLM)

제품 엔지니어링, 프로젝트 관리, 개발품질 관리

※ 클라우드 솔루션 (OneCloud)

2/3차 부품사에서 전산관리와 보안이슈까지 한번에 해결

※ 뷰어 솔루션 (3D WorkSpace)



시스템 사업실적 (최근 3개년)

STEP 01

AI 도입 배경

- 고객들의 요구
- 경쟁사

“현장 데이터를 분석해서 이상징후를 자동으로 알려줄 수 있을까요?”

“AI로 업무 효율화를 할 수 있을까요?”

STEP 02

AI에 관한 부정적 의견

- 현실적인 문제 (비용)
- 보안

내부시선

"이거 그냥 IF-ELSE 잘 짜면 되는 거 아냐?"
"결과가 매번 다르다며? 그럼 검증은 어떻게 하지?"
"실제로 우리 고객들이 잘 쓸까?"
"그거 해봤자 비용만 들고 실효성이 없다"

기술적문제

"내부 시스템 연동 시 인프라 문제, 보안 이슈"
"결과 해석이 불투명하고 신뢰가 어려움"

STEP 03

AI 인식의 변화

- AI 역할에 관한 재고
 - 결정을 도와 주는 보조 역할
- 내부 교육
 - Rule based vs AI
- 기능 중심의 적용
 - 작게 시작해서 빠르게 보여주자

제안

1. 조직의 인식 변화

2. 점진적인 학습 과정

(주)아이엠지

PART 2.

실무에서

LLM 적용 하기

Chapter 1 챗봇 적용

Chapter 2 시멘틱커널

Chapter 3 향후

시연 영상

1. 홈페이지 로그인

2. 페이지 이동 (주간보고)

3. 챗봇

iamz GW - [주간업무보고현황] x

localhost:5666

iamz (주)아이엠지

장아린로그아웃

HOMEBIZ

주간업무보고현황

주간업무보고현황

검색 할 메뉴를 입력하세요.

