



Part of the AgentCon World Tour by the Global AI Community

# AI Agents Developer Conference

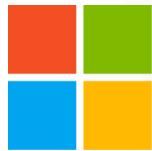
[agentcon.city/zurich](http://agentcon.city/zurich)



# Made Possible by



SwissLife



Microsoft







MCP IN .NET  
FROM ZERO TO

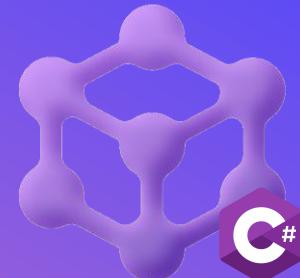


# CÉDRIC MENDELIN

Software Developer & Team Lead



isolutions<sup>®</sup>



# AGENDA

Motivation

C# ❤️ AI

Function Calling

MCP in .NET

Security

Conclusion



# MOTIVATION

---

+

•

○

dictatorial /diktətərēəl/ adj.  
like a dictator. 2 overbearing  
oritally adv. [Latin: related  
TATOR]

**diction** /'diks(ə)n/ n. manner  
ciation in speaking or singing  
*dictio* from *dico* dict- say]

**dictionary** /'diksənəri/ n. (pl.)  
book listing (usu. alphabetica)  
explaining the words of a lan-  
giving corresponding words in  
language. 2 reference book e-

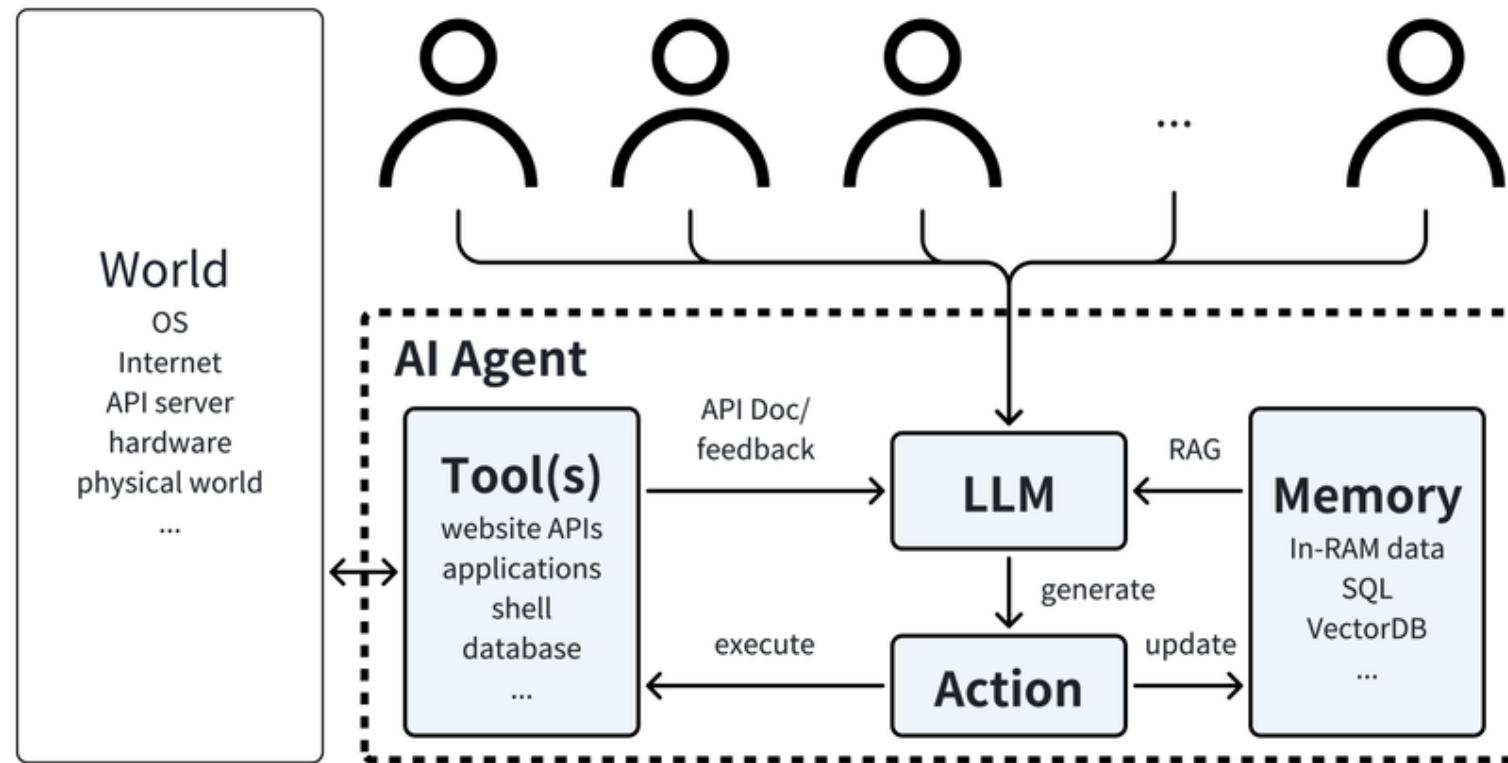
**What is the current date?**

141





# LLM BASED AI AGENT



C# ❤️ AI

---

+  
o  
•

# C# SDKS

**Microsoft.Extensions.AI (Middleware)**

**Microsoft.Extensions.AI.Abstractions**

**Microsoft.Extensions.VectorData**

**AI SDKS**  
**(OllamaSharp, OpenAI, Azure AI Inference)**

**Vector Store SDKs**

**Libraries**  
**(Semantic Kernel, AutoGen, Agent Framework)**

**MEAI.Evaluation**

# C# SDKS



+



=



Semantic Kernel

Autogen

Microsoft Agent  
Framework



ModelContextProtocol  
ModelContextProtocol.Core  
ModelContextProtocol.AspNetCore

