MockCert

Study guide | Salesforce | AI-201

Salesforce Certified Agentforce Specialist

Notes for every domain of the Salesforce Certified Agentforce Specialist exam, organized by the official exam guide's domains and weights.

By MockCert about 81 min read Updated November 19, 2026

CONTENTS

DOMAIN 1 · 20%
Prompt Engineering 2

DOMAIN 2 · 20%
Data 360 Fundamentals 18

DOMAIN 3 · 35%
AI Agents 23

DOMAIN 4 · 10%
Testing, Deployment, and Maintenance 41

DOMAIN 5 · 10%
Governance and Observability 49

DOMAIN 6 · 5%
Multi-Agent Orchestration 54

MockCert is an independent certification preparation platform. It is not affiliated with, endorsed by or operated by Salesforce. Exam domains and weights follow the official exam guide. These notes were last updated on November 19, 2026; the notes at mockcert.com are kept up to date as the exam changes.

Domain 1 | 20% of the exam | 23 min read

Prompt Engineering

Overview

This domain is about 20% of the exam, roughly 12 of its 60 questions.

Every point below comes from Salesforce's official documentation; each section lists its sources.

Given business requirements, identify when it's appropriate to use Prompt Builder.

Start here

THINK FIRST

When a business asks for AI-written content from CRM data, how do you tell that Prompt Builder is the right tool?

Why it matters: Questions describe a business need and ask whether a reusable prompt template built in Prompt Builder fits it.

Prompt Builder is where you build and manage reusable prompt templates. Each template has placeholders that fill with live CRM data before it reaches the LLM. Pick it when many users need consistent AI output grounded in records.

Key terms

Term	Meaning
Prompt template	A reusable, detailed prompt you create and manage in Prompt Builder.
Placeholder	A slot in a template for data that changes, such as a customer name or product price.
Prompt resolution	Filling a template's placeholders with real data to get a tailored prompt.
Merge field	A grounding reference to record fields, flows, Apex or uploaded files.
Einstein Trust Layer	Masks sensitive data with placeholder text before the prompt reaches the LLM.

What Prompt Builder is for

It builds and manages prompt templates that use your CRM data. Those templates power generative AI features and Agentforce agents. So a requirement for AI output that draws on CRM data matches what

Prompt Builder does.

Signals that point to a template

Choose a template when output must be consistent across many people. Prompts written by different users vary in style, so their output varies too. A reusable template fixes that and scales prompt design. That makes it a strong fit for B2C messages that need one voice at scale.

EXAM TRAP

Letting each user write their own prompt is not the scalable answer. Varied writing styles produce inconsistent output; a shared template solves that.

Grounding in current data

Placeholders keep each prompt tied to up-to-date record data. You ground a template with merge fields. They can point to record fields, flows, Apex and uploaded files. At run time, prompt resolution swaps each placeholder for the real value.

Template types match use cases

A template type gives you a starting point for one use case. For example, a Sales Email template helps sales reps draft personalized emails to a contact or lead. Match the business requirement to the type that serves it.

Build, test, activate

Prompt Builder guides a template from creation to testing to activation. In preview, turning Generate Response off shows only the resolved prompt; nothing goes to the LLM. Activation then makes the template available across the org, inside apps like Sales Emails and Lightning record pages.

EXAM TRAP

A template is not usable just because you saved it. Users get it in Salesforce apps only after you activate it.

Trust and edition checks

The Einstein Trust Layer masks sensitive data sent to the LLM. It sends placeholder text instead, then adds the real data back into the response. Also confirm licensing: Prompt Builder needs Enterprise, Performance or Unlimited Edition plus a qualifying Einstein or Agentforce add-on, or Agentforce Foundations.

Key takeaways

- Use Prompt Builder when many users need consistent, CRM-grounded AI output.
- Placeholders and merge fields keep prompts tied to current record data.
- A template reaches users in Salesforce apps only after activation.
- With Generate Response off, preview shows the resolved prompt and sends nothing to the LLM.

Sources: Prompt Builder | Salesforce Help (opens in a new tab) · Prompt Builder Key Concepts | Salesforce Help (opens in a new tab) · Get to Know Prompt Builder | Salesforce Help (opens in a new tab)

Identify access controls governing prompt templates.

Start here

THINK FIRST

How do you control who can build prompt templates, who can run them, and which version they get?

Why it matters: Questions describe a user who cannot build or run a prompt template and ask which permission or control fixes it.

Access to prompt templates is managed with permission sets. One set lets people build and manage templates, another lets them run templates. Activation then decides which version of a template users reach.

Key terms

Term	Meaning
Prompt Template Manager	The permission set for creating and managing prompt templates in Prompt Builder.
Prompt Template User	The permission set for accessing and running prompt templates outside Prompt Builder.
Execute Prompt Templates	The permission to run prompt templates, included in both permission sets.

Builders and runners need different sets

Manager builds templates; User runs them. Both permission sets include the right to run templates. Only Prompt Template Manager adds the right to manage them in Prompt Builder. Customize Application is an alternative for either job.

The two prompt template permission sets compared

Feature	Prompt Template Manager	Prompt Template User
Lets a user	Create and manage templates in Prompt Builder	Access and run templates outside Prompt Builder
Permissions included	Manage Prompt Templates and Execute Prompt Templates	Execute Prompt Templates
Alternative permission set	Customize Application	Customize Application

EXAM TRAP

Assigning Prompt Template Manager fails for a user without Setup access. Setup access is required before the user can be added to that set.

Controlling versions

Activation decides which template version users can reach. Activate the version you want people to use, and deactivate the ones you do not. This gives fine control over what runs for your users.

Standard templates

Copy a standard template to customize it. Some standard templates can also be overridden. The override works in place of the standard version.

Where templates appear

Template type decides where users meet it. Sales email templates are added to Einstein for Sales. Field generation templates are added to dynamic fields on Lightning App Builder pages. Before changing a template, view its references to see which flows, agents and components depend on it.

Key takeaways

- Permission sets control who can use prompt templates.
- Prompt Template Manager adds Manage Prompt Templates; Prompt Template User only runs templates.
- A user needs Setup access before joining the Prompt Template Manager permission set.
- Activation and deactivation control which template version users see.

Sources: How Users Work with a Prompt Template | Salesforce Help (opens in a new tab) · Enable Prompt Builder | Salesforce Help (opens in a new tab) · Manage Prompt Templates | Salesforce Help (opens in a new tab)

Identify the considerations for using a prompt template type such as field generation and flex types.

Start here

THINK FIRST

Given a content need, how do you choose between a Field Generation template and a Flex template, and what must be in place for each?

Why it matters: Questions describe where AI content is needed and ask which prompt template type fits and what setup it requires.

Template types fit different needs. Field Generation fills a record field when a user clicks a button. Flex covers any purpose the other types don't, with resources you define yourself, and you call it wherever you need it.

Key terms

Term	Meaning
Field Generation template	Creates content for a record field in Lightning Experience when a user clicks a button.
Flex template	Creates content for business purposes other template types don't cover.
Resource	An object you define as input to a Flex template; it can be marked required.
Generative AI-enabled field	A record field that uses a Field Generation template for summaries and descriptions.

Two types, two jobs

Pick the type by where the content goes and what feeds it. Field Generation serves one record field. Flex serves anything else, with inputs you choose.

Field Generation and Flex compared

Feature	Field Generation	Flex
Purpose	Content for a record field in Lightning Experience	Any business purpose other types don't cover
How it runs	A user clicks a button to fill the field	Called from custom UI or through the REST API
Where you use it	Record fields in Lightning Experience	Wherever you need it in your org

Field Generation in use

Generative AI-enabled fields give dynamic summaries and descriptions. Users can ask Einstein to rewrite a draft, through brief back-and-forth turns. No AI field icon means the field doesn't support a prompt template. On mobile, an admin must first enable Dynamic Forms on Mobile.

EXAM TRAP

Not every record field can take a template. If the generative AI-enabled field icon is missing, a prompt template isn't supported there.

Setting up Flex resources

A Flex template can have up to 5 resources. Give each resource object a unique Name and API Name. A checkbox marks a resource as required. In Preview Settings, pick an object for each resource to test.

EXAM TRAP

A Flex template cannot take unlimited inputs. It allows up to **5** resources.

Permissions to check

Check who builds templates and who runs them. Prompt Template Manager, or Customize Application, lets users create and manage templates. Prompt Template User lets them run templates outside Prompt Builder. If field generation templates aren't visible with a custom set, grant object access plus `ExecutePromptTemplate` and `ManagePromptTemplate`.

Key takeaways

- Use Field Generation for content in a record field; use Flex for anything else.
- Flex templates take up to **5** resources and run from custom UI or the REST API.
- Generative AI-enabled fields on mobile need Dynamic Forms on Mobile enabled first.
- Running templates outside Prompt Builder needs the Prompt Template User permission set.

Sources: Prompt Template Types | Salesforce Help (opens in a new tab) · Flex Prompt Templates in Action | Salesforce Help (opens in a new tab) · Create a Flex Prompt Template | Salesforce Help (opens in a new tab) · Field Generation Prompt Templates in Action | Salesforce Help (opens in a new tab)

Given a scenario, identify the appropriate grounding technique.

Start here

THINK FIRST

When should you ground AI in unstructured content through a search index, and when through metadata grounding?

Why it matters: Questions describe what the AI needs to know and ask which grounding technique or setting fits that need.

Unstructured content is grounded through a Data 360 search index: the data is chunked, turned into embeddings and cited. Org configuration is grounded through metadata grounding, which searches objects, fields and components in natural language.

Key terms

Term	Meaning
Chunk	A piece of content, sized by a token limit, that goes into the search index.
Vector embedding	A machine-readable form of a chunk that Data 360 creates for search.
Search index	A Data 360 index over chunked content that also creates a citation for each chunk.
Metadata grounding	Discovering, understanding and reusing org metadata through natural language.

Two grounding paths

Pick the technique by what the AI must know. Grounding in documents and other unstructured data runs through a search index. Grounding in how the org is configured runs through metadata grounding. The table compares the two.

Search index grounding and metadata grounding compared

Feature	Search index	Metadata grounding
Grounds the AI in	Unstructured data, split into chunks	Objects, fields and metadata components
Search behind it	A vector or hybrid index over chunk embeddings	Data 360 hybrid search, semantic and keyword
Key benefit	Citations to verify AI responses	Reuse working configurations as templates

Chunking comes first

Data must be chunked before it enters the search index. Data 360 offers several chunking strategies and picks one by content type. The chunks become vector embeddings. Chunking is available in every edition Data 360 supports.

EXAM TRAP

You do not have to pick a chunking strategy for each file. Data 360 selects one automatically from the content type.

Chunk size and the token limit

The max token limit sets how much text one chunk holds. It lives in the search index configuration. By default, the chunking max token limit is **512** tokens.

What a search index creates

A configuration creates a chunk object and an index object. Both are data model objects in Data 360. They store the chunk data and its generated embeddings. Each chunk gets a citation, so you can check AI answers against the content.