BOOKMARK

결재관리

기안함

대기함

진행함

완료함

반려함

회람함

결재선

기준관리

자산구분마스터

주간보고

주간업무보고관리

주간업무보고현황

주간업무이슈사항

연차관리

연차관리

연차현황

교육관리

QNA

교육관리

지출품의서관리

지출품의서

비품구입

자산관리

자산관리대장

라이선스관리

라이선스관리

인증서관리

기타관리

사이트관리

주간업무보고현황

기준일 2025-05-30 05 월 5 주차

작성자인차댓글

김병수 부장5주차0

기준일

제목

참여프로젝트

이번주보고내용

다음주보고내용

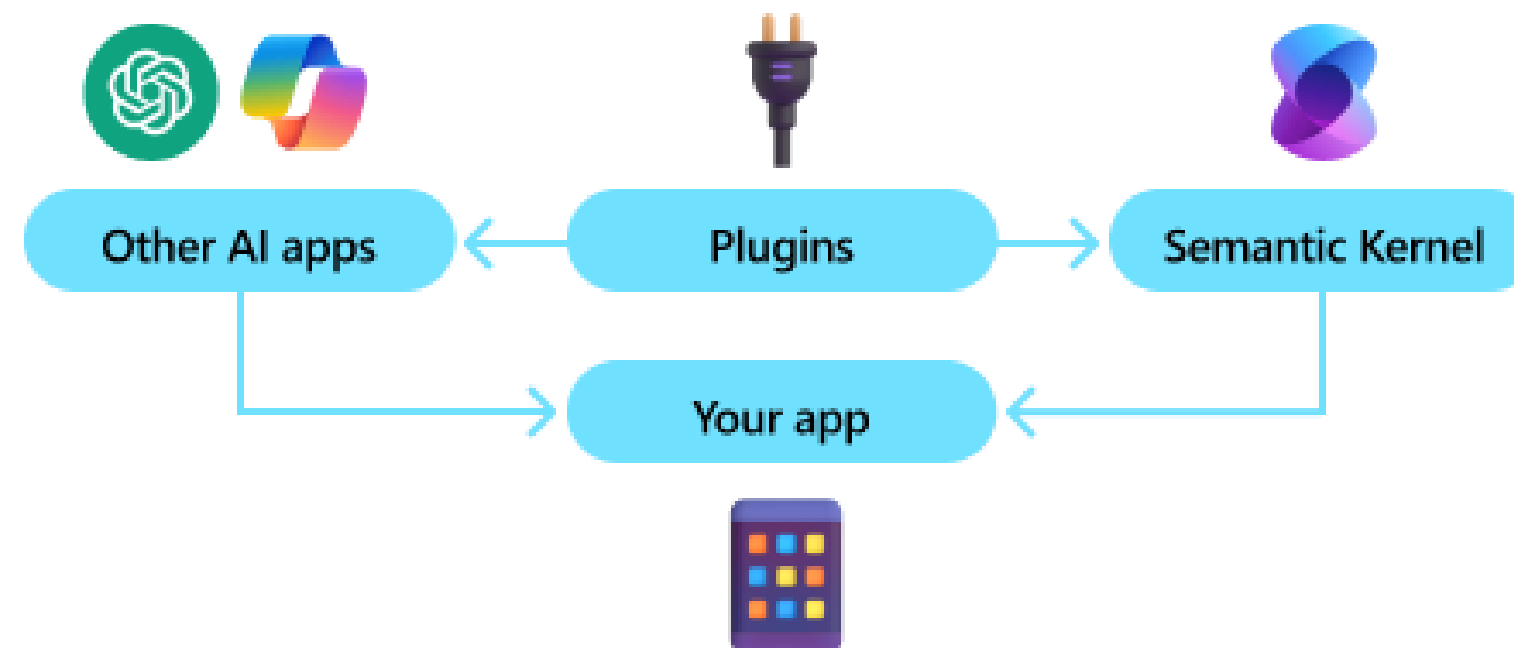
특이사항

비고

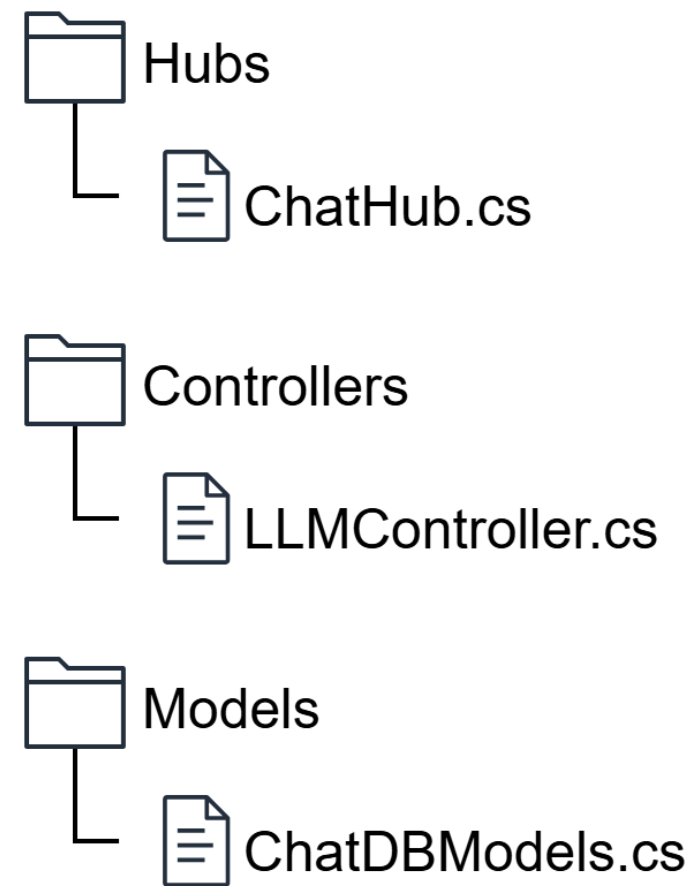
1건이 조회되었습니다.

