References:

https://learn.microsoft.com/en-us/microsoft-copilot-studio/publication-add-bot-to-microsoft-teams

https://www.microsoft.com/insidetrack/blog/how-our-employees-are-extending-enterprise-ai-with-custom-retrieval-agents/
https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/copilot-studio-experience

Question 1

Domain: Design AI-powered business solutions
Scenario: A development team is building a complex, multi-step agent using the Microsoft Agent Framework. This agent needs to receive a high-level user request, break it down into the required sequence of internal API calls and knowledge base, and manage the execution flow of these steps to achieve the final outcome.

Within the Agentic Core, which specific sub-component is primarily responsible for analyzing the user's intent, determining the optimal sequence of required tools/APIs and data sources (RAG), and managing the logical execution flow of these steps?

A.The Safety System

B.The Model Context Protocol (MCP) Interface

C.The Large Language Model (LLM)

D.The Planner and Orchestrator

right

Explanation:

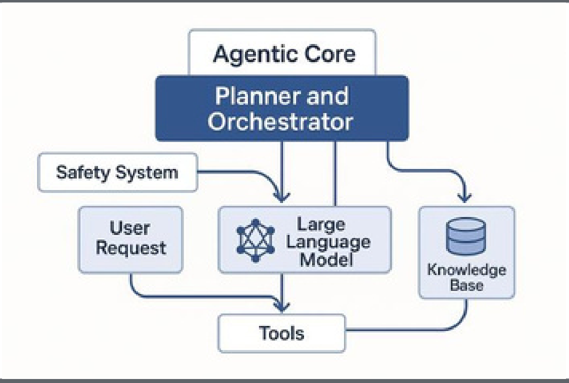
Correct Answer: D

Option D: The Planner and Orchestrator is correct because this component is the "brain" of the Agentic Core. Its function is to take the user prompt, use the LLM to assist in decomposition, build a logical plan (Planner) of steps to solve the request, and then manage the execution of those steps using the available Tools and data sources (Orchestrator).

Option A: The Safety System is incorrect because the Safety System's primary role is to enforce security, compliance, and responsible AI policies by monitoring the inputs and outputs, not to decompose the task or manage the operational sequence.

Option B: The Model Context Protocol (MCP) Interface is incorrect because the MCP is the standardized protocol used for communication between the Agentic Core and the Tools/APIs; it is not the component responsible for creating or managing the execution plan.

Option C: The Large Language Model (LLM) is incorrect because while the LLM provides the reasoning and natural language capabilities, the LLM itself is a resource utilized by the Planner to generate the plan; the Planner/Orchestrator is the component that handles the management and execution of the plan.



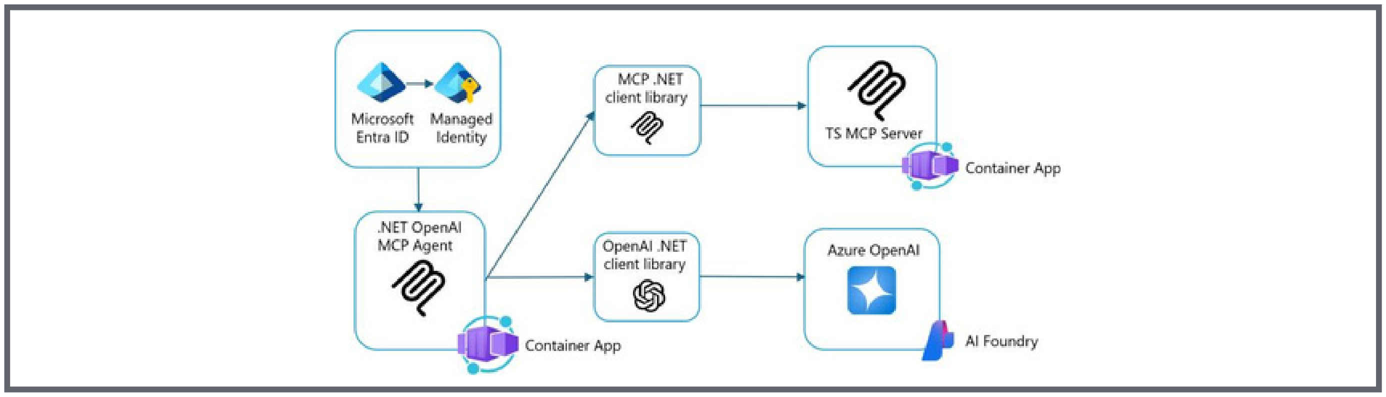
References:

- https://learn.microsoft.com/en-us/agent-framework/media/agent.svg
- https://techcommunity.microsoft.com/blog/educatordeveloperblog/ai-agents-planning-and-orchestration-with-the-planning-design-pattern---part-7/4399204
- https://www.microsoft.com/en-us/microsoft-365/planner/microsoft-planner

Question 2 Unattempted

Domain: Design AI-powered business solutions

An AI Agent needs to securely access Azure resources and external services (like an MCP Server) within a Microsoft-centric agentic solution. The diagram illustrates a key mechanism for managing the agent's identity and permissions.



Note: Drag and drop the following components into the correct sequential order to represent the authentication flow for the .NET OpenAI MCP Agent accessing secured resources, as depicted in the diagram

Correct Answer

1. C. Microsoft Entra ID
2. A. Managed Identity
3. B. .NET OpenAI MCP Agent

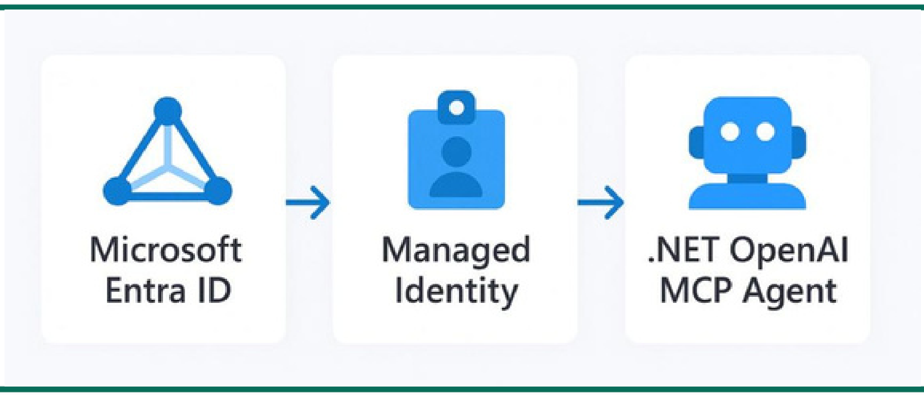
Explanation:

Correct Answers: C, A and B

Step 1 → C. Microsoft Entra ID

Step 2 → A. Managed Identity

Step 3 → B .NET OpenAI MCP Agent



Step 1: Microsoft Entra ID: This is the foundational cloud-based identity and access management service that centrally manages user identities, groups, and permissions. For Azure resources, it serves as the ultimate authority for issuing and validating identities. In the context of Managed Identities, Microsoft Entra ID is where the identity itself is registered and managed.

Step 2: Managed Identity: A Managed Identity is an identity registered with Microsoft Entra ID that Azure resources (like the Container App hosting the .NET OpenAI MCP Agent) can use. It eliminates the need for developers to manage credentials directly. Microsoft Entra ID provisions and manages this identity. The Agent's hosting environment then uses this Managed Identity to authenticate to other Azure services (like Azure OpenAI) or external services that trust Entra ID.

Step 3: .NET OpenAI MCP Agent: The .NET OpenAI MCP Agent, deployed within an Azure Container App, is configured to use the Managed Identity assigned to its hosting environment. This allows the Agent to automatically obtain access tokens from Microsoft Entra ID (via the Managed Identity) and present them when making calls to secured services, ensuring that its requests are authenticated and authorized without hardcoding secrets.

