Exam : **Professional Agentic Architect**

Title : Professional Agentic Architect

Version : DEMO

1.You are configuring an autonomous support agent using Agent Studio to handle multistep customer inquiries and invoke internal tools and connectors. During interactive testing in the console, the agent demonstrates inconsistent execution trajectories and fails to trigger the right tools. You need to improve the agent's reliability and accuracy.
What should you do first?
A. Enable Agent Search grounding in the Agent Studio tools pane, and lower the model temperature setting in the console configuration panel.
B. Configure Agent Platform Memory Bank in Agent Runtime to persist multiturn interaction logs, and increase the model's thinking_level parameter.
C. Use the side-by-side comparison view in Agent Studio to test alternative foundation models and select a Gemini Pro model variant.
D. Define imperatively structured system instructions in Agent Studio and add few-shot prompt templates that incorporate step-by-step reasoning examples for tools usage.
Answer: D
Explanation:
Defining clear, imperative system instructions in Agent Studio establishes explicit behavioral guidelines for how the agent should interpret customer requests, select appropriate tools, and execute multistep workflows in the correct sequence.
Adding few-shot examples that demonstrate successful tool invocations, expected parameter structures, and appropriate handling of tool responses helps the model learn reliable execution patterns. These examples provide concrete guidance for navigating complex customer support scenarios.
Google Cloud recommends starting with clear instructions and introducing targeted few-shot examples to address recurring behavioral failures. This improves tool selection accuracy, execution consistency, and overall agent reliability without requiring architectural changes or additional infrastructure.
Reference: https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/prompt-design-strategies https://docs.cloud.google.com/gemini-enterprise-cx/cx-agent-studio/instruction

2.You are designing a conversational agent in Gemini Enterprise to query proprietary internal engineering documentation stored across Cloud Storage and internal enterprise repositories. Your organization's security governance policy requires that individual users only receive search results from documents they are authorized to access based on their enterprise identity.
You need to connect this data source to your agent using a managed configuration that automatically enforces these access rules without custom code.
What should you do?
A. Store the enterprise documents in Cloud Storage, and configure a RAG Engine serverless pipeline using a basic similarity threshold with IAM condition filters.
B. Deploy an open-source vector database on Google Kubernetes Engine (GKE), and synchronize document metadata into a separate Firestore database collection.
C. Index the raw documents into an Agent Retrieval vector index, and configure client-side query filters matching each user's IAM group membership.
D. Connect the proprietary data source to an Agent Search enterprise data store with identity-aware access control list (ACL) synchronization enabled.
Answer: D
Explanation:

Google Cloud Agent Search provides managed enterprise data stores that support identity-aware, document-level access control through synchronized Access Control Lists (ACLs).
Connecting proprietary engineering documentation through an enterprise data store with ACL synchronization enabled allows the system to preserve document permissions from the original data sources. By integrating with the organization's identity provider, Agent Search maps authenticated users and groups to their corresponding document permissions.
During retrieval, Agent Search automatically enforces these permissions, ensuring users receive only documents they are authorized to access. This approach supports secure, identity-aware enterprise search without requiring custom authorization code, while simplifying data integration and centralized access governance.
Reference:
https://docs.cloud.google.com/gemini/enterprise/docs/iam-policy-for-apps-and-data-stores
https://docs.cloud.google.com/gemini/enterprise/docs/identity-mapping
https://docs.cloud.google.com/gemini/enterprise/docs/connectors/introduction-to-connectors-and-data-stores

3.You are building a support voice assistant using Customer Experience Agent Studio (CX Agent Studio) for a telecommunications organization. Inbound calls must be dynamically routed based on caller intent. If a caller requests billing details or account changes, PIN/OTP verification is required to establish a secure data-access boundary before any generative resolution steps can be executed. You want to configure the solution with minimal custom code.
What should you do?
A. Build a single generative agent with authentication and troubleshooting components configured as parallel tools.When a customer requests account-sensitive help, execute both tools concurrently to minimize latency, and configure safety filters to redact any diagnostic output if authentication fails.
B. Build a sequential cascade of two independent generative agents.Configure the first agent to handle intent classification and the OTP verification, and store state in an external database.Have the second agent poll this database for customer ID and continue the conversation, triggering a custom session initiation protocol (SIP) transfer webhook if needed.
C. Build a single generative agent.Use system instructions to classify intent.Use prompt-based guardrails and the model’s memory to enforce authentication before running troubleshooting steps.Execute a custom webhook tool to handle the OTP API call.
D. Build a hybrid architecture.Use a generative steering node to classify intent.Route account-sensitive queries to a deterministic flow component for PIN/OTP authentication.Pass the verified customer ID as a session parameter to the runtime.
Answer: D
Explanation:
A hybrid architecture in CX Agent Studio combines generative intent classification with deterministic authentication workflows, providing flexible conversational routing while enforcing strict security boundaries.
A generative steering node interprets the caller's intent and routes billing inquiries or account modification requests to a deterministic flow that performs PIN or OTP verification. The flow uses predefined validation steps and conditions to ensure that authentication succeeds before the caller can access sensitive account information or continue to generative resolution.