COPYRIGHT © 2024 iamz Inc. ALL RIGHTS RESERVED

Architecture



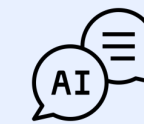
File Structure



LLM Orchestration



1. Prompting LLMs



2. Chaining LLMs

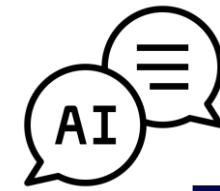


3. Managing LLM Resources

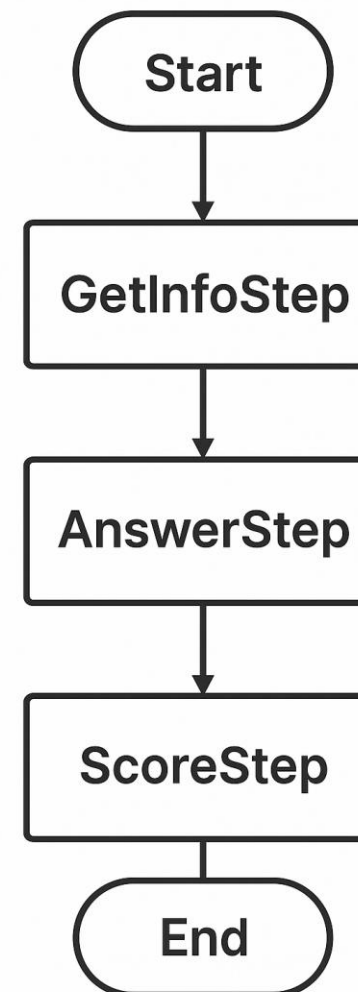


4. Monitoring LLM Resources

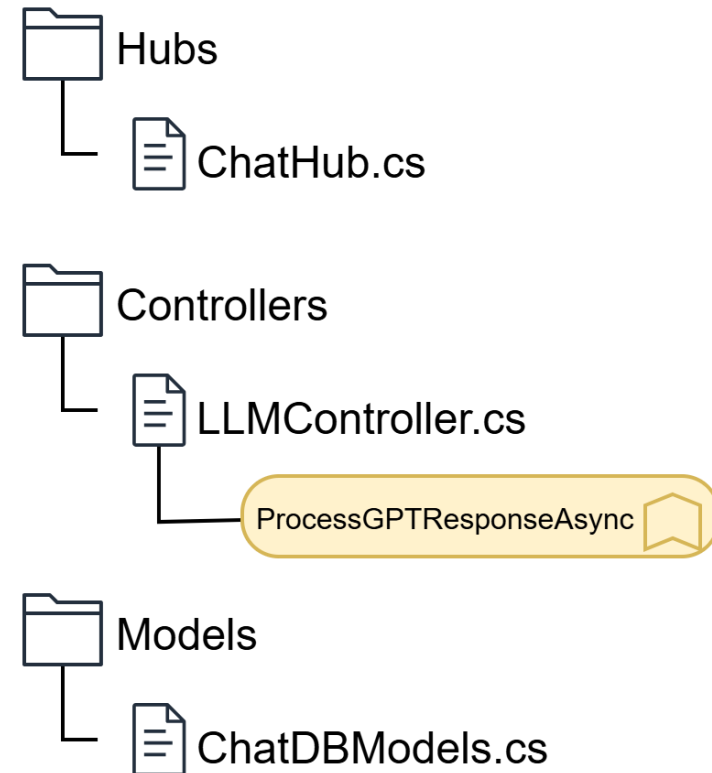




Prompting LLMs



Prompting LLMs



```
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
~ ~ ~

참조 2개
public async Task<JsonResult> ProcessGPTResponseAsync(string chat)
{
    try
    {
        var config = new ConfigurationBuilder()
            .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)