# MEAI – SETUP

```
1 var endpoint = "";
2 var apiKey = "";
3 var modelId = "";
4
5 var azureClient = new AzureOpenAIClient(
6     new Uri(endpoint),
7     new ApiKeyCredential(apiKey));
8
9 var chatClient = azureClient.GetChatClient(modelId).AsIChatClient();
```

# MEAI – EXECUTE PROMPT

```
1 // initialize history & settings
2 var prompt = "How do I get from Zurich HB to the Swisslife office?";
3 List<ChatMessage> history = [new ChatMessage(ChatRole.User, prompt)];
4 var options = new ChatOptions();
5
6 // execute prompt
7 var response = await chatClient.GetResponseAsync(history, options);
8 //response.Text
```

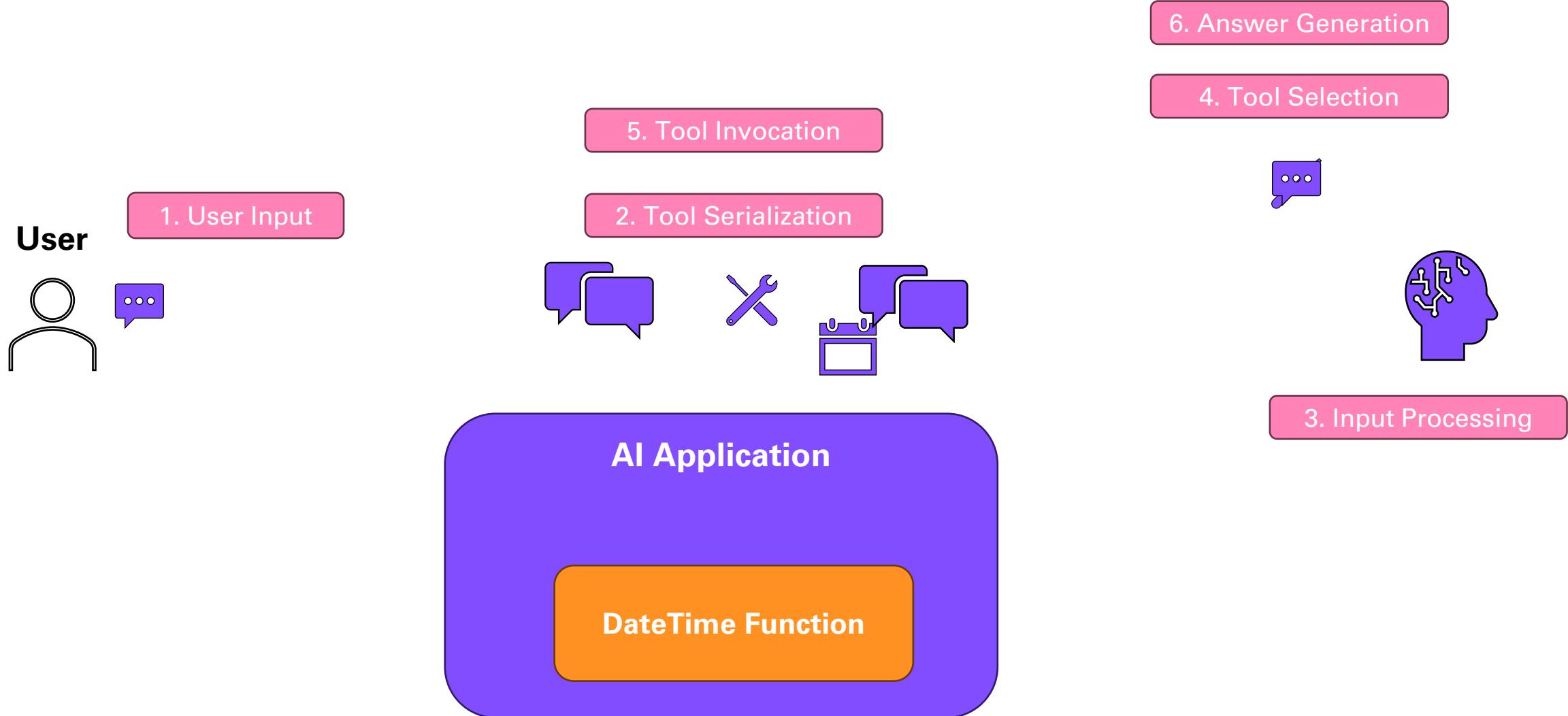
**What is the current date?**

# FUNCTION CALLING



---

# FUNCTION CALLING



# MEAI – FUNCTION CALLING

```
1 var functionCallClient = new ChatClientBuilder(chatClient)
2     .UseFunctionInvocation()
3     .Build();
4
5 var options= new ChatOptions
6 {
7     Tools = [AIFunctionFactory.Create(
8         () => DateTime.UtcNow,
9         "get_current_datetime",
10        "Gets the current UTC date and time")],
11     ToolMode = ChatToolMode.Auto // RequireAny, None
12 };
13
14 var response = await functionCallClient.GetResponseAsync(history, options);
```