After successful verification, the flow stores the verified customer ID and authentication status in session parameters, making this information available to subsequent agent interactions. This enables secure, context-aware customer support while minimizing custom code and ensuring that sensitive operations remain protected by explicit authentication controls.
Reference:
https://docs.cloud.google.com/dialogflow/cx/docs/concept/playbook/parameter
https://docs.cloud.google.com/gemini-enterprise-cx/cx-agent-studio/flow

4.You are configuring an agentic workflow in Agent Studio for an enterprise customer analytics platform. The agent was initialized with default system instructions, prompt templates, and a Gemini Flash model family. In staging, you observe that the agent frequently hallucinates when synthesizing data across multiple knowledge documents and produces inconsistent output formats.
You need to ensure the agent reliably uses verified source context and follows a standardized response format.
What should you do first?
A. Update the system instructions to restrict answers strictly to retrieved source data, and reconfigure prompts with chain-of-thought reasoning steps and few-shot input-output examples.
B. Update the system instructions to reduce the temperature parameter and maximum top-p sampling to zero, and change the model to a Gemini Pro model family.
C. Update the system instructions with explicit data redaction rules, and configure prompt templates to send all output through a Model Armor filter for schema enforcement.
D. Update the system instructions to route date parsing requests to a custom Cloud Run OpenAPI extension tool, and reconfigure prompt templates to increase chunk overlap in Agent Search.
Answer: A
Explanation:
Refining system instructions and prompt templates is the recommended first step for improving agent reliability, grounding, and output consistency in Agent Studio.
Explicitly instructing the agent to generate answers only from verified retrieved documents reduces hallucinations and prevents unsupported claims. Structured reasoning instructions help the agent systematically analyze information across multiple sources, while few-shot input-output examples demonstrate the expected response format, citation behavior, and synthesis patterns.
Google Cloud recommends using clear system instructions, structured prompts, and carefully selected few-shot examples to improve instruction adherence and response consistency. These changes strengthen grounding and standardize outputs without requiring additional infrastructure or model replacement.
Reference:
https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/system-instructions?hl=en
https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/prompt-design-strategies?authuser=0
https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/prompts/few-shot-examples?hl=en

5.You are designing an automated hotel reservation workflow using Customer Experience Agent Studio

(CX Agent Studio). The agent prompts the guest for check-in dates, room tier, and rewards membership status in a dedicated booking page. If a guest selects a luxury suite, the conversation flow should immediately branch to a specialized concierge flow only after the loyalty status parameter is confirmed. You need to configure this routing logic within the visual console without custom webhook middleware. What should you do?

A. Attach a custom JavaScript event handler to the start page to redirect the incoming user session.

B. Configure a global intent route that overrides active page form parameters during conversation turns.

C. Configure a conditional transition route on the page that evaluates form parameters before setting the target flow.

D. Configure a conditional transition route in a session event handler that resets form parameters before setting the target flow.

Answer: C

Explanation:

In Customer Experience Agent Studio (CX Agent Studio), a page-level conditional transition route enables deterministic conversation routing based on collected form parameters without requiring custom middleware.

Configure the booking page to collect the guest's room tier and loyalty membership status as form parameters. Then define a conditional transition route that checks whether the selected room tier is a luxury suite and whether the loyalty status has been confirmed.

Once both conditions are satisfied, the route automatically transitions the conversation to the specialized concierge flow. This ensures the concierge workflow begins only after the required information is available, providing reliable, parameter-driven routing entirely through the visual console.

Reference:

https://docs.cloud.google.com/dialogflow/cx/docs/concept/parameter

https://docs.cloud.google.com/dialogflow/cx/docs/reference/rest/v3beta1/TransitionRoute

https://docs.cloud.google.com/dialogflow/cx/docs/concept/handler