            .Build();

        // Create the process builder
        ProcessBuilder processBuilder = new("DocumentationGeneration");

        // Add the steps
        var step_1 = processBuilder.AddStepFromType<GetInfoStep>(); // 정보 수집
        var step_2 = processBuilder.AddStepFromType<AnswerStep>(); // 답변
        var step_3 = processBuilder.AddStepFromType<ScoreStep>(); // 점수 매기기

        processBuilder
            .OnInputEvent("Start")
            .SendEventTo(new(step_1));

        step_1
            .OnFunctionResult()
            .SendEventTo(new(step_2));

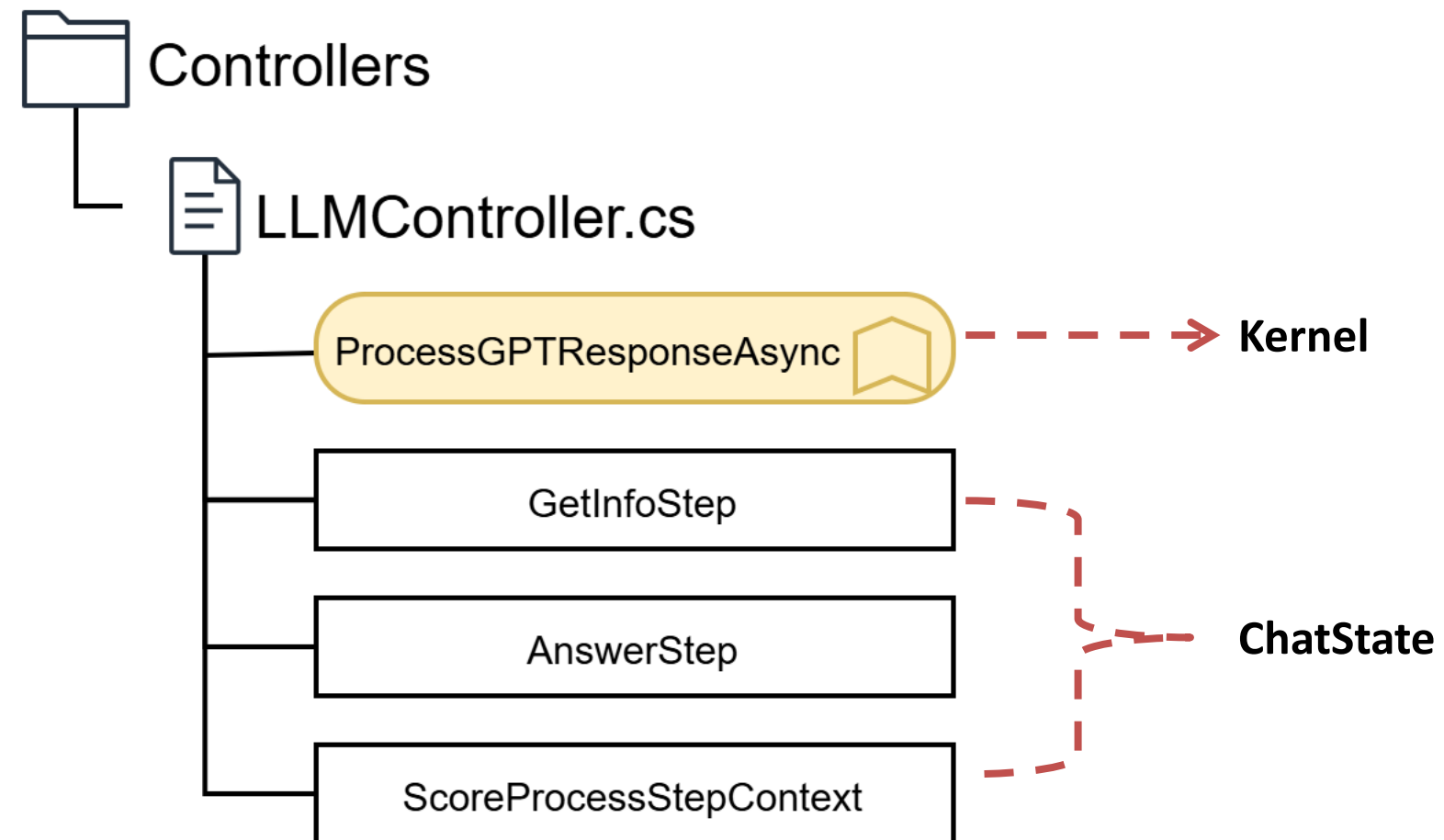
        step_2