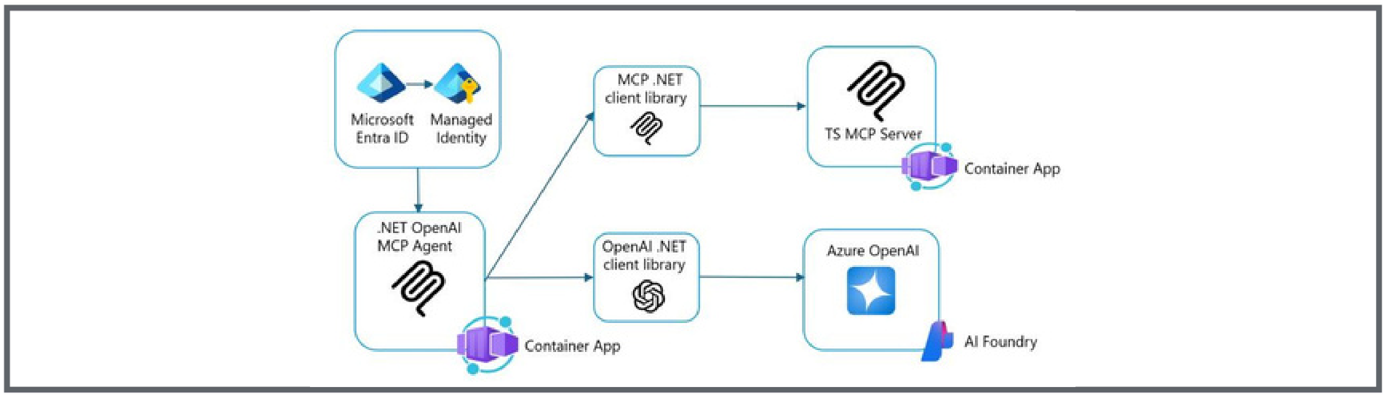
References:

- https://learn.microsoft.com/en-us/azure/developer/ai/intro-agents-mcp
- https://learn.microsoft.com/en-us/azure/developer/ai/build-mcp-server-ts?tabs=github-codespaces
- https://learn.microsoft.com/en-us/azure/developer/ai/build-openai-mcp-server-dotnet?tabs=github-codespaces

Question 3

Domain: Design AI-powered business solutions

The .NET OpenAI MCP Agent needs to interact with a custom business service exposed via a "TS MCP Server" (Tool Service Model Context Protocol Server). The diagram shows the communication path.



Drag and drop the following components into the correct sequential order, starting from the .NET OpenAI MCP Agent and ending with the TS MCP Server, to illustrate the communication path for invoking a tool/service.

Correct Answer

1. B..NET OpenAI MCP Agent
2. A. MCP .NET client library
3. C.TS MCP Server

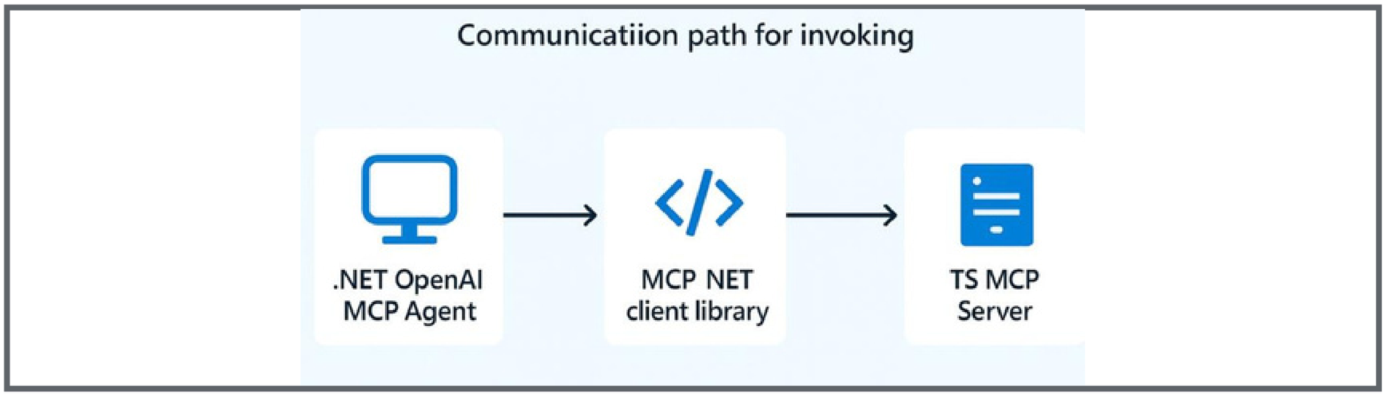
Explanation:

Correct Answers : B, A and C

Step 1 → B .NET OpenAI MCP Agent

Step 2 → A. MCP .NET client library

Step 3 → C. TS MCP Server



Step 1: .NET OpenAI MCP Agent: This is the initiating component. When the Agent's Planner (its "brain") determines that a specific task requires calling an external tool or service, it prepares the necessary arguments and initiates the call.