Enriching chunks

Vector and hybrid indexes can enrich chunks with metadata. When you build one from unstructured data, you can add extracted keywords, entities, topic overviews, questions answered and summaries to the chunks.

Grounding in metadata

Metadata grounding explains and reuses org configuration. It explains what a custom object is for, what a field name means and how security is set up. You can also reference a working configuration as a template

for a new field or object. It uses the Metadata Grounding MCP Server, which is in beta.

EXAM TRAP

Metadata grounding searches objects, fields and components, not documents. Grounding in unstructured content belongs in a chunked search index.

Key takeaways

- Unstructured data is chunked, embedded and indexed, and each chunk gets a citation.
- The chunking max token limit defaults to **512** and controls chunk size.
- Each search index configuration adds two data model objects: one for chunks and one for the index.
- Metadata grounding uses hybrid search over objects, fields and components through a beta MCP server.

Sources: Chunk Data | Salesforce Help (opens in a new tab) · Metadata Grounding Capabilities | Salesforce Help (opens in a new tab)

Explain the process for creating, activating, and executing prompt templates.

Start here

THINK FIRST

How does a prompt template move from a first draft to a live response that users can rely on?

Why it matters: Questions describe a stage in a template's life and ask what happens at design time, activation or run time.

You draft and preview a template in Prompt Builder, then activate one version for users. At run time, an agent, flow, Apex or app invokes it. The prompt is resolved securely before the LLM responds.

Key terms

Term	Meaning
Prompt Template Workspace	The Prompt Builder area where you draft a prompt and see its current version number.
Template version	A saved state of a template; only one version can be active at once.
Prompt resolution	The run-time step that adds merge field values from records, flows or Apex to the prompt.
Data masking	Run-time protection: pattern-based masking is automatic, field-based masking only works when enabled.

Who can build templates

Prompt Template Manager or Customize Application grants access. A user needs one of these two permission sets to create and manage templates in Prompt Builder. The Prompt Template Manager set carries the Manage Prompt Templates and Execute Prompt Templates permissions.

Drafting and saving

Your first save creates Version 1. You write the prompt in the Prompt section of the Prompt Template Workspace. After you click Save, the version number shows at the top of the workspace. Starting from a standard template? Copy it to customize it, or override it where Salesforce allows.

Previewing at design time

Preview simulates the whole run-time path. You give the template some input and check the resolved prompt. Then you see the response the LLM returns. So you can test a template end to end before anyone else uses it.

Activating a version

New templates stay unavailable to users until activated. Drafting is iterative, so users can't run a template by default. Only **one** version can be active at a time. With no active version, the template is unavailable to users.

What each version state allows

Feature	Never activated	Active now	Once active, now inactive
Users can run this version	No	Yes	No
Can be edited in place	Yes	No, save a new version	No, save a new version

EXAM TRAP

Deactivating a version does not make it editable again. A version that was ever active stays immutable, so you save a new version to change it.

Invoking at run time

An agent, flow, Apex or app invokes the template. You can also call it through an invocable action anywhere on the platform. Another route is Connect in Apex, or the Connect REST API. Prompt resolution then adds merge field values to the prompt.

EXAM TRAP

Merge data is not fetched with the template author's or an admin's access. The permissions of the user running the template decide what data the prompt can include.

Masking and the response

Pattern-based masking is automatic; field-based masking needs enabling. Both kinds of masking happen during run-time execution. Don't count on field-based masking unless it is turned on. The generated response then comes back in the user's language by default, or in a language you request.

Where users meet templates

The template type decides where it shows up. Custom sales email templates show up automatically as choices in the Lightning Experience email composer. Field generation templates go on dynamic fields in Lightning App Builder pages. Before changing a template, check its references to see which flows, agents and components depend on it.

Key takeaways

- Templates aren't available to users until you activate a version.
- Only **one** version is active at a time; with none active, the template is unavailable.
- A version that was ever active can't be edited; save a new version instead.
- Merge data at run time follows the executing user's permissions.
- Pattern-based masking is automatic, while field-based masking applies only if enabled.

Sources: Prompt Template Run-time Execution | Salesforce Help (opens in a new tab) · Activate and Deactivate Prompt Templates | Salesforce Help (opens in a new tab) · Manage Prompt Templates | Salesforce Help (opens in a new tab) · How Users Work with a Prompt Template | Salesforce Help (opens in a new tab)

Explain how to implement best practices for writing effective prompts.

Start here

THINK FIRST

What makes a prompt template easy for an LLM to follow and easy for your team to maintain?

Why it matters: Questions describe a prompt template that gives poor or inconsistent output and ask which best practice fixes it.

A good prompt keeps instructions and data clearly apart. It gives the model a role, plain wording and one clear job. Then you test it, gather feedback and refine it.

Key terms

Term	Meaning
Triple quotes	A container (""") that marks the text inside it as data or context, apart from the instructions.
Merge field	A field of record data in a prompt; it needs a readable label and may be empty for some records.

Term	Meaning
Fallback instruction	A line telling the AI what to do when a merge field is empty for a record.
Resolution panel	The panel where an automatic summary appears when a prompt is too large for the model.

Separate data from instructions

Wrap variable data in triple quotes. This is the single most important way to organize a complex prompt. Everything inside the quotes reads as data, not as orders. To mark the orders themselves, add a line reading `Instructions:` and wrap them in triple quotes too.

EXAM TRAP

Mixing record data into the instruction text makes it hard for the model to tell context from orders. Keep each in its own quoted block.

Structure long prompts

Long prompts need headers; very complex ones use several quoted blocks. Once a prompt runs past one screen, clear headers keep it scannable and easy to maintain. For very complex prompts, use one quoted block for variable data. Use a second block for a static example of tone and structure.

Label and guard merge fields

Label each merge field; add a fallback where one may be empty. A label tells the model what each value means, which helps most when field names are cryptic. A fallback instruction covers records where the field is empty. As a general rule, include each piece of data only once.

EXAM TRAP

Repeating a merge field to stress it does not help. Repeated data can make results inconsistent, so include each value once.

Give the model a role and one job

Assign a character, define its goal, and limit the output. Asking the model to act as a sales or support representative adds context. Direct wording such as "generate only the message to the customer" stops it from describing how it would write the content.

Write plainly and consistently

Use concise, natural language in one consistent style. Avoid industry jargon and technical terms. Pick one writing style for your templates and keep to it. A consistent style leads the model to produce consistent responses.

Start from examples and iterate

Begin with a library template, then test and refine. Salesforce suggests customizing a template from the Example Prompt Template Library. Try different templates for the same goal and collect end-user feedback. If a prompt is too large, the Resolution panel shows an automatic summary you can use, or you can shorten the prompt.

Key takeaways

- Triple quotes are the key technique for keeping variable data apart from instructions.
- Label each merge field, add a fallback for empty values, and include each value only once.
- A role with a goal, plus direct output instructions, keeps the response on target.
- Improve templates by trying alternatives and gathering end-user feedback.

Sources: Organize Complex Prompt Templates | Salesforce Help (opens in a new tab) · Best Practices for Building Prompt Templates | Salesforce Help (opens in a new tab) · Work with Large Prompts | Salesforce Help (opens in a new tab)

Identify the security and privacy features of the Trust Layer.

Start here

THINK FIRST

Which Trust Layer protections apply to a prompt, and how does that change when an agent sends it?

Why it matters: Questions describe how data reaches an LLM and ask which Trust Layer feature protects it, or whether it applies.

The Einstein Trust Layer brings together grounding by user permissions, data masking, zero data retention, toxicity detection and an audit trail. Agents, however, run with data masking turned off.

Key terms

Term	Meaning
Secure data retrieval	Grounding a prompt with only the data the running user is permitted to see.
Data masking	Detecting sensitive data and masking it before the prompt goes to the LLM.
Zero data retention	Third-party LLMs keep no data sent to them and do not train on it.
Toxicity score	A value from 0 to 1 for each category, where 1 is the most toxic.
Trust boundary	The Salesforce perimeter; an LLM hosted inside it keeps data within it.

What the Trust Layer is

A secure AI architecture built into the Salesforce platform. It combines agreements, security technology, and data and privacy controls. The steps below follow a prompt from grounding to the audit log.

Permission-aware grounding

Grounding respects the running user's permissions. Secure data retrieval keeps role-based controls and field-level security when it merges grounding data. So grounding data is limited by that user's access.

Data masking and agents

Masking hides sensitive data, but agents skip it. You choose what must and must not be masked. Pattern-based and field-based masking are disabled for agents. Where masking applies depends on how the prompt reaches the LLM.

When data masking applies

Situation	Data masking	Other protection
Prompt template used directly	Applied per Trust Layer settings	You select what is masked
Prompt template called by an agent action	Disabled	Zero data retention for third-party LLMs
Agent using an LLM inside the trust boundary	Disabled	Data stays in the boundary; zero retention applies

EXAM TRAP

Masking settings do not carry into agents. A prompt template run through an agent action is sent without masking.

Zero data retention

Third-party LLMs keep none of the data sent to them. That data is also never used for model training or product improvements.

Toxicity detection

Response checks are always on; prompt checks are optional. Detection in responses is on by default and cannot be changed. Detection in prompts is a beta feature that is off until you enable it. Models flag Violence, Sexual, Profanity, Hate and Physical content.

EXAM TRAP

Toxicity scores run from **0** to **1**, and **1** is the most toxic, not the safest. An overall score also combines all categories.

Prompt defense and audit

Prompt defense limits harm; the audit trail records it all. System policies lower the chance of hallucinations and harmful outputs. Prompts, responses, trust signals and toxicity scores are logged and stored in Data 360.

Key takeaways

- Secure data retrieval grounds prompts with the running user's permissions and field-level security.
- Data masking is disabled for agents, even with an LLM inside the trust boundary.
- Third-party LLMs retain no data and never train on it.
- Response toxicity detection is always on; prompt toxicity detection is a beta you enable.
- The audit trail stores prompts, responses and trust signals in Data 360.

Sources: Einstein Trust Layer | Salesforce Help (opens in a new tab) · Toxicity Detection | Salesforce Help (opens in a new tab) · Data Masking Limitations in Agentforce | Salesforce Help (opens in a new tab)

Explain how to manage and prevent specific models from being accessed.

Start here

THINK FIRST

When would you hide a model instead of deleting it, and what happens to the templates that use it?

Why it matters: Questions describe a model you want to retire or test and ask how to control who can select it.

In AI Models, you hide an LLM configuration to stop new selections in Prompt Builder. Templates already using it keep working. Deleting a model is permanent, so hiding is the reversible way to control access.

Key terms

Term	Meaning
AI Models	The Salesforce app, formerly Einstein Studio, where you connect and configure models.
BYOLLM	Bring Your Own LLM: connects an externally hosted foundation model to AI Models.
LLM configuration	A configured model, listed on the Configured Models tab, that template authors can pick.
Model Visibility	The column that tells you if Prompt Builder users can select a model or not.