            .OnFunctionResult()
            .SendEventTo(new(step_3));

        // Configure the kernel with your LLM connection details
        #pragma warning disable SKEXP0070
        Kernel kernel = Kernel.CreateBuilder()
            .AddOllamaChatCompletion(
                modelId: "llama3",
                endpoint: new Uri(config["Ollama:Endpoint"]!)
            ).Build();
        kernel.Data.Add("Question", new { question = chat });

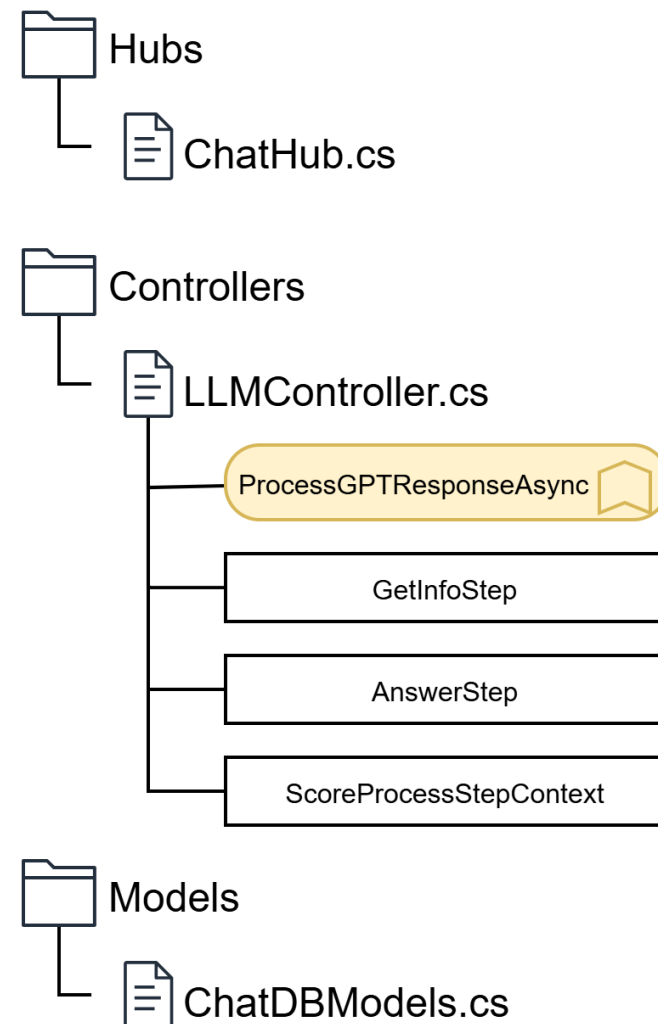
        // Build and run the process
        var process = processBuilder.Build();
        await process.StartAsync(kernel, new KernelProcessEvent { Id = "Start" });
    }
}
```