Step 2: MCP .NET client library: The .NET OpenAI MCP Agent does not directly call the raw API endpoint. Instead, it utilizes an MCP .NET client library. This library serves as an abstraction layer, handling the complexities of the Model Context Protocol (MCP) by translating the agent's high-level request into the standardized MCP format for communication with the tool service. This ensures consistent interaction regardless of the tool's underlying implementation.

Step 3: TS MCP Server: The MCP .NET client library then sends the MCP-formatted request to the TS MCP Server (Tool Service Model Context Protocol Server). This server, which exposes the custom business service capabilities, receives the request, processes it, performs the desired action (e.g., retrieving data, updating records), and then sends a response back through the MCP client library to the agent.

References:

- https://learn.microsoft.com/en-us/azure/developer/ai/intro-agents-mcp
- https://learn.microsoft.com/en-us/azure/developer/ai/build-mcp-server-ts?tabs=github-codespaces
- https://learn.microsoft.com/en-us/azure/developer/ai/build-openai-mcp-server-dotnet?tabs=github-codespaces

Question 11 Unattempted

Domain: Deploy AI-powered business solutions

An organization wants to deploy a new tool that their AI agent can use. This tool, an "MCP Tool Service," is implemented using Azure Functions to leverage serverless capabilities and serve as a remote MCP server, as described by Microsoft's guidance.

Note: Drag and drop the following components into the correct sequential order to represent the architectural flow, from a user's prompt to the execution of a tool service hosted on Azure Functions as a remote MCP server, as per Microsoft's recommendations.

Correct Answer

1. D. User Prompt
2. B. Agent (e.g., .NET OpenAI MCP Agent)
3. C. Model Context Protocol (MCP) Definition
4. A. Azure Function App (hosting MCP Server)

Explanation:

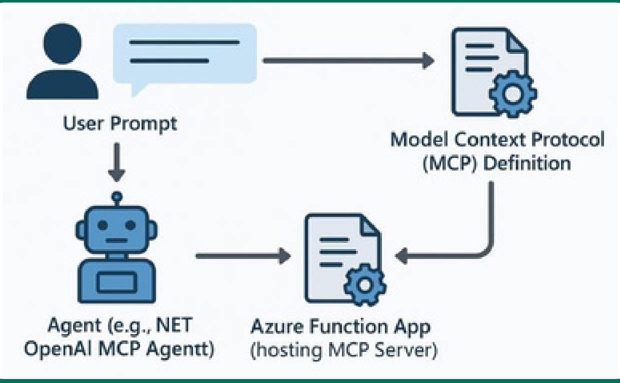
Correct Answers : D, B, C and A

Step 1 - D. User Prompt

Step 2 - B. Agent (e.g., .NET OpenAI MCP Agent)

Step 3 - C. Model Context Protocol (MCP) Definition

Step 4 - A. Azure Function App (hosting MCP Server)


User Prompt
Model Context Protocol (MCP) Definition
Agent (e.g., NET OpenAI MCP Agentt)
Azure Function App (hosting MCP Server)

Step 1: User Prompt: The entire agentic workflow begins when an end-user provides a natural language prompt or request to the AI agent, specifying a task they want to accomplish.

Step 2: Agent (e.g., .NET OpenAI MCP Agent): The Agent receives the user's prompt. Its internal components, particularly the Planner and Orchestrator (often working with an LLM), analyze the intent of the prompt and determine if any external tools are required to fulfill the request.