# MCP IN .NET

---

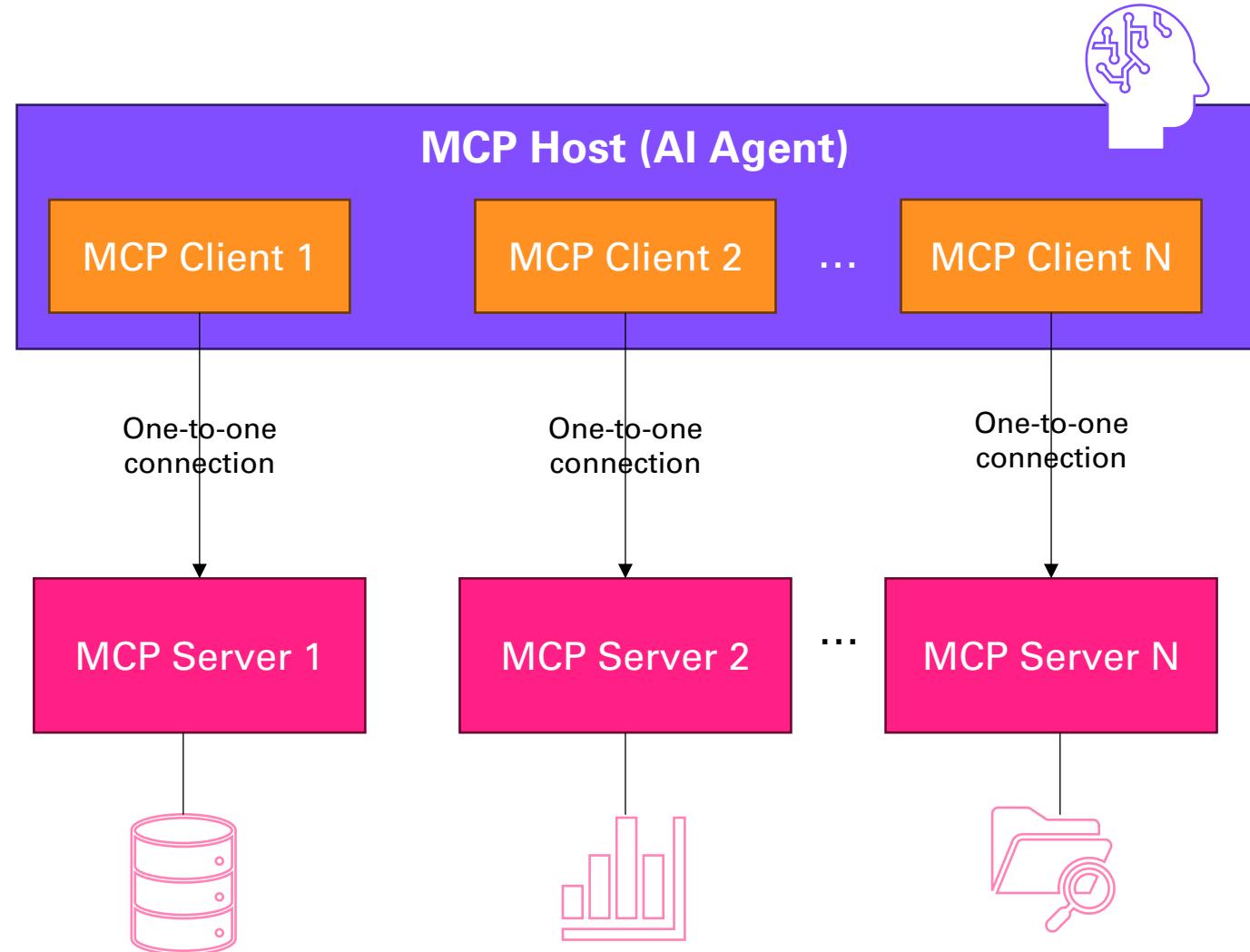
+

•

○

# MODEL CONTEXT PROTOCOL (MCP)

- Client – Server
- Transport
  - Stdio (local)
  - Http (remotely)
    - Server-Sent Events for Streaming
- JSON-RPC





# USE-CASES

## Developer Use Case

- Server locally – stdio
- Integrated in AI Coding assistant

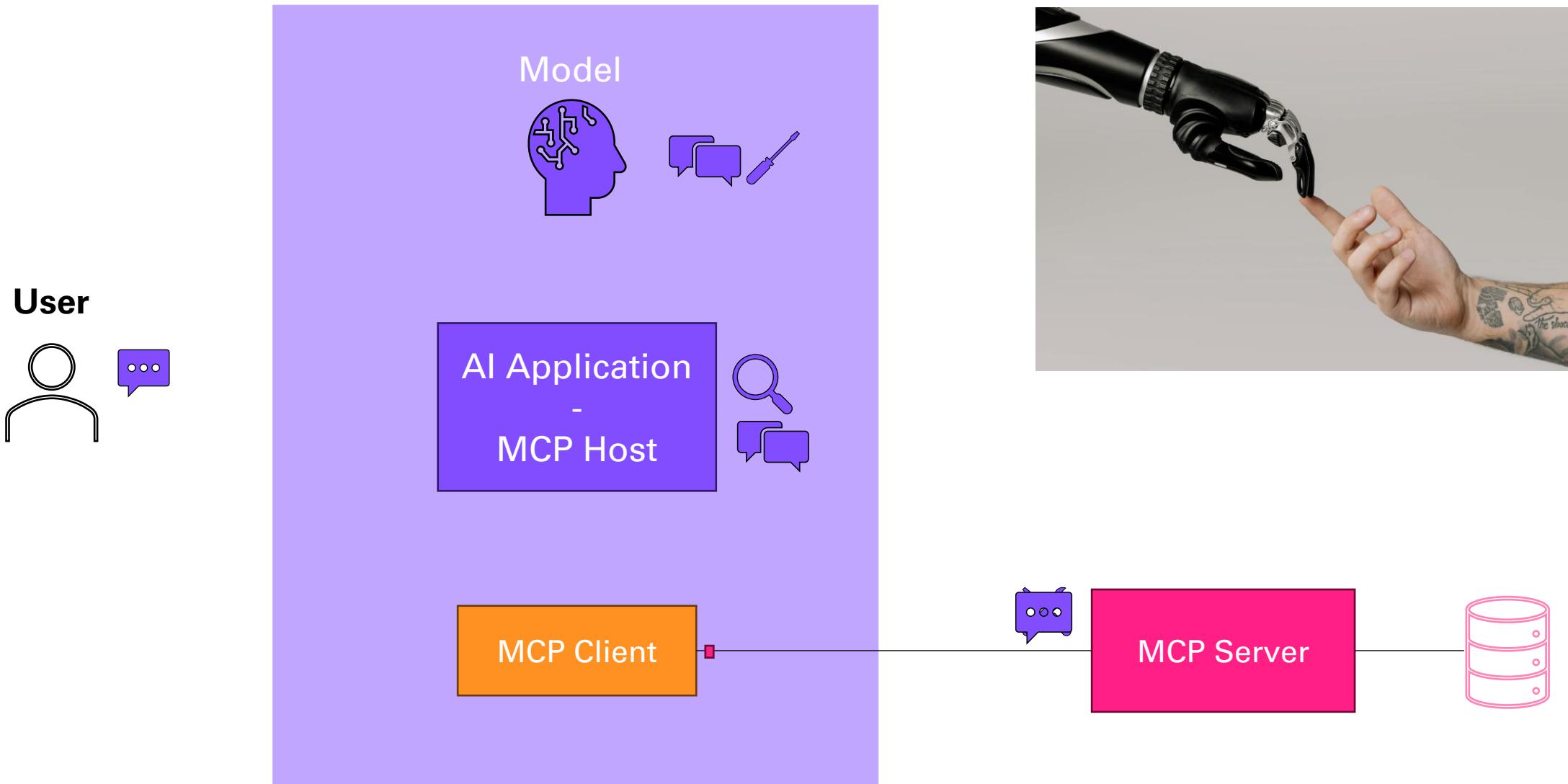


## Business Use Case

- Server remotely – http - API
- Integrated in your AI assistant



# MCP - FLOW



# MCP SERVER - TOOL

```
1 [McpServerToolType]
2 public class DateTools
3 {
4     [McpServerTool]
5     [Description("Returns the current date and time in ISO 8601 format.")]
6     public DateTime GetCurrentDateTime(IMcpServer mcpServer)
7         => DateTime.UtcNow;
8 }
9
```

# MCP SERVER

```
1 var builder = WebApplication.CreateBuilder(args);
2
3 builder.Services
4     .AddMcpServer()
5     .WithHttpTransport()
6     .WithTools<DateTools>();
7
8 // Add services to the container.
9 var app = builder.Build();
10
11 // map the mcp endpoint
12 app.MapMcp("/mcp");
13
14 await app.RunAsync();
```

# AI AGENT & MCP CLIENT

```
1 // create mcp client
2 var transport = new SseClientTransport(new()
3 {
4     Endpoint = new Uri("https://localhost: 7133/mcp"),
5     Name = "MCP Desktop Client",
6 },
7 httpClient);
8
9 await using IMcpClient mcpClient = await McpClientFactory.CreateAsync(transport);
10
11 // Retrieve the list of tools available on the MCP server
12 var mcpTools = await mcpClient.ListToolsAsync().ConfigureAwait(false);
13 var aiTools = mcpTools.Cast<AITool>();
14
15 // add tools to chatOptions and run execute prompt
16 ...
```

# MCP CAPABILITIES



## MCP Server

- Tools
- Resources
- Prompts
- Utilities
  - Completion
  - Logging
  - Pagination

## MCP Client

- Roots
- Elicitation
- Sampling

<https://modelcontextprotocol.io>

# DEMO

---

+

•

○

# MCP INSPECTOR

npx @modelcontextprotocol/inspector

MCP Inspector v0.16.8

Transport Type

Streamable HTTP

URL

<https://mcpoauthsecurity-hag0drcke>

Server Entry

Servers File

> Authentication

> Configuration

Reconnect

Disconnect

Connected

Resources Prompts Tools Ping Sampling Elicitations Roots Auth

Tools

List Tools

Clear

get\_random\_number

Generates a random number between the specified minimum and maximum values.

get\_current\_date\_time

Returns the current date and time in ISO 8601 format.

get\_random\_number\_from\_date\_time

Generates a random number based on a date.

get\_current\_date\_time

Returns the current date and time in ISO 8601 format.