Chaining LLMs



Chaining LLMs



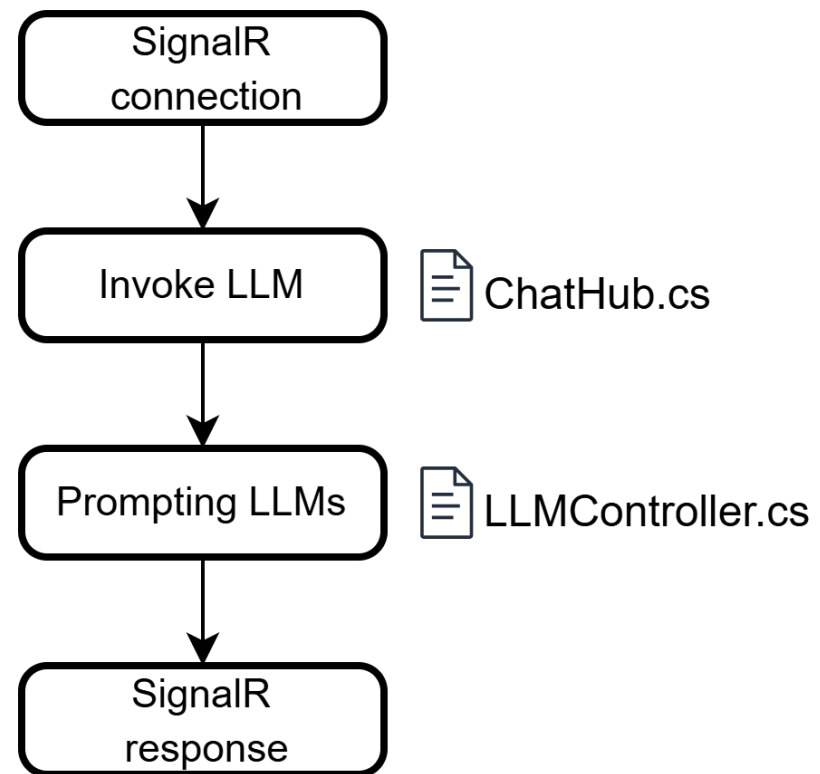
```
53  
54 [KernelFunction]  
55 참조 0개  
56 public async Task GetInfoAsync(Kernel kernel, KernelProcessStepContext context)  
57 {  
58     try  
59     {  
60         var config = new ConfigurationBuilder()  
61             .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)  
62             .Build();  
63         var chat = kernel.Data["Question"].ToString();  
64         bool isKorean = Regex.IsMatch(chat, @"[\u3131-\ud79d]");  
65         string langNote = isKorean ? "반드시 한국어로 답변해. 영어는 절대 사용하지 마." : "Respond strictly in English. Do not inclu  
66  
67         var agentKernel = Kernel.CreateBuilder()  
68             .AddOpenAIChatCompletion(  
69                 modelId: "gpt-4o",  
70                 apiKey: config["OpenAI:ApiKey"]  
71             ).Build();  
72  
73  
74         // 2. agent 생성  
75         var agent = new ChatCompletionAgent()  
76         {  
77             Name = "Host",  
78             Instructions = "Do not respond directly. Select and call a function that fits the user's question.\r\nSupposed you  
79             Kernel = agentKernel,  
80             Arguments = new KernelArguments(new PromptExecutionSettings()  
81             {  
82                 FunctionChoiceBehavior = FunctionChoiceBehavior.Required()  
83             }  
84         };  
85  
86         // 3. agent 커널 플러그인 연결  
87         agent.Kernel.Plugins.AddFromType<ChatDBModel>();  
88  
89         // 4. 결과 출력  
90         var toolResponse = "";  
91         var messages = new List<ChatMessageContent>  
92         {  
93             new(AuthorRole.User, chat)  
94         };  
95
```