Step 3: Model Context Protocol (MCP) Definition: Once the Agent identifies that a tool is needed, it consults the MCP Definition for that specific tool. This definition (often registered in an Agent Manifest or a tool registry) provides the agent with a standardized, machine-readable description of the tool's capabilities, its input parameters, and how to invoke it via the Model Context Protocol. This is crucial for the agent to know what the tool can do and how to call it.

Step 4: Azure Function App (hosting MCP Server): Based on the MCP Definition, the Agent constructs an MCP-compliant request and sends it. The Azure Function App is where the actual MCP Server is hosted. It receives this request, processes it by executing the underlying business logic of the tool (e.g., checking inventory, updating a database), and then returns an MCP-compliant response back to the Agent, allowing the Agent to complete the user's request.

Reference:

- https://techcommunity.microsoft.com/blog/appsonazureblog/build-ai-agent-tools-using-remote-mcp-with-azure-functions/4401059

Question 4 Unattempted

Domain: Deploy AI-powered business solutions
Your organization is planning to deploy a suite of AI-powered agents and applications across multiple departments. You want to ensure the applications follow a strong ALM (Application Lifecycle Management) strategy to maintain quality, traceability, and governance throughout development, testing, deployment, and updates.

According to Microsoft's best practices for ALM in Dynamics 365 and Power Platform applications, which FOUR of the following practices should be adopted as part of a robust ALM strategy? (Select 4)

Note: Drag the correct answer and drop into the corresponding answer area.

Correct Answer

A.Use version control and branching strategies in a system like Azure DevOps to manage code and solution changes throughout development and release cycles

C.Use managed solutions for non-development environments (e.g., Test, Production) and unmanaged solutions only in development environments

D.Configure release pipelines and build automation so that deployable packages are tested in non-production environments before being marked as release candidates for production

E.Use multiple development environments in parallel to support the current production version (for patches) while developing the next major version

Explanation:

Correct Answers: A, C, D and E

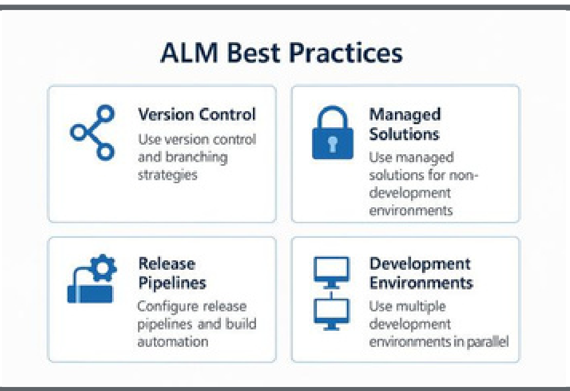
Option A: Use version control and branching strategies in a system like Azure DevOps to manage codeand solutionchangesthroughoutdevelopmentand releasecycles iscorrectbecause - Version control (like Git in Azure DevOps) is fundamental to ALM. It provides traceability, enables collaborative development, supports rollback capabilities, and is essential for managing changes across different development and release cycles.

Option C: Use managed solutions for non-development environments (e.g., Test, Production) and unmanagedsolutionsonlyindevelopmentenvironmentsiscorrectbecause this is a critical best practice in Power Platform ALM. Unmanaged solutions provide flexibility for active development, allowing direct component editing. Managed solutions are deployed to non-development environments (like Test, Staging, and Production) to ensure governance, controlled deployment, and proper cleanup/uninstallation of solution components.

Option D: Configure release pipelines and build automation so that deployable packages are tested in non-production environments before being marked as release candidates for production is correctbecause automation through CI/CD (Continuous Integration/Continuous Delivery) pipelines in tools like Azure DevOps is key to a robust ALM strategy. It ensures consistent builds, automates testing, reduces manual errors, and provides a reliable process for promoting validated solution packages through environments.

Option E: Use multiple development environments in parallel to support the current production version (for patches) while developing the next major version is correct because this parallel environment strategy (e.g., one environment/branch for hotfixes for the current production version and another for developing the next major feature release) is crucial for agility and stability. It allows critical patches to be developed and deployed quickly without being impacted by potentially unstable new features under development, thus minimizing disruption to the live system.