Run Tool

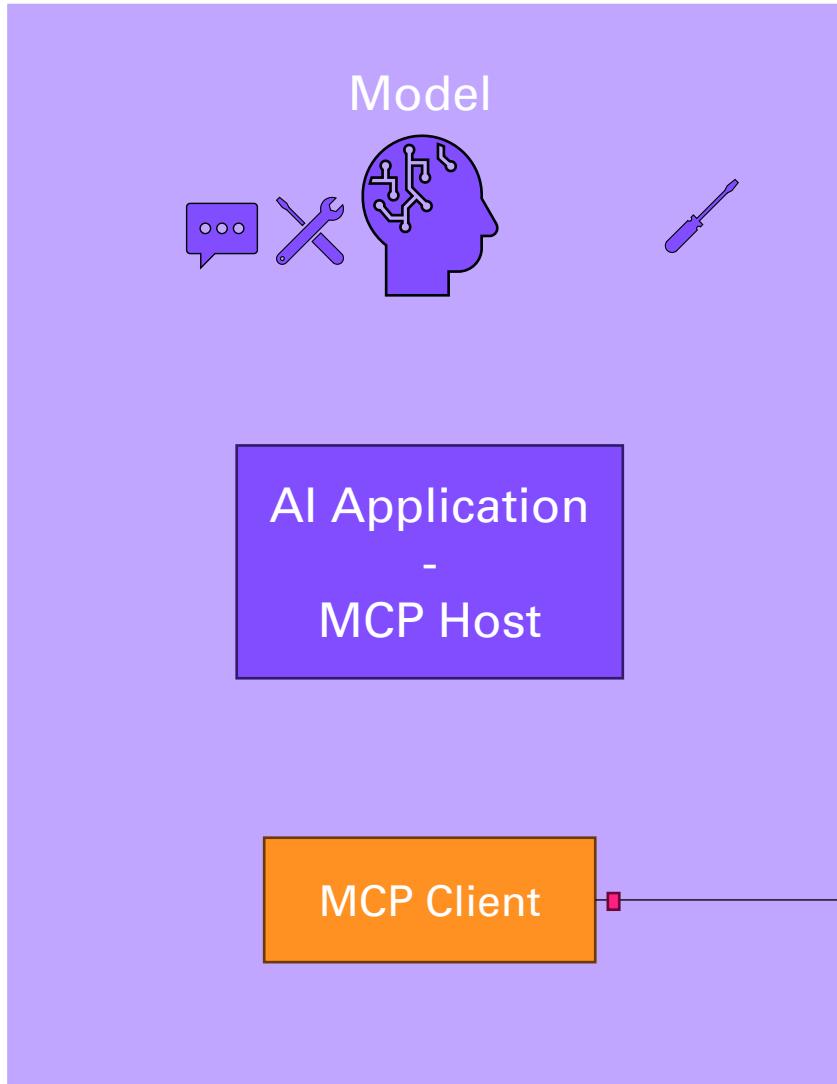
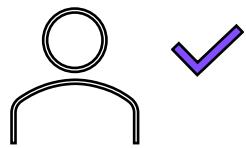
Copy Input

Tool Result: Success

"2025-09-22T11:42:44.1786351Z"

# MCP - ELICITATION

User



Human in the loop



# MCP SERVER – ELICITATION

```
1 [McpServerTool]
2 [Description("Returns the current date and time in ISO 8601 format.")]
3 public Task<string> GetCurrentDateTime(IMcpServer mcpServer)
4 {
5     if (mcpServer.ClientCapabilities?.Elicitation == null)
6     {
7         throw new McpException("Client does not support elicitation");
8     }
9
10    // start elicitation
11    var userAccepted = await mcpServer.ElicitAsync(new ElicitRequestParams()
12    {
13        Message = $"Do you want to execute tool '{nameof(GetCurrentDateTime)}'?",
14        RequestedSchema = new RequestSchema()
15        {
16            Properties = { ["answer"] = new BooleanSchema() }
17        }
18    });
19
20    if (userAccepted.Action != "accept" ||
21        userAccepted.Content?["answer"].ValueKind != JsonValueKind.True)
22    {
23        throw new McpException("User declined to proceed");
24    }
25
26    return DateTime.UtcNow.ToString("o"));
27 }
```

# MCP CLIENT – ELICITATION

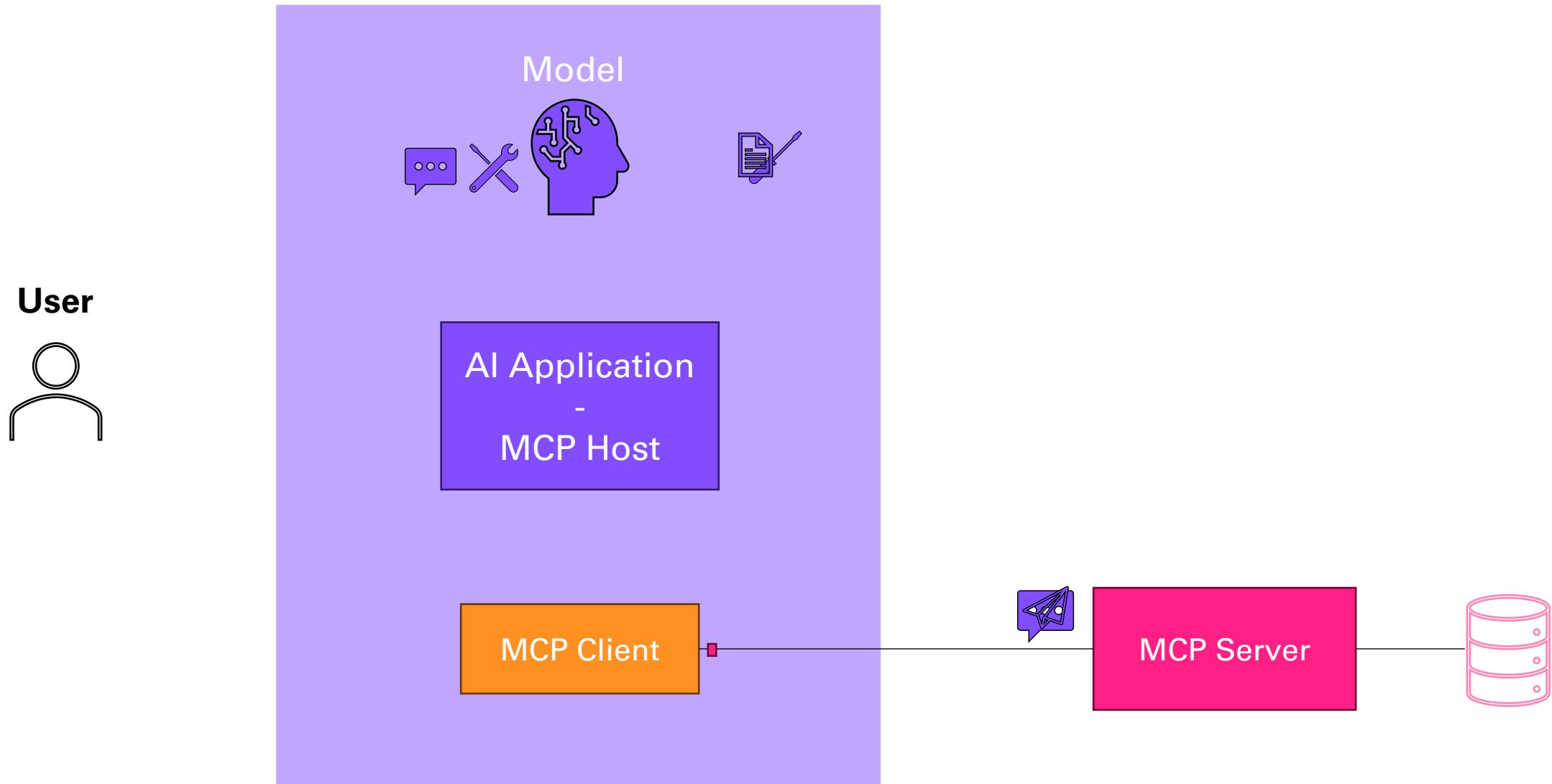
```
1 var transport = ...
2 await using IMcpClient mcpClient = await McpClientFactory.CreateAsync(
3     transport,
4     new McpClientOptions()
5     {
6         ClientInfo = new()
7         {
8             Name = "McpDemoClient",
9             Version = "1.0.0",
10        },
11        Capabilities = new()
12        {
13            Elicitation = new()
14            {
15                ElicitationHandler = HandleElicitationAsync,
16            }
17        }
18    });

```

# MCP CLIENT – ELICITATION HANDLER

```
1 public static ValueTask<ElicitResult> HandleElicitationAsync(
2     ElicitRequestParams? requestParams,
3     CancellationToken token)
4 {
5     Console.WriteLine(requestParams.Message);
6     Console.WriteLine($"Please allow/decline function execution with [y,n]");
7     var key = Console.ReadKey();
8     var userAllowance = key.KeyChar == 'y' || key.KeyChar == 'Y';
9     var content = new Dictionary<string, JsonElement>();
10    content["answer"] = JsonSerializer.Deserialize<JsonElement>(
11        JsonSerializer.Serialize(userAllowance));
12
13    return ValueTask.FromResult(new ElicitResult
14    {
15        Action = "accept",
16        Content = content
17    });
18 }
```

# MCP - SAMPLING



# DEMO

---

+

•

○

# SECURE MCP SERVERS

---

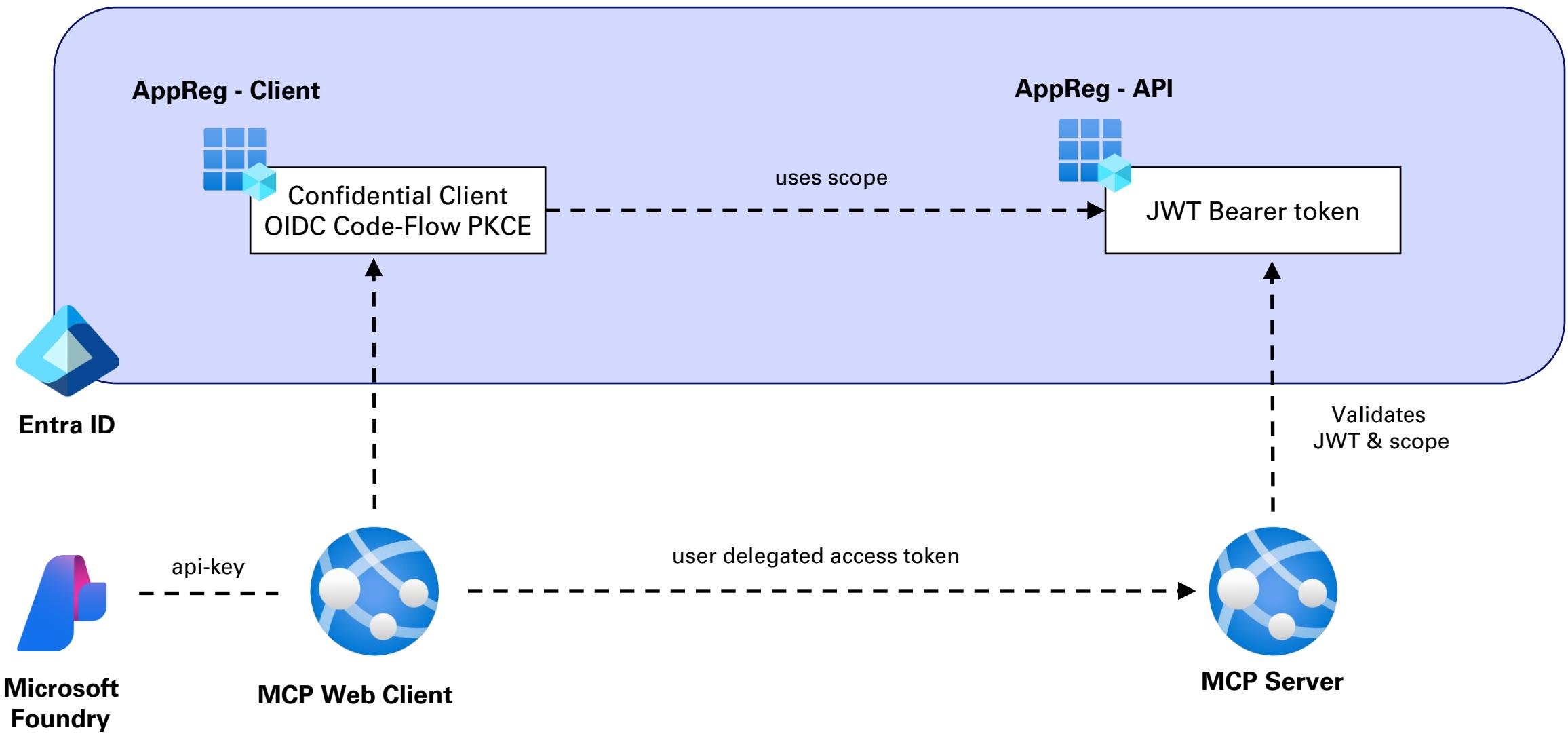


# SECURITY BEST PRACTICES

- MCP Server is basically an API – apply best-practices
- Client: Use OpenID Connect (OIDC)
  - Acquire the access token
- Server: Use delegated user access tokens only
  - Validate bearer token

The S in MCP stands for Security

# DEMO SETUP - MICROSOFT ENTRA ID



# MCP SERVER OAUTH SECURITY

```
1 builder.Services.AddMicrosoftIdentityWebApiAuthentication(builder.Configuration);
2 string httpMcpServerUrl = builder.Configuration["HttpMcpServerUrl"]!;
3
4 builder.Services.AddAuthentication()
5 .AddMcp(options =>
6 {
7     options.ResourceMetadata = new()
8     {
9         Resource = new Uri(httpMcpServerUrl),
10        ResourceDocumentation = new Uri("https://docs/health"),
11        ScopesSupported = ["mcp:tools"],
12    };
13 });
14
15 builder.Services.AddAuthorization();
16
```

# MCP SERVER OAUTH SECURITY

```
1 // change to scp or scope if not using magic namespaces from MS
2 // The scope must be validate as we want to force only delegated access tokens
3 // The scope is requires to only allow access tokens intended for this API
4 builder.Services.AddAuthorizationBuilder()
5     .AddPolicy("mcp_tools", policy =>
6         policy.RequireClaim("http://schemas.microsoft.com/identity/claims/scope",
7             "mcp:tools"));
```

# MCP SERVER OAUTH SECURITY

```
1 app.UseAuthentication();
2 app.UseAuthorization();
3
4 app.MapMcp("/mcp").RequireAuthorization("mcp_tools");
```

# CONFIDENTIAL OIDC MCP CLIENT

```
1 builder.Services.AddAuthentication(OpenIdConnectDefaults.AuthenticationScheme)
2     .AddMicrosoftIdentityWebApp(builder.Configuration.GetSection("AzureAd"))
3     .EnableTokenAcquisitionToCallDownstreamApi(
4         ["api://96b0f495-3b65-4c8f-a0c6-c3767c3365ed/mcp:tools"])
5     .AddInMemoryTokenCaches();
6
7 builder.Services.AddAuthorization(options =>
8 {
9     options.FallbackPolicy = options.DefaultPolicy;
10 });
11
12 builder.Services.AddRazorPages()
13     .AddMicrosoftIdentityUI();
```

# CONFIDENTIAL OIDC MCP CLIENT

```
1 var accessToken = await _tokenAcquisition  
2     .GetAccessTokenForUserAsync(["api://96b0f495-3b65-4c8f-a0c6-c3767c3365ed/mcp:tools"]);  
3  
4 await _chatService.EnsureSetupAsync(_clientFactory, accessToken);
```

# DEMO

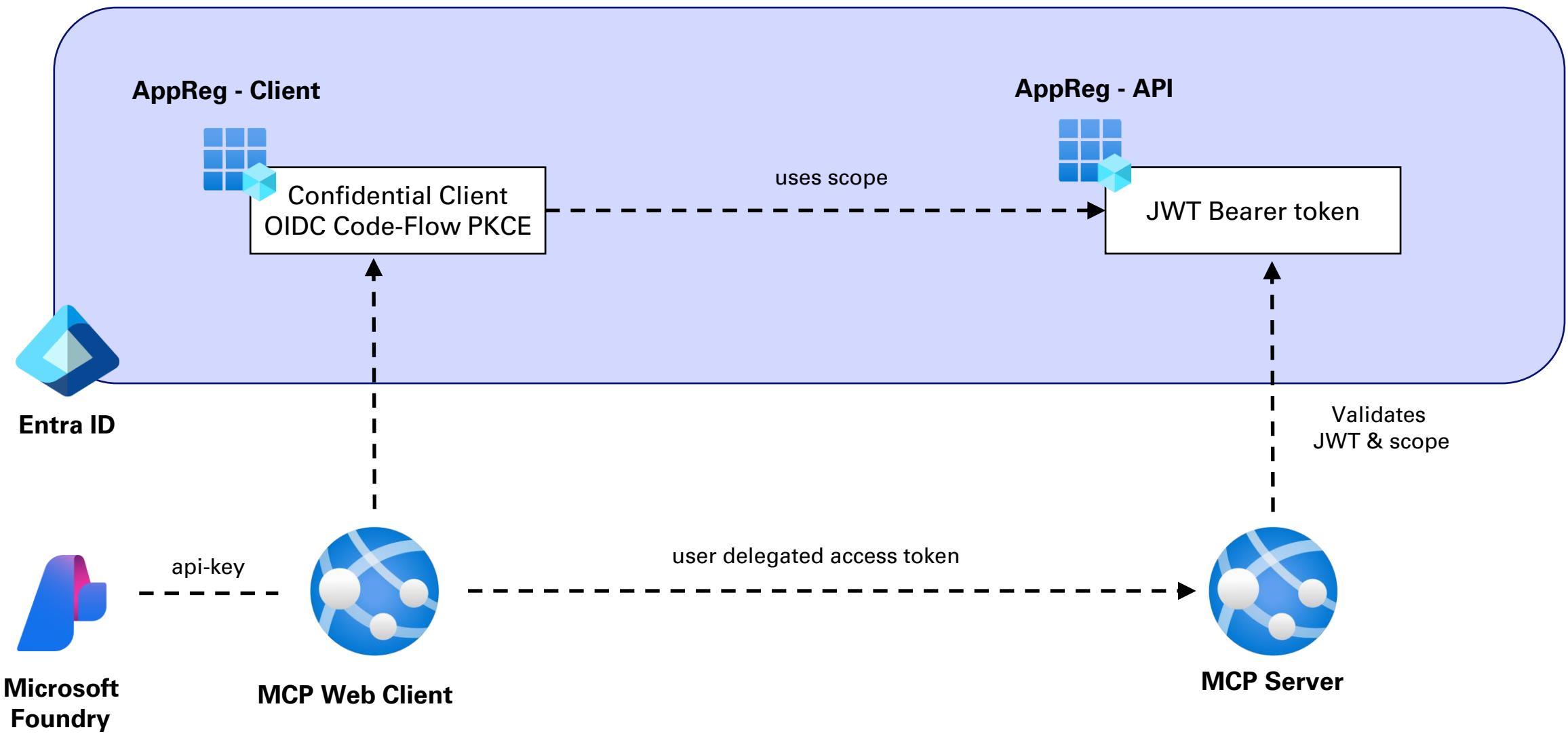
---

+

•

○

# DEMO SETUP - MICROSOFT ENTRA ID



# RECAP & RECOMMENDATIONS



---

# RECAP & RECOMMENDATIONS

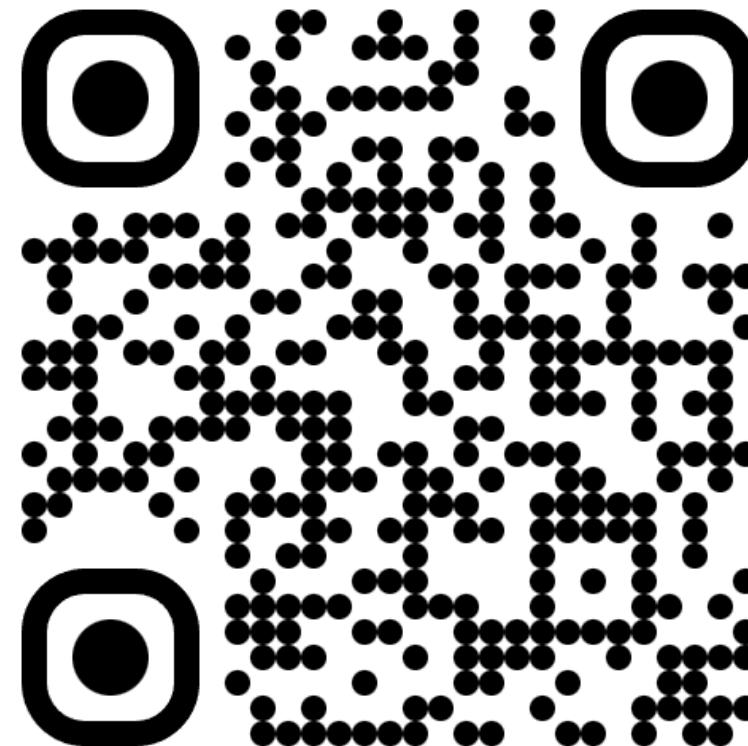
- Start thinking about **Business Use Case**
- Use advanced capabilities - Elicitation and Sampling
- C# ❤️ AI & MCP
- Human in-the-loop
- Apply security best-practices
- <https://github.com/damienbod/McpSecurity>

isolutions<sup>®</sup>

Security Meetup @isolutions  
05.02.2026

Continue the conversation after agentcon

## JOIN THE SWISS CHAPTER ON DISCORD



[gaic.io/discord](https://gaic.io/discord)

# REFERENCES

- [MEAI Documentation](#)
- [MCP C# SDK](#)
- [MCP Docs](#)
- [MCP Inspector](#)
- [MCP for beginners course](#)