Managing LLM Resources

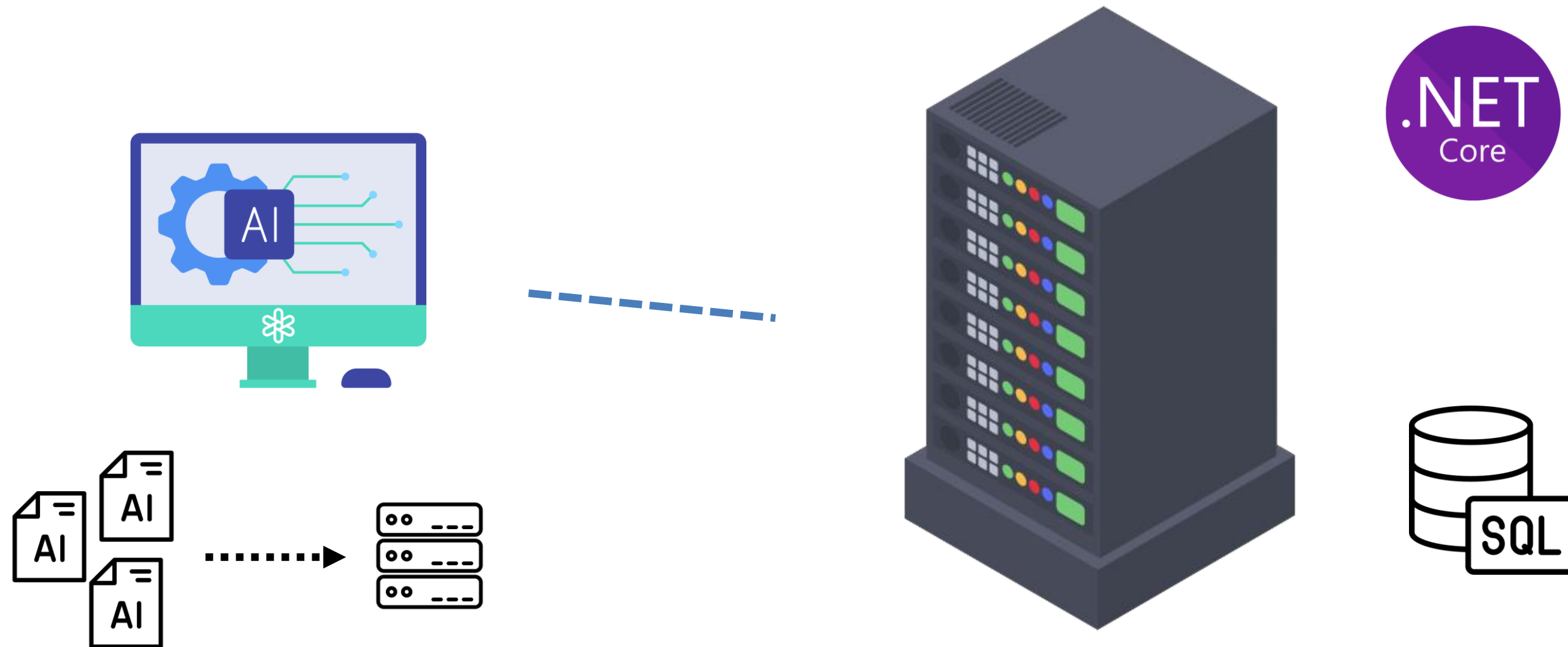


Managing LLM Resources



```
20  |  참조 1개
21  |  public class ChatHub : Hub
22  |  {
23  |      static string uuid; // 채팅 세션 키
24  |      static ChatHistory chatHistory;
25  |
26  |      // chat session 생성
27  |      참조 0개
28  |      public JsonResult CreateChatSession() ...
29  |
30  |      // llm 호출
31  |      참조 0개
32  |      public async Task<JsonResult> GetAIMessageAsync(string chat)
33  |      {
34  |          try
35  |          {
36  |              Console.WriteLine(">>GetAIMessageAsync");
37  |
38  |              chatHistory.AddUserMessage(chat);
39  |
40  |              LLMController lLMController = new LLMController();
41  |              JsonResult response = await lLMController.ProcessGPTResponseAsync(chat);
42  |              Console.WriteLine(JsonSerializer.Serialize(response));
43  |              chatHistory.AddAssistantMessage(JsonSerializer.Serialize(response.Value));
44  |
45  |              return response;
46  |          }
47  |          catch (Exception ex)
48  |          {
49  |              Console.WriteLine($"Error while waiting for notifications: {ex.Message}");
50  |              return new JsonResult(ex.Message);
51  |          }
52  |      }
53  |
54  |      // DB 저장
55  |      참조 0개
56  |      public JsonResult UpdateChatSession() ...
57  |
58  |  }
```

Monitoring LLM Resources



Part 1
과정과 문제

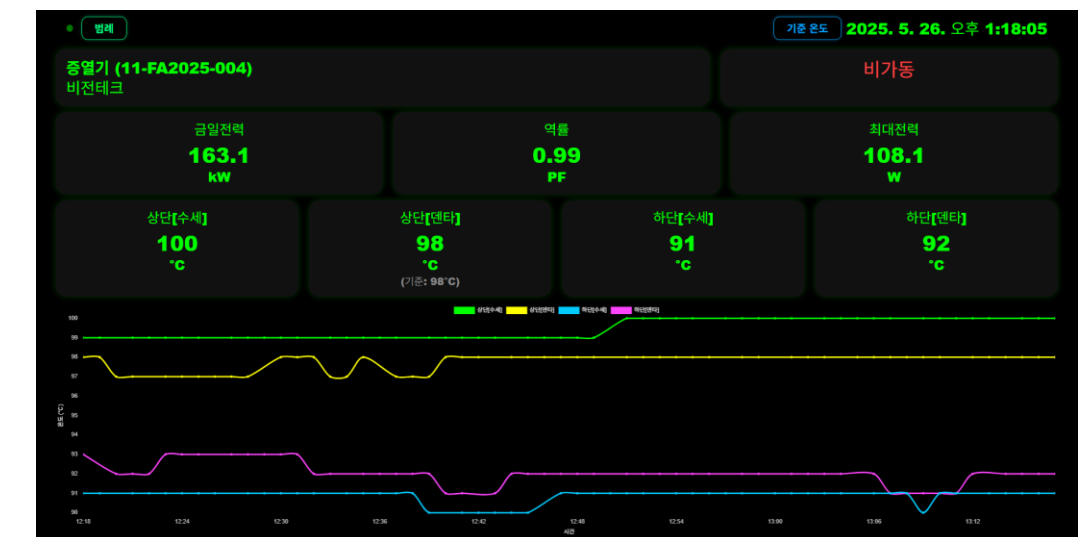


FEMS



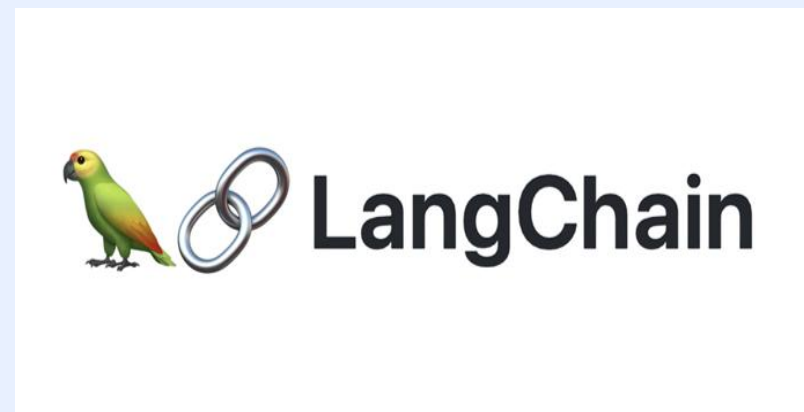
공장에너지관리시스템 (Factory Energy Management System)

- 설비 계측기의 유효전력을 통해 설비 가동/비가동 판단
- 설비 계측기의 역률을 통해 전력품질을 예측



Semantic Kernel으로 전환

V.1



LangChain

V.2



Semantic Kernel

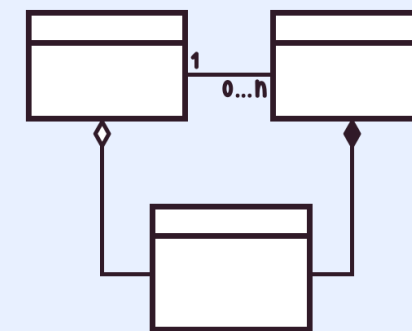
Semantic Kernel으로 전환

V.1



Vector DB

V.2



Relational DB

Thank you

㈜아이엠지
장아린 선임연구원/ 최영환 주임
E. arinee@iamz.co.kr
E. choeyh@iamz.co.kr

odinbox.co.kr