Why hide a model

Hiding steers Prompt Builder users to another model. Hide a configuration when you want template authors to move to a newer LLM version or another LLM. The replacement can even come from a different provider. Hiding also lets you hold back a new LLM until your testing is done.

How to hide it

Use Hide Model on the Configured Models tab. Open the Configured Models tab and pick Hide Model from that configuration's dropdown menu. The Model Visibility column then reads Hidden, and (Not Available) appears before the name. You can show it again later to make it selectable.

Impact on existing templates

Hiding never breaks templates already using the model. Prompt Builder marks such a template's model as hidden. The author can switch by choosing an available model from the Models list. Clones are stricter, as the table shows.

Existing templates and their clones after a model is hidden

Feature	Existing template	Cloned copy
Runs on the hidden model	Yes, the same as before	No, saving needs another model
How to switch models	Pick an available model from the Models list	Select another model before saving the copy

EXAM TRAP

Hiding a model doesn't stop existing templates from running. It removes the model from selection, and a clone must switch models before it can be saved.

Hide versus delete

Deleting a model can't be undone. Once a model is deleted, none of its versions or prediction jobs can be recovered. A hidden configuration, by contrast, can be shown again later. So hide a model when you may want it back.

EXAM TRAP

Don't delete a model just to pause access while you test a new one. Deletion loses every version and prediction job, while hiding can be reversed.

Versions and source models

Only one model version can be active at a time. Activating the latest version switches off any earlier versions automatically. A foundation model that serves as a source for other models is also locked for most edits. You must first delete all of its associated model configurations.

Who can manage models

Model management depends on your permissions. The Allow users to manage models in AI Models permission lets a user create, update and delete models. The Data Cloud Architect permission set goes further, with admin-level access to every AI Models feature, including activation.

Key takeaways

- Hide a model to steer authors elsewhere; existing templates keep running on it.
- Hide Model sits in the configuration's dropdown menu on the Configured Models tab.
- A cloned template with a hidden model must switch models before it can be saved.
- Deleting a model is permanent; hiding can be reversed by showing it again.
- Activating a model's latest version deactivates its earlier versions.

Sources: Manage Large Language Model (LLM) Access by Hiding Configurations | Salesforce Help (opens in a new tab) · Manage Models | Salesforce Help (opens in a new tab) · Bring Your Own Large Language Model | Salesforce Help (opens in a new tab) · Activate Your Model | Salesforce Help (opens in a new tab)

Domain 2 | 20% of the exam | 8 min read

Data 360 Fundamentals

Overview

This domain is about 20% of the exam, roughly 12 of its 60 questions.

Every point below comes from Salesforce's official documentation; each section lists its sources.

Explain the considerations of Agentforce Data Library and its concepts.

Start here

THINK FIRST

What do you need to decide and check before an Agentforce Data Library can ground your agent's answers?

Why it matters: Questions describe a grounding need and ask which data source, setting or limit of a data library applies.

A data library turns unstructured content into searchable chunks for AI. Saving one builds the Data 360 pieces for you, ending in a search index and a retriever. Your choices of source and settings decide what the agent can find.

Key terms

Term	Meaning
Data library	A store of knowledge articles, uploaded files or a custom retriever that grounds AI answers.
Chunk	A smaller part of a data source, made so search is faster and more relevant.
Search index	The organized chunks that search uses to find the parts most relevant to a query.
Retriever	Decides which Data 360 datasets AI tools can use; each data library has its own.
Intelligent Context	Advanced processing for file-based libraries that builds an optimal index and retrieval pipeline.

What a data library does

It makes unstructured content searchable for your agent. Web content, documents and large text fields become searchable information. When a user asks a question, the Answer Questions with Knowledge action answers from the matching data library.

What saving a library builds

Saving creates a data stream, then the objects, then the index. The data stream comes first, followed by data lake and data model objects. Those are mapped, and only then is the search index created. Each library also gets its own retriever, even when retrievers share an index.

Chunks and the search index

Sources are split into chunks, then indexed. Chunking makes search more efficient and handles very large datasets. The index lets search pick the most relevant chunks instead of reading a whole document.

Choosing the data source

Pick knowledge articles, uploaded files or a custom retriever. Choose carefully, because the source can't be changed after you select it. Each source has its own rules. A Data Cloud One companion org connects only to retrievers in its home org. So configure the library in the home org first.

The three data sources compared

Feature	Knowledge articles	Uploaded files	Custom retriever
Key requirement	Articles public and published (`IsVisibleInPkb` is True)	Up to **1,000** files; PDFs up to **100 MB**	An active retriever created in Data 360
Fields or content	Text or text area fields only	Text and HTML files up to **4 MB**	Search index and retriever you build first
In a Data Cloud One companion org	Not supported	Not supported	Required

EXAM TRAP

You cannot switch a library from files to Knowledge later. The data source is fixed once selected.

File libraries and Intelligent Context

File libraries get Intelligent Context processing automatically. It uses extra data credits. Turning it off may lower retrieval accuracy for tables and infographics, and you can't change it after saving. Keep the setup page open while files upload, or the upload fails.

Web search is not a new source

New data libraries don't support web search. To search the web, agents use the Search the Web standard action. Prompt templates rely on the web retriever in Prompt Builder instead. Existing web search libraries

are read-only.

EXAM TRAP

Creating a new data library for web content is not the answer. Use the Search the Web standard action in agents.

Licensing, credits and permissions

Data libraries need Data 360 and consume its credits. Express libraries are a type of data library that Agentforce creates in seconds. They are the exception: they don't consume Data 360 credits. Setting up a library needs Data Cloud admin and System Administrator permissions. Service reps need the Data Cloud User and Prompt Template User permission sets.

Sandboxes

Every sandbox type supports data libraries, but data does not copy. Data 360 sandboxes hold only metadata, so re-upload files to file-based libraries. Deploying data library changes from a sandbox to production or another sandbox isn't supported.

Key takeaways

- A library's data source is fixed once selected: Knowledge, files or a custom retriever.
- A library can't be used while its status is In Progress; builds take minutes to hours.
- Knowledge articles blocked by sharing or permissions never appear in AI responses.
- To give an agent several data libraries, create an ensemble retriever first.

Sources: What Are Data Libraries? | Salesforce Help (opens in a new tab) · Setting Up Data Libraries | Salesforce Help (opens in a new tab) · Use Uploaded Files | Salesforce Help (opens in a new tab) · Use a Custom Retriever | Salesforce Help (opens in a new tab) · Agentforce Data Library Troubleshooting | Salesforce Help (opens in a new tab) · Use Salesforce Knowledge | Salesforce Help (opens in a new tab) · Agentforce Data Library | Salesforce Help (opens in a new tab) · Create a Data Library | Salesforce Help (opens in a new tab)

Explain foundational concepts of Data 360 such as chunking, indexing, and retrievers.

Start here

THINK FIRST

How does content in Data 360 become the grounding an agent or prompt template can retrieve?

Why it matters: Questions describe grounding content or a data setup and ask which chunking, indexing or retriever behavior applies.

Data 360 brings data together into unified profiles. Grounding for agents and prompt templates rests on chunking, indexing and retrievers. A retriever returns relevant data to augment a prompt.

Key terms

Term	Meaning
Retriever	The component that fetches relevant data and adds it to a prompt.
Section-aware chunking	Chunking that splits a document at titles and headings, keeping each section's text together.
Passage extraction	Chunking that divides a document into passages by meaning and HTML elements, then splits overlong passages by window.
Overlap Tokens	The number of tokens from the end of one chunk repeated at the start of the next.
DLO and DMO	Data lake object and data model object; DLO fields map to DMO fields, and their data types must match.

Chunk, index, retrieve

Grounding runs through chunks, an index and a retriever. The exam expects you to ground agents and prompt templates with these three ideas. A retriever returns relevant data to add to a prompt.

Chunking strategies

Section-aware chunking is the default for HTML and PDF. That default applies in the advanced search index builder. Other strategies suit other content, as the table shows.

Data 360 chunking strategies compared

Strategy	How it splits content	Typical use
Section-aware	At titles and headings, until the next heading	Default for HTML and PDF files
Passage extraction	By meaning and HTML elements, then by window if too long	Documents with HTML structure
Conversation-based	New chunk past the Max Tokens or Max Turns limit	Transcripts, audio and video files
Custom function	Your own deployed chunking logic	Selected in the advanced setup

EXAM TRAP

Conversation-based chunking does not work on unstructured data model objects. Transcripts are uploaded as CSV files into DMOs instead.

How chunk boundaries are set

Grouped paragraphs stay whole when the token limit is reached. Data 360 can group paragraphs into one chunk within the token limit. At the limit, the chunk extends to the paragraph's end. Overlap Tokens repeat the end of one chunk at the start of the next. With passage extraction on HTML documents, HTML tags

are stripped by default; you can turn that off.

> **EXAM TRAP**
> A grouped chunk does not stop mid-paragraph at the token limit. Data 360 extends it to the end of the paragraph instead of splitting it.

Getting data into Data 360

Data is either ingested and stored, or connected and queried. Ingested data arrives as streaming or batch data; near-real-time web engagement is streaming. Zero copy federation queries physically separate systems as one logical system without duplicating data. Federation does not support array, struct or other nested types.

Harmonizing and unifying data

Data is mapped to a standard model and unified by rules. Identity resolution rulesets merge separate Individual profiles into one Unified Individual. When mapping a DLO to a DMO, field types must match. The one exception is a Number DLO field mapped to a Text DMO field.

> **EXAM TRAP**
> Plan DLO field types before you create the object. Once a DLO exists, its field data types cannot be changed.

Sharing and controlling data

Activation sends segments out; access policies govern who sees data. Activation publishes a segment to targets such as an advertising platform. Data Shares send data to external platforms like Snowflake. By default, an Allow All policy grants access to every object in a user's data spaces.

Key takeaways

- A retriever returns relevant data to augment a prompt.
- Section-aware chunking is the default for HTML and PDF in the advanced search index builder.
- The default retriever cannot be customized; build custom retrievers in AI Models.
- Overlap Tokens repeat the end of one chunk at the start of the next to preserve context.
- DLO and DMO field types must match, except Number to Text.

Sources: About Salesforce Data 360 | Salesforce Help (opens in a new tab) · Salesforce Certified Agentforce Specialist Exam Guide | Salesforce Help (opens in a new tab) · Chunking Strategies | Salesforce Help (opens in a new tab) · Data Types in Data 360 | Salesforce Help (opens in a new tab) · Data Flow | Salesforce Help (opens in a new tab) · Data Sources in Data 360 | Salesforce Help (opens in a new tab) · Associated Terms for Omnichannel Journeys Using Data 360 | Salesforce Help (opens in a new tab) · Associated Terms for Data Foundation | Salesforce Help (opens in a new tab)

Domain 3 | 35% of the exam | 26 min read

AI Agents

Overview

This domain is about 35% of the exam, roughly 21 of its 60 questions.

Every point below comes from Salesforce's official documentation; each section lists its sources.