Option B: Deploy all customizations directly to the production environment to speed up innovation andavoid re-testing insandboxenvironments isincorrectbecause this practice violates core ALM principles. Deploying directly to production without proper testing in lower environments (e.g., dev, test, UAT) significantly increases the risk of introducing errors, causing system instability, and impacting business operations. All changes should follow a controlled, validated path.


ALM Best Practices
Version Control
Use version control and branching strategies
Managed Solutions
Use managed solutions for non-development environments
Release Pipelines
Configure release pipelines and build automation
Development Environments
Use multiple development environments in parallel

References:

- https://learn.microsoft.com/en-us/dynamics365/guidance/implementation-guide/application-lifecycle-management-product
- https://learn.microsoft.com/en-us/power-platform/alm/overview-alm
- https://learn.microsoft.com/en-us/power-platform/alm/devops-build-tools

Question 5 Unattempted
Domain: Deploy AI-powered business solutions
Your company is onboarding employees to use Microsoft Copilot effectively for everyday productivity tasks. Many users are unsure which type of prompt to use for different activities such as creating new content, catching up on work, asking questions, or editing existing files.

You decide to test their understanding by providing a matching exercise that aligns each prompt type with a real-world example. You need to match each Copilot prompt type with the correct example or use cases.

Note: This is a Dra-Drop Matching question and for this you need to match prompt types with the correct examples by drag and drop the appropriate service into the corresponding answer area.

Correct Answers

A. Catch up

“What questions were asked during the meeting?”

B. Create

“Create a short presentation about time management”

C. Ask

“Give me ideas for a team building activity”

D. Edit

“Add an image of a target with arrows to this slide”

Explanation:

Correct Answers : 1-B, 2-C, 3-D and 4-A

Step 1 → Option B (Catch up → “What questions were asked during the meeting?”)

"Catch up" prompts are designed to quickly bring users up to speed on information they might have missed from meetings, emails, chats, or documents. For example, Copilot in applications like Teams or Outlook can efficiently extract key information, such as questions asked or decisions made, enabling users to get a concise overview without needing to review all content manually. This example directly asks for specific information from a past event, fitting the "Catch up" category perfectly.

Step 2 → Option C (Create → “Create a short presentation about time management.”)

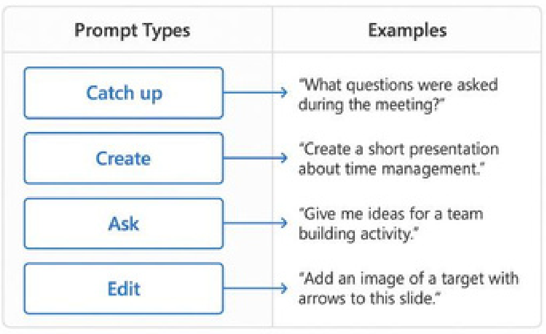
"Create" prompts instruct Copilot to generate entirely new content from scratch. This can include various formats such as documents, emails, presentations, project plans, or outlines, based on the user's natural-language instructions. The prompt "Create a short presentation about time management" clearly requests the generation of a new PowerPoint presentation, directly aligning with the "Create" prompt type.

Step 3 → Option D (Ask → “Give me ideas for a team building activity.”)

"Ask" prompts are open-ended questions aimed at seeking knowledge, suggestions, creative ideas, or general guidance. Unlike "Catch up" (which summarizes existing content) or "Edit" (which modifies existing content), "Ask" prompts are for exploratory purposes and decision support. The request "Give me ideas for a team building activity" is a classic example of asking Copilot for creative suggestions and brainstorming, making it an ideal match for the "Ask" prompt type.

Step 4 → Option A (Edit → “Add an image of a target with arrows to this slide.”)

"Edit" prompts are used to modify existing content. These modifications can range from rewriting text, improving clarity, adjusting tone, restructuring paragraphs, to inserting new elements like images or design features into an existing file. The instruction "Add an image of a target with arrows to this slide" explicitly directs Copilot to make a change to an already existing PowerPoint slide, which is a clear editing action.



Prompt Types	Examples
Catch up	“What questions were asked during the meeting?”
Create	“Create a short presentation about time management.”
Ask	“Give me ideas for a team building activity.”
Edit	“Add an image of a target with arrows to this slide.”

References: