



-Expert Verified, Online, **Free**.



CERTIFICATION TEST

-CertificationTest.net - Cheap & Quality Resources With Best Support

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -**Channels -**

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.
Knowledge sources must be used for unstructured content.
Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.
Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.

Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.

Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.

User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.

Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to configure the agent in Copilot Studio to meet Blue Vender's Responsible AI and data protection requirements.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Responsible AI and data protection requirements

Add custom prompt modifications.

Conduct adversarial testing.

Define Data Loss Prevention (DLP) policies.

Integrate with Microsoft Purview.

Disable generative answers.

Configure content moderation filters.

Answer Area

1

2

3

Suggested Answer:

Answer Area

1

Define Data Loss Prevention (DLP) policies.

2

Add custom prompt modifications.

3

Conduct adversarial testing.

 **Hatsapatsa** Highly Voted 1 month, 2 weeks ago

I think 3 should be Configure content moderation filters as stated in the case study.
upvoted 6 times

 **60ed5c2** Most Recent 1 month, 2 weeks ago

I agree the question specifically talks to configuration, testing would come later.
upvoted 1 times

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -**Channels -**

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.
Knowledge sources must be used for unstructured content.
Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.
Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.

Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.

Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.

User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.

Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to evaluate whether the current configuration decisions for the Blue Yonder Copilot agent comply with the company's security and governance policies.

Which compliance status should you assign to each configuration decision? To answer, move the appropriate compliance statuses to the correct configuration decisions. You may use each compliance status once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Compliance statuses	Compliance configuration	
Compliant	Configuration decision	Compliance status
	The agent allows anonymous access general inquiries such as baggage policies.	
Non-compliant	The agent uses an open web search to answer questions beyond internal knowledge.	
	The team relies only on default content filters without custom RAI instructions.	
	The agent uses one shared admin credential for all internal data access.	

Suggested Answer:

Compliance configuration	
Configuration decision	Compliance status
The agent allows anonymous access general inquiries such as baggage policies.	Compliant
The agent uses an open web search to answer questions beyond internal knowledge.	Non-compliant
The team relies only on default content filters without custom RAI instructions.	Non-compliant
The agent uses one shared admin credential for all internal data access.	Non-compliant

👤 **lucas_0102** Most Recent 1 month, 2 weeks ago

I'm struggling with this—why is this the correct answer?

upvoted 1 times

Case Study -

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -

Channels -

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.

Knowledge sources must be used for unstructured content.

Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.

Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.

Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.

Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.

User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.

Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to deploy the Blue Yonder Copilot agent to the public website and Microsoft Teams while ensuring compliance with the company's security and Responsible AI requirements.

Which two actions should you perform before making the agent available on both channels? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Configure prompt modifications to enforce tone, disclaimers, and refusal behavior at the system level.
- B. Manually add disclaimers to each topic before publishing.
- C. Embed the web channel and then rely on channel-level settings to enforce content moderation.
- D. Configure Power Platform DLP policies to restrict unauthorized data connectors.
- E. Publish the agent and then enable Responsible AI filters individually for each channel.

Suggested Answer: A, D

Community vote distribution

AD (100%)

🗲️ **moodin** Most Recent 3 weeks ago

Selected Answer: AD

Brief states "Power Platform DLP policies must be enforced to block unauthorized data flows." Additionally, E is configuring per-channel rather than system level, bad practice.

upvoted 1 times

🗲️ **690471c** 1 month, 2 weeks ago

Selected Answer: AD

las políticas DLP de Power Platform permiten bloquear conectores, canales o fuentes de datos no autorizados. Es una medida esencial antes de publicar un agente conectado a sistemas empresariales.

upvoted 2 times

Case Study -

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -

Channels -

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.

Knowledge sources must be used for unstructured content.

Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.

Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.

Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.

Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.

User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.

Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to configure the Blue Yonder Copilot agent's responses in accordance with the company's content control and platform requirements.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Add required disclaimer text inside each individual topic.
- B. Configure prompt instructions that include disclaimer text.
- C. Use Markdown syntax within response content.
- D. Insert HTML formatting directly into topic responses.
- E. Duplicate disclaimer text across reusable topics.

Suggested Answer: B, C

Community vote distribution

BC (100%)

 moodin Most Recent 3 weeks ago

Selected Answer: BC

"Disclaimers must be applied consistently across all generative responses" for B, and "Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported" for C

upvoted 2 times

Case Study -

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -

Channels -

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.

Knowledge sources must be used for unstructured content.

Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.

Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.

Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.

Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.

User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.

Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to configure the agent in Copilot Studio to use internal and external partner knowledge sources to answer user questions about the airline services.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Use a Microsoft Graph connector to index the partner's travel advisory content.
- B. Write individual Q&A pairs for each document as separate topics.
- C. Enable unrestricted web search for the agent.
- D. Add the internal policy documents as a knowledge source.

Suggested Answer: A, D

Community vote distribution

AD (100%)

 **faisalronnie** Most Recent 3 weeks ago

Selected Answer: AD

Travel Advisory Content: Uses REST API with partner services.> if this is true then A cannot be the option
upvoted 1 times

Case Study -

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -

Channels -

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.

Knowledge sources must be used for unstructured content.

Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.

Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.

Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.

Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.

User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.

Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to ensure that every AI-generated response from the agent in Copilot Studio includes a disclaimer that complies with the company's security and governance policies.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Add a greeting message that includes the disclaimer.
- B. Add a prompt modification in the generative answers node settings.
- C. Create a disclaimer topic that always runs first.
- D. Disable generative answers and use only pre-authored responses.
- E. Edit every topic's first message to include the disclaimer.

Suggested Answer: A, B

Community vote distribution

BC (100%)

🗉 👤 **Hatsapatsa** Highly Voted 👍 1 month, 2 weeks ago

Selected Answer: BC

I would say 'B and C' because A only runs once, case study requires it to be added to each generative response and therefore a reusable separate topic would be the better option.

upvoted 5 times

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -**Channels -**

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.
Knowledge sources must be used for unstructured content.
Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.
Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.
Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.
Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.
User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.
Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.

You need to determine which authentication model should be applied to each Blue Yonder Copilot interaction scenario to comply with the company's security and governance requirements.

Which authentication requirement should you apply to each scenario? To answer, move the appropriate authentication requirements to the correct user scenarios. You may use each authentication requirement once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Authentication requirements	Authentication requirements	Authentication requirements
Allow anonymous access.	User scenario	Authentication requirement
Require customer sign-in using an account.	A customer asks, "What is the baggage allowance for international flights?"	
Require internal agent sign-in using Microsoft Entra ID.	A customer asks, "Can I review my upcoming trips?"	
Use agent author credentials for authenticated data calls.	An internal support agent accesses the Copilot agent from Microsoft Teams to look up a loyalty member's profile.	

Suggested Answer:

Authentication requirements	Authentication requirement
User scenario	
A customer asks, "What is the baggage allowance for international flights?"	Allow anonymous access.
A customer asks, "Can I review my upcoming trips?"	Require customer sign-in using an account.
An internal support agent accesses the Copilot agent from Microsoft Teams to look up a loyalty member's profile.	Require internal agent sign-in using Microsoft Entra ID.

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -

A company has an existing custom connector that is approved and available in the environment. A builder wants an agent in Copilot Studio to call the connector during a conversation to retrieve information from an internal system.

To meet the business needs, the solution must meet the following requirements:

The agent must make the connector available for topic steps.

The agent must run the connector call with a valid connection.

The connector call must receive the required input values at runtime.

You need to configure the agent so the custom connector can be used as a tool.

What should you configure for each requirement? To answer, move the appropriate configurations to the correct requirements. You may use each configuration once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Configurations

- Map topic variables to the tool inputs.
- Add the connector as a tool in the agent.
- Add the connector as a knowledge source.
- Create or reuse a connection for the connector.

Custom connectors

Requirement

The agent must make the connector available for topic steps.

The agent must run the connector call with a valid connection.

The connector call must receive the required input values at runtime.

Configuration

Suggested Answer:

Custom connectors

Requirement

The agent must make the connector available for topic steps.

The agent must run the connector call with a valid connection.

The connector call must receive the required input values at runtime.

Configuration

- Add the connector as a tool in the agent.
- Create or reuse a connection for the connector.
- Map topic variables to the tool inputs.

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -

A company needs to ground an agent in Copilot Studio answers using live enterprise data from a supported system via a Microsoft Power Platform connector.

To meet the business needs, the builder must meet the following requirements:

Authenticate using a valid connection.

Add the connector as a real-time knowledge source.

Select the tables for knowledge grounding.

You need to add a Power Platform connector as a real-time knowledge source for the agent.

In which order should you perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

REST API

Select Sign in and create a new connection or reuse an existing connection.

Select the tables to include as the knowledge source.

Select Add knowledge from the agent Knowledge experience.

Select the real-time connector from the Add knowledge dialog.

Select Add to agent to complete the connection.

Answer Area

1

2

3

4

5

Suggested Answer:

Answer Area

1

Select Add knowledge from the agent Knowledge experience.

2

Select the real-time connector from the Add knowledge dialog.

3

Select Sign in and create a new connection or reuse an existing connection.

4

Select the tables to include as the knowledge source.

5

Select Add to agent to complete the connection.

Currently there are no comments in this discussion, be the first to comment!

HOTSPOT -

You deploy an agent in Microsoft Copilot Studio that triggers a cloud flow to update customer records in Microsoft Dataverse.

You observe the following after deploying the agent:

Several runs fail due to a temporary authentication issue.

The flow continues to trigger and generate additional failed runs.

You must:

Stop additional failed executions while troubleshooting.

Process a failed execution after resolving the authentication issue.

Validate if the most recent executions succeeded immediately after resolution.

You need to use the appropriate monitoring and management actions to restore normal flow operation and validate successful execution.

Which action should you perform for each requirement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Flow monitoring actions**Requirement**

Stop additional failed executions while troubleshooting.

Process a failed execution after resolving the authentication issue.

Validate if the most recent executions succeeded immediately after resolution.

Action

Delete the flow.
Disable the flow.
Cancel the most recent run.
Modify the trigger condition.

Restart the flow.
Refresh the flow.
Resubmit the run.
Recreate the trigger.

Republish the flow.
Review the Activity history.
Export the flow to a solution.
Inspect the Analytics summary.

Suggested Answer:

Flow monitoring actions

Requirement

Stop additional failed executions while troubleshooting.

Action

Delete the flow.
Disable the flow.
Cancel the most recent run.
Modify the trigger condition.

Process a failed execution after resolving the authentication issue.

Restart the flow.
Refresh the flow.
Resubmit the run.
Recreate the trigger.

Validate if the most recent executions succeeded immediately after resolution.

Republish the flow.
Review the Activity history.
Export the flow to a solution.
Inspect the Analytics summary.

🗂️ 👤 **lucas_0102** Most Recent 1 month, 2 weeks ago

Could someone help me understand the logic behind this answer?

upvoted 1 times

DRAG DROP -

A company is preparing an agent flow so that it can be invoked by an agent during conversations.

The agent flow must meet the following requirements:

The agent must be able to trigger the flow.

The agent flow must be verified.

You need to prepare an agent flow so that it can be used by the agent.

In which order should you perform the actions to prepare the agent flow? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Agent flow readiness steps

Configure the invocation trigger that allows the agent to call the agent flow.

Make the agent flow available for agent usage.

Validate the agent flow behavior with representative test runs.

Configure the actions that implement the required behavior.

Create a new agent flow artifact in the environment.

Answer Area

1

2

3

4

5

Suggested Answer:

Answer Area

1

Create a new agent flow artifact in the environment.

2

Configure the invocation trigger that allows the agent to call the agent flow.

3

Configure the actions that implement the required behavior.

4

Validate the agent flow behavior with representative test runs.

5

Make the agent flow available for agent usage.

Currently there are no comments in this discussion, be the first to comment!

HOTSPOT -

A company uses an agent flow that occasionally requires human input before continuing execution.

Some automated actions must pause until a human provides a decision or additional information. The flow must be configured to:

Capture a human response for use in later steps.

Continue processing within the same flow run after the response is submitted.

Wait for a manual decision before proceeding.

You need to configure a human-in-the-loop agent flow.

Which setting should you configure for each requirement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Human-in-the-loop flow configuration**Requirement**

Capture a human response for use in downstream logic.

Continue processing within the same flow run after a response is submitted.

Wait for a manual decision before proceeding.

Configuration

Send a notification to a user.

Use the action's response outputs in subsequent steps.

Enable analytics logging for the response.

Republish the flow after the decision is submitted.

Trigger a new flow run when a response is received.

Allow the action to wait for and return a response before continuing.

Add a human approval action.

Add a manual trigger to the flow.

Send a message that asks for input.

Suggested Answer:**Human-in-the-loop flow configuration****Requirement**

Capture a human response for use in downstream logic.

Continue processing within the same flow run after a response is submitted.

Wait for a manual decision before proceeding.

Configuration

Send a notification to a user.

Use the action's response outputs in subsequent steps.

Enable analytics logging for the response.

Republish the flow after the decision is submitted.

Trigger a new flow run when a response is received.

Allow the action to wait for and return a response before continuing.

Add a human approval action.

Add a manual trigger to the flow.

Send a message that asks for input.

Currently there are no comments in this discussion, be the first to comment!

HOTSPOT -

A company uses an agent that invokes an agent flow to exchange information during a conversation.

The company requires that the agent send data into the flow and receive structured results back from the same flow run. To support this business need, the flow must be configured to do the following:

Capture the data provided by the agent.

Return data results to the agent.

You need to configure the flow so that it can exchange data with the agent.

What should you configure for each requirement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Agent flow input and output configuration**Requirement****Configuration**

Capture the data provided by the agent.

Use an internal variable.
Add a text input parameter.
Store values only in the topic.

Return data results to the agent.

Log values to run history.
Store values only in the flow.
Add a text output parameter.

Suggested Answer:**Agent flow input and output configuration****Requirement****Configuration**

Capture the data provided by the agent.

Use an internal variable.
Add a text input parameter.
Store values only in the topic.

Return data results to the agent.

Log values to run history.
Store values only in the flow.
Add a text output parameter.

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -

You are configuring an agent in Copilot Studio for an organization. The organization uses Microsoft Dataverse, Dynamics 365, and SharePoint Online.

The agent must retrieve information from enterprise systems and internal documents. The agent must meet the following requirements:

Must retrieve the current balance of a customer account stored in Dataverse.

Must retrieve the real-time shipping status of an order that updates throughout the day in Dynamics 365.

Must answer questions based on the content of a static PDF policy document.

Must provide a welcome message that does NOT depend on enterprise data.

You need to configure the agent to meet the requirements.

How should the agent retrieve the data? To answer, move the appropriate solutions to the correct requirements. You may use each solution once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Solutions

- Create a manual response in the topic.
- Connect to a Power Platform connector.
- Upload a file to Agents Knowledge Base.
- Configure a generative answer without enterprise data.

Enterprise data retrieval methods

Requirement

Retrieve the current balance of a customer account stored in Dataverse.

Retrieve the real-time shipping status of an order that updates throughout the day in Dynamics 365.

Answer questions based on the content of a static PDF policy document.

Provide a welcome message that does **NOT** depend on enterprise data.

Solution

Suggested Answer:

Enterprise data retrieval methods

Requirement

Retrieve the current balance of a customer account stored in Dataverse.

Retrieve the real-time shipping status of an order that updates throughout the day in Dynamics 365.

Answer questions based on the content of a static PDF policy document.

Provide a welcome message that does **NOT** depend on enterprise data.

Solution

- Connect to a Power Platform connector.
- Connect to a Power Platform connector.
- Upload a file to Agents Knowledge Base.
- Create a manual response in the topic.

Currently there are no comments in this discussion, be the first to comment!

An agent calls a flow. The agent requires structured output values to be returned.

The agent receives unexpected or empty values.

You need to configure the agent so that data is exchanged correctly between the agent and the flow.

What should you do?

- A. Validate parameter definitions.
- B. Change the trigger schema.
- C. Review the flow run history.
- D. Add a Parse JSON inside the flow.
- E. Republish the agent.

Suggested Answer: A

 **faisalronnie** Most Recent 3 weeks ago

A. Validate parameter definitions ☐

The problem is about data being exchanged between the agent and the flow. If the agent receives empty or unexpected values, the first thing to check is that:

Agent inputs match flow inputs Flow outputs are properly defined Parameter names and data types match Returned values are mapped correctly
upvoted 2 times

 **lucas_0102** 1 month, 2 weeks ago

Why is this answer justified? Can someone elaborate?

upvoted 1 times

A company is building an agent in Copilot Studio.

The agent must meet the following requirements:

Responses must be grounded in approved internal content.

The agent must be able to retrieve information from configured data sources in a topic.

You need to configure the agent so that topic responses are grounded from custom data sources.

What should you configure?

- A. Enable analytics to review which topics users trigger most often.
- B. Add a custom prompt that tells the agent to answer only from company data.
- C. Configure knowledge sources in a generative answers node.
- D. Add SharePoint as a knowledge source only at the agent level.
- E. Increase the response length limit to improve answer completeness.

Suggested Answer: C

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -

A company is building an agent in Copilot Studio topic that must retrieve real-time status information from an external REST API using a Send HTTP request node.

To meet the business needs, the topic must meet the following requirements:

The request must include the required authentication header.

The request must call the endpoint using the correct HTTP method.

The response must be configured with an appropriate response data type based on a schema and saved so the topic can reuse the returned values.

You need to configure the Send HTTP request node.

Which configuration should you use for each requirement? To answer, move the appropriate configurations to the correct requirements. You may use each configuration once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Configurations

Add required headers to the HTTP request.

Set the method to GET in the HTTP request.

Select JSON data type and save response to a variable.

Set the method to POST and send JSON in a request body.

HTTP Request node configuration**Requirement**

Configure authentication for the request.

Select the correct HTTP method for retrieving data.

Configure structured data handling and storage of data for reuse in the topic.

Configuration

Suggested Answer:**HTTP Request node configuration****Requirement**

Configure authentication for the request.

Select the correct HTTP method for retrieving data.

Configure structured data handling and storage of data for reuse in the topic.

Configuration

Add required headers to the HTTP request.

Set the method to GET in the HTTP request.

Select JSON data type and save response to a variable.

Currently there are no comments in this discussion, be the first to comment!

A company is deploying an agent for employees.

The company has the following requirements:

Access must be restricted to the internal employees only.

End-user authentication must be enforced.

You need to configure the agent based on the company requirements.

What should you do?

- A. Select Authenticate with Microsoft Entra ID for the agent.
- B. Enable Require Secure Access in the Web channel security.
- C. Publish the agent as a public website experience.
- D. Select Authenticate with personal account for the agent.

Suggested Answer: A

  **lucas_0102** Most Recent 1 month, 2 weeks ago

Can someone explain why this answer is correct?

upvoted 1 times

DRAG DROP -

A company is integrating an agent in Copilot Studio into a custom application by using the Microsoft 365 Agents SDK.

The company has the following requirements for the application:

The deployment workflow must follow a dependency driven sequence.

The agent must be published before configuration values are retrieved.

End-to-end validation must occur before production deployment.

You need to plan the deployment workflow to integrate the agent.

In which order should you perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Deployment workflow steps

- Configure and enable the agent's target integration channel.
- Retrieve current agent configuration values.
- Deploy the web app to production.
- Implement the Microsoft 365 Agents SDK in the app using the agent configuration.
- Publish the agent.
- Validate the end-to-end conversation workflow in a test environment.

Answer Area

1	
2	
3	
4	
5	
6	

Suggested Answer:**Answer Area**

1	Publish the agent.
2	Retrieve current agent configuration values.
3	Configure and enable the agent's target integration channel.
4	Implement the Microsoft 365 Agents SDK in the app using the agent configuration.
5	Validate the end-to-end conversation workflow in a test environment.
6	Deploy the web app to production.

 **faisalronnie** Most Recent 1 month ago

Order Action 1 Publish the agent. 2 Configure and enable the agent's target integration channel. 3 Retrieve current agent configuration values. 4 Implement the Microsoft 365 Agents SDK in the app using the agent configuration. 5 Validate the end-to-end conversation workflow in a test environment. 6 Deploy the web app to production.

upvoted 4 times

 **lucas_0102** 1 month, 2 weeks ago

I'm not following the reasoning—can someone explain?

upvoted 1 times

An agent in Copilot Studio already exists in a development environment.

You need to prepare the agent so it can be moved to a production environment using the supported solution-based approach.

You need to create an exportable package that contains the existing agent.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Remove agent topics directly from the solution to prevent export errors.
- B. Convert the solution to a managed solution so it can be exported.
- C. Export the unmanaged solution that contains the agent.
- D. Publish the agent to a channel and export the channel configuration.
- E. Create a custom solution and add the existing agent to the solution.

Suggested Answer: C, E

Community vote distribution

CE (100%)

🗲️ 👤 **Joselito79** Most Recent 2 days ago

Selected Answer: CE

☐ C. Export the unmanaged solution that contains the agent. ☐ E. Create a custom solution and add the existing agent to the solution.

upvoted 1 times

🗲️ 👤 **faisalronnie** 1 month ago

Selected Answer: CE

The supported ALM process is:

Create/Open an unmanaged solution. Add Existing → Copilot Agent to the solution. Include dependencies (flows, connectors, environment variables, etc.). Export the unmanaged solution (or later export as managed for production deployment depending on the scenario).

upvoted 2 times

HOTSPOT -

A company wants to use content from a non-Microsoft enterprise knowledge base to ground agent responses.

To meet the business needs, the knowledge source must meet the following requirements:

Support grounding from indexed data.

Use existing access controls from the source system.

Support read-only operations.

You need to determine the correct behavior to meet the requirements.

Which behavior meets each requirement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Copilot connector knowledge behavior**Requirement**

Support grounding from indexed data.

Use existing access controls from the source system.

Support read-only operations.

Configuration

Retrieve indexed content
Execute runtime operations.
Retrieve live data at runtime.

Respect existing permissions.
Bypass permissions for faster retrieval.
Respect existing permissions from agent authors.

Prevent transactional updates.
Allow authenticated write operations.
Enable updates for enhanced performance.

Suggested Answer:**Copilot connector knowledge behavior****Requirement**

Support grounding from indexed data.

Use existing access controls from the source system.

Support read-only operations.

Configuration

Retrieve indexed content
Execute runtime operations.
Retrieve live data at runtime.

Respect existing permissions.
Bypass permissions for faster retrieval.
Respect existing permissions from agent authors.

Prevent transactional updates.
Allow authenticated write operations.
Enable updates for enhanced performance.

Currently there are no comments in this discussion, be the first to comment!

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.

An agent uses a flow that calls an external service which can occasionally fail or time out.

When a failure occurs, the agent must meet the following requirements:

Must not terminate silently.

Must send a notification containing the error details.

You need to configure the agent flow so that failures are handled in a controlled and predictable way.

Solution: Retry the failing action without handling failure.

Does the solution meet the goal?

A. Yes

B. No

Suggested Answer: B

Currently there are no comments in this discussion, be the first to comment!

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.

An agent uses a flow that calls an external service which can occasionally fail or time out.

When a failure occurs, the agent must meet the following requirements:

Must not terminate silently.

Must send a notification containing the error details.

You need to configure the agent flow so that failures are handled in a controlled and predictable way.

Solution: Configure run after conditions for failure paths.

Does the solution meet the goal?

A. Yes

B. No

Suggested Answer: A

Currently there are no comments in this discussion, be the first to comment!

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.

An agent uses a flow that calls an external service which can occasionally fail or time out.

When a failure occurs, the agent must meet the following requirements:

Must not terminate silently.

Must send a notification containing the error details.

You need to configure the agent flow so that failures are handled in a controlled and predictable way.

Solution: Send a notification with relevant error details.

Does the solution meet the goal?

A. Yes

B. No

Suggested Answer: A

Community vote distribution

A (100%)

 **faisalronnie** Most Recent 1 month ago

Selected Answer: A

Sending a notification with the error details explicitly alerts someone that the failure occurred, so the failure is not silent, and it also satisfies the requirement to send a notification containing the error details.

upvoted 3 times

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.

A company is configuring an agent in Copilot Studio that uses a generative answers node inside multiple topics.

The company requires responses to meet the following requirements:

Must use an executive summary format.

Must apply only within a specific topic.

Must continue using configured knowledge sources.

Must comply with connector security controls.

You need to configure a custom prompt to meet the formatting requirements.

Solution: Remove the generative answers node and rely only on the custom prompt to produce formatted responses.

Does the solution meet the goal?

A. Yes

B. No

Suggested Answer: B

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -**Channels -**

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.
Knowledge sources must be used for unstructured content.
Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.
Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.
Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.
Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.
User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.
Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.
You need to determine how the agent in Copilot Studio should integrate with two different systems in the Blue Yonder environment, based on their characteristics and the tools available.
Which integration mechanism should you use for each system? To answer, move the appropriate integrations to the correct system or data sources. You may use each integration once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.

Integrations

HTTP request

Custom connector

Dataverse connector

Generative AI knowledge

Copilot agent integration

System or data source	Integration
Internal Reservation	
Flight Status and Weather Service	

Suggested Answer:

Copilot agent integration

System or data source	Integration
Internal Reservation	Custom connector
Flight Status and Weather Service	HTTP request

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Blue Yonder Airlines is a global carrier headquartered in Los Angeles, California, operating domestic and international flights. The company serves millions of passengers annually through its website, mobile app, and call centers. To improve customer service efficiency and reduce call center volume, Blue Yonder is deploying an AI agent in Microsoft Copilot Studio.

The agent will handle customer inquiries across multiple channels – web chat, mobile app, and Microsoft Teams (for internal support staff). It will answer questions, retrieve data from enterprise systems, and escalate to human agents when needed.

The project is led by a cross-function team:

Product manager: Defines requirements and success metrics.

Lead agent author: Designs topics, intents, and generative behavior.

Flow designers: Build agent flows and integrations.

IT/security and compliance: Oversees identity, data protection, and Responsible AI (RAI) compliance.

Current environment -**Channels -**

Public website: Embedded web chat

Mobile app: In-app chatbot -

Microsoft Teams: Internal support agent access

Identity and access -

Customers: Anonymous access for general inquiries (e.g., flight status, baggage policy).

Authentication is required for personal data access (e.g., bookings, loyalty points).

Internal staff: Authenticate via Microsoft Entra ID.

Data sources -

Reservation and Ticketing System (internal): REST API, no prebuilt connector with custom enterprise database.

Flight Status and Weather APIs (external): REST APIs with API keys.

Customer Support Knowledge Base: SharePoint library with PDFs and policy documents.

Loyalty Program Data: Stored in Dynamics 365 and Dataverse.

Travel Advisory Content: Uses REST API with partner services.

Integration mechanisms -

Custom connectors must be used for internal APIs that lack prebuilt connectors.

HTTP request nodes may be used for lightweight external APIs.
Knowledge sources must be used for unstructured content.
Agent flows must be used to encapsulate reusable logic (e.g., rebooking).

Business requirements -

Omnichannel support -

Deploy the agent across web, mobile, and Teams with a consistent user experience. The Teams deployment must also support internal staff.

Self-service capabilities -

The agent must handle common inquiries such as:

Flight status -

Booking and rebooking -

Loyalty program questions -

Travel policies and baggage rules

Human escalation -

If the agent cannot resolve an issue or the user requests help, it must:

Escalate to a human agent.

Transfer the conversation transcript and relevant context.

Redact any sensitive personal data before escalation.

Knowledge integration -

The agent must use scalable methods for knowledge integration and must not rely on manually authored Q&A topics for each document.

Performance metrics -

First-contact resolution: +25%

Tier-1 call deflection: $\geq 20\%$

Response time: 90% of queries answered within 30 seconds

Accuracy: $\geq 95\%$ for known FAQs -

CSAT: $\geq 85\%$ for AI-handled interactions

Technical requirements -

Platform constraints -

No custom code is permitted; only Copilot Studio's built-in tools may be used.

All backend logic must be implemented using agent flows.

Markdown must be used for formatting (e.g., bold, bullet points); HTML is not supported.

Authentication -

Sign-in is required for personal data access.

Anonymous access is allowed for general inquiries.

User identity must be used for data access; shared or builder credentials must not be used.

Compliance and security -

Power Platform DLP policies must be enforced to block unauthorized data flows.

Responsible AI content moderation filters must be enabled.

Prompt modifications must be added to enforce tone, disclaimers, and refusal behavior.

Disclaimers must be applied consistently across all generative responses. Manual edits to individual topics must be avoided.

Monitoring and maintenance -

All conversations and actions must be logged for auditing.
Weekly reviews of transcripts and metrics must be conducted.

Issues and constraints -

API rate limits: External APIs (e.g., flight status) have usage limits. Agent flows must handle retries and caching to avoid exceeding quotas.
Knowledge base limits: Copilot Studio has limits on the number and size of indexed documents. Large files must be split or summarized.
Generative answer risks: Generative responses must be constrained to avoid policy violations. Prompt modifications and filters must be used to enforce tone, safety, and compliance.
User input variability: Users phrase questions in diverse ways. Topics must include varied trigger phrases and fallback handling.
Authentication UX: The agent must clearly explain when sign-in is required and handle transitions smoothly across channels.

Problem statement -

Blue Yonder Airlines must deploy a secure, scalable, and policy-compliant AI agent using Microsoft Copilot Studio. The agent must deliver accurate, helpful, and safe responses across multiple channels, integrate with enterprise systems, and support both anonymous and authenticated users. It must adhere to strict data protection and Responsible AI standards while improving customer service efficiency and satisfaction.
You need to determine an appropriate implementation related to the use of agent flows that is based on Blue Vender's design and compliance standards.
Which implementation method should you use for each scenario? To answer, move the appropriate implementation methods to the correct scenarios. You may use each implementation method once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.

Implementation methods

Use a tool node.

Use a generative answers node.

Use a Go to another topic node.

Prompt the user for manual input.

Manually rebuild the logic in the topic.

Agent flow implementations

Scenario

A topic includes a series of steps that replicate logic already defined in another flow.

A topic provides a natural language explanation of a process but does not execute it.

A topic includes a step that triggers a reusable process to update booking details.

Suggested Answer:

Agent flow implementations

Scenario

A topic includes a series of steps that replicate logic already defined in another flow.

A topic provides a natural language explanation of a process but does not execute it.

A topic includes a step that triggers a reusable process to update booking details.

Use a Go to another topic node.

Use a generative answers node.

Use a tool node.

faisalronnie

Most Recent

3 weeks ago

Scenario Implementation method A topic includes a series of steps that replicate logic already defined in another flow. Use a tool node. A topic provides a natural language explanation of a process but does not execute it. Use a generative answers node. A topic includes a step that triggers a reusable process to update booking details. Use a tool node.

upvoted 1 times

Case Study -

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Fabrikam Inc. is a Canada-based manufacturer with a growing service organization that supports field technicians and internal operations teams. Fabrikam Inc. plans to launch a new internal agent solution named Operations Concierge to reduce time spent searching policy content, retrieving operational metrics, and executing routine transactions.

The agent will be used by three groups:

- Service coordinators who triage incoming service requests

- Field technicians who need guided procedures and parts availability

- Operations managers who monitor KPIs and exceptions

The agent solution must work in real-world operational conditions. Users often ask questions mid-call with a customer or while coordinating parts shipments. The agents require quick, reliable outcomes. As a result, Fabrikam Inc. requires the solution to:

- Provide grounded answers with traceability when it provides guidance.

- Retrieve real-time metrics when users ask for operational status.

- Execute authenticated updates when users initiate a flow (such as creating a parts request).

Fabrikam Inc. also expects the solution to be maintained by multiple makers and developers across the year. The company has experienced duplicated logic and inconsistent behavior across different agents. This project emphasizes reuse, governance, and maintainability across teams.

Current environment -

Fabrikam Inc. runs three Microsoft Power Platform environments for agent development and release: Dev, Test, and Prod.

The team plans to build the agent and validate it in Dev and Test, then promote to Prod by using a controlled release process that supports repeatable deployments.

Fabrikam Inc. already has two assets the team wants to reuse:

- A partially completed Copilot Studio agent named Service Desk Agent, used by IT to create internal tickets and route requests

- A Microsoft Foundry agent created by a central AI team that performs specialized summarization and classification for long-form text (for example, summarizing call transcripts into an incident narrative)

Fabrikam Inc. also has operational and knowledge data sources:

- A curated policy library (internal SOPs, service warranty rules, escalation criteria, and standard operating procedures)

- A set of indexed documents and procedures in an Azure AI Search service that supports vector search for the policy library

- A Microsoft Fabric workspace that includes a semantic model used by operations leadership for reporting

Business requirements -

Fabrikam Inc. requires Operations Concierge to meet the following business requirements:

Traceability requirement: When the agent provides policy guidance or procedural recommendations, users must be able to see where the answer came from.

Metrics requirement: When users ask about service performance (backlog, SLA risk, parts shortages, dispatch delays), the solution must return up-to-date metrics in a structured format that operations managers can use in weekly reviews.

Transaction requirement: The solution must support authenticated updates initiated during conversations, including creating a parts request and

updating a service case status.

In addition, Fabrikam Inc. wants to avoid duplicating common assets across agents:

The team must reuse the same set of escalation topics, MCP tool definitions, and a standard safety disclaimer across three different agents.

Only the platform engineering group is allowed to edit shared assets. However, all agent authors must be able to use them.

Technical requirements -

The Fabrikam Inc. solution architecture uses a multi-agent approach so that specialist responsibilities are isolated and can evolve independently.

The Operations Concierge (primary agent) must coordinate the following specialist capabilities:

Policy and procedure Q&A: Use an enterprise knowledge source that supports indexed retrieval across the curated policy library and service procedures.

Operational metrics: Delegate metric queries to a Fabric Data Agent that reads governed business data through the Fabric semantic model.

Authenticated updates: Use tools exposed by an existing internal Model Context Protocol (MCP) server that provides transactional operations for the service organization.

Specialized processing: Delegate summarization and classification requests to an existing Microsoft Foundry agent.

Fabrikam Inc. will onboard two MCP servers as tools:

PartsOps MCP server: exposes tools for parts availability checks and parts request creation. The server requires per-user authentication because actions must be traceable to the requesting user.

WarrantyRules MCP server: exposes a read-only tool for validating warranty coverage. The server uses an API key shared by the agent team.

Fabrikam Inc. has also defined a collaboration requirement with the existing Service Desk Agent:

The primary agent must delegate IT-specific requests to the existing Service Desk Agent rather than reimplement ticket creation logic.

Finally, Fabrikam Inc. plans to support a partner integration:

For shipment tracking inquiries, Fabrikam Inc. will delegate to a partner-provided agent that is only available through a standardized agent-to-agent endpoint.

Issues and constraints -

During early testing, Fabrikam Inc. found three recurring problems:

Makers are copying and modifying the same components across agents, resulting in inconsistent disclaimers and duplicated tools.

Users can obtain a correct answer, but the response is not consistently traceable to a source when the agent uses knowledge.

The primary agent can route some requests, but specialist capabilities are not consistently delegated (for example, some metric questions are answered generatively instead of being routed to the Fabric Data Agent).

You are part of the engineering team responsible for correcting the design and configuration to meet the preceding requirements and constraints.

You need to enable Operations Concierge to delegate shipment tracking inquiries according to Fabrikam Inc.'s defined architecture and technical requirements.

Which integration approach should you use?

- A. Configure the partner capability as an enterprise knowledge source with indexed retrieval enabled.
- B. Configure the partner capability as an MCP tool and invoke it as a transactional operation.
- C. Configure the partner capability as an A2A integration and supply its endpoint information.
- D. Configure the primary agent to handle shipment tracking through generative responses after publishing.

Suggested Answer: C

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Fabrikam Inc. is a Canada-based manufacturer with a growing service organization that supports field technicians and internal operations teams. Fabrikam Inc. plans to launch a new internal agent solution named Operations Concierge to reduce time spent searching policy content, retrieving operational metrics, and executing routine transactions.

The agent will be used by three groups:

Service coordinators who triage incoming service requests

Field technicians who need guided procedures and parts availability

Operations managers who monitor KPIs and exceptions

The agent solution must work in real-world operational conditions. Users often ask questions mid-call with a customer or while coordinating parts shipments. The agents require quick, reliable outcomes. As a result, Fabrikam Inc. requires the solution to:

Provide grounded answers with traceability when it provides guidance.

Retrieve real-time metrics when users ask for operational status.

Execute authenticated updates when users initiate a flow (such as creating a parts request).

Fabrikam Inc. also expects the solution to be maintained by multiple makers and developers across the year. The company has experienced duplicated logic and inconsistent behavior across different agents. This project emphasizes reuse, governance, and maintainability across teams.

Current environment -

Fabrikam Inc. runs three Microsoft Power Platform environments for agent development and release: Dev, Test, and Prod.

The team plans to build the agent and validate it in Dev and Test, then promote to Prod by using a controlled release process that supports repeatable deployments.

Fabrikam Inc. already has two assets the team wants to reuse:

A partially completed Copilot Studio agent named Service Desk Agent, used by IT to create internal tickets and route requests

A Microsoft Foundry agent created by a central AI team that performs specialized summarization and classification for long-form text (for example, summarizing call transcripts into an incident narrative)

Fabrikam Inc. also has operational and knowledge data sources:

A curated policy library (internal SOPs, service warranty rules, escalation criteria, and standard operating procedures)

A set of indexed documents and procedures in an Azure AI Search service that supports vector search for the policy library

A Microsoft Fabric workspace that includes a semantic model used by operations leadership for reporting

Business requirements -

Fabrikam Inc. requires Operations Concierge to meet the following business requirements:

Traceability requirement: When the agent provides policy guidance or procedural recommendations, users must be able to see where the answer came from.

Metrics requirement: When users ask about service performance (backlog, SLA risk, parts shortages, dispatch delays), the solution must return

up-to-date metrics in a structured format that operations managers can use in weekly reviews.

Transaction requirement: The solution must support authenticated updates initiated during conversations, including creating a parts request and updating a service case status.

In addition, Fabrikam Inc. wants to avoid duplicating common assets across agents:

The team must reuse the same set of escalation topics, MCP tool definitions, and a standard safety disclaimer across three different agents.

Only the platform engineering group is allowed to edit shared assets. However, all agent authors must be able to use them.

Technical requirements -

The Fabrikam Inc. solution architecture uses a multi-agent approach so that specialist responsibilities are isolated and can evolve independently.

The Operations Concierge (primary agent) must coordinate the following specialist capabilities:

Policy and procedure Q&A: Use an enterprise knowledge source that supports indexed retrieval across the curated policy library and service procedures.

Operational metrics: Delegate metric queries to a Fabric Data Agent that reads governed business data through the Fabric semantic model.

Authenticated updates: Use tools exposed by an existing internal Model Context Protocol (MCP) server that provides transactional operations for the service organization.

Specialized processing: Delegate summarization and classification requests to an existing Microsoft Foundry agent.

Fabrikam Inc. will onboard two MCP servers as tools:

PartsOps MCP server: exposes tools for parts availability checks and parts request creation. The server requires per-user authentication because actions must be traceable to the requesting user.

WarrantyRules MCP server: exposes a read-only tool for validating warranty coverage. The server uses an API key shared by the agent team.

Fabrikam Inc. has also defined a collaboration requirement with the existing Service Desk Agent:

The primary agent must delegate IT-specific requests to the existing Service Desk Agent rather than reimplement ticket creation logic.

Finally, Fabrikam Inc. plans to support a partner integration:

For shipment tracking inquiries, Fabrikam Inc. will delegate to a partner-provided agent that is only available through a standardized agent-to-agent endpoint.

Issues and constraints -

During early testing, Fabrikam Inc. found three recurring problems:

Makers are copying and modifying the same components across agents, resulting in inconsistent disclaimers and duplicated tools.

Users can obtain a correct answer, but the response is not consistently traceable to a source when the agent uses knowledge.

The primary agent can route some requests, but specialist capabilities are not consistently delegated (for example, some metric questions are answered generatively instead of being routed to the Fabric Data Agent).

You are part of the engineering team responsible for correcting the design and configuration to meet the preceding requirements and constraints.

You need to integrate Fabrikam Inc.'s existing Foundry agent so Operations Concierge can delegate summarization requests.

Which action should you perform for each requirement? To answer, move the appropriate actions to the correct requirements. You may use each action once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Actions

Add an external agent of type Microsoft Foundry.

Publish the Copilot Studio agent to enable external calls.

Add the Foundry output as an indexed knowledge source.

Enter the Foundry project endpoint URL to create the connection.

Foundry agent integrations

Requirement	Action
The main agent must be able to call an external Foundry agent.	
The external Foundry agent must reference the correct Foundry project and agent.	

Suggested Answer:

Foundry agent integrations

Requirement	Action
The main agent must be able to call an external Foundry agent.	Add an external agent of type Microsoft Foundry.
The external Foundry agent must reference the correct Foundry project and agent.	Enter the Foundry project endpoint URL to create the connection.

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Fabrikam Inc. is a Canada-based manufacturer with a growing service organization that supports field technicians and internal operations teams. Fabrikam Inc. plans to launch a new internal agent solution named Operations Concierge to reduce time spent searching policy content, retrieving operational metrics, and executing routine transactions.

The agent will be used by three groups:

- Service coordinators who triage incoming service requests

- Field technicians who need guided procedures and parts availability

- Operations managers who monitor KPIs and exceptions

The agent solution must work in real-world operational conditions. Users often ask questions mid-call with a customer or while coordinating parts shipments. The agents require quick, reliable outcomes. As a result, Fabrikam Inc. requires the solution to:

- Provide grounded answers with traceability when it provides guidance.

- Retrieve real-time metrics when users ask for operational status.

- Execute authenticated updates when users initiate a flow (such as creating a parts request).

Fabrikam Inc. also expects the solution to be maintained by multiple makers and developers across the year. The company has experienced duplicated logic and inconsistent behavior across different agents. This project emphasizes reuse, governance, and maintainability across teams.

Current environment -

Fabrikam Inc. runs three Microsoft Power Platform environments for agent development and release: Dev, Test, and Prod.

The team plans to build the agent and validate it in Dev and Test, then promote to Prod by using a controlled release process that supports repeatable deployments.

Fabrikam Inc. already has two assets the team wants to reuse:

- A partially completed Copilot Studio agent named Service Desk Agent, used by IT to create internal tickets and route requests

- A Microsoft Foundry agent created by a central AI team that performs specialized summarization and classification for long-form text (for example, summarizing call transcripts into an incident narrative)

Fabrikam Inc. also has operational and knowledge data sources:

- A curated policy library (internal SOPs, service warranty rules, escalation criteria, and standard operating procedures)

- A set of indexed documents and procedures in an Azure AI Search service that supports vector search for the policy library

- A Microsoft Fabric workspace that includes a semantic model used by operations leadership for reporting

Business requirements -

Fabrikam Inc. requires Operations Concierge to meet the following business requirements:

Traceability requirement: When the agent provides policy guidance or procedural recommendations, users must be able to see where the answer came from.

Metrics requirement: When users ask about service performance (backlog, SLA risk, parts shortages, dispatch delays), the solution must return

up-to-date metrics in a structured format that operations managers can use in weekly reviews.

Transaction requirement: The solution must support authenticated updates initiated during conversations, including creating a parts request and updating a service case status.

In addition, Fabrikam Inc. wants to avoid duplicating common assets across agents:

The team must reuse the same set of escalation topics, MCP tool definitions, and a standard safety disclaimer across three different agents.

Only the platform engineering group is allowed to edit shared assets. However, all agent authors must be able to use them.

Technical requirements -

The Fabrikam Inc. solution architecture uses a multi-agent approach so that specialist responsibilities are isolated and can evolve independently.

The Operations Concierge (primary agent) must coordinate the following specialist capabilities:

Policy and procedure Q&A: Use an enterprise knowledge source that supports indexed retrieval across the curated policy library and service procedures.

Operational metrics: Delegate metric queries to a Fabric Data Agent that reads governed business data through the Fabric semantic model.

Authenticated updates: Use tools exposed by an existing internal Model Context Protocol (MCP) server that provides transactional operations for the service organization.

Specialized processing: Delegate summarization and classification requests to an existing Microsoft Foundry agent.

Fabrikam Inc. will onboard two MCP servers as tools:

PartsOps MCP server: exposes tools for parts availability checks and parts request creation. The server requires per-user authentication because actions must be traceable to the requesting user.

WarrantyRules MCP server: exposes a read-only tool for validating warranty coverage. The server uses an API key shared by the agent team.

Fabrikam Inc. has also defined a collaboration requirement with the existing Service Desk Agent:

The primary agent must delegate IT-specific requests to the existing Service Desk Agent rather than reimplement ticket creation logic.

Finally, Fabrikam Inc. plans to support a partner integration:

For shipment tracking inquiries, Fabrikam Inc. will delegate to a partner-provided agent that is only available through a standardized agent-to-agent endpoint.

Issues and constraints -

During early testing, Fabrikam Inc. found three recurring problems:

Makers are copying and modifying the same components across agents, resulting in inconsistent disclaimers and duplicated tools.

Users can obtain a correct answer, but the response is not consistently traceable to a source when the agent uses knowledge.

The primary agent can route some requests, but specialist capabilities are not consistently delegated (for example, some metric questions are answered generatively instead of being routed to the Fabric Data Agent).

You are part of the engineering team responsible for correcting the design and configuration to meet the preceding requirements and constraints.

You need to plan how Fabrikam Inc. will reuse shared components across multiple agents while limiting who can change the shared components.

Which approach should you use for each requirement? To answer, move the appropriate components to the correct requirements. You may use each component once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Components	Reusable components	
Restrict component collection to the primary agent. Create a component collection and move shared components into it. Export the primary agent as a solution and copy assets into other agents. Share the component collection with teammates while controlling edit access.	Requirement	Component
	Prevent shared assets from being modified by most builders.	
	Reuse the same set of components across multiple agents without copying.	
	Limit a shared set of components so it can be used only by its intended primary agent.	

Suggested Answer:

Reusable components	
Requirement	Component
Prevent shared assets from being modified by most builders.	Share the component collection with teammates while controlling edit access.
Reuse the same set of components across multiple agents without copying.	Create a component collection and move shared components into it.
Limit a shared set of components so it can be used only by its intended primary agent.	Restrict component collection to the primary agent.

Currently there are no comments in this discussion, be the first to comment!

Case Study -

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Fabrikam Inc. is a Canada-based manufacturer with a growing service organization that supports field technicians and internal operations teams. Fabrikam Inc. plans to launch a new internal agent solution named Operations Concierge to reduce time spent searching policy content, retrieving operational metrics, and executing routine transactions.

The agent will be used by three groups:

- Service coordinators who triage incoming service requests

- Field technicians who need guided procedures and parts availability

- Operations managers who monitor KPIs and exceptions

The agent solution must work in real-world operational conditions. Users often ask questions mid-call with a customer or while coordinating parts shipments. The agents require quick, reliable outcomes. As a result, Fabrikam Inc. requires the solution to:

- Provide grounded answers with traceability when it provides guidance.

- Retrieve real-time metrics when users ask for operational status.

- Execute authenticated updates when users initiate a flow (such as creating a parts request).

Fabrikam Inc. also expects the solution to be maintained by multiple makers and developers across the year. The company has experienced duplicated logic and inconsistent behavior across different agents. This project emphasizes reuse, governance, and maintainability across teams.

Current environment -

Fabrikam Inc. runs three Microsoft Power Platform environments for agent development and release: Dev, Test, and Prod.

The team plans to build the agent and validate it in Dev and Test, then promote to Prod by using a controlled release process that supports repeatable deployments.

Fabrikam Inc. already has two assets the team wants to reuse:

- A partially completed Copilot Studio agent named Service Desk Agent, used by IT to create internal tickets and route requests

- A Microsoft Foundry agent created by a central AI team that performs specialized summarization and classification for long-form text (for example, summarizing call transcripts into an incident narrative)

Fabrikam Inc. also has operational and knowledge data sources:

- A curated policy library (internal SOPs, service warranty rules, escalation criteria, and standard operating procedures)

- A set of indexed documents and procedures in an Azure AI Search service that supports vector search for the policy library

- A Microsoft Fabric workspace that includes a semantic model used by operations leadership for reporting

Business requirements -

Fabrikam Inc. requires Operations Concierge to meet the following business requirements:

Traceability requirement: When the agent provides policy guidance or procedural recommendations, users must be able to see where the answer came from.

Metrics requirement: When users ask about service performance (backlog, SLA risk, parts shortages, dispatch delays), the solution must return up-to-date metrics in a structured format that operations managers can use in weekly reviews.

Transaction requirement: The solution must support authenticated updates initiated during conversations, including creating a parts request and

updating a service case status.

In addition, Fabrikam Inc. wants to avoid duplicating common assets across agents:

The team must reuse the same set of escalation topics, MCP tool definitions, and a standard safety disclaimer across three different agents.

Only the platform engineering group is allowed to edit shared assets. However, all agent authors must be able to use them.

Technical requirements -

The Fabrikam Inc. solution architecture uses a multi-agent approach so that specialist responsibilities are isolated and can evolve independently.

The Operations Concierge (primary agent) must coordinate the following specialist capabilities:

Policy and procedure Q&A: Use an enterprise knowledge source that supports indexed retrieval across the curated policy library and service procedures.

Operational metrics: Delegate metric queries to a Fabric Data Agent that reads governed business data through the Fabric semantic model.

Authenticated updates: Use tools exposed by an existing internal Model Context Protocol (MCP) server that provides transactional operations for the service organization.

Specialized processing: Delegate summarization and classification requests to an existing Microsoft Foundry agent.

Fabrikam Inc. will onboard two MCP servers as tools:

PartsOps MCP server: exposes tools for parts availability checks and parts request creation. The server requires per-user authentication because actions must be traceable to the requesting user.

WarrantyRules MCP server: exposes a read-only tool for validating warranty coverage. The server uses an API key shared by the agent team.

Fabrikam Inc. has also defined a collaboration requirement with the existing Service Desk Agent:

The primary agent must delegate IT-specific requests to the existing Service Desk Agent rather than reimplement ticket creation logic.

Finally, Fabrikam Inc. plans to support a partner integration:

For shipment tracking inquiries, Fabrikam Inc. will delegate to a partner-provided agent that is only available through a standardized agent-to-agent endpoint.

Issues and constraints -

During early testing, Fabrikam Inc. found three recurring problems:

Makers are copying and modifying the same components across agents, resulting in inconsistent disclaimers and duplicated tools.

Users can obtain a correct answer, but the response is not consistently traceable to a source when the agent uses knowledge.

The primary agent can route some requests, but specialist capabilities are not consistently delegated (for example, some metric questions are answered generatively instead of being routed to the Fabric Data Agent).

You are part of the engineering team responsible for correcting the design and configuration to meet the preceding requirements and constraints.

You need to configure Operations Concierge so that prompt-driven summarization behavior aligns with Fabrikam Inc.'s architectural requirements.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Configure Azure AI Search as the grounding source for summarization prompts.
- B. Publish the agent to ensure the selected model is enforced for summarization.
- C. Select a model from the Foundry model catalog in the custom prompt configuration.
- D. Enable citations to control how summarization output is generated.
- E. Create a reusable custom prompt template and apply it where summarization responses are generated.

Suggested Answer: C, E

Currently there are no comments in this discussion, be the first to comment!

DRAG DROP -**Case Study -**

This is a case study. Case studies are not timed separately from other exam sections. You can use as much exam time as you would like to complete each case study. However, there might be additional case studies or other exam sections. Manage your time to ensure that you can complete all the exam sections in the time provided. Pay attention to the Exam Progress at the top of the screen so you have sufficient time to complete any exam sections that follow this case study.

To answer the case study questions, you will need to reference information that is provided in the case. Case studies and associated questions might contain exhibits or other resources that provide more information about the scenario described in the case. Information provided in an individual question does not apply to the other questions in the case study.

A Review Screen will appear at the end of this case study. From the Review Screen, you can review and change your answers before you move to the next exam section. After you leave this case study, you will NOT be able to return to it.

To start the case study -

To display the first question in this case study, select the "Next" button. To the left of the question, a menu provides links to information such as business requirements, the existing environment, and problem statements. Please read through all this information before answering any questions. When you are ready to answer a question, select the "Question" button to return to the question.

Background -

Fabrikam Inc. is a Canada-based manufacturer with a growing service organization that supports field technicians and internal operations teams. Fabrikam Inc. plans to launch a new internal agent solution named Operations Concierge to reduce time spent searching policy content, retrieving operational metrics, and executing routine transactions.

The agent will be used by three groups:

- Service coordinators who triage incoming service requests

- Field technicians who need guided procedures and parts availability

- Operations managers who monitor KPIs and exceptions

The agent solution must work in real-world operational conditions. Users often ask questions mid-call with a customer or while coordinating parts shipments. The agents require quick, reliable outcomes. As a result, Fabrikam Inc. requires the solution to:

- Provide grounded answers with traceability when it provides guidance.

- Retrieve real-time metrics when users ask for operational status.

- Execute authenticated updates when users initiate a flow (such as creating a parts request).

Fabrikam Inc. also expects the solution to be maintained by multiple makers and developers across the year. The company has experienced duplicated logic and inconsistent behavior across different agents. This project emphasizes reuse, governance, and maintainability across teams.

Current environment -

Fabrikam Inc. runs three Microsoft Power Platform environments for agent development and release: Dev, Test, and Prod.

The team plans to build the agent and validate it in Dev and Test, then promote to Prod by using a controlled release process that supports repeatable deployments.

Fabrikam Inc. already has two assets the team wants to reuse:

- A partially completed Copilot Studio agent named Service Desk Agent, used by IT to create internal tickets and route requests

- A Microsoft Foundry agent created by a central AI team that performs specialized summarization and classification for long-form text (for example, summarizing call transcripts into an incident narrative)

Fabrikam Inc. also has operational and knowledge data sources:

- A curated policy library (internal SOPs, service warranty rules, escalation criteria, and standard operating procedures)

- A set of indexed documents and procedures in an Azure AI Search service that supports vector search for the policy library

- A Microsoft Fabric workspace that includes a semantic model used by operations leadership for reporting

Business requirements -

Fabrikam Inc. requires Operations Concierge to meet the following business requirements:

Traceability requirement: When the agent provides policy guidance or procedural recommendations, users must be able to see where the answer came from.

Metrics requirement: When users ask about service performance (backlog, SLA risk, parts shortages, dispatch delays), the solution must return