Explain how an agent works and its basic building blocks of agent script.

Start here

THINK FIRST

How do subagents, actions and Agent Script work together so an agent behaves both flexibly and predictably?

Why it matters: Questions describe an agent's behavior or a build goal and ask which building block or script element controls it.

An agent is built from subagents, each a job to be done with its own actions and instructions. Agent Script adds rules that always run the same way. Natural language guides the flexible parts.

Key terms

Term	Meaning
Subagent	A category of actions for one job to be done, with instructions; formerly called a topic.
Agent Router	A special subagent that uses each subagent's name and description to pick which one to use.
Reasoning actions	The actions and utilities the LLM may choose from while it reasons.
Transition	A utility or script step that moves the conversation to another subagent.
Resource	A reference to a subagent, action or variable that resolves like a merge field in instructions.

Subagents organize the work

Each subagent covers one job to be done. Its parts are a name, a description, reasoning instructions, reasoning actions and agent actions. Since **April 2026**, topics are called subagents, with no change in how

they work.

Routing to a subagent

The Agent Router picks a subagent by name and description. That is the default, and you can customize routing. Filters add conditions, so a subagent or action is only available when they are met.

EXAM TRAP

A vague subagent description is not harmless. By default, the name and description are what the Agent Router uses to choose it.

Script logic and natural language

Agent Script fixes the rules; prompts guide the conversation. Expressions set if/else conditions, variables and transitions. Before reasoning starts, each condition is evaluated against variable values, so only matching instructions reach the prompt. Branches are checked in order, and the first match wins.

Instructions versus reasoning actions

Where you place an action decides when it runs. Put it in reasoning instructions to run it deterministically, before reasoning. Put it in reasoning actions to let the LLM choose it. Follow-up actions are steps the agent takes after it runs an action during reasoning.

An action placed in reasoning instructions or in reasoning actions

Feature	Reasoning instructions	Reasoning actions
When the action runs	Deterministically, before any reasoning	When the LLM selects it during reasoning
Inputs and outputs	You set the variables manually	Follow-up actions (steps after it runs) can save outputs to variables
Where a new action goes	Not added by default	Added by default, with its action definition

EXAM TRAP

Setting a variable after a handoff transition fails. Anything in the script after a transition is skipped.

Actions get things done

Use standard actions for common tasks, custom ones for your own. Reasoning actions also include utilities: transitions, setting variables and escalation to a representative. Follow-up actions can save outputs, switch subagents or run another action. Escalation needs a Messaging or Enhanced Chat v2 connection.

Canvas view and Script view

Action definitions show in Script view, not in Canvas view. In Canvas view, type @ to insert resources. Errors made in Script view can block Canvas view until you fix them. You can commit an agent with warnings, but not with errors.

Orchestrating several agents

Orchestrators hand off by default, or can supervise. In the start_agent agent_router section, connected agents act as tools you can call with conditions. A chained agent gets context only through its inputs block.

Handoff mode and supervisor mode compared

Feature	Handoff mode	Supervisor mode
Default?	Yes	No
Who talks to the user	The connected subagent, after the handoff	Only the orchestrator agent
Orchestrator's role	Passes control to the subagent	Delegates, gathers and combines responses

Key takeaways

- Subagents (formerly topics) group the actions and instructions for one job.
- Conditions in Agent Script are evaluated before reasoning, so only matching instructions reach the LLM.
- Reasoning instructions run actions deterministically; reasoning actions let the LLM choose.
- Script after a transition is skipped.
- Handoff is the default orchestration mode; supervisor mode keeps the orchestrator in charge.

Sources: The Building Blocks of Agents | Salesforce Help (opens in a new tab) · Subagents | Salesforce Help (opens in a new tab) · Building Agents in Canvas View | Salesforce Help (opens in a new tab) · Agent Script | Salesforce Help (opens in a new tab) · Example: More Reliable Agent Instructions with Agent Script | Salesforce Help (opens in a new tab) · Agent Script in Multi-Agent Solutions | Salesforce Help (opens in a new tab)

Explain the components and benefits of hybrid reasoning, including how Agent Script functions in Canvas and Script View.

Start here

THINK FIRST

Which parts of an agent's behavior should be left to the LLM, and which should be fixed in script, and why does mixing them help?

Why it matters: Questions describe an agent's workflow and ask what hybrid reasoning, Agent Script, Canvas view or Script view does in it.

Hybrid reasoning mixes programmatic expressions with natural language instructions. The Atlas engine runs the deterministic parts first, then hands the LLM only the instructions that apply. You write it all in Agent Script, through Canvas view or Script view.

Key terms

Term	Meaning
Hybrid reasoning	Combining programmatic expressions with natural language, for both control and LLM flexibility.
Agent Script	The language for building agents in Agentforce Builder, in any view.
Atlas reasoning engine	A graph-based engine that works like a flowchart of nodes, variables and transitions.
Canvas view	A view that summarizes Agent Script into blocks you can expand.
Script view	A view for writing Agent Script directly, with highlighting, autocompletion and validation.

Why hybrid reasoning exists

Natural-language-only instructions grew large and inconsistent. In the legacy builder, big prompts could exceed LLM context windows and were often applied unevenly. The whole prompt went to the LLM every turn. Hybrid reasoning keeps LLM flexibility but adds enterprise predictability.

How Atlas keeps paths predictable

Atlas separates the big-picture workflow from conversational skills. It works like a flowchart, so agents follow specific, predictable paths. Natural language handles conversation, while programmatic expressions handle business rules.

Deterministic first, then the LLM

Reasoning instructions resolve in order, with no LLM at that stage. Conditionals are evaluated from variable values before reasoning begins. Only the instructions that apply go into the prompt. That improves instruction adherence and reduces latency.

EXAM TRAP

The LLM does not read every instruction and pick the relevant ones. Conditionals are resolved first, so it receives only the applicable instructions.

Variables hold state reliably

Variables store the agent's state instead of LLM context memory. Their values can feed action inputs, conditionals and filters downstream. Actions placed early in reasoning instructions run before any LLM reasoning, so their outputs can hydrate the prompt.

Transitions between subagents

Switch subagents deterministically, or let the LLM decide. Each subagent in Agent Script has its own scope and reasoning instructions. You can expose a transition to the LLM as an action. Any script after a transition is skipped, so set variables before it. The starting subagent uses the `start_agent` prefix; by

default it is the Agent Router.

> **EXAM TRAP**
> Setting a variable after a handoff transition does nothing. Script after the transition is skipped, so set it first.

Checks after an action

After an action runs, the agent checks the result in a reasoning loop. It then runs another action, asks the user more or replies. This continues for up to **seven** reasoning loops. A Service agent also validates its reply for grounding and scope before sending it.

Canvas view and Script view

Both views edit the same Agent Script and stay in sync. Canvas fits builders who prefer blocks; Script view fits advanced users who write code directly.

The two editing views compared

Feature	Canvas view	Script view
How script appears	Summarized blocks you can expand	The full script, edited directly
Editing aids	Type / for expressions, @ for resources	Syntax highlighting, autocompletion, validation
Switching	May be unavailable after some Script view edits	Fix errors here before returning to Canvas

Key takeaways

- Hybrid reasoning pairs programmatic control with LLM flexibility through Agent Script.
- Conditionals resolve before reasoning, so the LLM gets only the instructions that apply.
- Store state in variables, not LLM memory, so later steps can use it deterministically.
- Edits stay consistent across Canvas and Script view, but Script errors can block Canvas.

Sources: How Agents Work | Salesforce Help (opens in a new tab) · Example: More Reliable Agent Instructions with Agent Script | Salesforce Help (opens in a new tab) · Agent Script | Salesforce Help (opens in a new tab) · Agent Script in Multi-Agent Solutions | Salesforce Help (opens in a new tab) · Building Agents in Canvas View | Salesforce Help (opens in a new tab)

Given a use case, manage deterministic behavior for the agent using mechanisms like filters, variables, and template expressions.

Start here

THINK FIRST

When must an agent behave the same way every time, and which tools make that happen instead of relying on instructions?

Why it matters: Questions describe a rule an agent must always follow and ask which filter, variable or expression enforces it.

Instructions guide the LLM, but filters are deterministic. Variables hold facts about the session, and filters check those facts before a subagent or action can run. Template expressions put variable values into instructions and responses.

Key terms

Term	Meaning
Filter	A rule that blocks a subagent or action unless a variable meets its conditions.
Context variable	A variable mapped to an object field, prefixed with `$Context`, and typically read-only.
Custom conversation variable	A variable that stores an action's output, for use in a filter or as action input.
System variable	A predefined, read-only variable that holds a value only during the session.
Template expression	A merge syntax such as `{!@variables.userName}` that inserts a variable's value.

Filters versus instructions

Filters are deterministic; subagent and action instructions are not. A filter is applied the same way in every conversation. So when a rule must never be skipped, put it in a filter rather than in instructions.

EXAM TRAP

Writing "only do this for verified users" in instructions does not guarantee it. Only a filter is applied consistently across conversations.

How a filter is built

Each condition names a variable, an operator and a value. You can add several conditions and require any or all of them. Conditions can use context, conversation and custom variables. A filter on a subagent or action applies in that agent only.

Filters for authentication

On customer channels, use filters to require authentication. Salesforce recommends this for subagents and actions that act on a user's behalf. In Agent Script, an `available when` clause does the same job, such as `available when @variables.isVerified == False`.

EXAM TRAP

Do not authenticate with context variables. Salesforce does not recommend it; use a filter on a verification variable instead.

Three kinds of variables

Context, custom and system variables differ in source and editability. Pick the variable kind by where its value comes from and whether the agent may change it.

Variable kinds compared

Feature	Context variable	Custom conversation variable	System variable
Value comes from	An object field, such as a Messaging Session field	An action's output	Predefined by the platform
Can it change?	Typically read-only; End User Language is editable	Yes, by mapping action outputs to it	Read-only

Setting and clearing custom variables

Map an action output to store a value; map an empty one to clear it. The variable's data type follows the action output's type: Boolean, Number or Text. In Agent Script, a `set` statement stores an output, such as `set @variables.order_id=@outputs.order_id`.

Locking action inputs

An assigned variable is the only value the agent can use for that input. This removes guesswork: the LLM cannot invent the input. Context and custom conversation variables can both be assigned this way.

Protecting sensitive variables

Keep verification variables hidden from the LLM and from APIs. For a variable used for security, consider leaving Allow LLM to use value and Allow value to be set by API unselected. By default, custom variables are internal only; making one external lets the Agent API set it.

Template expressions in instructions

Expressions insert a variable's value into instructions. Use `{!$Context.VariableAPIName}` for a context variable and `{!VariableAPIName}` for a conversation variable. In Agent Script, write `{!@variables.userName}`. You can also branch on values, such as checking `@system_variables.current_modality` for text or voice.

Key takeaways

- Use a filter, not instructions, for any rule that must hold in every conversation.
- A filter condition is a variable, an operator and a value; require any or all conditions.
- To clear a custom variable, map an empty action output to it.

- Context variable use cases can't be tested in Agentforce Builder; test them on a connected channel.

Sources: Example: Improve Your Agent's Memory with Filters and Variables | Salesforce Help (opens in a new tab) · Variables in the Legacy Builder | Salesforce Help (opens in a new tab) · Variables | Salesforce Help (opens in a new tab) · Create a Filter to Control Access to Subagents and Actions in the Legacy Builder | Salesforce Help (opens in a new tab) · Use Context Variables in Messaging Conversations | Salesforce Help (opens in a new tab) · Create a Custom Variable for Your Agent in the Legacy Builder | Salesforce Help (opens in a new tab)

Given a scenario, select and configure standard topics, custom topics, standard Agent actions, and custom Agent actions.

Start here

THINK FIRST

When do you reuse a standard subagent or action, and when do you build a custom one for your agent?

Why it matters: Questions describe an agent's job and ask which subagent or action to add, and which setting makes it behave correctly.

Topics are now called subagents, and they work the same way. You can add standard subagents from the asset library or create custom ones. Custom actions can call flows, Apex, prompt templates and other functions.

Key terms

Term	Meaning
Subagent	The new name for an agent topic since April **2026**, with no change in function.
Asset library	Where standard and custom subagents live for adding to agents.
Reference action	The flow, prompt template or other function a custom agent action is built on.
Reasoning actions	Actions the LLM may choose during reasoning, where you add filters and follow-ups.

Topics are now subagents

Since April 2026, agent topics are called subagents. Nothing about how they work changed. If a question says topic or subagent, treat them as the same thing.

Standard versus custom subagents

Reuse standard subagents; create custom ones for your own needs. Where you create a custom subagent decides who can reuse it.

Where a subagent comes from, and what follows

Feature	Standard subagent	Custom in asset library	Custom inside an agent
How you add it	New, then Add from Asset Library	Created in the asset library	New, then New Subagent in Agentforce Builder
Who can use it	Any agent you add it to	Multiple agents and versions	Only that agent

Copies do not sync

An agent gets its own copy of each subagent and action you add. Later changes in the asset library don't reach that copy. To apply them, add the updated subagent to the agent again.

EXAM TRAP

Editing a template subagent in the asset library does not update agents that already use it. You must add the updated subagent again.

How a subagent gets picked

The Agent Router uses a subagent's name and description by default. So the description should say which user requests belong to it. A custom subagent also needs at least one instruction.

Building a custom action

Create custom actions from Agentforce Assets in Setup. Pick a reference action type, such as a flow or prompt template. Custom actions can also call Apex, external services, MuleSoft APIs or retrievers (beta). Then assign the action to a subagent to add it to an agent.

Where an action sits in a subagent

Reasoning instructions run actions first; reasoning actions let the LLM choose. Put an action in reasoning instructions to run it deterministically before any reasoning. An action you add to a subagent goes into reasoning actions automatically.

Input and output settings

Action settings control inputs, outputs and confirmation. Require input and Collect data from user govern inputs. Show in conversation decides whether outputs appear in the response. Require user confirmation guards record changes.

Common agent action settings

Setting	What it does	Note
Require input	The action can't run if the input is missing	Read-only when the reference action requires it
Collect data from user	The user must provide this parameter	Set per parameter
Show in conversation	Includes the output in the response	Off for all outputs: the LLM summarizes them

Setting	What it does	Note
Require user confirmation	Asks the user to confirm a record change first	For actions that change records
Filter from agent action	Excludes a reference action parameter	At least one output must be used

Key takeaways

- Custom subagents made in the asset library are reusable; ones made inside an agent are not.
- Asset library changes never sync to an agent's copy; add the updated subagent again.
- A flow-based action follows the flow's permissions, field-level security and sharing.
- You can change an agent only in a draft state; create a new draft of a committed agent.

Sources: Create a Custom Subagent | Salesforce Help (opens in a new tab) · Add a Subagent from the Asset Library in the Legacy Builder | Salesforce Help (opens in a new tab) · Create a Custom Action | Salesforce Help (opens in a new tab) · Create a Custom Agent Action in the Legacy Builder | Salesforce Help (opens in a new tab) · Add an Action to a Subagent from the Asset Library | Salesforce Help (opens in a new tab) · Create a Custom Subagent in the Legacy Builder | Salesforce Help (opens in a new tab) · Subagents | Salesforce Help (opens in a new tab) · Manage Subagents | Salesforce Help (opens in a new tab)

Explain the process for connecting agents to various channels such as digital experience, email, voice, and Slack.

Start here

THINK FIRST

When you connect an agent to a channel, what changes for every version of that agent, and what stays separate?

Why it matters: Questions describe a channel change or channel experience and ask how it affects the agent and its versions.

For a Help Agent, channels belong to the agent, not to a version. So channel changes hit every version at once, while grounding and detail changes create a new version.

Key terms

Term	Meaning
Agent version	A version of a Help Agent. Only one is active at a time, and editing grounding sources or agent details creates a new one.
Escalation flow	A flow assigned to a channel; like channels, it is shared by all versions.
Welcome recommendations	Suggested prompts that appear when the Agentforce panel opens from most supported pages.

Term	Meaning
In-conversation recommendations	Suggestions that show up mid-conversation, prompted by what the user types.

Channels belong to the agent

A Help Agent's channels are assigned to the agent, not a version. Change channels on any version, and the update applies to all of them. Adding or removing a messaging channel takes effect immediately, even on the active version.

EXAM TRAP

Removing a messaging channel from a draft version does not protect the live one. Channel changes apply at once to every version, including the active one.

What creates a new version

Channel edits are shared, but grounding edits make a new version. Updating grounding sources or agent details creates a new version. The active, customer-facing version stays as it was.

How Help Agent changes behave

Change	Effect on versions	Reaches customers
Add or remove a messaging channel	Applies to all versions	Immediately, on the active version
Escalation flow on a channel	Shared by all versions	The same for every version
Grounding sources or agent details	Creates a new version	Not until that version is active

Different behavior per channel

Use separate agents, not versions, for different channel behavior. All versions share the same channels, so versions can't A/B test configurations across channels. Besides, just one version can be live at once.

EXAM TRAP

Two versions of one Help Agent cannot serve two channels differently. Create separate agents instead.

Monitoring the voice channel

Supervisors watch voice-enabled agents in Omni Supervisor. They read call transcripts in its Agentforce tab. In a supervisor configuration, confirm that tab and the Service agents are visible. Agent Optimization adds call recordings, quality scores and knowledge gaps.

Employee agent panel experience

Recommendations let employees start with a prompt, not a blank box. Welcome recommendations appear when the Agentforce panel opens. In-conversation recommendations appear as users chat, and they need Data Cloud. Separate settings control each kind.

Key takeaways

- Help Agent channels and escalation flows are shared by every version of the agent.
- Changing grounding sources or agent details creates a new version and leaves the live one alone.
- For different behavior on different channels, create separate agents.
- Welcome recommendations work without Data Cloud; in-conversation ones need it.

Sources: Help Agent Versions and Channels | Salesforce Help (opens in a new tab) · Manage Conversation Recommendations in the Legacy Builder | Salesforce Help (opens in a new tab) · Monitor Your Voice-Enabled Agent | Salesforce Help (opens in a new tab)

Explain the security context in which the agent is actually running, and how it impacts agent action execution.

Start here

THINK FIRST

Whose permissions does an agent use when it runs an action, and how does that change between employee and customer channels?

Why it matters: Questions describe an agent, its channel and an action that fails or overreaches, and ask which user context or permission explains it.

An agent runs either as the logged-in user or as a dedicated agent user. Logged-in channels apply that user's own access. Open channels such as Messaging use the agent user, and Salesforce recommends filters and variables there to limit record access.

Key terms

Term	Meaning
Agent user	A Salesforce user record holding every permission the agent needs on open channels.
Private action	An action the triggering user must be authorized for, with identity securely confirmed.
MessagingSession	The record of a messaging conversation, which can store a customer's confirmed identity.
AuthSession	An authenticated portal session that, once linked to the MessagingSession, confirms who the user is.

Two security contexts

The channel decides whose access the agent uses. Channels limited to logged-in users run as that user. Open channels run as the agent user. The table compares the two.

Logged-in user context and agent user context compared

Feature	Logged-in user context	Agent user context
Typical channel	Lightning Experience	Messaging channels, such as Enhanced Chat
Where you assign permissions	Individual users or user profiles (Employee agent)	The agent's user record (Service agent)
Limits on subagents and actions	Applied by default from the end user's context	Need extra setup: filters and variables
Org-wide sharing defaults	Depend on whether the user is internal or external	Internal user defaults apply

Flows that call an agent

The agent type sets the context for a calling flow. A flow calling an Employee agent or Agentforce (Default) runs as the authenticated end user. A flow calling a Service agent runs as the agent user. Grant that agent user record access with a permission set, such as Read on Case.

Permissions actions need

Flow actions need Run Flows plus object access. The user or the agent user needs both at minimum. A flow action also follows the flow's own permissions, field-level security and sharing. Prompt templates in actions need the Prompt Template User permission set.

> **EXAM TRAP**
> Do not give the agent user or end users permissions to create or manage agents. Neither one should hold them.

Who can talk to an agent

Service agents need no special user permissions. Employee agents are different: each user must be given access to the individual Employee agent before they can use it.

Confirming identity for private actions

Confirm who the user is before a private action runs. Save the confirmed identity on the MessagingSession so the agent can check it. In the secure example, an AuthSessionId field links the session to an AuthSession. The AuthSession can end when the portal conversation ends.

EXAM TRAP

Agent context variables are not a safe way to authenticate. Salesforce does not recommend using them for authentication.

Authenticating on an Experience site

Reauthenticate with a parameterized URL to a secure portal method. The Apex method takes MessagingSession.SessionKey, a GUID unique to active sessions. An AuthRequestTime field limits how long external authentication is allowed.

EXAM TRAP

Logging in on the portal does not resume the chat by itself. No event reaches the agent session, so the user must keep the conversation going.

Limits to watch

Some contexts block custom agent invocable actions. They are not supported for Experience Cloud site users or the Platform Integration User. On a custom site with its own login, no AuthSession is tracked, so authentication lasts until the token expires.

Key takeaways

- Logged-in channels run as the end user; open channels run as the agent user.
- On agent-user channels, limit record access with filters and variables.
- A flow calling a Service agent runs as the agent user, not the end user.
- Flow actions need Run Flows plus access to the objects the flow uses.
- Confirm identity before a private action; context variables are not recommended for authentication.

Sources: Common User Access for Standard Agent Actions | Salesforce Help (opens in a new tab) · Maintain Trust with Agentforce Actions in the Legacy Builder | Salesforce Help (opens in a new tab) · Example Implementation of a Secure Agentforce Service Agent | Salesforce Help (opens in a new tab) · Call an Agent from a Flow or Apex Class | Salesforce Help (opens in a new tab) · Create a Custom Action | Salesforce Help (opens in a new tab) · Use Context Variables in Messaging Conversations | Salesforce Help (opens in a new tab) · Add User Identification to Agentforce Actions | Salesforce Help (opens in a new tab)

Given a scenario, identify when to use an Employee or Service agent.

Start here

THINK FIRST

Given a request for an AI agent, how do you tell whether it should be an Employee agent or a Service agent?

Why it matters: Questions describe who the agent serves and where, and ask whether an Employee or a Service agent fits.

Start with the audience. Employee agents serve internal staff and act as the logged-in user. Service agents serve customers on messaging and other channels, act as a dedicated agent user and escalate to reps when needed.

Key terms

Term	Meaning
Employee agent	An agent for internal employees that runs in the logged-in user's context.
Service agent	An agent that supports customers on messaging and other channels and escalates to reps.
Agent user	The Salesforce integration user a Service agent runs as, holding the permissions it needs.
Inbound conversation	A conversation the user starts; the only kind Agentforce Service Agent supports.
Omni-Channel flow	A flow that routes messaging work. An inbound one sends conversations to a Service agent; an outbound one carries transfers.

Audience decides the type

Internal staff point to Employee; customers point to Service. Employee agents help staff look up information, finish tasks and get personalized support. Service agents resolve common customer support cases and requests, and can hand off to a rep.

Employee and Service agents compared

Feature	Employee agent	Service agent
Who it serves	Internal employees	Customers
Runs as	The logged-in user	An agent user (integration user)
Typical channels	Agentforce panel in Lightning, Slack, Salesforce mobile app	Messaging and other customer channels, including Bring Your Own Channel (BYOC)
Usage metering	Metered or unmetered, by the user's permissions	Metered by action or by resolution

Whose permissions apply

An Employee agent acts with the logged-in user's access. A Service agent acts as its agent user instead. Give that user only what the job needs: to update a case on email, it needs View and Update case permissions.

EXAM TRAP

A Service agent does not borrow the customer's or a rep's access. It runs as its own agent user, so grant permissions to that user record.

Channels each type supports

Employee agents live in employee channels; Service agents in customer ones. Employee agents appear in the Agentforce panel, Slack and the mobile app. Enhanced messaging channels, including Bring Your Own Channel, route to Service agents, not Employee agents.

EXAM TRAP

Routing a customer messaging channel to an Employee agent is wrong. Enhanced messaging channels, including BYOC, route only to Service agents.

Service agent conversation rules

Service agents handle only conversations the user starts. A session closes **72** hours after it starts, or when the user ends it, whatever the channel timeout. To transfer a messaging conversation, it hands off through the Escalation subagent plus an outbound Omni-Channel flow.

Building and managing each type

Each type needs its own build permissions. Employee agents need Manage AI Agents or Customize Application. Service agents need Manage Agentforce Service Agents plus Manage AI Agents, or Customize Application. With several Employee agents, describe each one's purpose to avoid overlapping subagents.

Key takeaways

- Internal users in Lightning, Slack or mobile point to an Employee agent.
- Customers on messaging channels, with escalation to reps, point to a Service agent.
- Service agents run as an agent user; Employee agents run as the logged-in user.
- A Service agent session closes **72** hours after it starts, whatever the channel timeout.

Sources: Create an Agent from an Agentforce Employee Agent Template | Salesforce Help (opens in a new tab) · Create an Agent from an Agentforce Service Agent Template | Salesforce Help (opens in a new tab) · Considerations for Agentforce Service Agent | Salesforce Help (opens in a new tab) · Connect a Service Agent to Enhanced Messaging Channels | Salesforce Help (opens in a new tab) · Considerations for Agentforce Employee Agent | Salesforce Help (opens in a new tab) · Considerations for Agentforce Service Agent on Email | Salesforce Help (opens in a new tab)

Given a scenario, identify when it's appropriate to use Agent API.

Start here

THINK FIRST

When is a standard channel enough for an agent, and when do you need to reach it programmatically through an API or code instead?

Why it matters: Questions describe where and how an agent must run and ask whether a standard channel or programmatic access, such as Agent API, fits.

First check whether a standard channel and connection already cover the need. They give the agent channel context and ready-made response formats. When the agent must run from a flow or Apex class, in the background or without Builder, use the programmatic routes instead.

Key terms

Term	Meaning
Connection	A standard link that gives an agent context about specific channels.
Adaptive response format	A response format included with a standard connection, built for specific channels or apps.
Invocable action	Lets a flow or Apex class call a Service agent, an Employee agent or Agentforce (Default).
Headless development	Building, deploying and running agents without Agentforce Builder or Setup.
MCP	Links agents to outside services: Agentforce's built-in MCP client uses tools from registered MCP servers, with no hand-built integration.

Standard channels come first

Agents deploy to a set of standard channels. Standard channels include the Agentforce panel, the mobile app, Slack, messaging platforms and email. Support varies by agent type. For Employee agents, deploying to Lightning Experience and mobile just means activating the agent.

Rich responses on channels

Adaptive response formats come with standard connections. Rich choice, rich link and time picker responses are for Service agents only. On Enhanced Chat v2, custom Lightning Web Components take the place of these formats.

Calling an agent from code

Use flows or Apex when no person starts the conversation. Invocable actions let a flow or Apex class call an agent. So an agent can handle background or event-driven tasks wherever a flow or Apex can run. For deeper cases, Salesforce points to its API and Developer guides.

EXAM TRAP

A background, event-driven task does not need a chat channel. Call the agent from a flow or Apex class instead.

Working without Builder

Headless development skips Agentforce Builder and Setup. Agents are still built, deployed and operated, just without either of those two tools.

Agents reaching other systems

MCP replaces hand-built integrations to outside services. Without it, you handle API connectivity, authentication and data access yourself. Agentforce has a built-in MCP client. Gateway policies protect connections to external APIs and MCP servers.

EXAM TRAP

Hand-coding API connectivity to a third-party service is not the only option. A registered MCP server gives agents its tools through the built-in client.

Key takeaways

- Standard connections give an agent context about specific channels, and channel support varies by agent type.
- Background or event-driven agent work runs from a flow or Apex class.
- Headless development builds and runs agents without Agentforce Builder or Setup.
- Salesforce recommends a separate Service agent for Telephony and for Service Email.

Sources: Deploy Your Agent to Channels | Salesforce Help (opens in a new tab) · Standard Agent Connection Reference | Salesforce Help (opens in a new tab) · Standard Adaptive Response Format Reference | Salesforce Help (opens in a new tab) · Extend Your Agentforce Implementation | Salesforce Help (opens in a new tab) · MCP for Agentforce | Salesforce Help (opens in a new tab)

Domain 4 | 10% of the exam | 12 min read

Testing, Deployment, and Maintenance

Overview

This domain is about 10% of the exam, roughly 6 of its 60 questions.

Every point below comes from Salesforce's official documentation; each section lists its sources.

Given a scenario, test an agent using Testing Center.

Start here

THINK FIRST

How do you plan an agent test in Testing Center so the results are trustworthy and nothing breaks along the way?

Why it matters: Questions describe a testing goal and ask which test type, test case source or Testing Center rule applies.

AI agents are non-deterministic, so you test them across many scenarios. Testing Center builds those scenarios for you and runs them at turn level or conversation level. Run it in a sandbox, because tests can change real CRM data.

Key terms

Term	Meaning
Turn-level test	Checks one agent response on its own, such as after a prompt change.
Conversation-level test	Checks a full multi-turn conversation for whether the user's goal is met.
Test suite	What you run against a chosen agent version; switching that version clears the worksheet results.
Testing Center API	Automates test runs so testing fits your development workflow.

Why broad testing matters

Agents are non-deterministic, so one passing chat proves little. Test across a wide range of scenarios to confirm consistent, accurate answers. Do it in a sandbox, since testing agents can modify CRM data.

EXAM TRAP

Running Testing Center in production is risky. Tests can change CRM data, so use a sandbox.

Turn level or conversation level

Match the test type to what you need to prove. A single response calls for turn level; a whole interaction calls for conversation level.

The two test types compared

Feature	Turn-level	Conversation-level
What it evaluates	One response in isolation	A full multi-turn conversation
Use it to	Validate prompt changes or knowledge retrieval	See if the agent achieves the user's goal

Conversation-level restrictions

Draft versions and Employee agents can't use conversation-level tests. Escalation isn't supported either; a triggered escalation ends the test conversation with a transfer message. In beta, an LLM can generate up to **20** test cases, and a CSV can supply up to **1,000**. Conversations run up to **20** turns, with at most **3** suites running at once.

Where test cases come from

Generate cases, build Q&A from knowledge, or upload your own. Testing Center can draw cases from the agent's subagents and actions. For knowledge tests, link the agent to a data library holding that content. You can also automate runs with the Testing Center API.

Versions, permissions and cost

Only committed agent versions appear in Testing Center. Switching versions clears the worksheet, but run history keeps past results. Creating tests needs Manage Agentforce Grids and Manage Agentforce Testing. Since Summer **'26**, Testing Center runs are unmetered, but the same suite run in Agentforce Grid consumes credits.

EXAM TRAP

Unmetered testing covers runs from Testing Center only. Running the suite in Agentforce Grid consumes Einstein Requests or Flex Credits.

Key takeaways

- Use turn-level tests for one response and conversation-level tests for a whole interaction.
- Commit an agent version before you look for it in Testing Center.
- Escalation isn't supported in conversation-level tests.
- Data 360 queries from Testing Center are still metered, even though test runs are not.

Sources: Agentforce Testing Center | Salesforce Help (opens in a new tab) · Create a Test Suite | Salesforce Help (opens in a new tab) · Considerations for Testing Center | Salesforce Help (opens in a new tab) · Test Agents | Salesforce Help (opens in a new tab)

Explain how Testing Center evaluations work.

Start here

THINK FIRST

How does Testing Center decide whether an agent's answer passed, and what can you add beyond the defaults?

Why it matters: Questions describe a test result or a testing goal and ask which evaluation or scorer explains or measures it.

Every test runs a set of default scorers, chosen by whether the test is turn-level or conversation-level. In the legacy Testing Center, you can add quality evaluations to the defaults. The response evaluation and custom scorers use an LLM as the judge.

Key terms

Term	Meaning
Scorer	A check that grades the agent's behavior in a test, such as the action or subagent check.
LLM judge	An AI model that compares or assesses responses and assigns a score.
Expected Response	The field holding the reply the agent should give, used to grade its actual reply.
Custom scorer	A scorer whose criteria you write in a prompt template for the AI model to apply.
Turn-level vs conversation-level	The two test levels. Each decides which scorers are available; Task Resolution, for example, is a conversation-level scorer.

Default scorers always run

Default scorers run on every test. Which scorers you get depends on the test level. The subagent and action checks are named differently at each level, so read the name to tell which level a result came from.

The same two checks, named by test level

Check	Turn-level test	Conversation-level test
Subagent check	Subagent Assertion	Subagent Evaluation
Action check	Action Assertion	Actions Evaluation

How response grading works

An LLM judge compares the expected and actual reply. The judge works out the expected response, compares it with the reply the agent recorded, and assigns a score. In conversation-level tests, Task

Resolution returns pass or fail on whether the user's issue was resolved.

> **EXAM TRAP**
> Writing instructions such as "answer politely with the order date" in Expected Response produces inaccurate scores. Write the actual reply the agent should give.

Action checks

The action check wants the right actions and all of them. It confirms that the agent picks the correct actions for the utterance. It also confirms that none of the expected actions is missing.

Quality evaluations in legacy Testing Center

Three evaluations are defaults; quality evaluations are added. Legacy Testing Center runs response accuracy, subagent assertion and action assertion by default. You can add quality evaluations on top of them.

Legacy Testing Center evaluations and what each measures

Evaluation	What it measures	Default?
Completeness	All important information from the utterance is covered	No, added
Coherence	Grammatical, conversational language, not raw data like JSON	No, added
Instruction adherence	Subagent instructions understood and fully followed	No, added
Latency	Test execution time, in milliseconds	No, added

Custom scorers

Custom scorers put your criteria in a prompt template. The AI model applies that template to assess each session. A Labels scorer keeps outputs within a predefined set of approved values. For free-form output, pick Boolean, DateTime, Number, Text or URL as the data type.

> **EXAM TRAP**
> A failed run is not left blank. When the model cannot decide or the prompt fails, the scorer applies its Fallback Label.

Key takeaways

- Default scorers always run; the test level decides which ones are available.
- Subagent and action checks are Assertions at turn level and Evaluations at conversation level.
- Expected Response holds the reply itself, never instructions about the reply.
- Custom scorers are LLM-as-judge checks defined in a prompt template, with a Fallback Label.

Sources: Default Scorers | Salesforce Help (opens in a new tab) · Choose Evaluations | Salesforce Help (opens

in a new tab) · Create Custom Scorers | Salesforce Help (opens in a new tab)

Identify the considerations for deploying an agent from sandbox to production.

Start here

THINK FIRST

What carries over automatically when you move an agent and its Data 360 setup from sandbox to production, and what must you redo?

Why it matters: Questions describe a sandbox-to-production move and ask which step is missing or what will not carry over.

A Data 360 sandbox holds metadata but no data. Library files and connection authorization do not copy over either. Data 360 changes move to production in a DevOps data kit inside a change set, but data library changes cannot be deployed.

Key terms

Term	Meaning
Data kit	The DevOps tool that, with a change set, deploys Data 360 changes to production; its components deploy in sequence.
Change set	The package that carries changes into the target org over a deployment connection.
Deployment connection	A connection set up in the target org so it can accept inbound change sets.
Data library	Agentforce Data Library, which can be file-based and works in every sandbox type.
Data space	A Data 360 space whose components a data kit can deploy; the data kit does not create the space itself.

What a sandbox contains

A Data 360 sandbox holds metadata only, no data. This is true even in Partial Copy and Full Copy sandboxes. Connections are copied, but their authorization data is not. You must configure each connection again in the sandbox.

EXAM TRAP

A Full Copy sandbox still has no Data 360 data. Only metadata is replicated when Data 360 is provisioned there.

Moving Data 360 changes

Use a DevOps data kit inside a change set. Create the deployment connection in the production org, because it is the target that accepts inbound change sets. Before deploying, create any sandbox data space in production by hand. The data kit does not create it.

EXAM TRAP

The deployment connection is not set up in the sandbox. It belongs in the production org that receives the change set.

When a data kit deploy goes wrong

A failed component stops every component after it. Components publish in sequence, so the rest of the sequence is skipped. If the same data kit arrives in another change set, only the kit installs. Open it and click Deploy Data Kit to deploy its components.

Data libraries in a sandbox

Data libraries work in every sandbox type, but files do not copy. Re-upload the files to each file-based data library. Each such library uses a retriever, which you manage in AI Models. Deactivate and reactivate that retriever so it works in the sandbox. Library changes made in a sandbox cannot be deployed to production or another sandbox.

EXAM TRAP

Do not plan to build data library changes in a sandbox and deploy them. That path is not supported, whether the target is production or another sandbox.

Agent versions and assets

Only committed versions of new-builder agents are listed. Uncommitted versions do not show in the Setup list view or the legacy builder. Changes to a standard asset do not update agents already built from it. Recreate the agent, or re-add the asset and redo your customizations.

Voice agents and trusted URLs

Voice agents need the livekit.cloud URLs in Trusted URLs. Starting in **May 2026**, Salesforce adds these URLs so voice agents can be previewed and deployed.

Key takeaways

- Sandboxes get Data 360 metadata only; re-authorize connections and re-upload library files.
- Deploy Data 360 changes with a data kit in a change set, through a connection made in production.
- Create sandbox data spaces in production manually before deploying the data kit.
- Data library changes cannot be deployed from a sandbox.

Sources: Test Data Libraries in a Sandbox | Salesforce Help (opens in a new tab) · Deploy Data 360 Changes Across Orgs | Salesforce Help (opens in a new tab) · Agentforce Builder Considerations | Salesforce Help

(opens in a new tab)

Identify the considerations for deploying a template from sandbox to production.

Start here

THINK FIRST

How do you move a prompt template and the Data 360 setup behind it from a sandbox into production?

Why it matters: Questions describe a template built in a sandbox and ask how to deploy it or what will not move with it.

Prompt templates move like other metadata: in change sets, packages or with Salesforce CLI, with dependencies in mind. The Data 360 setup they ground on moves separately, in a data kit. Data, files and sandbox-only objects stay behind.

Key terms

Term	Meaning
Change set	A way to deploy prompt templates, limited by how component dependencies behave.
Data kit	A DevOps package that, with a change set, moves Data 360 sandbox changes to production.
Component dependency	Metadata a template relies on, which affects deployment by change set or CLI.

Three ways to deploy a template

Use a change set, a package or Salesforce CLI. A prompt template fits in managed, unmanaged or unlocked packages. With change sets and with the CLI, component dependencies shape what deploys. Check what the template relies on before you deploy.

EXAM TRAP

Packaging is not limited to one type. A prompt template can go in a managed, an unmanaged or an unlocked package.

Setting up Data 360 in a sandbox

Turn on and configure Data 360 in production first. Production needs Data Cloud licenses and an active Data 360 tenant before a sandbox can activate it. Sandbox usage then consumes production credits. Sandboxes follow production limits, so the **200-MB** Developer storage limit does not apply.

What the sandbox holds

A Data 360 sandbox has metadata, not data. All sandbox types are supported, yet even Full Copy sandboxes replicate no data. File-based data libraries also lose their files, so re-upload them in the sandbox.

EXAM TRAP

Choosing a Full Copy sandbox does not bring Data 360 data along. Only metadata is replicated in every sandbox type.

Deploying the Data 360 side

Data 360 changes go to production in a data kit and change set. Objects created directly in the sandbox stay there and never deploy back. Data library changes made in a sandbox cannot be deployed to production or another sandbox either.

EXAM TRAP

Creating a Data 360 object straight in the sandbox does not get it to production. Such objects remain local to the sandbox.

Cloning a configured sandbox

Clone only a sandbox in Active status. A sandbox being refreshed, maintained or deleted cannot be cloned. Cloning a configured sandbox brings over its identity resolution rules, data stream definitions, data model objects and their relationships.

Key takeaways

- Deploy prompt templates by change set, package or Salesforce CLI, minding component dependencies.
- Data 360 must be on and configured in production before a sandbox can use it.
- Sandboxes carry Data 360 metadata only, and file-based libraries need their files re-uploaded.
- Objects created directly in a sandbox and data library changes never deploy back.

Sources: Prompt Builder Limits and Considerations | Salesforce Help (opens in a new tab) · Data 360 in a Sandbox | Salesforce Help (opens in a new tab) · Test Data Libraries in a Sandbox | Salesforce Help (opens in a new tab) · Clone a Data 360 Sandbox | Salesforce Help (opens in a new tab)

Domain 5 | 10% of the exam | 6 min read

Governance and Observability

Overview

This domain is about 10% of the exam, roughly 6 of its 60 questions.

Every point below comes from Salesforce's official documentation; each section lists its sources.

Explain the process for managing and monitoring agents.

Start here

THINK FIRST

Once an agent is live, how do you watch how it performs and decide what to improve next?

Why it matters: Questions describe a monitoring or management need and ask which tool or setting gives the answer.

Monitoring happens in layers. Testing Center catches weak answers before users see them. Omni Supervisor shows live voice calls, while Agent Analytics and Agent Optimization show performance and gaps over time.

Key terms

Term	Meaning
Agent Analytics	Records every step of a conversation and shows agent performance over time.
Agent Optimization	Lets you review call recordings, conversation details and quality scores.
Omni Supervisor	Where supervisors watch voice-enabled agents on active calls, in its Agentforce tab.
Scorer	Automatically evaluates each agent response in Testing Center.
Recommendations	Suggestions an Employee agent shows: welcome ones when its panel opens, and in-conversation ones.

Pick the tool for the question

Each monitoring tool answers a different question. Ask whether you need live oversight, long-term trends or a test before release.

Monitoring tools compared

Tool	What it shows	Use it when
Omni Supervisor	Voice-enabled agents on active calls and their transcripts	You must watch calls as they happen
Agent Analytics	Every step of each conversation, gathered in one place	You need performance over time
Agent Optimization	Call recordings, conversation details, quality scores	You hunt for knowledge gaps
Testing Center	Scored responses across many test scenarios	You validate before users see it

Watching live voice calls

Supervisors read call transcripts in Omni Supervisor's Agentforce tab. That tab lists the agents on active calls. In a supervisor configuration, first confirm the Agentforce tab and the Service agents are visible there.

EXAM TRAP

Don't mix up the two jobs. Omni Supervisor is for voice calls in progress; analytics reports and dashboards track performance over time.

Finding gaps over time

Agent Analytics and Agent Optimization reveal trends and gaps. Agent Analytics gathers every session's details so you can see performance over time. Agent Optimization lets you review recordings and quality scores, and spot where the AI lacked the right answer.

Testing before users see it

Scorers in Testing Center evaluate each response automatically. That shows where the agent falls short before release. Test cases can come from AI, a Data Library, manual uploads or the Testing Center API.

Managing an agent's settings

Manage agents from Agentforce Agents in Setup. Managing Employee agents needs Manage AI Agents or Customize Application. Welcome and in-conversation recommendations have separate settings. In-conversation recommendations need Data Cloud; welcome recommendations don't.

EXAM TRAP

Turning off welcome recommendations does not turn off in-conversation ones. Each has its own setting.

Key takeaways

- Use Omni Supervisor for live voice calls and analytics for trends over time.
- Agent Optimization helps find knowledge gaps where the AI lacked the right answer.
- Welcome recommendations are toggled on the agent's System Messages tab.

- In-conversation recommendations don't appear without Data Cloud.

Sources: Test Agents | Salesforce Help (opens in a new tab) · Monitor Your Voice-Enabled Agent | Salesforce Help (opens in a new tab) · Manage Conversation Recommendations in the Legacy Builder | Salesforce Help (opens in a new tab)

Explain agent analytics and agent optimization.

Start here

THINK FIRST

How do Agent Analytics and Agent Optimizer help you see how an agent performs and decide what to fix?

Why it matters: Questions describe an analytics need, a metric or a setup problem and ask which feature, metric or requirement applies.

Agent Analytics reports on every turn and event of agent sessions, through a unified session tracing data model. It scores sessions on outcomes such as engagement and deflection. Agent Optimizer diagnoses groups of sessions, and you configure it separately.

Key terms

Term	Meaning
STDM	The unified Session Tracing Data Model that Agent Analytics is built on.
Engagement rate	The share of sessions where the customer sent at least one message.
Deflection rate	The share of sessions that ended without an escalation.
Agent Optimizer	A pilot or beta service that diagnoses groups of agent sessions.

Where Agent Analytics lives

Open it in Agentforce Studio under Observe & Optimize. Go to Analytics, then Observe & Optimize, and choose Service Agent Analytics or Employee Agent Analytics. Its reports rest on the Session Tracing Data Model, which captures every turn and event across sessions.

KPI dimensions

Six dimension tabs load the KPI lenses. They are Effectiveness, Usage, Quality, Health, Trust and Voice. Picking a tab loads the KPI lens for that dimension.

Reading session metrics

Each session gets one outcome, checked in order. Once a condition matches, later ones are not evaluated. The table defines the metrics most questions ask about.

Key Agent Analytics metrics and how each is defined

Metric	Definition	Detail to remember
Engagement rate	Share of sessions with one or more customer messages	No customer messages means unengaged
Deflection rate	Share of sessions that ended without escalation	Escalated sessions do not count
Deflected outcome	Customer sent a message and LLM Deflection Score is 4 or 5	Both conditions must hold
Quality score	Rating from an external evaluator	Scale of 1 to 5

EXAM TRAP

A session with no customer messages is not deflected. It counts as unengaged, because deflection also needs a customer message.

Before you install

Provision Data Cloud, turn on Einstein and have an active agent. You also need Standard Data Model version **1.130** or higher. Install with System Administrator plus Data Cloud Admin, or Agentforce Service Agent Configuration plus Data Cloud Architect.

EXAM TRAP

Turning on every optional metric is not free. Feedback and Knowledge/RAG metrics provision extra data model objects that add overhead.

Giving users access

Share the workspaces, then assign the right permission sets. Agentforce 1 users need Tableau Next Platform Analyst and Data Cloud User. Salesforce Foundations users need TabNext Limited Consumer and Data Cloud User. For CRM data, give the Platform Integration User the Data Cloud Salesforce Connector permission set.

Agent Optimizer

Optimizer diagnoses groups of sessions without using credits. A diagnosis samples about **100** sessions by default, up to **1,000** per analysis. Users need two permission sets: Data Cloud User and Access Agentforce Optimization.

EXAM TRAP

Installing an Agent Analytics app does not set up Agent Optimization. Each one must be configured on its own.

Sandboxes and usage

Agent Analytics is available in sandbox orgs. Sandbox usage records are not stored in the sandbox. They are written to your home production org instead. Check consumption in the production org's Digital Wallet dashboard, the source of truth for all usage.

Key takeaways

- Agent Analytics is built on the Session Tracing Data Model and captures every turn and event.
- Deflection rate counts sessions that ended without escalation; engagement needs a customer message.
- Each session gets one outcome, from the first condition that matches.
- Agent Optimizer samples about **100** sessions by default, uses no Agentforce credits and is configured separately.

Sources: About Agent Analytics | Salesforce Help (opens in a new tab) · Get Insights with Agent Analytics | Salesforce Help (opens in a new tab) · Improve Agent Performance with Agent Optimizer in Observability (Beta) | Salesforce Help (opens in a new tab) · Set Up Agent Analytics | Salesforce Help (opens in a new tab)

Domain 6 | 5% of the exam | 6 min read

Multi-Agent Orchestration

Overview

This domain is about 5% of the exam, roughly 3 of its 60 questions.

Every point below comes from Salesforce's official documentation; each section lists its sources.

Given a scenario, determine whether a Multi Agent architecture is appropriate for scalability and control.

Start here

THINK FIRST

When is splitting work across an orchestrator and connected agents worth the extra latency and limits?

Why it matters: Questions describe a business need and ask whether a multi-agent design fits and which routing mode suits it.

A multi-agent solution puts an orchestrator in front of specialized, reusable agents in the same org. It suits work split across business domains. It costs more latency and has firm limits on depth, timeouts and agent types.

Key terms

Term	Meaning
Orchestrator agent	The agent that handles the first conversation, then routes and delegates tasks.
Connected subagent	A complete, independent agent with its own expertise and identity, linked to an orchestrator.
Handoff mode	The default routing pattern: a connected subagent takes over the conversation directly.
Supervisor mode	A routing pattern where the orchestrator combines results from several agents before replying.
ASA and AEA	Short names for two agent types: Agentforce Service Agent (ASA) and Agentforce Employee Agent (AEA).

What multi-agent orchestration does

Specialized Agentforce agents in one org work together. An orchestrator greets the user and delegates. Reusable agents can represent domains such as Sales, Human Resources and Finance, across many solutions.

EXAM TRAP

A connected subagent is not the same as a standard subagent. A connected one is a full agent; a standard one belongs to the agent definition.

Choosing a routing mode

Handoff is the default; supervisor mode synthesizes answers. Pick handoff when a domain expert should own the conversation end to end. Pick supervisor when one reply must combine several agents' results.

Handoff mode and supervisor mode compared

Feature	Handoff mode	Supervisor mode
Default?	Yes	No
Best for	A specialist, such as billing or HR, owning the conversation	Combining results from several agents first
Speed	Faster: skips the orchestrator's synthesis step	Includes the orchestrator's synthesis step

Where it does not fit

Orchestration stays inside a single Salesforce org. It does not work across orgs or with external, non-Agentforce agents. Delegation goes only **one** level deep in this release.

EXAM TRAP

A connected subagent cannot delegate to a further layer of agents. Only one level of delegation is supported.

Supported agent pairings

Only three orchestrator and connected subagent pairings work. ASA stands for Agentforce Service Agent and AEA for Agentforce Employee Agent. The pairings are ASA with ASA, AEA with AEA, and an AEA orchestrator with ASA connected subagents. An ASA orchestrator with AEA subagents is not on the list.

Latency, timeouts and scale

Multi-agent is slower than single-agent by design. Coordination and extra LLM calls add latency. Orchestrators time out at **120** seconds and connected subagents at **30**. No cap on connected subagents is enforced. Internal tests with **8** of them showed no latency impact.

Keeping control

Clear descriptions and one-way variables keep routing predictable. Routing relies on each connected subagent's description, so vague or overlapping ones cause inconsistent behavior. Variables flow only from the orchestrator to connected subagents, never back.

Key takeaways

- Use handoff for specialists who own the conversation; supervisor to combine several answers.
- Orchestration works in one org, with one level of delegation, and only with Agentforce agents.
- Timeouts are **120** seconds for orchestrators and **30** for connected subagents.
- Write distinct subagent descriptions, because routing depends on them.

Sources: Multi-Agent Orchestration | Salesforce Help (opens in a new tab) · Considerations and Limitations in Multi-Agent Solutions | Salesforce Help (opens in a new tab)

Explain the purpose of existing open standard multi-agent protocols such as MCP and A2A.

Start here

THINK FIRST

Why would you connect an agent to an external system through an open protocol instead of building a custom integration?

Why it matters: Questions describe an agent that needs an outside system's tools and ask what MCP or multi-agent orchestration provides.

MCP is an open standard for connecting AI models to external systems and their tools. It replaces hand-built integrations with one standard interface. In Agentforce, connected server tools become agent actions you add like any other.

Key terms

Term	Meaning
Model Context Protocol (MCP)	An open standard that defines how AI models link to outside systems and their tools.
MCP server	Holds what an external system makes available: tools, prompts and resources.
MCP client	Connects to one or more MCP servers to reach an external system's tools and data.
MCP host	The app that coordinates the process and manages the MCP client.
Orchestrator agent	Handles a user's first conversation, then routes and delegates tasks.

The problem MCP solves

Without MCP, every integration is built by hand. You would handle API connectivity, authentication, data access and code yourself. MCP acts as a translator between systems, so that manual work is done for you. Anthropic, which developed it, likens it to a USB-C port for AI.

Servers, clients and hosts

MCP relies on three roles: servers, clients and hosts. The table shows what each role does, with one more detail for each role.

The three MCP roles compared

Role	What it does	More detail
Server	Exposes an external system's tools, prompts and resources	Agentforce connects third-party, MuleSoft or Salesforce-hosted servers
Client	Connects to one or more servers for tools and data	Agentforce has an MCP client built in
Host	Coordinates the process and manages the client	Provides the interface for using server tools and data

Tools become agent actions

Allowlisted server tools appear as agent actions. MCP tools are executable functions an LLM calls to act in external systems. Once you connect a server, the tools you allowlist go into the Agentforce asset library. You then add them to an agent like any other action.

What Agentforce supports

MCP for Agentforce supports server tools only. Server apps, prompts and resources aren't supported. Servers must use the Streamable HTTP transport; HTTP+SSE is unsupported. Connect third-party servers through Agentforce Registry, and MuleSoft or Salesforce-hosted servers through API Catalog.

EXAM TRAP

An MCP server's prompts and resources do not come across to Agentforce. Only its tools are supported.

Multi-agent orchestration

Orchestration gives customers one front door to many agents. Multi-Agent Orchestration provides an interoperability framework and a single front door "superagent". An orchestrator agent handles the first conversation, then routes and delegates tasks to other agents.

Key takeaways

- MCP standardizes how AI models reach external tools, replacing custom integrations.
- An MCP setup has three roles: server, client and host.
- Agentforce supports only MCP server tools over Streamable HTTP.

- Third-party servers connect through Agentforce Registry; MuleSoft and Salesforce-hosted ones through API Catalog.

Sources: MCP for Agentforce | Salesforce Help (opens in a new tab) · Connect Agentforce Solutions to MCP Servers | Salesforce Help (opens in a new tab) · Multi-Agent Orchestration | Salesforce Help (opens in a new tab) · Considerations for MCP for Agentforce | Salesforce Help (opens in a new tab)

MockCert

Take a Salesforce Certified Agentforce Specialist mock exam the way the real one feels

5 timed mock exams · 60 questions in 105 minutes each · one is free

- Free to start: study notes, explained quizzes, flashcards and the first mock exam
- Timed mock exams built to the official exam guide
- A verifiable Certificate of Assessment when you pass a premium mock

Explore Salesforce Certified Agentforce Specialist on MockCert



Open it on your phone

mockcert.com/salesforce-agentforce-specialist/

Scan the code with your phone's camera.