Google Cloud Certified

Generative AI Leader

Detailed study guide

Gen AI fundamentals, Google Cloud's gen AI offerings, techniques to improve model output, and business strategy for successful, secure, and responsible adoption. Built from the official exam guide and Google Cloud sources.

~30%	~35%	~20%	~15%
Fundamentals of gen AI	Google Cloud's gen AI offerings	Techniques to improve gen AI model output	Business strategies for a successful gen AI solution

Edition: October 2026. Independent study material, not affiliated with Google. Practice items are original and are not real exam questions. Google renamed many products in 2026 (for example Vertex AI is now Gemini Enterprise Agent Platform): confirm details on cloud.google.com before exam day.

Contents

START HERE — **The Generative AI Leader exam and how to use this guide**

0.1 Who the exam is for
0.2 Sections and weighting
0.3 Exam mechanics
0.4 How this guide is organised
0.5 Three-week study plan

FOUNDATION — **Google Cloud gen AI at a glance**

F.1 The five layers
F.2 Names you must recognise
F.3 Need-to-offering map

SECTION 1 — **Fundamentals of gen AI**

1.1 Core gen AI concepts and use cases
1.2 Data types in gen AI and their business implications
1.3 The gen AI landscape: core layers and business implications
1.4 Google's foundation models: use cases and strengths
1.5 Additional practice questions

SECTION 2 — **Google Cloud's gen AI offerings**

2.1 Google Cloud's strengths in the field of gen AI
2.2 Prebuilt gen AI offerings that enable AI-powered work
2.3 How Google Cloud's gen AI offerings improve the customer experience
2.4 How Google Cloud empowers developers to build with AI
2.5 The purpose and types of tooling for gen AI agents
2.6 Key terms
2.7 Additional practice questions

SECTION 3 — **Techniques to improve gen AI model output**

3.1 Proactively overcoming foundation model limitations
3.2 Prompt engineering techniques
3.3 Grounding techniques and their use cases
3.4 Additional practice questions

SECTION 4 — **Business strategies for a successful gen AI solution**

4.1 Implementing a transformational gen AI solution
4.2 Secure AI: protecting AI systems from attacks and misuse
4.3 Responsible AI in business
4.4 Additional practice questions

CROSS-SECTION PRACTICE — **Case study practice**

C.1 A method for scenario questions
C.2 Case study: Cymbal Retail

EXAM DAY — **Exam technique and signal words**

E.1 Signal words that map to answers
E.2 Distractor patterns
E.3 Time management

APPENDIX A One-page cheat sheet

APPENDIX B Glossary

APPENDIX C Hands-on exploration plan

APPENDIX D Final readiness checklist

APPENDIX E Official resources

START HERE

The Generative AI Leader exam and how to use this guide

Orientation, exam mechanics, and a study plan

The Generative AI Leader certification validates business-level knowledge of generative AI and of Google Cloud's gen AI products. Google describes the certified professional as someone who understands how gen AI can transform a business, can identify use cases across functions and industries, can hold meaningful conversations with technical and non-technical teams, and can lead organisations toward innovative and responsible AI adoption. The expertise tested is **strategic leadership and influence, not technical implementation**, although you need a sound conceptual understanding of how the technology works.

90 min	50–60	None	3 years
Exam length	Multiple choice questions	Prerequisites; any job role	Validity period

0.1 Who the exam is for

The certification is for anyone in any job role, with or without hands-on technical experience: business leaders, product managers, sales and pre-sales specialists, consultants, transformation and innovation leads, and technical leaders who need to speak the language of business value. Questions are scenario based. A typical item describes an organisation, its goal, and a constraint, then asks which approach, product, or practice best fits.

Three kinds of knowledge earn points:

- **Concepts.** What a foundation model, a diffusion model, grounding, RAG, or an agent is, and what each can and cannot do.
- **Product fit.** Which Google offering matches a need: the Gemini app, Gemini Enterprise, Gemini for Google Workspace, Customer Engagement Suite, Gemini Enterprise Agent Platform, Agent Search, Model Garden, or a pre-built API.
- **Judgement.** Choosing the right technique to fix a model limitation, the right first use case, the right security control, and the responsible-AI practice that a situation demands.

0.2 Sections and weighting

Section	Weight	Relative emphasis	Chapter
1. Fundamentals of gen AI	~30%		Section 1
2. Google Cloud's gen AI offerings	~35%		Section 2

Section	Weight	Relative emphasis	Chapter
3. Techniques to improve gen AI model output	~20%		Section 3
4. Business strategies for a successful gen AI solution	~15%		Section 4

EXAM TIP

Section 2 is the largest and the most "memorisable": many items reduce to matching a business need to the right Google product. Build your own need-to-product table as you read Chapter 2 and revise it daily.

0.3 Exam mechanics

Item	Detail
Length and format	90 minutes; 50 to 60 multiple choice questions (single answer and multiple select).
Fee	USD 99 plus tax where applicable.
Languages	English, Japanese, Spanish, Portuguese.
Delivery	Online-proctored or onsite-proctored.
Validity and renewal	Valid for 3 years; renew within the renewal eligibility period (see Google's renewal FAQs).
Official preparation	The exam guide, Google Cloud Skills Boost Generative AI Leader learning path, and Google's official sample questions.

NOTE ON PRODUCT NAMES

In 2026 Google renamed Vertex AI to **Gemini Enterprise Agent Platform** (the exam guide shortens this to "Agent Platform"). Vertex AI Studio became **Agent Studio**, Vertex AI Search became **Agent Search**, and Google Agentspace became **Gemini Enterprise**. Older articles and training may still use the former names; this guide uses the current names and gives the former name once.

0.4 How this guide is organised

- **Google Cloud gen AI at a glance** maps the product family and its renames.
- **Sections 1 to 4** follow the exam guide item by item with explanations, comparison tables, decision tables, callouts, original practice questions with answers, and key takeaways.
- **Case study practice** applies everything to one organisation's gen AI journey, followed by exam technique and signal words.
- **Appendices** contain a cheat sheet, glossary, a study and hands-on exploration plan, a readiness checklist, and official resources.

EXAM TIP

Patterns that frequently decide an answer.

COMMON TRAP

Plausible distractors and misconceptions.

LEADER'S LENS

The business implication: value, risk, cost, change management, or governance.

0.5 Three-week study plan

Week	Read	Explore	Checkpoint
1	Orientation; Section 1	Try Gemini in the Gemini app; compare a short and a long-context task; generate an image and a short video description of what Imagen and Veo would do; sketch the five-layer landscape for your organisation.	Section 1 practice questions at 80%.
2	Section 2	Tour Agent Studio and Model Garden in the Google Cloud console (free trial); read the Gemini Enterprise, Workspace, and Customer Engagement Suite product pages; build your need-to-product table.	Section 2 practice questions at 80%.
3	Sections 3 and 4; Case study; Appendices	Practise zero-shot, few-shot, role, chain-of-thought, and chained prompts; try Grounding with Google Search in Agent Studio; read SAIF and Google's AI Principles; complete Google's official sample questions.	All practice sets at 85%.

FOUNDATION

Google Cloud gen AI at a glance

The product family, its layers, and its new names

Before the detail, place every offering on one map. The exam guide organises gen AI into five layers, and Google's offerings sit at each one.

F.1 The five layers



Layer	What it provides	Google Cloud examples
Infrastructure	Compute, storage, and networking that train and serve models	AI Hypercomputer, Tensor Processing Units (TPUs), GPUs, Google data centres and global network
Models	Pre-trained foundation models and model access	Gemini, Gemma, Imagen, Veo, plus partner and open models through Model Garden
Platform	Tools to access, customise, ground, evaluate, deploy, and govern models	Gemini Enterprise Agent Platform (formerly Vertex AI): Agent Studio, Model Garden, Agent Search, RAG Engine, Evals, Model Registry, Feature Store, AutoML
Agents	Systems that reason, plan, and use tools to achieve goals	Agents built with Agent Studio or the Agent Development Kit (ADK) and run on Agent Runtime; Conversational Agents; Gemini Enterprise agents
Applications	End-user products with gen AI built in	Gemini app, Gemini Enterprise, Gemini for Google Workspace, Customer Engagement Suite

F.2 Names you must recognise

Current name	Former name	One-line purpose
Gemini Enterprise Agent Platform ("Agent Platform")	Vertex AI Platform	Unified platform to build, ground, tune, evaluate, deploy, and govern models and agents.
Agent Studio	Vertex AI Studio	Console workspace to prototype prompts, compare models, and build agents with low code.
Gemini Enterprise Agent Platform Model Garden	Vertex AI Model Garden	Catalogue of Google, partner, and open models.

Current name	Former name	One-line purpose
Agent Search	Vertex AI Search	Google-quality search and out-of-the-box RAG over enterprise data.
RAG Engine on Gemini Enterprise Agent Platform	(unchanged)	Managed RAG building blocks for custom pipelines.
Agent Runtime	Vertex AI Agent Engine	Managed runtime to deploy and scale agents (with Sessions and Memory Bank).
Gemini Enterprise Agent Platform Evals	Gen AI evaluation service	Evaluate model and agent quality.
Feature Store, Model Registry, Pipelines, Endpoints on Agent Platform	Vertex AI equivalents	MLOps building blocks.
Gemini Enterprise	Google Agentspace	Enterprise app for search across company data, agents, and NotebookLM-style research.
Google AI Studio	(unchanged)	Fast, free-tier developer playground for the Gemini API.

COMMON TRAP

Agent Studio (enterprise, in the Google Cloud console, governed by your project's security and data controls) and **Google AI Studio** (lightweight developer playground for the Gemini API) are different products. Enterprise data, governance, and production deployment point to Agent Studio; quick individual experimentation points to Google AI Studio.

F.3 Need-to-offering map

If the organisation needs...	Start with...	Chapter
Personal productivity assistant for individuals	Gemini app (with Gems for repeatable tasks)	2
Gen AI inside Gmail, Docs, Sheets, Slides, Meet	Gemini for Google Workspace	2
Employees searching and acting across company data and apps	Gemini Enterprise	2
Customer-facing search over a website or catalogue	Agent Search	2
Virtual agents, live-agent assistance, contact-centre analytics	Customer Engagement Suite (Conversational Agents, Agent Assist, Conversational Insights, Contact Center as a Service)	2
Build custom gen AI apps or agents with governance	Gemini Enterprise Agent Platform (Agent Studio, ADK, Agent Runtime)	2

If the organisation needs...	Start with...	Chapter
Choose among many models, including open and partner models	Model Garden	2
Answers grounded in company documents	Agent Search (prebuilt RAG) or RAG Engine (custom)	2, 3
Answers grounded in fresh public information	Grounding with Google Search	3
Task-specific AI with no gen AI needed (OCR, translation, speech)	Pre-built APIs: Document AI, Translation, Speech-to-Text, Text-to-Speech, Vision, Video Intelligence, Natural Language	2
An open model to run or fine-tune on your own infrastructure	Gemma	1
Protect AI systems from attack and misuse	Secure AI Framework (SAIF), IAM, Security Command Center, workload monitoring	4

References

- Generative AI Leader certification (Google Cloud)
- Generative AI Leader exam guide
- Gemini Enterprise Agent Platform name changes

SECTION 1

Fundamentals of gen AI

Exam weight: ~30% of the exam

Section 1 is the vocabulary and mental-model section of the Generative AI Leader exam. It checks that you can explain what generative AI is, how it relates to artificial intelligence and machine learning, how models are built and run, why data quality decides success, how the gen AI market is layered, and what each of Google's flagship model families is good at. The exam is written for leaders, not engineers: you will not be asked to write code or configure a service, but you will be asked to pick the right concept, approach, model, or product for a business scenario and to recognise the business consequence of that choice.

Study it in three passes. First, learn the definitions until you can explain each term to a non-technical colleague in two sentences. Second, practise mapping scenarios to concepts (for example "customer segmentation with no predefined groups" maps to unsupervised learning; "run a model inside an air-gapped plant" maps to an open model such as Gemma). Third, learn the Google Cloud names. Google renamed Vertex AI in 2026; the exam guide already uses the new names, so this chapter does too.

> **NOTE**
>
> **Naming used in this book.** Google's official name-change table states that Vertex AI is now **Gemini Enterprise Agent Platform** (shortened in the exam guide to **Agent Platform**). Related renames you will meet in this chapter: Agent Studio (formerly Vertex AI Studio), Model Garden on Agent Platform (formerly Vertex AI Model Garden), Agent Search (formerly Vertex AI Search), Agent Runtime (formerly Vertex AI Agent Engine), Managed Training (formerly Vertex AI Training), and the Agent Platform versions of Model Registry, Endpoints, Pipelines, Feature Store, Model Monitoring, AutoML and Evals (formerly the Vertex AI equivalents). After this note the book uses only the current names.

Official objective (exam guide, Section 1)	Covered in
1.1 Define core gen AI concepts: AI, NLP, ML, generative AI, foundation models, multimodal foundation models, diffusion models, prompt tuning, prompt engineering, LLMs	1.1.1 – 1.1.11
1.1 Describe ML approaches: supervised, unsupervised, reinforcement	1.1.12 – 1.1.14
1.1 Identify ML lifecycle stages (data ingestion, data preparation, model training, model deployment, model management) and the Google Cloud tools for each	1.1.15 – 1.1.20
1.1 Choose the appropriate foundation model for a business use case (modality, context window, security, availability and reliability, cost, performance, fine-tuning and customisation)	1.1.21, 1.4.5
1.1 Identify business use cases where gen AI can create, summarise, discover and automate (text, image, code, video, data analysis, personalised UX)	1.1.22
1.2 Describe how data types are used in gen AI and the business implications	1.2.1
1.2 Data quality and accessibility (completeness, consistency, relevance, availability, cost, format)	1.2.2, 1.2.3
1.2 Structured vs unstructured data, with real-world examples	1.2.4

Official objective (exam guide, Section 1)	Covered in
1.2 Labelled vs unlabelled data	1.2.5
1.3 Core layers of the gen AI landscape: infrastructure, models, platforms, agents, applications, and the business implications	1.3.1 – 1.3.7
1.4 Use cases and strengths of Google's foundation models: Gemini, Gemma, Imagen, Veo	1.4.1 – 1.4.5
Agent fundamentals implied by the guide (agents layer, Section 2 agent topics)	1.4.6 – 1.4.8

1.1 Core gen AI concepts and use cases

Google's own learning path for this certification opens with the same idea: generative AI is one branch of a much larger family of techniques, and a leader must be able to place it correctly. The "Gen AI: Unlock Foundational Concepts" course on Google Skills describes its goal as exploring "the differences between AI, ML, and gen AI" and understanding "how various data types enable generative AI to address business challenges". The relationship is a set of nested circles.



Read the diagram as "each box is a subset of the one before it". Natural language processing sits beside machine learning as another subfield of AI, and today most NLP is done with machine learning and, increasingly, with large language models.

1.1.1 Artificial intelligence (AI)

Google Cloud defines artificial intelligence as "a broad field, which refers to the use of technologies to build machines and computers that have the ability to mimic cognitive functions associated with human intelligence", such as seeing, understanding and responding to language, analysing data and making recommendations. Google stresses that AI is not a single system but "a set of technologies implemented in a system to enable it to reason, learn, and act to solve a complex problem".

Google also distinguishes capability levels: **Artificial Narrow Intelligence** (the specialised AI in use today, such as image recognition), **Artificial General Intelligence** (human-level AI capable of any intellectual task, a research goal) and **Artificial Super Intelligence** (theoretical). Every product in this exam is narrow AI, however capable it appears.

EXAM TIP

If an answer option describes AI as "a machine learning algorithm" or ML as "the broader field", it has the hierarchy upside down. AI is the umbrella; ML is one way of achieving AI.

1.1.2 Machine learning (ML) and deep learning

Machine learning is "a subset of artificial intelligence that automatically enables a machine or system to learn and improve from experience. Instead of explicit programming, machine learning uses algorithms to analyze large amounts of data, learn from the insights, and then make informed decisions." The product of training is a

model: the learned pattern that is then used to make predictions or generate content on new data.

Deep learning is a subset of ML that trains multi-layer artificial neural networks. Google explains that networks with more than three layers are called deep neural networks and that modern networks can have hundreds or thousands of layers. Deep learning is what made modern generative AI possible, because deep networks can learn from huge volumes of unstructured data such as text, images and audio.

Aspect	Traditional software	Machine learning
How behaviour is defined	Programmers write explicit rules	The system learns patterns from example data
What improves it	New code	More and better data, retraining, tuning
Typical failure mode	Bugs in the rules	Poor or biased data, drift when the world changes
Business example	Tax calculation engine	Fraud detection that adapts to new fraud patterns

1.1.3 Natural language processing (NLP)

Natural language processing is the subfield of AI that enables computers to understand, interpret and generate human language, written or spoken. Google lists NLP alongside deep learning, robotics and expert systems as major subfields under the AI umbrella. Classic NLP tasks include sentiment analysis, entity extraction, classification, translation, speech recognition and summarisation.

Historically each NLP task needed its own specialised model. Large language models changed that: a single LLM can translate, summarise, classify and answer questions from a prompt alone. Google Cloud still offers task-specific pre-trained APIs (such as the Natural Language API, Translation API and Speech-to-Text), which are covered in Section 2; for this section, remember that LLMs are a modern way of doing NLP, not a separate discipline.

1.1.4 Generative AI

Generative AI is the branch of AI whose models **create new content** (text, images, audio, video, code or structured data) that resembles the data they learned from, in response to an input called a **prompt**. Traditional ("discriminative" or "predictive") ML answers questions such as "Is this transaction fraudulent?" or "How many units will we sell next week?". Generative AI answers requests such as "Draft a reply to this complaint", "Create a product image on a white background" or "Write a SQL query that returns last quarter's churned customers".

Business need	Predictive (traditional) ML	Generative AI
Output	A label, score or number	New content: text, image, video, audio, code
Typical question	"Which customers will churn?"	"Write a personalised retention offer for each at-risk customer"
Data needed to start	Usually a labelled dataset you collect and train on	Often none: start from a pre-trained foundation model and a prompt
Quality measure	Accuracy, precision, recall against known answers	Helpfulness, groundedness, factuality, safety, brand fit (often judged by humans or model-based evaluation)

COMMON TRAP

Gen AI is not the answer to every AI question. If the scenario needs a precise numeric forecast or a yes/no risk score from tabular history, a predictive model (for example AutoML on Agent Platform or BigQuery ML) is usually the better, cheaper choice. Choose gen AI when the output is content, language or reasoning over unstructured information.

1.1.5 Foundation models

Google Cloud defines foundation models as "a form of AI model that undergoes pre-training on a large amount of data to do a range of tasks". They are usually trained with **self-supervised learning**, in which the model creates its own training signal (for example by predicting masked or missing parts of the input), so it does not need humans to label the data. The scale of this training "can lead to emergent capabilities, where the model can complete tasks it wasn't explicitly trained to do". Google calls the shift "from specialized tools to adaptable, general-purpose models" the hallmark of the foundation model paradigm.

Google describes a typical foundation model life: data collection and preparation, architecture selection (commonly transformers), self-supervised pre-training, optional **fine-tuning** on task-specific data, **alignment and safety training** (for example reinforcement learning from human feedback), then evaluation and deployment. The business consequence is that most organisations never pre-train a foundation model: they **consume** one (via API), **adapt** it (prompting, grounding, tuning) and **govern** it.

Google offers a useful analogy: the foundation model is the **engine**, generative AI is the **function** it performs. Not every foundation model is used generatively (one could be adapted for classification), but foundation models power the current wave of gen AI.

Pre-training	The expensive first phase in which a model learns general patterns from a very large, broad dataset.
Self-supervised learning	Training in which labels are derived automatically from the data itself (for example "predict the next word"), avoiding costly manual labelling.
Emergent capability	An ability the model shows at scale that it was not explicitly trained for, such as translating between language pairs it rarely saw.
Model Garden	The catalogue on Agent Platform where organisations discover, test, customise and deploy Google, third-party and open foundation models (Google cites more than 200 models).

1.1.6 Large language models (LLMs)

Google Cloud defines a large language model as "a statistical language model, trained on a massive amount of data, that can be used to generate and translate text and other content, and perform other natural language processing (NLP) tasks". LLMs are typically based on the **Transformer** architecture, which Google researchers introduced in 2017. Typical uses are text generation, machine translation, summarisation, question answering and conversational chatbots.

The relationship to foundation models is "parent and child": **all LLMs are foundation models, but not all foundation models are LLMs**. LLMs are trained mainly on text and code; the broader foundation model category also includes models trained on images, audio, video or several modalities at once.

1.1.7 Multimodal foundation models

A **modality** is a type of data: text, image, audio, video, code. A multimodal foundation model is trained on more than one modality and can accept and/or produce more than one. Google describes Gemini as "capable of understanding virtually any input, combining different types of information, and generating almost any output". In practice that means one model can read a scanned invoice image and output JSON, watch a product video and write a description, or listen to a call recording and summarise it.

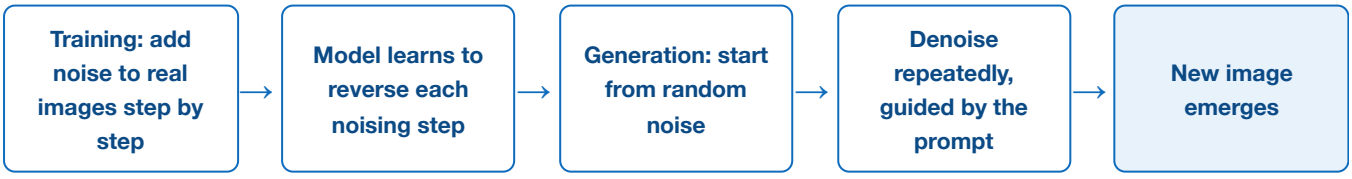
Model type	Typical input	Typical output	Google example
LLM (text)	Text, code	Text, code	Gemini used for text tasks; Gemma
Multimodal understanding	Text, image, audio, video, PDF	Text (including structured JSON)	Gemini Pro, Flash and Flash-Lite tiers
Text-to-image	Text (and reference images)	Images	Imagen; Gemini image models ("Nano Banana")
Text/image-to-video	Text, images	Video with audio	Veo
Embeddings	Text or images	Numeric vectors	Gemini embedding and multimodal embedding models

LEADER'S LENS

Multimodality removes a costly integration step. Previously a claims process might chain OCR, an image classifier and a text model; a single multimodal model can now take the photo, the form and the adjuster's notes together. Fewer components means fewer failure points and faster time to value, but it also concentrates risk in one model, so evaluation and human review matter more.

1.1.8 Diffusion models

Diffusion models are the family of generative models behind most modern image (and many video) generators. Google's "Introduction to Image Generation" course explains that diffusion models "draw inspiration from physics, specifically thermodynamics" and "underpin many state-of-the-art image generation models and tools on Google Cloud". The idea in plain terms:



Because the model learns the general structure of images rather than copying any one image, it can produce novel results that match a text description. The same principle is being applied to text: Google DeepMind lists **DiffusionGemma**, "a text diffusion model designed to maximize generation speed", in the Gemma family.

EXAM TIP

Associate "diffusion" with **image and video generation** and "transformer/LLM" with **language**. If a question asks which model type turns random noise into an image guided by a prompt, the answer is a diffusion model.

1.1.9 Tokens, context windows, embeddings and hallucinations

Four supporting terms appear throughout the exam even though the guide does not list them separately.

Token	The unit a model reads and writes: a word, part of a word, a piece of an image or a slice of audio. Model limits, latency and pricing are measured in tokens.
Context window	The maximum number of tokens the model can consider in one request. Google's analogy is **short-term memory**. Many Gemini models accept 1 million or more tokens; Google equates 1 million tokens to roughly 50,000 lines of code, 8 average-length novels or transcripts of over 200 podcast episodes.
Embedding	A list of numbers (a vector) that represents the meaning of text, an image or other content, so that similar items sit close together. Embeddings power semantic search, recommendations, clustering and retrieval-augmented generation (RAG).
Hallucination	Output that is fluent and confident but factually wrong or unsupported by the provided sources. Mitigations (grounding, RAG, prompt design, human review) are tested in Section 3.

Long context changes design choices. Google notes that smaller context windows "often require strategies like arbitrarily dropping old messages, summarizing content, using RAG with vector databases", whereas a 1M-token window lets you provide "all relevant information upfront". For Gemini 4 Argon the 1,048,576-token window is shared across input, thinking and output tokens; earlier Gemini 3 and 2.5 models use a separate 1M input and 64K output budget.

> **COMMON TRAP**
> A big context window is not a substitute for current, authoritative data. It lets you *supply* more information per request, but the model still only knows what you put in the prompt plus what it learned before its knowledge cutoff. Frequently changing enterprise data is still best served by grounding or RAG.

1.1.10 Prompt engineering

Google defines prompt engineering as "the art and science of designing and optimizing prompts to guide AI models, particularly LLMs, towards generating the desired responses". A **prompt** is "the input you provide to the model to elicit a specific response". Good prompts supply context, clear instructions, examples, constraints and the desired output format. Prompt engineering changes nothing inside the model: it is the cheapest, fastest lever and Google recommends it as the starting point.

Google lists common prompt types: **zero-shot** (instruction with no examples), **one-shot, few-shot and multi-shot** (include worked examples), and **chain-of-thought** (ask the model to reason step by step). Section 3 of this book treats techniques in depth; here you need the definition and its place in the customisation ladder.

```
Role: You are a customer-care writer for an airline.
Task: Draft a 120-word apology email for a 3-hour delay.
Context: Passenger is a Gold member; offer 5,000 bonus miles.
Format: Subject line, then three short paragraphs. Friendly, no jargon.
```

1.1.11 Prompt tuning, fine-tuning and the customisation ladder

Prompt tuning is a **parameter-efficient** customisation technique. Instead of a person writing better words (prompt engineering), a small set of learnable "soft prompt" values is trained on example data and prepended to the model's input, while the foundation model's own weights stay frozen. The result behaves like a highly optimised, machine-learned prompt for a specific task. Because only a tiny number of parameters is learned, it is quick and inexpensive, and one base model can serve many tasks by swapping soft prompts.

Prompt tuning belongs to the wider family that Google calls **parameter-efficient tuning** (also called adapter tuning), which "updates a relatively small subset of the model's parameters". The alternative is **full fine-tuning**, which updates all parameters and can reach higher quality on complex tasks but "demands higher computational resources for both tuning and serving". On Agent Platform, Gemini models are customised with **supervised fine-tuning**, which uses labelled input-output examples (Google suggests "think 100 examples or more").

Lever	What changes	Data needed	Cost and speed	Use when
Prompt engineering	Only the input text	None (a few examples optional)	Lowest; minutes	Always first; prototyping; limited labelled data
Grounding / RAG (Section 3)	Adds retrieved facts to the prompt at run time	Your documents or data sources	Low to medium	Answers must reflect current or private facts
Prompt tuning / parameter-efficient tuning	A small set of added parameters; base model frozen	Labelled examples	Medium; fast to train	Consistent task behaviour, style or format at scale
Full fine-tuning	All model weights	Larger labelled dataset	Highest compute for tuning and serving	Complex, highly specialised tasks where other levers fall short

Google's tuning guidance is explicit: "We recommend starting with prompting to find the optimal prompt. Then, move on to fine-tuning (if required) to further boost performances or fix recurrent errors." Tuning also has a cost benefit: it can "save cost at inference by removing few-shot examples, long instructions in the prompts".

> **COMMON TRAP**
> Do not confuse **prompt engineering** (a human crafts the text; no training) with **prompt tuning** (a training process that learns soft prompt parameters from labelled data). Also, fine-tuning teaches *behaviour, style and task skill*; it is a poor way to keep a model up to date on facts that change weekly. Use grounding for fresh facts.

1.1.12 Supervised learning

Supervised learning "uses labeled datasets to train algorithms to predict outcomes and recognize patterns". Each training example contains inputs (called **features**) and the correct output (the **label**), and the model learns the mapping between them. Google splits it into two task types:

- **Classification**: predict a category. Google's example is the email spam filter trained on labelled spam and legitimate messages.

- **Regression**: predict a continuous number, such as a salary from work experience, or next month's demand.

Business examples Google gives include loan default **risk assessment**, **image classification**, **fraud detection** trained on labelled fraudulent and legitimate transactions, and **recommendation systems** trained on past behaviour. In gen AI, supervised fine-tuning of Gemini is also supervised learning: the "labels" are the ideal responses you want the model to imitate.

1.1.13 Unsupervised learning

Unsupervised learning trains on **unlabelled** data; the models are "allowed to discover patterns and insights without any explicit guidance or instruction". Google lists three task types:

- **Clustering**: group similar items, used for customer segmentation, fraud detection and image analysis (k-means is the classic algorithm).
- **Association**: find "if-then" co-occurrence rules, such as products frequently bought together (market basket analysis).
- **Dimensionality reduction**: compress many variables into fewer while keeping the important structure, for visualisation or as a pre-processing step.

Unsupervised learning is useful when you do not know in advance what groups exist. A human usually interprets the discovered clusters afterwards (Google's example: weather data grouped by temperature, which an analyst recognises as the four seasons).

1.1.14 Reinforcement learning (and the hybrids)

Reinforcement learning (RL) is "a type of machine learning where an 'agent' learns optimal behavior through interaction with its environment", by trial and error, "receiving feedback in the form of rewards or penalties for its actions". Google describes a feedback loop of four elements, **agent, environment, actions and rewards**: the agent observes the state, acts according to its policy, the environment changes, a reward signal arrives, and the policy is updated to favour actions that earned rewards. Google's well-known example is the system that learned to play the game Go.

Business uses include dynamic pricing, supply-chain and inventory optimisation, robotics, recommendation sequencing, and resource scheduling (for example data-centre cooling). RL is also central to gen AI itself: **reinforcement learning from human feedback (RLHF)** is a standard alignment step in which human ratings of model responses steer the model towards helpful, harmless outputs. Agent Platform offers this as Reinforcement Learning on Gemini Enterprise Agent Platform (formerly RLHF on Vertex AI).

Two hybrids complete the picture. **Semi-supervised learning** uses a small labelled set plus a large unlabelled set. **Self-supervised learning**, used to pre-train foundation models, generates labels automatically from the data itself.

Approach	Training data	Learns to	Business example	Scenario clue words
Supervised	Labelled (inputs plus known correct outputs)	Predict a label or value	Predict loan default; classify support tickets; forecast sales	"historical outcomes", "known answers", "predict", "classify"

Approach	Training data	Learns to	Business example	Scenario clue words
Unsupervised	Unlabelled	Find hidden structure	Segment customers; detect unusual transactions; bundle products	"no predefined categories", "discover groups", "patterns"
Reinforcement	Rewards from interacting with an environment	Choose a sequence of actions that maximises reward	Dynamic pricing; warehouse robots; tuning a chatbot with human ratings (RLHF)	"trial and error", "reward", "feedback loop", "optimise decisions over time"
Self-supervised	Unlabelled; labels derived from the data	General representations	Pre-training foundation models	"predict masked or next token", "pre-training at scale"

EXAM TIP

The fastest discriminator is the data. Labels present: supervised. No labels and the goal is grouping or discovery: unsupervised. No dataset at all, just an environment and a reward: reinforcement.

1.1.15 The ML lifecycle at a glance

The exam guide names five lifecycle stages. Google's Architecture Center describes the same journey in more detail (data extraction, data analysis, data preparation, model training, model evaluation, model validation, model serving, model monitoring) and stresses that "the real challenge isn't building an ML model, the challenge is building an integrated ML system and to continuously operate it in production". The practice of automating and monitoring every step is **MLOps**, "an ML engineering culture and practice that aims at unifying ML system development (Dev) and ML system operation (Ops)".



The lifecycle is a loop, not a line: monitoring in the management stage detects drift, which triggers new data ingestion and retraining. Google calls automatic retraining **continuous training (CT)**, a property unique to ML systems alongside the familiar continuous integration and continuous delivery.

Stage	Business question	Main Google Cloud tools
Data ingestion	How do we collect data from every source, in batch or real time?	Pub/Sub, Cloud Storage, BigQuery, Dataflow, Managed Service for Apache Spark (formerly Dataproc)
Data preparation	Is the data clean, consistent, labelled, governed and ready to use?	BigQuery, Dataflow, Managed Service for Apache Spark, Data Labeling on Agent Platform, Feature Store on Agent Platform, Knowledge Catalog (data quality, lineage), Colab Enterprise and Workbench notebooks
Model training	Do we build, tune or adopt a model, and on what compute?	Managed Training on Agent Platform, AutoML on Agent Platform, BigQuery ML, supervised fine-tuning in Agent Studio, Model Garden, Experiments on Agent Platform, TPUs and GPUs

Stage	Business question	Main Google Cloud tools
Model deployment	How do people and applications use the model, at what latency and scale?	Model Registry, Agent Platform Endpoints (online inference), Agent Platform Batch Inference, Agent Runtime for agents, Cloud Run, GKE
Model management	Is it still accurate, safe, versioned and governed?	Model Monitoring, Agent Platform Pipelines, Model Registry versions and aliases, Model Evaluation and Agent Platform Evals, Feature Store feature monitoring

1.1.16 Stage 1: Data ingestion

Ingestion moves raw data from where it is created into a place where it can be processed. Google's Architecture Center calls this **data extraction**: selecting and integrating relevant data from various sources.

- **Pub/Sub**: "an asynchronous and scalable messaging service that decouples services producing messages from services processing those messages". Use it for real-time streams such as clickstreams, IoT telemetry and application events. Google's typical pattern forwards events to Pub/Sub, processes them with Dataflow and lands them in BigQuery, Bigtable or Cloud Storage.
- **Cloud Storage**: object storage for files of any type and size, the natural landing zone for unstructured data such as PDFs, images, audio and video, and for training datasets.
- **BigQuery**: Google's "fully managed, AI-ready data platform". It supports continuous ingestion of structured and unstructured data in real time and open table formats such as Apache Iceberg.
- **Dataflow**: "unified stream and batch data processing at scale", used for data movement and ETL into BigQuery, built on open-source Apache Beam.
- **Managed Service for Apache Spark** (formerly Dataproc): managed Spark and Hadoop for organisations with existing open-source data pipelines, integrated with BigQuery and Cloud Storage.

1.1.17 Stage 2: Data preparation

Preparation turns raw data into model-ready data: cleaning errors and duplicates, standardising formats, joining sources, engineering features, labelling examples and splitting into training, validation and test sets. Google's fine-tuning guide summarises the same steps for gen AI: collect, clean and format, then split.

- **BigQuery** for SQL-based exploration, cleansing and transformation; **Dataflow** and **Managed Service for Apache Spark** for large-scale transformation pipelines.
- **Data Labeling on Agent Platform** to add human labels to data for supervised learning.
- **Feature Store on Agent Platform**: "a centralized repository for organizing, storing, and serving ML features". The current version manages feature data in BigQuery (the offline store) and serves the latest values online at low latency, so the same feature definitions are used for training and for live predictions, which reduces training-serving skew.
- **Knowledge Catalog** (the evolution of Dataplex Universal Catalog) to discover data, add business context, track lineage and profile and validate data quality.
- **Colab Enterprise** and **Workbench** notebooks, integrated with BigQuery, for exploratory analysis.

1.1.18 Stage 3: Model training

Training fits a model to the prepared data; for foundation models it usually means adapting an existing model rather than training from scratch. Options range from no-code to full control:

Option	Skill needed	Best for
Use a pre-trained foundation model from Model Garden as-is	Low	Most gen AI use cases; start here
Supervised fine-tuning of Gemini (Agent Studio or API)	Low to medium	Consistent style, format or domain task once prompting plateaus
AutoML on Agent Platform	Low (no code)	Predictive models on tabular, image or other data with your labels
BigQuery ML	SQL	Analysts training predictive models where the data already lives
Managed Training on Agent Platform (custom training)	High (data scientists and ML engineers)	Full control of framework, code and hyperparameters; serverless or reserved Managed Training Clusters

Experiments on Agent Platform tracks and compares training runs so teams can pick the best model. Training compute comes from Google's AI infrastructure (TPUs and GPUs), discussed in 1.3.2.

1.1.19 Stage 4: Model deployment

Deployment makes a model available to applications and users. The Architecture Center lists three forms: a REST API for online predictions, an embedded model on an edge or mobile device, or a batch prediction system.

- **Model Registry on Agent Platform**: "a central repository where you can manage the lifecycle of your ML models". From the registry you can version a model, assign aliases (such as the default production alias), evaluate it, deploy it to an endpoint or run batch inference. It also supports BigQuery ML models without export.
- **Agent Platform Endpoints** and **Online Inference** for real-time, low-latency predictions; **Batch Inference** for large offline jobs (for example scoring every customer overnight).
- **Agent Runtime** (formerly Vertex AI Agent Engine) for deploying and scaling agents rather than single models.
- **Cloud Run** or **GKE** when the model or application must run in containers that you control.

For Google's managed foundation models (such as Gemini) you do not deploy anything: you call the model through the Agent Platform API on a pay-as-you-go or Provisioned Throughput basis. Deployment work is needed for custom, tuned or open models you host yourself.

1.1.20 Stage 5: Model management

Models decay. The Architecture Center warns that "models can decay in more ways than conventional software systems" because "constantly evolving data profiles" change the world the model sees. Management keeps a deployed model trustworthy:

- **Model Monitoring on Agent Platform** "monitors models for training-serving skew and inference drift and sends you alerts when the incoming inference data skews too far from the training baseline", so you can decide whether to retrain.
- **Agent Platform Pipelines** "helps you automate, monitor, and govern your ML workflows", turning manual steps into repeatable pipelines (the move from MLOps maturity level 0 to automated levels).

- **Model Registry** versioning and aliases for controlled rollouts and rollbacks; ML metadata for lineage and audit.
- **Model Evaluation** and **Agent Platform Evals** (formerly the Gen AI evaluation service) to measure quality before and after changes.

Training-serving skew	A difference between the data a model was trained on and the data it receives in production, for example a feature computed differently in the two places.
Drift	A change over time in production input data (or in the relationship between inputs and outcomes) that erodes model quality, such as new customer behaviour after a pricing change.
MLOps	Applying DevOps principles (automation, testing, monitoring, continuous delivery) to ML systems, plus continuous training.

LEADER'S LENS

Budget for the whole lifecycle, not just the build. Google points out that only a small fraction of a real-world ML system is model code; configuration, data collection and verification, serving infrastructure and monitoring make up the rest. A pilot that ignores management becomes an unmaintained liability once the data shifts.

EXAM TIP

Map the verb to the tool: **stream/ingest events** to Pub/Sub; **store files** to Cloud Storage; **analyse with SQL** to BigQuery; **transform at scale** to Dataflow; **share features** to Feature Store; **version and deploy** to Model Registry and Endpoints; **detect drift** to Model Monitoring; **automate the workflow** to Pipelines.

1.1.21 Choosing the right foundation model for a business use case

There is no single best model. The exam guide lists the factors a leader should weigh; Agent Platform's Model Garden exists precisely so organisations can compare Google, third-party and open models against them. Treat model choice as a portfolio decision, matching each use case to the smallest model that meets the requirement.

Factor	Question to ask	What it points to
Modality	What goes in and what must come out: text, images, audio, video, code?	Multimodal Gemini for mixed inputs; Gemini image models or Imagen for images; Veo for video; embeddings for search
Context window	How much material must the model consider in one request?	Long-context Gemini (around 1M tokens) for whole contracts, codebases or hours of video; smaller windows are fine for short chats
Security and data governance	Where may the data be processed? Who can access the model? Are there residency or air-gap requirements?	Managed models on Agent Platform with enterprise controls; open models such as Gemma that can run on your own infrastructure or devices
Availability and reliability	What uptime, regions, quotas and throughput guarantees are needed? Is the model GA or Preview?	Generally available models for production; Provisioned Throughput for guaranteed capacity; check regional availability

Factor	Question to ask	What it points to
Cost	What is the cost per request at expected volume?	Flash-Lite or Flash tiers, smaller open models, batch inference and context caching for high-volume work; frontier tiers only where they earn it
Performance	What quality, reasoning depth and latency does the task need?	Frontier or Pro tier for complex reasoning; Flash for balanced speed and quality; Flash-Lite for lowest latency
Fine-tuning and customisation	Will we need to tune it, and is that supported?	Models that support supervised fine-tuning on Agent Platform; open-weight models (Gemma) for full control of tuning

Notice the trade-off triangle of **quality, latency and cost**. Google's own model descriptions express it: Gemini 3.1 Flash-Lite is "our most cost-efficient model, optimized for low latency use cases for high-volume, cost-sensitive LLM traffic", while the frontier model Gemini 4 Argon "excels at deep work in complex, real-world tasks". Google's fine-tuning guide adds that "for high-volume use cases, fine-tuning a smaller model can be more cost-effective than using a larger, general-purpose model for each request".

COMMON TRAP

"Pick the most powerful model" is rarely the best answer. If a scenario stresses millions of requests, tight budgets or real-time latency, the correct choice is usually a smaller, faster tier that meets the quality bar. If it stresses that data must never leave the premises, a managed cloud API is wrong regardless of quality.

LEADER'S LENS

Avoid lock-in to one model version. Model families evolve quickly (Google released several Gemini generations within two years). Design applications so the model is a configurable component, evaluate new versions with your own test set, and plan for scheduled model upgrades and retirements.

1.1.22 Business use cases: create, summarise, discover, automate

The exam guide frames gen AI value with four verbs. A useful habit is to classify every proposed use case under one of them; it clarifies the success metric.

Verb	What gen AI does	Examples	Typical success metric
Create	Generates new content	Marketing copy, product descriptions, campaign images, training videos, code, synthetic test data	Time to first draft, content volume, brand-compliance rate
Summarise	Condenses large inputs into the essentials	Call and meeting summaries, contract digests, research briefs, incident reports	Reading time saved, accuracy of key points, handle time
Discover	Finds and synthesises information, answers questions over data	Enterprise search over policies, semantic product search, insight extraction from reviews, natural-language questions over BigQuery	Time to answer, search success rate, self-service resolution

Verb	What gen AI does	Examples	Typical success metric
Automate	Performs tasks or multi-step workflows, often via agents	Invoice processing, ticket triage and routing, code migration, claims intake, scheduling	Throughput, straight-through processing rate, cost per transaction, error rate

The guide also lists the content types to recognise:

Use case type	Example scenario	Google model or offering to think of
Text generation	Drafting personalised customer emails and FAQs	Gemini; Gemini for Google Workspace
Image generation	Product imagery and ad variants without a photo shoot	Gemini image models ("Nano Banana"), Imagen
Code generation	Generating, explaining and migrating code; writing SQL from questions	Gemini (for example through Gemini Code Assist)
Video generation	Short promotional or training clips from a script or storyboard	Veo
Data analysis	Asking questions of sales data in natural language; extracting fields from documents into tables	Gemini with BigQuery; multimodal extraction to JSON
Personalised user experience	Tailored recommendations, adaptive shopping assistants, content in each customer's language and tone	Gemini with grounding on customer data; agents

LEADER'S LENS

Prioritise use cases where value is high and risk is manageable: internal "summarise" and "discover" use cases with a human reviewing outputs are usually the safest first wins. Externally facing "create" and fully autonomous "automate" use cases deliver larger returns but need stronger grounding, evaluation, safety settings and human oversight.

1.2 Data types in gen AI and their business implications

Every capability in this book rests on data. Google's machine learning overview puts it simply: "Assuming the training data is of high quality, the more training samples the machine learning algorithm receives, the more accurate the model will become", and it warns that ML is "often only as good as the data it's being fed". Section 1.2 tests whether you can judge data as a business asset: its quality, its accessibility, its structure and whether it is labelled.

1.2.1 How data is used across a gen AI solution

Data enters a gen AI solution at several points, and each use has different quality requirements.

Where data is used	Purpose	Business implication
Pre-training (by the model provider)	Gives the foundation model broad knowledge and skills	You inherit its strengths, gaps, biases and knowledge cutoff; choose providers with transparent, responsible practices

Where data is used	Purpose	Business implication
Prompts and context	Supplies task instructions and the facts needed now	Prompt data may contain confidential or personal information; apply access control and privacy policies
Grounding and RAG sources	Anchors answers in your documents, databases or the web	Answer quality can never exceed source quality; outdated or contradictory documents produce outdated or contradictory answers
Tuning datasets	Teaches a style, format or specialised task	Needs curated, labelled, representative examples; Google stresses "high-quality, well-labeled data is crucial for good performance and better than quantity"
Evaluation datasets	Measures whether the model meets the bar before and after changes	Without a trusted test set you cannot prove ROI or detect regressions
Production logs and feedback	Monitoring, drift detection, continuous improvement	Must be retained and governed in line with privacy obligations

1.2.2 Data quality characteristics

The exam guide names completeness, consistency and relevance as quality dimensions (with availability, cost and format covering accessibility, next section). Learn each with a symptom you would see in a gen AI project.

Characteristic	Meaning	Symptom when it fails	Typical remedy
Completeness	All required records and fields are present; no important gaps	A support assistant cannot answer about a product line whose manuals were never loaded; a churn model ignores customers with missing fields	Gap analysis against the use case; fill or source missing data; document known gaps
Consistency	The same thing is represented the same way across sources and over time	Conflicting answers because regions store prices in different currencies, dates in different formats, or two policy versions disagree	Standardise formats, master data management, de-duplicate, retire superseded documents
Relevance	The data actually relates to the task and the population the model will serve	A model tuned on consumer emails performs poorly on B2B contracts; retrieval returns off-topic passages	Curate sources to the use case; filter noise; ensure training data reflects production prompts
Accuracy (related)	Values are correct	Confidently wrong answers grounded on wrong records	Validation rules, data quality scans, source-of-truth ownership
Timeliness (related)	Data is current enough for the decision	Assistant quotes last year's policy	Scheduled refresh, streaming ingestion, grounding on live sources

LEADER'S LENS

Poor data quality is also a responsible AI issue. Google notes that ML's bias potential comes from the data it is fed: if a dataset under-represents a customer group (a completeness and relevance failure), the model will serve that group worse. Data quality reviews should therefore include representativeness, not only missing values.

1.2.3 Data accessibility: availability, cost and format

High-quality data has no value if the model cannot reach it, legally and practically. The guide groups three accessibility characteristics.

Characteristic	Leader's question	Example issue	Google Cloud help
Availability	Can the right systems and people access the data when needed, with permission?	Knowledge locked in departmental silos or on laptops; data owner unknown; legal restrictions on use	Central storage in Cloud Storage and BigQuery; Knowledge Catalog for discovery and governance; IAM for controlled access
Cost	What does it cost to acquire, store, clean, label, move and process the data?	Manual labelling of 100,000 images; repeated egress; reprocessing petabytes	Serverless processing (BigQuery, Dataflow); start with prompting before paying for labelled tuning data; reuse features via Feature Store
Format	Is the data in a form the model and tools can consume?	Scanned faxes, proprietary file types, inconsistent schemas, audio without transcripts	Multimodal Gemini reads PDFs, images, audio and video directly; Dataflow and Spark convert formats; open table formats in BigQuery

EXAM TIP

When a scenario says "the data exists but teams cannot find it or are not permitted to use it", the problem is **accessibility (availability)**, not quality. When it says "the records contradict each other", the problem is **consistency**. When it says "half the fields are blank", it is **completeness**.

1.2.4 Structured, semi-structured and unstructured data

Google Cloud defines **unstructured data** as "information that doesn't follow a predefined format, model, or structure. It doesn't fit neatly into rows and columns." It makes up an estimated 70% to 90% of the information a business collects. **Structured data** "relies on SQL, enforces strict schemas, and is easily searchable". **Semi-structured data**, such as JSON or XML, "contains some organizational tags but lacks a rigid relational schema".

	Structured	Semi-structured	Unstructured
Data model	Strict, predefined schema	Flexible, self-describing	No predefined model
Typical storage	Relational databases, data warehouses (BigQuery, Cloud SQL, AlloyDB)	Document and NoSQL stores (Firestore), JSON in BigQuery	Object storage (Cloud Storage), data lakes, NoSQL
Query method	SQL	NoSQL APIs, SQL extensions	AI, ML, NLP, search indexes, embeddings

	Structured	Semi-structured	Unstructured
Real-world examples	Financial ledgers, inventory tables, CRM records, transaction histories, spreadsheets with fixed columns	JSON API responses, XML feeds, CSV exports, application logs with fields	Emails, chat transcripts, PDFs and contracts, images, call recordings, video, social media posts, sensor waveforms

Gen AI's big business shift is that unstructured data, previously expensive to use, becomes directly usable. Google's examples: support tickets land in Cloud Storage, Gemini reads them to find the core issue and writes structured tags into a database for routing; insurance claim photos are analysed by a model and a structured repair estimate is stored for downstream systems. The pattern is **unstructured in, structured out**.

COMMON TRAP

A PDF invoice is unstructured even though it contains a table, because it has no machine-enforced schema. A CSV file is usually classed as semi-structured by Google. A relational table of orders is structured. Watch for scanned forms and recordings: always unstructured.

1.2.5 Labelled and unlabelled data

Labelled data pairs each example with the correct answer ("this email is spam", "this X-ray shows a fracture", "the ideal reply to this ticket is..."). **Unlabelled data** is raw examples with no answer attached. The distinction drives which ML approach is possible and how much the project costs.

	Labelled data	Unlabelled data
Enables	Supervised learning; supervised fine-tuning of gen AI models; evaluation against known answers	Unsupervised learning (clustering, association); self-supervised pre-training of foundation models; grounding and RAG sources
Cost	Expensive and slow: needs human experts, guidelines and quality checks	Cheap and abundant
Risk	Labeller inconsistency or bias becomes model bias	Hidden noise and irrelevant content
Examples	Transactions marked fraud or legitimate; product photos tagged by category; call summaries approved by supervisors	Raw clickstream logs, an archive of untagged images, a document repository
Google Cloud help	Data Labeling on Agent Platform; AutoML on labelled data	Foundation models used with prompting and grounding need no labels to start

LEADER'S LENS

Foundation models changed the economics of AI projects. Because they are pre-trained with self-supervised learning on unlabelled data, many use cases can begin with prompting and grounding and no labelling budget at all. Labelled data becomes a targeted investment, mainly for evaluation sets and for fine-tuning once you have proven value.

1.3 The gen AI landscape: core layers and business implications

The third course of Google's learning path, "Gen AI: Navigate the Landscape", explores "the different layers of building gen AI solutions, Google Cloud's offerings, and the factors to consider when selecting a solution". The exam guide names five layers. Each layer depends on the ones below it; as you move up, abstraction increases and the audience shifts from engineers to business users.



1.3.1 The five layers at a glance

Layer	What it is	Google offerings	Who typically engages
Infrastructure	Compute, accelerators, storage, networking and data centres that train and serve models	AI Hypercomputer, TPUs (including TPU 8t for training and TPU 8i for inference), NVIDIA GPUs, Compute Engine, GKE, Cloud Storage, Google's global network	Infrastructure and platform engineers; model builders
Models	Foundation models and specialised models that generate and understand content	Gemini, Gemma, Imagen and Gemini image models, Veo, Lyria, embeddings; plus third-party and open models in Model Garden	Data scientists, ML engineers, solution architects
Platforms	Managed environment to access, customise, ground, evaluate, deploy, govern and monitor models and agents	Gemini Enterprise Agent Platform: Model Garden, Agent Studio, tuning, RAG Engine, Agent Search, Vector Search, Evals, Pipelines, Model Registry, Model Monitoring; Google AI Studio for rapid prototyping	Developers and technical teams
Agents	Systems that use models to reason, plan and take actions with tools towards a goal	Agent Development Kit (ADK), Agent Studio, Agent Garden, Agent Runtime, Memory Bank, Agent Registry, Agent Identity, Agent Gateway; prebuilt agents in Gemini Enterprise and Customer Engagement Suite	Developers, business process owners
Applications	End-user products where people experience gen AI	Gemini app, Gemini Enterprise, Gemini for Google Workspace, Gemini Code Assist, Customer Engagement Suite, Google Search experiences, custom apps	Employees, customers, business users

1.3.2 Infrastructure layer

Infrastructure is the physical and virtual foundation: specialised accelerators, high-bandwidth networking, storage and the data centres that house them. Google's **AI Hypercomputer** is its integrated system for training, serving and operating AI, combining Google-designed **Tensor Processing Units (TPUs)** with NVIDIA GPUs and software co-designed with Google DeepMind. Google currently highlights TPU 8t for training and TPU 8i for inference. Section 2 covers infrastructure benefits in detail.

Business implications: infrastructure determines training speed, serving latency, cost per token and energy efficiency. Very few organisations need to manage AI infrastructure directly; most consume it indirectly through managed models and platforms. Organisations that train or host their own models (including open models such as Gemma) make infrastructure decisions about accelerators, capacity reservations and regions.

1.3.3 Models layer

The models layer contains the foundation models themselves. Agent Platform provides access to "over 200 foundation models" in Model Garden, including Google's first-party models, open models such as Gemma, and third-party models such as Anthropic's Claude family.

Business implications: model choice drives quality, cost and risk (see 1.1.21). An **open approach** (choice of first-party, third-party and open models) protects against lock-in and lets each use case use the best-fit model. Models also change quickly, so plan for versioning, evaluation and migration.

1.3.4 Platforms layer

A platform turns raw models into enterprise solutions. Google describes Gemini Enterprise Agent Platform as "a unified platform to build, deploy, govern, and optimize enterprise-grade AI agents and model-based solutions", organised around four pillars: **Build, Scale, Govern and Optimize**. It is "an evolution of Vertex AI" and "supports the complete AI lifecycle from accessing over 200 foundation models to deploying and managing your agents". It offers low-code tools (Agent Studio) and code-based tools (Colab Enterprise notebooks, Agent Development Kit).

Business implications: the platform is where security, privacy, governance, evaluation and MLOps live. Choosing a platform rather than stitching together point tools reduces integration effort, gives one place for access control and auditing, and speeds the path from prototype to production.

1.3.5 Agents layer

Agents use models plus **tools**, **memory** and **instructions** to pursue goals and complete tasks. On Google Cloud they are built with the Agent Development Kit or Agent Studio, deployed on Agent Runtime and governed with Agent Registry, Agent Identity and Agent Gateway. 1.4.6 to 1.4.8 cover agent fundamentals.

Business implications: agents move gen AI from answering to **doing**, so the value (automation of multi-step work) and the risk (actions taken in real systems) both increase. Governance questions become central: what identity does the agent act under, what tools can it call, what needs human approval, how are its actions logged?

1.3.6 Applications layer

Applications are where users meet gen AI: chat assistants, productivity suites, contact-centre tools, search experiences and custom apps. Google examples include Gemini for Google Workspace, Gemini Enterprise (described by Google as working "across your email, docs, code, data, and business apps"), Gemini Code Assist and Customer Engagement Suite.

Business implications: buying a ready-made application is the fastest path to value and needs the least technical skill, but offers the least differentiation and customisation. Building a custom application on lower layers gives control and competitive advantage at higher cost and time.

1.3.7 Business implications: where should your organisation play?

The layers frame a classic build-versus-buy decision. The higher you engage, the faster and simpler; the lower you engage, the more control and differentiation (and responsibility) you take on.

Situation	Engage at	Example
Need productivity gains for all staff quickly, minimal IT effort	Applications	Roll out Gemini for Google Workspace to draft documents and summarise email
Want to automate a specific business process using enterprise systems	Agents (on the platform)	An agent that triages supplier invoices, checks the ERP and routes exceptions
Need a differentiated, customer-facing product with your data and brand	Platforms plus models	Build a grounded shopping assistant on Agent Platform with Gemini and Agent Search
Require a specialised or self-hosted model, or strict data residency	Models plus infrastructure	Fine-tune Gemma and run it on GKE or on-premises hardware
Building frontier models or very large-scale training	Infrastructure	Reserve TPU capacity through AI Hypercomputer

EXAM TIP

Expect "which layer?" questions. TPUs, GPUs and data centres are **infrastructure**; Gemini, Gemma, Imagen and Veo are **models**; Agent Platform, Model Garden and Agent Studio are **platforms**; ADK-built or prebuilt systems that take actions with tools are **agents**; Gemini for Workspace and the Gemini app are **applications**.

COMMON TRAP

"Gemini" names both a model family (models layer) and user-facing products such as the Gemini app and Gemini Enterprise (applications layer). Read the scenario: if users interact with it directly to get work done, it is the application; if developers call it through an API, it is the model.

1.4 Google's foundation models: use cases and strengths

The exam guide names four Google model families. Learn each by its **strength** and the **scenario clue** that points to it. Version numbers change every few months; the exam tests the families and their positioning, not the latest release number. The versions below are those listed on Google's official model pages in October 2026.

1.4.1 Gemini: Google's multimodal, long-context model family

Gemini models, built by Google DeepMind, are Google's flagship **natively multimodal** models: they understand text, images, audio, video, PDFs and code, and reason across them in one request. Google's generative AI page describes Gemini as "capable of understanding virtually any input, combining different types of information, and generating almost any output".

Key strengths

- **Multimodality**: one model for documents, images, audio and video understanding, with text (including structured JSON) output; specialised Gemini variants generate images, speech and video.
- **Long context**: mainline models offer a context window of about 1 million tokens (1,048,576), enabling whole-document, whole-codebase and long-video analysis. Google notes Gemini "is the first model capable of accepting 1 million tokens".

- **Reasoning and "thinking"**: configurable thinking levels let you trade depth of reasoning for speed and cost.
- **Agentic capability**: function calling, tool use (for example Google Search grounding and code execution) and long-horizon task execution.
- **Enterprise availability**: offered on Agent Platform with enterprise security and governance, and in Google AI Studio and the Gemini API for developers.

Tiers. Google positions Gemini models in tiers that trade capability against speed and cost:

Tier	Positioning in Google's words	Current examples (October 2026)	Choose when
Frontier / Pro	Gemini 4 Argon: "Google's frontier model that excels at deep work in complex, real-world tasks across domains including coding, finance, legal, cybersecurity, and science". Gemini 3.1 Pro: "reasoning-first model optimized for complex agentic workflows and coding"	Gemini 4 Argon (GA); Gemini 3.1 Pro PREVIEW; Gemini 2.5 Pro	Hardest reasoning, high-stakes analysis, complex multi-step agents, where quality outweighs cost
Flash	Gemini 3.8 Flash: "Our most intelligent workhorse model yet, built for long-horizon coding and autonomous agents"	Gemini 3.8, 3.7, 3.6, 3.5 Flash; Gemini 2.5 Flash	Most production workloads: balanced quality, speed and price
Flash-Lite	Gemini 3.1 Flash-Lite: "our most cost-efficient model, optimized for low latency use cases for high-volume, cost-sensitive LLM traffic"	Gemini 3.5 Flash-Lite; Gemini 3.1 Flash-Lite; Gemini 2.5 Flash-Lite	Classification, extraction, routing, translation at very high volume and low latency
Specialised	Live (real-time voice), TTS (speech), image generation ("Nano Banana"), Omni Flash (video) and embeddings	Gemini 3.8 Live; Gemini 3.8 Flash TTS PREVIEW; Nano Banana 2.1; Gemini Omni Flash PREVIEW	Voice agents, narration, image and video creation, semantic search

Typical use cases: summarising long contracts and reports, extracting fields from invoices and forms into JSON, answering questions over video or audio, code generation and migration, multilingual customer service, data analysis, and powering agents. Gemini models understand more than 100 languages.

> **EXAM TIP**
> Scenario clues for Gemini: "multiple data types in one request", "very long documents or videos", "complex reasoning", "managed and enterprise-ready". If cost and volume dominate, the answer is a Flash or Flash-Lite tier; if reasoning depth dominates, the Pro or frontier tier.

1.4.2 Gemma: lightweight open models

Gemma is "a family of lightweight, state-of-the-art open models built from the same research and technology used to create the Gemini models". Gemma models are "provided with open weights and permit responsible commercial use, allowing you to tune and deploy them in your own projects and applications". Google DeepMind describes them as models that "run wherever users need them, from cloud servers to laptops and even phones".

Key strengths

- **Run anywhere**: on your own hardware, on-premises, at the edge, on mobile devices or hosted services; download from Kaggle or Hugging Face, or deploy from Model Garden on Agent Platform.
- **Full control and customisation**: open weights allow fine-tuning with your preferred frameworks and techniques, and inspection of the model you deploy.
- **Efficiency**: Gemma 4 is designed to "maximize intelligence-per-parameter" and comes in sizes from E2B and E4B (effective-parameter models built for mobile, edge and browser) through 12B and a 31B dense model to a 26B mixture-of-experts model. Small models have a 128K context window and medium models up to 256K.
- **Capabilities**: reasoning with configurable thinking, multimodal input (text and image on all sizes; audio on some), function calling for agentic workflows, and support for over 140 languages.
- **Specialised variants**: MedGemma (medical text and image comprehension), ShieldGemma 2 (safety classification of inputs and outputs), CodeGemma (coding), PaliGemma (vision-language), EmbeddingGemma (embeddings for RAG and search), TxGemma (therapeutics) and others.

Typical use cases: on-device assistants that work offline, processing sensitive data that must stay on-premises or in a sovereign environment, low-cost high-volume tasks on owned infrastructure, research and custom domain models, and embedding AI in products where per-call API costs or connectivity are constraints. Google cites real examples such as an offline AI micro-server for educators and a lightweight model improved for Swahili.

> **COMMON TRAP**
> "Open" does not mean "free of responsibility". With Gemma, your organisation hosts, secures, scales, monitors and updates the model. Choose Gemma when control, locality or customisation justify that operational effort; choose managed Gemini when you want Google to run the model and want frontier capability.

1.4.3 Imagen: text-to-image generation and editing

Imagen is Google's family of **text-to-image diffusion models**. Google's image documentation summarises what it enables: "Generate novel images using only a text prompt (text-to-image AI generation)", "Edit or expand an uploaded or generated image using a mask area you define" (inpainting and outpainting), and "Upscale existing, generated, or edited images". The Imagen 4 generation was positioned for "professional branding, intricate scene composition, and high-fidelity design tasks that require precise text rendering and complex lighting", in Standard, Ultra and Fast variants that trade quality against speed and cost.

Typical use cases: marketing and advertising creative, product imagery and lifestyle shots, design concepts and mood boards, localised campaign variants, and editing existing images (removing or inserting objects, extending backgrounds).

NOTE

Current status. The exam guide still names Imagen, so learn its strengths. However, Google's developer documentation now describes Imagen as "Google's legacy image generation model" that is shut down in the Gemini API, and directs users to the **Gemini image models**, nicknamed **Nano Banana** (for example Nano Banana 2.1, Nano Banana Pro and Nano Banana 2 Lite). On Agent Platform, image generation is documented with these Gemini models, which add conversational (multi-turn) editing, multi-image fusion, character consistency and strong text rendering, with reasoning from Gemini's world knowledge. For exam purposes: **"generate or edit images from text" maps to Imagen**; in practice today, Google's recommended image models are the Gemini image family.

1.4.4 Veo: video generation

Veo is Google DeepMind's **video generation** model family. Google describes Veo 3 as bringing "greater realism and fidelity, made possible by Veo 3's real world physics and audio", with improved prompt adherence and creative control, and the ability to "add sound effects, ambient noise, and even dialogue", generating all audio natively. Veo 3.1 is "our latest line of video generation models".

On Agent Platform, Veo 3.1 supports:

- Text-to-video, image-to-video, and video from first and last frames.
- Generating videos from reference ("ingredient") images for consistent subjects and styles.
- Extending existing videos and generating sound.
- Content Credentials (C2PA) to label AI-generated media.
- Quality and speed options: Veo 3.1 Generate, Veo 3.1 Fast and Veo 3.1 Lite PREVIEW for lower cost.

Typical use cases: short marketing and social clips, product demonstrations, storyboarding and previsualisation for film and games, training and onboarding videos, and personalised video at scale. Google also uses Veo in creative tools such as Google Flow (AI filmmaking) and Google Vids (video creation for work).

LEADER'S LENS

Generated media raises brand, copyright and trust questions. Put review workflows in place before publishing, use provenance features such as Content Credentials, and align usage with responsible AI policies (Section 4).

1.4.5 Matching Google's models to business scenarios

Scenario	Best fit	Why
Analyse a 500-page contract plus scanned exhibits and answer questions	Gemini (Pro or Flash tier)	Long context and multimodal document understanding
Classify 10 million support messages per day at minimum cost	Gemini Flash-Lite	Lowest latency and cost per request for simple, high-volume tasks
Run an assistant offline on field technicians' tablets	Gemma (small E2B or E4B sizes)	Open weights, on-device execution, no connectivity required
Keep patient data inside a hospital's own data centre while using a medical model	Gemma (for example MedGemma) self-hosted	Runs on owned infrastructure; customisable

Scenario	Best fit	Why
Create campaign images and edit backgrounds of product photos	Imagen (exam name); Gemini image models in current practice	Text-to-image generation and mask-based or conversational editing
Produce an 8-second product teaser with music and sound effects from a script	Veo	Text or image to video with native audio
Build an agent that reasons over multiple systems and long tasks	Gemini (Flash or frontier tier) on Agent Platform	Function calling, tool use and long-horizon reasoning

1.4.6 What is an AI agent?

Google Cloud defines AI agents as "software systems that use AI to pursue goals and complete tasks on behalf of users. They show reasoning, planning, and memory and have a level of autonomy to make decisions, learn, and adapt." Their capabilities are made possible "in large part by the multimodal capacity of generative AI and AI foundation models".

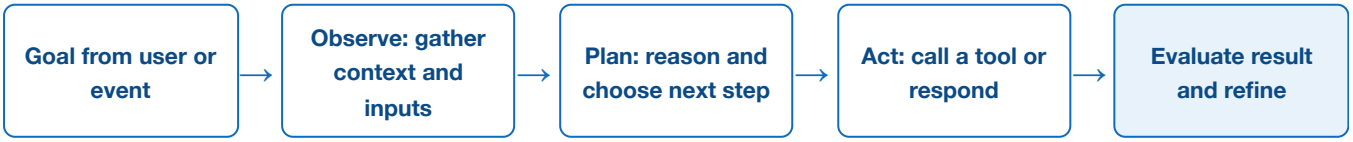
Google describes four building blocks of an agent:

Model	The LLM is the agent's "brain", providing the ability to understand, reason and act.
Tools	"Functions or external resources that an agent can utilize to interact with its environment", such as APIs, databases, search and code execution.
Memory	Short-term memory for the current interaction and long-term memory of past interactions, so the agent maintains context and improves (on Agent Platform: Sessions and Memory Bank).
Persona and instructions	The defined role, goals, communication style and rules that keep the agent's behaviour consistent.

	AI agent	AI assistant	Bot
Purpose	Autonomously and proactively perform tasks	Assist users with tasks	Automate simple tasks or conversations
Capabilities	Complex, multi-step actions; learns and adapts; can decide independently	Responds to requests; recommends actions, but the user decides	Follows predefined rules; limited learning
Interaction	Proactive, goal-oriented	Reactive to user requests	Reactive to triggers or commands

1.4.7 The agent loop: observe, plan, act

Google lists the key features of an agent as reasoning, acting, observing, planning, collaborating and self-refining, with reasoning and acting at the core (the ReAct pattern). In operation they form a loop that repeats until the goal is achieved or a human is needed:



Example: a "rebook disrupted passengers" agent **observes** the cancellation event and passenger records, **plans** which alternative flights meet each passenger's constraints, **acts** by calling the booking API and sending notifications, then checks results and escalates exceptions to staff.

Google also distinguishes agents by interaction: **interactive partners** (surface agents that converse with users, such as customer service agents) and **autonomous background processes** (background or workflow agents driven by events with limited human interaction).

1.4.8 Single-agent and multi-agent systems

A **single agent** operates independently on a well-defined task using its tools. A **multi-agent system** has "multiple AI agents that collaborate or compete to achieve a common objective or individual goals"; importantly, "each agent can have different foundation models that best fit their needs".

Benefit group	Multi-agent benefits Google lists
Efficiency and productivity	Increased output (agents divide tasks like specialised workers); simultaneous execution; automation of repetitive tasks
Improved decision-making	Collaboration and debate between agents; adaptability as situations change; more robust reasoning through feedback
Enhanced capabilities	Complex problem-solving by combining strengths; natural language communication; tool use; learning and self-improvement

LEADER'S LENS

Start with the simplest design that works: one well-scoped agent with a few tools and clear human checkpoints. Move to multi-agent designs when a process naturally splits into specialised roles (for example research, drafting and compliance review). Each agent needs an identity, least-privilege access to tools and audit logging; Agent Platform's Agent Identity, Agent Registry and Agent Gateway exist for this.

COMMON TRAP

A chatbot that only answers questions from a script is a **bot**, and an assistant that drafts while the user decides is an **assistant**. The word "agent" in an answer option implies goal-directed autonomy and tool use. Also note that the agent in reinforcement learning (1.1.14) is a learning concept, not the same as a gen AI agent product.

1.5 Additional practice questions

These original questions follow the style of the exam: a business scenario with one best answer unless the item asks you to choose more than one. Answers and explanations follow the last question.

Practice 1.1 MULTIPLE CHOICE

A national retailer has three years of purchase histories but no predefined customer categories. The marketing director wants to discover natural groupings of shoppers so that campaigns can be tailored to each group. Which machine learning approach best fits this goal?

A. Supervised learning using a classification model trained on campaign response labels
B. Unsupervised learning using clustering on the purchase data
C. Reinforcement learning that rewards the model for each completed purchase
D. Prompt tuning a foundation model with labelled customer segments

Practice 1.2 MULTIPLE CHOICE

A logistics company wants to feed real-time location events from 40,000 delivery vans into its analytics and ML pipeline. Events must be captured reliably even when downstream processing is temporarily slow. Which Google Cloud service is designed for this ingestion step?

A. Model Registry on Agent Platform
B. Feature Store on Agent Platform
C. Pub/Sub
D. Agent Studio

Practice 1.3 MULTIPLE SELECT (CHOOSE TWO)

A law firm wants a gen AI solution that reviews complete merger agreements of up to 900 pages together with scanned signature pages and exhibit images, and answers partners' questions about obligations across the whole set in a single request. Which two model characteristics are most important when selecting the foundation model?

A. A very large context window
B. The lowest available price per token
C. Multimodal input support for images and documents
D. Native video generation with audio
E. The ability to run on a mobile phone

Practice 1.4 MULTIPLE CHOICE

A manufacturer wants an AI assistant inside an air-gapped production facility with no internet connection. Engineering also wants to fine-tune the model on proprietary maintenance manuals and keep full control of the weights. Which Google model family is the best fit?

A. Gemini Flash-Lite accessed through the Agent Platform API
B. Veo
C. Imagen
D. Gemma

Practice 1.5 MULTIPLE CHOICE

A consumer brand needs dozens of short promotional clips with matching sound effects and background music, generated from text descriptions and a few product reference images. Which Google model family should the team evaluate first?

A. Gemma
B. Veo
C. Gemini embedding models
D. Imagen

Practice 1.6 MULTIPLE CHOICE

An insurer's claims-triage model performed well at launch, but accuracy has declined over six months after the company entered a new market with different customer behaviour. The model itself has not changed. Which capability would have alerted the team to this problem earliest?

A. Model Monitoring on Agent Platform tracking training-serving skew and inference drift
B. Data Labeling on Agent Platform
C. Increasing the model's context window
D. Moving the model to a larger TPU configuration

Practice 1.7 MULTIPLE CHOICE

A grounded product assistant gives contradictory answers about prices and delivery dates. Investigation shows that European systems store prices in euros with day-month-year dates, while North American systems use dollars with month-day-year dates, and both are loaded as sources. Which data quality characteristic is primarily at fault?

A. Completeness
B. Relevance
C. Consistency
D. Availability

Practice 1.8 MULTIPLE SELECT (CHOOSE TWO)

A bank is inventorying data sources for a gen AI pilot. Which two of the following are examples of unstructured data?

A. A relational table of account balances with fixed columns
B. Recorded customer service phone calls
C. Scanned, handwritten mortgage application forms
D. A data warehouse table of daily foreign-exchange rates

Practice 1.9 MULTIPLE CHOICE

A CIO wants her developers to have one managed environment where they can compare Google, third-party and open models, tune and evaluate them, ground them on company data, deploy agents and apply central security and governance. In the gen AI landscape, which layer is she describing, and which Google offering corresponds to it?

A. Infrastructure layer: AI Hypercomputer
B. Applications layer: Gemini for Google Workspace
C. Models layer: Gemini 4 Argon
D. Platforms layer: Gemini Enterprise Agent Platform

Practice 1.10 MULTIPLE CHOICE

A product team is prototyping a Gemini-based tool that writes product descriptions. Outputs are good but sometimes ignore the required format. The team has 30 approved example descriptions and a deadline in two weeks. Following Google's recommended approach, what should they do first?

A. Improve the prompt with explicit format instructions and a few of the approved examples, then evaluate
B. Run full fine-tuning on all model parameters using the 30 examples
C. Pre-train a new foundation model on the company's product catalogue
D. Switch to a diffusion model, which follows formats more reliably

Practice 1.11 MULTIPLE SELECT (CHOOSE TWO)

An operations director is considering replacing one large agent with a multi-agent system for procurement, where separate agents handle supplier research, contract drafting and compliance checks. Which two statements correctly describe benefits of the multi-agent design?

A. It guarantees that no agent will produce hallucinations
B. Specialised agents can work on different parts of the task simultaneously
C. It removes the need for access control because agents check each other
D. Each agent can use the foundation model that best fits its role

Practice 1.12 MULTIPLE CHOICE

A telecom provider wants to tag every incoming customer message by intent and urgency: about 15 million short messages a day, with responses needed in well under a second. The task is simple and the finance team is focused on cost per message. Which choice best balances the requirements?

A. The frontier Gemini model with the highest thinking level
B. Veo with audio generation disabled
C. A Gemini Flash-Lite model
D. Train a new LLM from scratch on the company's message history

Answers and explanations

1.1: B. No labels exist and the goal is to discover natural groupings, which is the definition of unsupervised learning; clustering is its classic technique for customer segmentation. A needs labels and predicts known categories. C learns actions from rewards in an environment, not segments. D is a model customisation method and would require segments that do not yet exist.

1.2: C. Pub/Sub is Google's asynchronous, scalable messaging service for ingesting streams of events, and it decouples producers from consumers so slow downstream processing does not lose data. Model Registry (A) manages model versions, Feature Store (B) serves prepared ML features, and Agent Studio (D) is a low-code agent and prompt environment.

1.3: A and C. Reasoning across a 900-page agreement in one request needs a very large context window, and scanned pages and exhibit images need multimodal input. Price (B) matters but is not the deciding characteristic for this high-value task. Video generation (D) and on-device execution (E) are irrelevant.

1.4: D. Gemma models have open weights, run on your own hardware without connectivity and can be fine-tuned with full control. A managed API (A) needs network access to Google Cloud. Veo (B) and Imagen (C) generate video and images, not assistant responses.

1.5: B. Veo generates video from text and reference images and produces native audio such as sound effects and music. Imagen (D) creates still images, Gemma (A) is a text-centric open model family, and embedding models (C) produce vectors for search.

1.6: A. The model has not changed, but production data has moved away from the training baseline: classic drift. Model Monitoring detects training-serving skew and inference drift and raises alerts so the team can retrain. Labelling (B) supports training, not detection. Context window (C) and compute (D) do not address changing data.

1.7: C. The same facts are represented differently across sources (currencies and date formats), which is a consistency failure. The data is present (so not completeness), on-topic (so not relevance) and reachable (so not availability).

1.8: B and C. Audio recordings and scanned handwritten forms have no predefined schema, so they are unstructured. Account balances (A) and exchange-rate tables (D) live in fixed-column relational tables and are structured.

1.9: D. A managed environment for model choice, tuning, evaluation, grounding, agent deployment and governance is the platforms layer, which on Google Cloud is Gemini Enterprise Agent Platform. AI Hypercomputer (A) is infrastructure, Gemini for Google Workspace (B) is an end-user application, and Gemini 4 Argon (C) is a single model.

1.10: A. Google recommends starting with prompting, then moving to fine-tuning only if needed to fix recurring errors. Clear format instructions and few-shot examples are fast and cheap. Full fine-tuning (B) is costly and 30 examples are fewer than Google's suggested "100 examples or more". Pre-training (C) is unrealistic, and diffusion models (D) are for image and video generation.

1.11: B and D. Google lists simultaneous execution by specialised agents and the ability for each agent to use the foundation model that best fits its needs as multi-agent benefits. No architecture guarantees zero hallucinations (A), and every agent still needs identity and least-privilege access control (C).

1.12: C. Gemini Flash-Lite is positioned by Google as the most cost-efficient tier for low-latency, high-volume, cost-sensitive traffic, and intent tagging is a simple classification task. The frontier model with high thinking (A) adds latency and cost the task does not need. Veo (B) generates video. Training an LLM from scratch (D) is slow, expensive and unnecessary.

Key takeaways

- AI is the umbrella; ML learns from data; deep learning uses multi-layer neural networks; generative AI creates new content; foundation models are large pre-trained models adaptable to many tasks; LLMs are text-focused foundation models; multimodal models handle several data types.
- Diffusion models generate images and video by learning to reverse a noising process; transformers underpin LLMs.
- Prompt engineering changes only the input and is always the first lever; prompt tuning and other parameter-efficient methods learn a small set of parameters with the base model frozen; full fine-tuning changes all weights at higher cost. Google recommends prompting first, tuning only if needed.
- Supervised learning needs labels (classify, predict), unsupervised finds structure in unlabelled data (cluster, associate), reinforcement learns actions from rewards (and aligns LLMs through RLHF).
- ML lifecycle and tools: ingestion (Pub/Sub, Cloud Storage, BigQuery, Dataflow, Managed Service for Apache Spark), preparation (BigQuery, Dataflow, Data Labeling, Feature Store, Knowledge Catalog), training (Managed Training, AutoML, BigQuery ML, tuning), deployment (Model Registry, Endpoints, Batch Inference, Agent Runtime), management (Model Monitoring, Pipelines, Evals).
- Choose models on modality, context window, security, availability and reliability, cost, performance and customisation; the smallest model that meets the bar usually wins.
- Gen AI value falls into create, summarise, discover and automate, across text, image, code, video, data analysis and personalised experiences.
- Data quality (completeness, consistency, relevance) and accessibility (availability, cost, format) decide outcomes; unstructured data is the majority of enterprise data and gen AI makes it usable; labelled data is costly and mainly needed for supervised learning, tuning and evaluation.
- The landscape has five layers: infrastructure, models, platforms, agents, applications. Engaging higher is faster; engaging lower gives more control and responsibility.
- Gemini: multimodal, long context, tiers from frontier/Pro through Flash to Flash-Lite. Gemma: open weights, run anywhere, fine-tune freely. Imagen: text-to-image and mask-based editing (Gemini image models are now Google's recommended image generators). Veo: text and image to video with native audio.
- AI agents pursue goals with a model, tools, memory and instructions in an observe, plan, act loop; multi-agent systems add specialisation and parallelism but need identity and governance for each agent.

Official Google references

- Generative AI Leader certification exam guide
- Generative AI Leader certification page
- Google Skills: Generative AI Leader Certification learning path
- Google Skills: Gen AI: Beyond the Chatbot
- Google Skills: Gen AI: Unlock Foundational Concepts
- Google Skills: Gen AI: Navigate the Landscape
- Google Skills: Introduction to Image Generation (diffusion models)
- Gemini Enterprise Agent Platform name changes from Vertex AI
- Agent Platform overview
- Google Cloud blog: Introducing Gemini Enterprise Agent Platform
- Gemini Enterprise Agent Platform product page
- Generative AI on Google Cloud

- Google models on Agent Platform
- Gemini 4 Argon on Agent Platform
- Gemini 3.1 Flash-Lite on Agent Platform
- Gemini API: Models
- Gemini API: Long context
- Google DeepMind: Gemini
- Gemma models overview
- Gemma 4 model overview
- Google DeepMind: Gemma
- Google DeepMind: Gemma 4
- Image generation with Gemini and Imagen overview
- Gemini API: Imagen 4
- Gemini API: Generate images using Imagen (legacy, migrate to Nano Banana)
- Agent Platform: Generate images with Gemini
- Google DeepMind: Gemini image models
- Google DeepMind: Veo
- Veo 3.1 on Agent Platform
- Generate videos with Agent Platform
- Artificial intelligence versus machine learning
- What is machine learning?
- What is supervised learning?
- What is unsupervised learning?
- What is reinforcement learning?
- What are foundation models?
- Large language models on Google Cloud
- Prompt engineering: overview and guide
- Fine-tuning LLMs and AI models
- Introduction to tuning on Agent Platform
- What is unstructured data?
- What is an AI agent?
- Architecture Center: MLOps, continuous delivery and automation pipelines in machine learning
- MLOps on Agent Platform
- Introduction to Model Registry
- Feature Store on Agent Platform overview
- Pub/Sub overview
- Dataflow overview
- Managed Service for Apache Spark on clusters overview
- BigQuery overview
- Knowledge Catalog overview
- Choose models and infrastructure for your generative AI application
- AI Hypercomputer
- Gemini Enterprise

SECTION 2

Google Cloud's gen AI offerings

Exam weight: ~35% of the exam

This is the largest section of the exam. It tests whether you can look at a business need and name the Google offering that meets it, explain why Google Cloud is a credible strategic partner for generative AI, and describe how agents use tools to act in the real world. You are not expected to configure anything. You are expected to recognise products by their purpose, users, and business value, and to avoid confusing products that sound alike.

Study this chapter in three passes. First, learn the **layers**: infrastructure (AI Hypercomputer, TPUs, GPUs), models (Gemini, Gemma, partner models in Model Garden), the developer platform (Agent Platform), and ready-made applications (Gemini app, Gemini Enterprise, Gemini for Google Workspace, Customer Engagement Suite). Second, learn the **decision tables** in each section, because most exam items are "a company needs X, which offering should it choose?". Third, learn the **vocabulary of agents and tools**: extensions, functions, data stores, plugins, and the pre-built AI APIs an agent can call.

Official objective (exam guide, Section 2)	Covered in
2.1 AI-first approach and commitment to future innovation	2.1.1
2.1 Enterprise-ready AI platform (responsible, secure, private, reliable, scalable)	2.1.2
2.1 Comprehensive AI ecosystem (gen AI integrated across Google products and services)	2.1.3
2.1 Benefits of Google Cloud's open approach	2.1.4
2.1 AI-optimised infrastructure (hypercomputer, TPUs, GPUs, data centres, cloud computing)	2.1.5
2.1 Control over data (security, privacy, governance, first-party and open models, pre-built and customisable solutions, agents)	2.1.6
2.1 Democratising AI development (low-code and no-code tools, pre-trained models, APIs)	2.1.7
2.2 Gemini app and Gemini Advanced (e.g., Gems)	2.2.1
2.2 Gemini Enterprise (Gemini Notebook API, multimodal search, custom agents)	2.2.2 to 2.2.5
2.2 Gemini for Google Workspace	2.2.6, 2.2.7
2.3 External search offerings (Agent Search on Gemini Enterprise Agent Platform, Google Search)	2.3.1, 2.3.2
2.3 Customer Engagement Suite (Conversational Agents, Agent Assist, Conversational Insights, Google Cloud Contact Center as a Service)	2.3.3 to 2.3.8
2.4 Agent Platform (Model Garden, Agent Search, Agent Platform AutoML)	2.4.1 to 2.4.4
2.4 RAG offerings (prebuilt RAG with Agent Search, RAG APIs)	2.4.5, 2.4.6
2.4 Using Agent Platform to build custom agents	2.4.7, 2.4.8

Official objective (exam guide, Section 2)	Covered in
2.5 How agents use tools (extensions, functions, data stores, plugins)	2.5.1 to 2.5.5
2.5 Google Cloud services and pre-built AI APIs for agent tooling	2.5.6 to 2.5.8
2.5 When to use Agent Studio and Google AI Studio	2.5.9

NOTE: PRODUCT NAMES IN THIS CHAPTER

In 2026 Google Cloud launched **Gemini Enterprise Agent Platform** as "the evolution of Vertex AI", and renamed many Vertex AI products. The official name-change page is the source of truth used here. The exam guide uses the new names, often shortened to "Agent Platform". Each product is introduced once as "current name (formerly ...)" and then referred to by the exam-guide name.

Current name (used in this book)	Formerly
Gemini Enterprise Agent Platform ("Agent Platform")	Vertex AI Platform / Vertex AI
Agent Studio on Gemini Enterprise Agent Platform ("Agent Studio")	Vertex AI Studio
Gemini Enterprise Agent Platform Model Garden ("Model Garden")	Vertex AI Model Garden
Agent Search on Gemini Enterprise Agent Platform ("Agent Search")	Vertex AI Search
AutoML on Gemini Enterprise Agent Platform ("Agent Platform AutoML")	Vertex AI AutoML
RAG Engine on Gemini Enterprise Agent Platform ("RAG Engine")	Name unchanged (Vertex AI RAG Engine)
Vector Search on Gemini Enterprise Agent Platform; Agent Retrieval	Vertex AI Vector Search; Vertex AI Vector Search 2.0
Agent Runtime on Gemini Enterprise Agent Platform ("Agent Runtime")	Vertex AI Agent Engine
Gemini Enterprise Agent Platform Memory Bank and Sessions	Vertex AI Agent Engine Memory Bank and Sessions
Feature Store, Model Registry, Pipelines, Endpoints on Gemini Enterprise Agent Platform	Vertex AI Feature Store, Model Registry, Pipelines, Endpoints
Gemini Enterprise Agent Platform Evals	Gen AI evaluation service on Vertex AI
Gemini Enterprise	Google Agentspace
Gemini Notebook / Gemini Notebook Enterprise	NotebookLM / NotebookLM Enterprise
Customer Experience Agent Studio (Conversational Agents)	Dialogflow CX (the docs describe CX Agent Studio as its evolution)
Customer Experience Insights	Conversational Insights
Cloud Run functions	Cloud Functions

2.1 Google Cloud's strengths in the field of gen AI

Objective 2.1 is about the "why Google Cloud?" conversation that a leader has with a board, a CIO, or a sceptical business unit. The exam expects you to recognise seven themes and connect each to concrete evidence. Google itself summarises the argument as a **full-stack** advantage: "Our chips are designed for our models. Our models are grounded in your data. The agent is built on both, and secured by the infrastructure underneath."



2.1.1 AI-first approach and commitment to future innovation

Google describes itself as an AI-first company: AI research is not a bolt-on but the foundation of its products. Three facts support the claim and are useful in exam scenarios.

- **Research leadership.** Google DeepMind builds the Gemini family of models and co-designs hardware with Google Cloud. Google Cloud's launch of Gemini Enterprise explicitly credits "the pioneering research of Google DeepMind" and a "complete, full-stack approach to innovation".
- **Proven at planetary scale.** TPUs are described by Google Cloud as "the engine behind Gemini and Google's flagship billion-user products, including Search, Photos, and Maps". Enterprises buy the same technology that runs Google's own services.
- **Rapid, continuous delivery.** The cadence of new models (Gemini, Gemma, Imagen, Veo, Lyria) and platforms (Agent Platform, Gemini Enterprise, TPU generations) means customers inherit innovation without rebuilding. Google Cloud states that all future Vertex AI roadmap work is delivered through Agent Platform, so investing in the platform is investing in the roadmap.

Adoption is part of the evidence. Google Cloud reports that nine of the top ten AI labs and most AI unicorns use Google Cloud, and that a large share of its customers already use its AI products. For a leader, the message is: choosing Google Cloud is choosing a partner whose own business depends on AI working at scale.

> **LEADER'S LENS**
> "AI-first" is a strategic argument, not a feature. When you make it to a board, pair it with risk reduction: the same models, chips, and security controls that serve billions of Google users are offered to enterprises, and new capabilities arrive as managed services rather than as projects your teams must build.

2.1.2 An enterprise-ready AI platform

The exam guide lists five qualities. Learn them as a mnemonic (**R-S-P-R-S**) and attach evidence to each.

Quality	What it means for a business	Google Cloud evidence
Responsible	AI behaves in line with company values and regulation.	Google's AI Principles; configurable safety filters in Agent Studio; Model Armor protections against prompt injection and data leakage; Governance Policies with Content Protection and Semantic Governance; AI Content Detection API for identifying AI-generated media.
Secure	Systems resist attack and misuse.	Secure-by-design infrastructure; IAM; VPC Service Controls; Agent Identity (unique cryptographic ID per agent), Agent Gateway (central policy enforcement for tool calls), Agent Threat Detection and an Agent Security dashboard powered by Security Command Center.
Private	Your data stays yours.	Training Restriction in the Service Specific Terms; data residency, customer-managed encryption keys (CMEK), Access Transparency; options to reach zero data retention (see 2.1.6).
Reliable	Services are available and predictable.	24/7 enterprise support and SLAs for Agent Platform (not offered for the consumer Gemini Developer API); Agent Evaluation, Agent Simulation and Agent Observability to test and trace agent behaviour.
Scalable	The solution grows from pilot to global production.	AI Hypercomputer; TPU clusters of up to one million chips in a single logical training cluster; GKE clusters of up to 130,000 nodes; Agent Runtime with sub-second cold starts and long-running agents.

EXAM TIP

If a question asks why a **regulated** organisation (bank, hospital, public sector) should build on Agent Platform rather than a consumer tool, look for answers about enterprise terms, compliance certifications (for example HIPAA and SOC 2 support), IAM, data residency, CMEK, and SLAs. Google's own comparison says regulated customers should use Agent Platform rather than the Gemini Developer API.

2.1.3 A comprehensive AI ecosystem

"Comprehensive ecosystem" has two meanings on the exam.

1. **Gen AI is integrated across Google's products.** Gemini appears in Gmail, Docs, Sheets, Slides, Meet, Drive, Chat and Vids (Gemini for Google Workspace), in the Gemini app, in Gemini Enterprise, in contact-centre tools, in BigQuery and in security products. Users meet AI inside tools they already use, which lowers training cost and speeds adoption.
2. **A partner and integration ecosystem.** Gemini Enterprise launched with an ecosystem Google describes as "over 100,000 partners", over 100 connectors (Google Workspace, Microsoft 365, Salesforce, ServiceNow, Jira, Confluence, SAP and more), and an Agent Marketplace inside the Agent Gallery where employees can request partner-built agents (for example from Accenture, Oracle and ServiceNow).

The business benefit is **compounding value**: an agent built on Agent Platform can be published to employees through the Gemini Enterprise app, governed by the same identity and gateway controls, grounded in the same enterprise data, and surfaced alongside Workspace content. Each investment reinforces the others.

2.1.4 The open approach

Google Cloud positions openness as protection against lock-in and as freedom to use "the best model for the job". Know these five dimensions.

Dimension of openness	What Google Cloud offers	Business benefit
Model choice	Model Garden provides access to more than 200 models: Google first-party models (Gemini, Imagen, Veo, Lyria), Google open models (Gemma) and third-party and open models such as Anthropic's Claude family, Llama and others, with Hugging Face models scanned before deployment.	Pick models by cost, quality, latency or licence; switch without changing platform.
Open models	Gemma is Google's family of open models that customers can download, tune and run where they choose.	Run on-premises or at the edge; full control of weights.
Open frameworks and protocols	The Agent Development Kit (ADK) is open source and model-agnostic; Google contributed the Agent2Agent (A2A) protocol to the Linux Foundation in June 2025 with AWS, Cisco, Microsoft, Salesforce, SAP and ServiceNow; Gemini Enterprise supports bring-your-own Model Context Protocol (MCP) servers.	Agents built by different vendors can interoperate; skills are portable.
Open software on the infrastructure	TPUs support PyTorch and JAX natively and the vLLM inference engine; GKE is based on open-source Kubernetes.	Developers use familiar tools; workloads remain portable.
Multicloud and hybrid	Gemini Enterprise works with Microsoft 365 and SharePoint; Agent Platform is available on Google Distributed Cloud; Cross-Cloud Network connects workloads across clouds.	Use AI where the data already lives; no forced migration.

COMMON TRAP

"Open" does not mean "free" or "unmanaged". The open approach means **choice and interoperability** (models, frameworks, protocols, clouds). An answer that says Google Cloud requires you to use only Gemini models is wrong; so is an answer that says choosing third-party models removes enterprise governance. Partner models in Model Garden run under the same platform controls.

2.1.5 AI-optimised infrastructure

The exam names five components: the hypercomputer, Google's custom TPUs, GPUs, data centres, and cloud computing. Together they explain why Google can offer high performance at a competitive cost.

AI Hypercomputer
: Google Cloud's integrated supercomputing architecture that "brings together purpose-built hardware, open software, and flexible consumption models". It is not a single machine; it is a system design covering accelerators, storage, networking, software (for example Pathways, JAX, PyTorch, vLLM, GKE) and purchasing options, tuned to train, serve and operate AI.

Tensor Processing Unit (TPU)
: Google's custom-designed AI accelerator, built over more than a decade and co-designed with open software. TPUs run Gemini and Google's consumer services.

Graphics Processing Unit (GPU)
: General-purpose parallel accelerators. Google Cloud partners closely with NVIDIA and offers the latest GPUs (Google states it will be among the first to offer instances based on NVIDIA Vera Rubin NVL72).

Goodput
: The share of accelerator time that does useful training work. Google cites up to 97% goodput on its training stack.

TPU generation (current at time of writing)	Positioning	Status on cloud.google.com/tpu
TPU 8t (8th generation)	Large-scale pre-training; up to 9,600 chips in a superpod; nearly 3x compute per pod versus the previous generation.	Announced, "coming soon"
TPU 8i (8th generation)	Post-training and low-latency inference; about 80% better performance per dollar than previous generations.	Announced, "coming soon"
Ironwood (7th generation)	Energy-efficient training, reasoning and inference; 9,216 liquid-cooled chips per pod; 42.5 exaFLOPS per pod; 4x per-chip performance over Trillium.	Generally available
Trillium (6th generation)	Training and inference; 4.7x peak compute per chip versus TPU v5e and 67% better energy efficiency.	Generally available

Data centres and cloud computing complete the picture. Google states that its data centres deliver six times more computing power per unit of electricity than five years ago, which matters for both cost and sustainability reporting. Cloud consumption models let organisations rent rather than build: Spot VMs (up to 91% savings for fault-tolerant jobs), Dynamic Workload Scheduler (up to 50% for jobs with flexible start dates) and committed use discounts (up to 50%).

EXAM TIP

TPU versus GPU is rarely an either-or question for a leader. The expected reasoning is: TPUs are Google's purpose-built accelerators that give price-performance and energy efficiency for large AI workloads; GPUs offer broad framework compatibility; AI Hypercomputer lets you use either, and Google notes you can switch between them (for example with native PyTorch support on TPUs) without rewriting code.

LEADER'S LENS

Most enterprises will never buy a TPU directly. They benefit indirectly: when they call Gemini through Agent Platform or use Gemini Enterprise, those services run on Google's infrastructure. The infrastructure argument is therefore about the **cost, speed and reliability of the services you consume**, as well as the option to train or serve your own models at scale if you need to.

2.1.6 Control over your data

Leaders are asked "will the AI vendor learn from our confidential data?" more than any other question. Google Cloud's answer has three layers.

1. **Contractual commitments.** Under the "Training Restriction" in the Google Cloud Service Specific Terms, "Google won't use your data to train or fine-tune any AI/ML models without your prior permission or instruction". Google states this applies to all managed models on Agent Platform, including GA and pre-GA models. For Gemini Enterprise, Google states that your data, "including prompts, outputs, and model tuning, is never used to train Google models or models for any other customer", that Google never sells customer data, and does not use it for advertising. For Gemini in Google Workspace, content stays within the organisation and is "not human reviewed or used for Generative AI model training outside your domain without permission".
2. **Technical controls.** Data residency, CMEK, Access Transparency, VPC Service Controls, IAM, audit logging, Model Armor, and permission-aware retrieval (Gemini Enterprise answers respect each user's

existing access rights in source systems).

3. **Governance and choice.** Organisation policies can restrict which Model Garden models may be used; Agent Registry curates approved agents and tools; Agent Gateway enforces which endpoints agents may reach. Customers choose among first-party and open models, pre-built and customisable solutions, and their own or Google's agents.

NOTE: ZERO DATA RETENTION NUANCES

Google publishes guidance on achieving zero data retention on Agent Platform. Some features store limited data by design: for example, Grounding with Google Search stores derived query logs for up to three days for debugging, and this cannot be disabled; Google recommends **Web Grounding for Enterprise** for customers who need zero data retention. Prompt logging for abuse monitoring can be exempted on request. Leaders should know that "not used for training" and "never stored" are different commitments.

COMMON TRAP

Do not assume consumer and enterprise products share the same data terms. The consumer Gemini app with a personal account, and the Gemini Developer API used via Google AI Studio, are governed by different terms from Agent Platform, Gemini Enterprise and Workspace business editions. When a scenario involves confidential corporate data, the correct answer is usually the enterprise offering.

2.1.7 Democratising AI development

Democratisation means that more people (not only machine learning specialists) can create value with AI. Google Cloud does this through a spectrum of tools.

Skill level	Tool	Example
Any employee, no code	Gemini app, Gemini in Workspace, Gemini Enterprise Agent Designer, Gemini Notebook	A marketing manager builds an agent in natural language that drafts campaign briefs from approved brand guidelines.
Business technologist, low code	Agent Studio, CX Agent Studio, AutoML, Gemini in AppSheet	An analyst trains a churn classifier on a table with AutoML; a CX lead builds a support agent on a visual canvas.
Developer, API	Pre-trained APIs (Vision, Speech-to-Text, Translation, Document AI, Natural Language), Gemini API on Agent Platform	A developer adds receipt extraction to an expense app with a single Document AI call.
Engineer, full code	ADK, Colab Enterprise, custom training, Agent Runtime	A platform team builds a multi-agent claims-handling system with custom tools.

Pre-trained models are the heart of democratisation: instead of collecting data and training from scratch, teams call a model that already understands language, images, speech or documents, and optionally adapt it with prompts, grounding or light tuning.

LEADER'S LENS

Democratisation shifts the bottleneck from scarce ML talent to **governance and use-case selection**. If everyone can build agents, you need an approval path (for example Agent Registry and Agent Gallery publishing), clear data-access rules, and a way to measure value. Plan these before you open the floodgates.

2.2 Prebuilt gen AI offerings that enable AI-powered work

Objective 2.2 covers products that people use directly, with no development effort. Three offerings matter: the **Gemini app** (individual assistant), **Gemini Enterprise** (the company-wide agentic front door), and **Gemini for Google Workspace** (AI inside productivity apps). The exam tests whether you can tell them apart.

2.2.1 The Gemini app, Google AI plans and Gems

The **Gemini app** (gemini.google.com and mobile apps) is Google's conversational AI assistant. It helps with writing, planning, learning, analysis and creative work. Core features listed on Google's plans page include image generation and editing, **Deep Research** (multi-step research reports), **Gemini Live** (spoken conversation), **Canvas** (co-editing documents and code), and **Gems**.

Gemini Advanced, named in the exam guide, was the premium tier of the Gemini app. Google now sells premium access as **Google AI plans**: Google AI Plus, Google AI Pro and Google AI Ultra, each giving progressively higher usage limits, access to more capable models, and extras such as Gemini in Gmail and Docs for personal accounts, Google Flow credits for video creation, and more storage. For exam purposes, treat "Gemini Advanced" as "the paid Gemini app experience with Google's most capable models and higher limits".

Gem	A saved, customised version of Gemini with its own instructions (and optionally reference files) for a repeated task, such as a "brand voice editor" or "interview coach". Gems remove the need to re-type long prompts and give consistent results. Gemini for Workspace business editions also include Gems for building "a team of AI experts" for repeatable tasks.
Skill (Gemini app)	Google's successor to Gems: reusable custom shortcuts invoked in chat (for example `/prep-for-meetings`) that can be stacked. Google states that support for Gems will be removed starting November (2026) for personal accounts and March 2027 for Workspace business and enterprise customers, with Gems migrated automatically to skills.

NOTE

The exam guide still refers to Gems. Answer Gem questions on the concept (persistent instructions plus context for a repeatable task). Be aware that the product is transitioning to skills, which serve the same purpose with more flexibility.

Business value. The Gemini app raises individual productivity: faster first drafts, research synthesis, brainstorming, learning new topics, and simple data analysis. Gems (and skills) turn one expert's best prompt into a reusable asset for a team. Used with a Workspace business account, the Gemini app inherits enterprise-grade data protections; used with a personal account it does not.

2.2.2 Gemini Enterprise (formerly Google Agentspace)

Gemini Enterprise (formerly Google Agentspace) is Google Cloud's "front door for AI in the workplace". The documentation defines it as "an intranet search, AI assistant, and agentic platform" that uses data from across the organisation. At launch Google described six components behind one chat interface:

1. **Gemini models**, the "brains" of the system.
2. A **no-code workbench** so any team can analyse information and orchestrate agents.
3. A **taskforce of pre-built Google agents**, such as Deep Research and Data Insights.
4. **Secure connections to company data** wherever it lives (Google Workspace, Microsoft 365, Salesforce, SAP and others).
5. A **central governance framework** to visualise, secure and audit agents.
6. An open **partner ecosystem**.

Since Google Cloud Next 2026 the Gemini Enterprise app has gained long-running agents that work for hours or days in secure cloud sandboxes, an **Inbox** for supervising agent activity, reusable **Skills**, **Projects** (shared team workspaces with agents), **Canvas** for co-editing Docs and Slides (with export to Microsoft Office formats), an **Agent Marketplace** within the Agent Gallery, and bring-your-own MCP integrations. The Gemini Enterprise app is built on Agent Platform, so Agent Identity, Agent Registry and Agent Gateway govern its agents too.

> **NOTE**
> Google's marketing page for Gemini Enterprise now presents the product simply as "Gemini" for business ("one universal agent for the way you work"), while the documentation and console continue to use "Gemini Enterprise". Some capabilities on that page are labelled early access.

Edition	Seats	Pooled storage and indexing per user
Gemini Enterprise Business	1–500 users	25 GiB
Gemini Enterprise Standard	1 or more	30 GiB
Gemini Enterprise Plus	1 or more	75 GiB
Gemini Enterprise Pay-as-you-go	1 seat minimum	Pay-as-you-go
Gemini Enterprise Frontline	150 or more users of Standard or Plus	2 GiB

Features differ by edition (for example, Frontline users can only use agents provisioned by administrators and cannot upload custom skills). You do not need to memorise edition matrices for this exam; you should know that editions exist so that organisations can match cost to user needs.

Typical use cases from Google's documentation: create images, videos and reports; handle administrative tasks ("help me book a conference room"); research, ideate and plan ("summarise our best-performing campaigns and give me three new ideas"); and learn about people and projects ("find all documentation related to our go-to-market strategy").

2.2.3 Gemini Notebook Enterprise and the Notebook API

Gemini Notebook (formerly NotebookLM) is an AI research and writing tool that grounds its answers in the sources you add: PDFs, Google Docs and Slides, website URLs, YouTube videos, audio and more. It produces summaries, Audio Overviews (podcast-style discussions), video overviews, mind maps, reports, slide decks, infographics, flashcards and other study aids. Google renamed NotebookLM to Gemini Notebook in July 2026.

Gemini Notebook Enterprise is the "highly compliant, enterprise-ready version" that runs in your Google Cloud project. Documents uploaded are subject to Google Cloud terms, data stays within the project, and administrators can apply CMEK, Model Armor, sensitive-data protection and usage audit logging. It is licensed alongside Gemini Enterprise; several editions let users create, publish and chat with notebooks.

The **Notebook API** (the exam guide's "Gemini Notebook API") lets developers manage notebooks programmatically: create, retrieve, list recently viewed, delete and share notebooks, and generate podcasts (Audio Overviews) through a standalone API method. The business value is automation at scale: for example, automatically creating a notebook for every new client engagement, populated with the contract and briefing documents, and shared with the account team.

> **EXAM TIP**
> If the need is "employees want to ask questions of a **specific, curated set of documents** and get source-grounded answers, summaries and audio briefings", think Gemini Notebook (Enterprise). If the need is "search and act across **all** company systems", think Gemini Enterprise. If the need is "create notebooks automatically from another system", think the Notebook API.

2.2.4 Multimodal, permission-aware enterprise search

Gemini Enterprise gives employees "a single, multimodal search interface offering permissions-aware access to enterprise information". Three ideas matter.

- **Unified:** one search box across Google Workspace, Microsoft 365, Confluence, Jira, SharePoint, ServiceNow, Salesforce, BigQuery and more than 100 connectors, without moving the data. Gemini indexes selected sources "while keeping data where it lives".
- **Multimodal:** it can process text, images, PDFs and structured data, and generate images and video.
- **Permission-aware:** answers respect each user's existing access rights in source systems. A user cannot see a summary of a document they could not open directly.

Answers can blend search with generation, ground on enterprise data and (depending on edition) on Google Search, and include citations. Under the hood, Gemini Enterprise uses the same Discovery Engine technology as Agent Search.

2.2.5 Custom agents, Agent Designer and the Agent Gallery

Gemini Enterprise lets organisations add their own agents in three ways.

Route	Who builds	How
No-code agents PREVIEW	Business users	**Agent Designer** (the current documentation calls this tool **Workflow Builder**): describe an agent in natural language or use a visual editor; combine generative steps with deterministic nodes and flows; add human-in-the-loop approval checkpoints; connect data sources and tools; test, publish and share with specific users or groups; schedule recurring runs.

Route	Who builds	How
High-code agents	Developers and IT	Build with ADK on Agent Platform, deploy to Agent Runtime, and publish into the Gemini Enterprise app for the business to use.
Partner agents	Third parties	Browse and request through the **Agent Marketplace** in the Agent Gallery; once approved they appear for users.

The **Agent Gallery** is "the unified destination for every agent your teams need, whether built by Google, created internally by your company, or discovered through our partners". Google-made agents include **Deep Research** (long-running, citation-backed research across web and internal data) and **Data Insights** (generates SQL and search queries across structured and unstructured data to explain the "why" behind business outcomes).

LEADER'S LENS

Gemini Enterprise answers the "agent sprawl" problem: rather than each department buying a separate AI tool, employees get one interface, IT gets one governance plane, and builders (citizen or professional) publish to one gallery. Measure success by adoption, time saved on high-volume tasks, and the number of governed agents in production, not by the number of pilots.

2.2.6 Gemini for Google Workspace

Gemini for Google Workspace puts generative AI directly inside the apps where people already work. Since January 2025, Google has included these AI features in Workspace Business and Enterprise editions rather than selling them as separate Gemini add-ons. Highlights listed by Google:

App	Example capabilities	Business value
Gmail	"Help me write" drafts and refines emails; side panel summarises threads and answers questions about the inbox.	Less time on routine correspondence; faster catch-up.
Docs	"Help me write" drafts, rewrites and changes tone; side panel summarises and answers questions about files.	Faster first drafts of proposals, policies, briefs.
Sheets	Side panel helps create tables, formulas and analyses from natural language.	Analysis without spreadsheet expertise.
Slides	Generate images and draft content for presentations.	Faster, more visual decks.
Meet	"Take notes for me" captures notes and action items; real-time speech translation for business customers.	Better meetings; multilingual collaboration.
Drive and Chat	Side panel summarises and finds content across files and conversations.	Knowledge retrieval in context.
Vids	Turn documents or presentations into videos with AI-generated scripts and voiceover.	Training and communications at lower cost.
Gemini app and Gemini Notebook	Included for Workspace users, with enterprise-grade protections and Gems.	A governed assistant and research tool for every employee.
AppSheet and Workspace Studio	Build apps or automations by describing a process in natural language.	Citizen automation of business processes.

Administrators can manage access to Gemini features (with more granular service-level controls in Enterprise editions), turn the Gemini app, Gemini Notebook and Vids on or off, and control meeting note-taking. Existing Workspace data protections, such as data loss prevention and data residency, apply automatically to AI features.

2.2.7 Choosing the right prebuilt offering

Business need	Best fit	Why not the others
An individual wants a general-purpose AI assistant for writing, planning and research.	Gemini app (with a Google AI plan for higher limits)	No company-wide data connections needed.
A team repeats the same prompt-heavy task (e.g., reformatting status reports).	A Gem (or skill) in the Gemini app	No development; reusable instructions are enough.
Employees want AI help while drafting emails, documents and meeting notes in the tools they already use.	Gemini for Google Workspace	The help must be in-app, in the flow of work.
Employees need one place to search and act across Workspace, Microsoft 365, Jira, Salesforce and other systems, with permission-aware answers and agents.	Gemini Enterprise	Workspace alone does not unify third-party systems; the Gemini app has no enterprise connectors.
A legal team wants source-grounded Q&A, summaries and audio briefings over a defined set of case files, with enterprise compliance.	Gemini Notebook Enterprise	The scope is a curated source set, not all company data.
A business unit wants to build its own no-code agent that follows an approval workflow.	Gemini Enterprise Agent Designer	No developers needed; governance included.
A company wants to embed AI in its own customer-facing product.	Not a prebuilt productivity tool: use Agent Platform (2.4) or Customer Engagement Suite (2.3)	Prebuilt offerings serve employees, not external products.

COMMON TRAP

Gemini for Google Workspace and Gemini Enterprise overlap but are not the same. Workspace puts AI **inside** Google's productivity apps. Gemini Enterprise is a **separate platform** that connects to many systems (including Microsoft 365) and hosts agents. A company that runs Microsoft 365 can still adopt Gemini Enterprise.

2.3 How Google Cloud's gen AI offerings improve the customer experience

Objective 2.3 moves from employees to **customers**. Two families of offerings matter: external search (Agent Search and Google Search grounding) and the Customer Engagement Suite for service and contact centres.

2.3.1 External search with Agent Search

Agent Search on Gemini Enterprise Agent Platform (formerly Vertex AI Search) helps organisations "build Google-quality search experiences for websites, structured and unstructured data". Google's product page identifies two opportunities:

1. **Better search experiences** on customer-facing websites, apps and intranets, moving "from frustrating keyword matching to modern conversational search experiences similar to Google's new generative search experience". This can be as simple as adding a search widget to a web page.
2. **Better generative AI applications**, by using Agent Search as an out-of-the-box RAG system that grounds model answers in enterprise data (covered in 2.4.5).

Out of the box, Agent Search provides natural language understanding and semantic search, synonym understanding, spell correction and auto-suggest, generative summaries and conversational (multi-turn) search, multimodal search, self-learning ranking, parsers for PDF, HTML and TXT, and controls such as boost and bury, filters and redirects. Summaries come with citations and links back to the sources.

Industry offerings are tuned for specific data: Agent Search for media (personalised content recommendations), Agent Search for healthcare (clinical data repositories), and commerce search. Commerce search has moved under customer experience branding as **AI Commerce Search on Gemini Enterprise for Customer Experience** (formerly Vertex AI Search for Commerce), providing search, browse and recommendations for retail sites.

Business value: customers find answers and products faster, which increases conversion and self-service, reduces support contacts, and improves satisfaction. Because it is a managed service, organisations get Google-quality relevance without building search infrastructure. Google states that customer data in Agent Search is not used to train models, and that the service meets compliance standards such as HIPAA, ISO 27000-series and SOC 1/2/3.

> **EXAM TIP**
> "Our website search returns irrelevant keyword matches; customers want to ask questions in natural language and get summarised answers with links" points to **Agent Search**. "Employees need one search across internal tools" points to **Gemini Enterprise**. Both share Google's search technology, but they serve different audiences.

2.3.2 Google Search as a grounding source

The exam guide lists "Google Search" alongside Agent Search as an external search offering. In the context of gen AI on Google Cloud this means **Grounding with Google Search**: connecting a model to "world knowledge and a wide possible range of topics using results from Google's search engine". Grounding reduces hallucinations, anchors responses to sources, and provides auditability through links.

Grounding source on Agent Platform	Use it when you need
Grounding with Google Search	Fresh, public, world knowledge (news, prices, events) beyond the model's training cutoff.
Web Grounding for Enterprise	Web knowledge with compliance controls suitable for highly regulated industries (and a zero-data-retention path).
Grounding with Google Maps	Location and geospatial context.

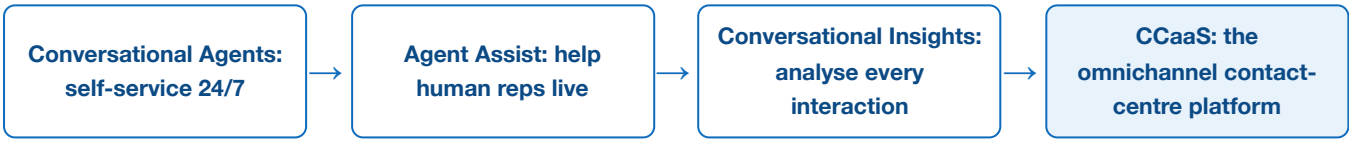
Grounding source on Agent Platform	Use it when you need
Grounding with Agent Search	Your own websites or document collections indexed in Agent Search.
Grounding with RAG Engine	Your private data through a configurable managed RAG service.
Grounding with Elasticsearch or your own search API	Existing enterprise search indexes you want to keep.

COMMON TRAP

Grounding with Google Search does **not** make your private documents searchable, and Agent Search does not give the model knowledge of today's public news. If a scenario needs both (for example, a financial assistant that compares internal research with today's market news), the answer combines enterprise grounding with web grounding.

2.3.3 Customer Engagement Suite: overview

Customer Engagement Suite is Google Cloud's conversational AI solution "for web, mobile apps, call centers and point of sale" that "works alongside your customer service reps to answer questions via chat and voice and take actions". It has four components named in the exam guide.



Google now also markets these capabilities under **Gemini Enterprise for Customer Experience (CX)**, "an agentic solution designed to bring shopping and customer service together on a single intelligent interface". Its documentation lists Customer Experience Agent Studio, Agent Assist, Customer Experience Insights, Commerce agents (such as a Shopping agent PREVIEW and a Food Ordering agent) and AI Commerce Search. The CCaaS product page still refers to "Customer Engagement Suite building blocks". The exam uses the Customer Engagement Suite names, so learn those first and recognise the newer labels.

Customer evidence cited by Google includes Commerzbank (a chatbot handling over two million chats and resolving 70% of inquiries) and Mercari (a projected 500% ROI from reducing representative workloads by at least 20%).

2.3.4 Conversational Agents

Conversational Agents are AI-powered virtual agents that hold natural conversations with customers by chat or voice, answer questions, and take actions such as checking an order, rebooking a flight or updating an address. They provide 24/7 self-service and escalate to a human when needed.

The builder has evolved. **Customer Experience Agent Studio** ("CX Agent Studio") is described as "a minimal code conversational agent builder" that "represents the evolution of Dialogflow CX"; Dialogflow CX and Dialogflow ES documentation is now labelled "legacy conversational agents". CX Agent Studio is built on ADK and adds an intuitive visual builder for non-technical teams, AI-assisted building and evaluation (for example, generating an agent with Gemini and refining instructions), low-latency streaming voice, asynchronous tool calls that avoid awkward silences, and versioning with one-click rollback. Agents can mix **generative**

behaviour (flexible, natural answers) with **deterministic** flows (fixed steps for compliance-sensitive tasks such as identity verification or payments). Google states that next-generation agents support more than 40 languages and can be configured once for telephony, web, mobile, email and chat.

Business value: higher containment (more issues resolved without a human), lower cost per contact, round-the-clock availability, and consistent answers.

2.3.5 Agent Assist

Agent Assist "provides in-the-moment coaching and next-best action guidance to customer care representatives, helping them resolve customer issues faster and with greater accuracy". It supports human agents rather than replacing them.

- **Generative knowledge assist:** answers from a central knowledge base suggested during the call or chat.
- **Smart reply** and ready-to-send responses.
- **Real-time transcription** and automatic **summarisation** after the interaction (reducing after-call work).
- **Live translation** between representative and customer.
- **AI Coach** (step-by-step guidance) and **AI Trainer** (personalised role-play practice to onboard and upskill staff).

Google's product page claims Agent Assist can help service representatives handle 28% more conversations. Business benefits include lower average handle time, higher first-contact resolution, faster onboarding and reduced burnout and turnover.

2.3.6 Conversational Insights

Customer Experience Insights (formerly Conversational Insights) "analyzes real-time data from across your customer operations to provide operations managers and quality assurance teams with KPIs, inquiry topic categories to prioritize, and areas of improvement". It identifies **call drivers** (why customers contact you), **sentiment**, and trends, and lets managers explore service trends in natural language. In the CCaaS platform it works with Quality AI for quality management.

Business value: every conversation becomes data. Leaders can see which products generate complaints, which policies confuse customers, which representatives need coaching, and which topics are candidates for automation by Conversational Agents.

2.3.7 Google Cloud Contact Center as a Service (CCaaS)

Google Cloud Contact Center as a Service is "a turnkey, enterprise-grade, omnichannel contact center solution that is native to the cloud". Where the other three components are AI capabilities, CCaaS is the **contact-centre platform** itself: routing, queues, channels, agent desktop and workforce tools, with Google AI built in.

- Omnichannel engagement: voice (WebRTC and PSTN), chat, SMS, email and social, with web and mobile SDKs that embed support inside your app and preserve context when customers switch channels.
- AI-driven predictive routing and **visual IVR** (self-service menus on screen instead of by phone keypad).
- Co-browsing so representatives see what customers see.
- Workforce and quality management: recording, coaching, scheduling, forecasting, adherence.
- An embeddable Agent Adapter so representatives can work inside the system of record (for example the CRM), or a standalone agent desktop.

- Virtual agents built with Conversational Agents, with seamless escalation to humans.

Business value: Google lists higher CSAT, higher agent satisfaction, and lower costs through platform simplification, automation of routine tasks and a single system of record.

2.3.8 Choosing the right customer experience offering

Business need	Best fit
Customers cannot find products or answers using the website's keyword search.	Agent Search (or AI Commerce Search for retail catalogues)
A chatbot must answer questions about today's public events or prices.	Grounding with Google Search
Automate common customer requests 24/7 by chat and phone, escalating complex cases.	Conversational Agents (CX Agent Studio)
Human representatives need real-time suggested answers, transcription, translation and automatic call summaries.	Agent Assist
Managers need to know why customers call, how they feel, and where quality is slipping.	Conversational Insights (Customer Experience Insights)
Replace an ageing on-premises contact centre with a cloud-native omnichannel platform.	Google Cloud Contact Center as a Service
An online store wants an agent that guides shoppers from discovery to checkout.	Commerce agents in Gemini Enterprise for CX (Shopping agent, Preview) with AI Commerce Search

COMMON TRAP

Agent Assist helps **human** representatives; Conversational Agents talk to **customers** directly. Conversational Insights **analyses** conversations after (and during) the fact; it does not converse. CCaaS is the **platform** that hosts all of them. Exam distractors often swap these roles.

LEADER'S LENS

A mature contact-centre roadmap uses all four together: Insights finds the top call drivers, Conversational Agents automate the simple ones, Agent Assist speeds up the complex ones handled by people, and CCaaS ties channels, routing and reporting together. Start with Insights to build the business case from real data.

2.4 How Google Cloud empowers developers to build with AI

Objective 2.4 is about the developer platform. Leaders need to understand what their technical teams will use, what each component is for, and when a managed component saves months of work.

2.4.1 Agent Platform: the unified developer platform

Gemini Enterprise Agent Platform (formerly Vertex AI) is "a unified platform to build, deploy, govern, and optimize enterprise-grade AI agents and model-based solutions". Google calls it "the evolution of Vertex AI": it keeps model selection, model building, and agent building, and adds agent integration, DevOps, orchestration and security. All future Vertex AI services and roadmap are delivered through Agent Platform. It is organised into four pillars.

Pillar	Key components
Build	Agent Development Kit (ADK); Agent Studio (low-code canvas); Agent Garden (prebuilt agents and templates); Model Garden; RAG Engine; Vector Search; Managed Agents API
Scale	Agent Runtime (formerly Vertex AI Agent Engine); Sessions; Memory Bank; Code Execution in a sandbox
Govern	Agent Registry; Agent Identity; Agent Gateway with Model Armor; Governance Policies; AI threat and vulnerability scanning; AI Content Detection API
Optimize	Agent Evaluation; Agent Simulation; Observability (trace viewer); Agent Optimizer (suggests refined system instructions from clustered failures)

Agent Platform also retains the full predictive ML toolkit: Colab Enterprise and Workbench notebooks, Managed Training, Pipelines, Model Registry, Feature Store, Model Monitoring and Endpoints for online and batch inference.

> **NOTE**
> Some Google pages and code samples abbreviate the platform further as "Gemini platform". Treat "Agent Platform", "Gemini Enterprise Agent Platform" and "Gemini platform" as the same developer platform. Do not confuse it with the Gemini Enterprise **app** used by employees (2.2.2): the app is built on the platform.

2.4.2 Model Garden

Gemini Enterprise Agent Platform Model Garden (formerly Vertex AI Model Garden) is "an AI/ML model library that helps you discover, test, customize, and deploy models and assets from Google and Google partners". It offers more than 200 models in three categories:

Category	Description	Examples
Foundation models	Pretrained multitask large models that can be tuned or customised.	Gemini, Imagen, Veo, Lyria; partner models such as Claude offered as managed APIs (Model as a Service)
Fine-tunable models	Models you fine-tune with a notebook or pipeline.	Open models such as Gemma and Llama
Task-specific solutions	Prebuilt models, mostly ready to use, often customisable.	Models for specific vision, speech or language tasks

Advantages listed by Google: all models in one place, a consistent deployment pattern, built-in integration with tuning, evaluation and serving, and managed deployment that removes the difficulty of serving gen AI models. Administrators can set an **organisation policy** that allows only vetted models. Google scans its serving containers and partner checkpoints; Hugging Face models are scanned and unsafe ones are blocked.

Business value: freedom to choose the right model for each use case (quality versus cost versus latency), easy comparison, and no lock-in, all within enterprise governance.

2.4.3 Agent Search for developers

For developers, Agent Search is a **retrieval engine**: a managed index over websites, unstructured documents (for example PDFs in Cloud Storage) and structured data (for example BigQuery tables) that can power search apps, recommendations and RAG. Developers create a data store, connect it to a search or recommendations app, and call it through APIs or an embeddable widget. Behind the scenes, Agent Search uses the Discovery Engine API, which Gemini Enterprise also uses. Developers can also attach Agent Search as a grounding source for Gemini models in Agent Studio or ADK agents.

2.4.4 Agent Platform AutoML

AutoML on Gemini Enterprise Agent Platform (formerly Vertex AI AutoML) trains custom machine learning models on your own labelled data **with minimal or no code**. You supply data, choose the objective, and the service handles data splitting, feature engineering, model selection and hyperparameter tuning, then helps you evaluate, deploy and explain the model. Supported data types include tabular data (classification, regression and forecasting), images (classification and object detection) and other modalities.

AutoML is a **predictive** AI tool, not a generative one. It belongs in this section because it is part of Agent Platform's democratisation story: domain experts can build accurate, task-specific models (for example, predicting churn, classifying defect photos or forecasting demand) without data science teams.

Situation	Choose
A general task such as summarising, drafting or answering questions in natural language.	A foundation model (Gemini) with prompting and grounding
A standard perception task (OCR, translation, transcription, label detection) with no custom labels.	A pre-trained API (2.5.7)
A prediction or classification specific to your business, you have labelled data, and you lack ML specialists.	Agent Platform AutoML
Full control over architecture, framework and training code.	Custom training on Agent Platform

COMMON TRAP

Do not choose AutoML to "build a chatbot" or "generate marketing copy". AutoML creates predictive models from labelled examples. Generative tasks use foundation models from Model Garden.

2.4.5 Prebuilt RAG with Agent Search

Retrieval-augmented generation (RAG) retrieves relevant facts from a knowledge source and adds them to the prompt, so the model answers from your data rather than only from what it learned in training. RAG reduces hallucinations, keeps answers current, and enables citations.



Building that pipeline yourself is complex. Google states that "Agent Search functions as an out-of-the-box RAG system", handling "ETL, OCR, chunking, embedding, indexing, storing, input cleaning, schema adjustments, information retrieval, and summarization" in "just a few clicks". This is the exam's **prebuilt RAG with Agent Search**. Choose it when you want Google-quality retrieval with the least engineering and do not need to customise each step.

2.4.6 RAG APIs and RAG Engine

When teams need more control, Google offers component-level options. The exam calls these **RAG APIs**.

Option	What it gives you	Typical user
Agent Search builder APIs (DIY grounding)	The components of Agent Search's RAG system exposed as separate APIs: the **Document AI Layout Parser** (parse and chunk documents while preserving structure), the **ranking API** (re-rank retrieved passages by relevance), the **grounded generation API** (generate answers grounded in supplied facts or sources), and the **check grounding API** (score how well a response is supported by given facts).	Teams that want to plug Google's best components into their own pipeline.
RAG Engine on Gemini Enterprise Agent Platform	A managed, configurable "data framework for developing context-augmented LLM applications". It handles ingestion (from local files, Cloud Storage, Google Drive), transformation (chunking), embedding, indexing into a **corpus**, retrieval and generation, with choices of parsers, embeddings and vector databases. Exposed through the RAG API.	Developers who want a managed RAG pipeline they can tune.
Vector Search and **Agent Retrieval** (formerly Vertex AI Vector Search and Vector Search 2.0)	A high-performance vector database for storing and searching embeddings at scale.	Advanced teams building custom retrieval, recommendations or ad serving.

Need	Choose
Fastest route to grounded answers over company documents and websites, minimal engineering.	Prebuilt RAG with Agent Search
Keep the existing pipeline but improve relevance of retrieved passages.	Ranking API
Verify that a generated answer is supported by the source facts before showing it.	Check grounding API
Complex PDFs with tables and headings are chunked badly.	Document AI Layout Parser
Managed but configurable RAG (choose chunking, embeddings, vector store) for a custom agent.	RAG Engine
Direct control of a vector database with custom embeddings.	Vector Search

2.4.7 Building custom agents on Agent Platform

An **AI agent** is an application that uses a model to reason about a goal, plan steps, call tools to act, observe results and continue until the goal is met. Agent Platform supports the whole agent lifecycle.

Stage	Agent Platform capability	Business value
Start from a template	**Agent Garden**: curated prebuilt agents and templates (for example code modernisation, financial analysis, invoice processing).	Faster time to first value.
Design (low code)	**Agent Studio**: prompt and agent design on a visual canvas, then export logic to ADK for deeper customisation.	Business and technical users collaborate.
Build (code)	**Agent Development Kit (ADK)**: open-source, model-agnostic framework in Python, TypeScript, Go and Java for single agents and multi-agent systems, with workflow agents for predictable pipelines and dynamic routing for adaptive behaviour, and a rich tool ecosystem.	Flexibility without lock-in.
Deploy and scale	**Agent Runtime**: managed runtime with sub-second cold starts and long-running agents that maintain state for days; ADK agents can also run on Cloud Run or GKE.	Production without managing infrastructure.
Remember	**Sessions** (context within an interaction) and **Memory Bank** (long-term memories across sessions).	Personalised, context-aware experiences.
Govern	**Agent Identity**, **Agent Registry**, **Agent Gateway**, Model Armor.	Least privilege, auditability, protection against prompt injection and data leakage.
Evaluate and improve	**Agent Simulation**, **Agent Evaluation**, **Observability**, **Agent Optimizer**.	Quality and safety before and after launch.
Deliver to users	Publish to the **Gemini Enterprise app**, embed in your products, or expose to other agents via A2A.	Reach employees and customers through governed channels.

Google cites customers such as Comcast (rebuilding the Xfinity Assistant as a multi-agent system with ADK and Agent Runtime), Color Health (a virtual cancer clinic assistant) and Payhawk (using Memory Bank so a finance agent remembers user habits, cutting expense submission time by over 50%).

LEADER'S LENS

Custom agents are justified when the workflow is differentiating, touches your own systems, or must be embedded in your products. If an off-the-shelf agent in Gemini Enterprise or a Customer Engagement Suite component already fits, buy before you build. When you do build, budget for evaluation, observability and governance from day one; these are what separate a demo from a production agent.

2.4.8 Build or buy: a decision table

Scenario	Recommended offering
Employees need an AI assistant across company data, quickly, with no development.	Gemini Enterprise (prebuilt)
A contact centre wants virtual agents and representative assistance.	Customer Engagement Suite (prebuilt, configurable)
A product team wants to add a Gemini-powered feature to its own mobile app with enterprise SLAs.	Gemini models on Agent Platform

Scenario	Recommended offering
Developers want to compare Gemini, Claude and Gemma for cost and quality on the same task.	Model Garden with Agent Platform Evals
A bank needs a multi-step agent that reads internal policies, checks a core system via API and drafts a decision for human approval.	Custom agent with ADK, RAG Engine or Agent Search, deployed on Agent Runtime, governed by Agent Identity and Gateway
A retailer needs to predict which customers will churn from historical tables, with no ML team.	Agent Platform AutoML

2.5 The purpose and types of tooling for gen AI agents

A foundation model on its own can only produce content from what it learned in training and what is in its prompt. It cannot check today's stock level, book a meeting or update a CRM record. **Tools** close that gap. Objective 2.5 asks you to explain how agents use tools, to recognise the tool types named in the exam guide, to match Google Cloud services and pre-built APIs to agent tasks, and to choose between Agent Studio and Google AI Studio.

2.5.1 How agents use tools

A common way to describe an agent is in three parts:

Model	The language model that reasons and decides what to do next (for example Gemini).
Orchestration layer	The loop that manages instructions, memory, planning and the cycle of reasoning, acting and observing (for example a ReAct-style loop). In Google Cloud this is provided by ADK, Agent Studio, CX Agent Studio or Gemini Enterprise's agent tools.
Tools	The connections that let the agent retrieve information or take actions in the outside world: APIs, functions, databases, search indexes and other agents.



Tools give agents two broad abilities: **knowledge** (fetch fresh or private information, such as policy documents or order status) and **action** (change something in the world, such as creating a ticket or sending an email). The exam guide names four tool types: extensions, functions, data stores and plugins.

2.5.2 Extensions

An **extension** is a standardised bridge between an agent and an external API that the **agent side executes directly**. The extension describes the API and gives examples of how to use it; the agent decides when to call it and the platform makes the call. Classic examples were a code interpreter extension (run model-generated code) and a search extension.

When to use: when you want the agent to call an API itself, with little or no custom application code, and the API is well defined.

NOTE: STATUS OF VERTEX AI EXTENSIONS

The specific product called Vertex AI Extensions was deprecated on 26 May 2026 and will be shut down after 26 November 2026; Google recommends migrating to Agent Platform. The **concept** of an agent-executed extension remains valid and appears in the exam guide. On Agent Platform, equivalent capabilities include built-in tools (such as Code Execution in a secure sandbox and grounding tools), ADK tool integrations, and MCP servers governed through Agent Registry and Agent Gateway.

2.5.3 Functions (function calling)

With **function calling**, the developer describes available functions (name, purpose, parameters) to the model. When the model decides a function is needed, it does **not** run it. It returns structured output naming the function and suggested arguments. The **application** (client side) executes the call, applies its own security and business logic, and sends the result back to the model, which then composes the answer.

Aspect	Extension	Function
Who executes the API call	The agent or platform	Your application code
Control and security	Platform-managed	Full developer control (authentication, validation, approval steps, private networks)
Good when	Standard APIs, quick integration	Sensitive systems, custom logic, human approval, APIs not reachable from the agent

Example: a travel agent asks "change my flight to Friday". The model outputs `change_booking(booking_id, new_date)`; the airline's backend validates the passenger, checks fare rules and performs the change, then returns the confirmation for the model to explain.

EXAM TIP

If a scenario stresses that the company must keep **control over when and how** a backend system is called (for example payments, regulated records, or calls that require human approval), function calling is the better fit, because execution stays in the company's own code.

2.5.4 Data stores

A **data store** gives an agent access to information beyond its training data, typically documents, websites or structured records converted into a searchable index (often using vector embeddings). The agent retrieves relevant content at run time and uses it to ground its answer. This is RAG used as a tool.

On Google Cloud, data stores are created in **Agent Search** (websites, unstructured documents, structured data from BigQuery or Cloud Storage), in **RAG Engine** corpora, or in **Vector Search**. Gemini Enterprise connectors play the same role for employees, indexing Workspace, Microsoft 365, Jira and other sources with permission awareness.

When to use: the agent must answer from private, frequently changing or large bodies of knowledge, such as product manuals, HR policies or a support knowledge base.

2.5.5 Plugins

A **plugin** is a packaged, reusable integration that adds a set of capabilities (often tools plus their configuration and authentication) to an agent or an AI application, typically for a third-party service. Where a function is a single declared operation, a plugin bundles everything needed to use a service so that it can be installed and governed as a unit.

In today's Google Cloud products, this idea appears as **connectors and actions** in Gemini Enterprise (for example letting an agent create a Jira issue or send an email), **bring-your-own MCP servers** that expose tools from your own systems, prebuilt **integrations in ADK**, and partner agents available through the Agent Marketplace. Treat "plugin" on the exam as "a pre-packaged connector that extends what an agent can do with a specific system".

Agent needs to...	Tool type
Call a well-defined public API directly with minimal custom code	Extension
Trigger an operation in a sensitive backend where your code must validate and execute it	Function
Answer from thousands of internal documents that change weekly	Data store
Gain a ready-made set of actions for a SaaS application (for example a ticketing system)	Plugin (connector or MCP server)

2.5.6 Google Cloud services as agent tools

Agents become useful when they can reach enterprise data and logic. The exam guide lists these Google Cloud services.

Service	Role as an agent tool	Example
Cloud Storage	Object storage for files: documents, images, audio, video. A common source for data stores and RAG corpora, and a place to write outputs.	An agent reads contracts from a bucket, or saves generated reports.
Databases (for example Cloud SQL, AlloyDB, Spanner, Firestore, BigQuery for analytics)	Structured, authoritative records the agent queries or updates, usually through functions.	Look up an order status; write a case note; query sales figures.
Cloud Run functions (formerly Cloud Functions)	Event-driven, single-purpose serverless code; an easy way to wrap business logic or a legacy API as a callable tool.	A function that checks credit limits when the agent calls it.
Cloud Run	Serverless containers for APIs, services and whole agents; scales automatically, including to zero. ADK agents can be deployed here.	Host a custom pricing API or an MCP server the agent uses.
Agent Platform	Models, grounding, RAG Engine, Agent Search, Code Execution, Agent Runtime and governance; also hosts other agents an agent can delegate to.	An orchestrator agent calls a specialist summarisation agent.

2.5.7 Pre-built AI APIs as agent tools

Google Cloud's pre-trained APIs (listed by Google as "Pre-trained APIs on Gemini Enterprise Agent Platform") give agents specialised perception and language skills through a simple call, with no model training. They are often cheaper, faster or more deterministic than asking a general model to do the same task, and they return structured output.

API	What it does	Agent use case
Speech-to-Text API	Converts audio to text, in real time or from files, in over 125 languages; can use Chirp 3, Google's speech foundation model.	A voice agent transcribes a caller; meeting audio is turned into searchable text.
Text-to-Speech API	Converts text into natural-sounding speech with lifelike voices.	A phone agent speaks responses; accessible audio versions of content.
Translation API (Cloud Translation)	Translates text dynamically across many languages using neural machine translation, custom models and Gemini-powered Adaptive Translation.	A support agent replies in the customer's language.
Document Translation (Cloud Translation feature)	Translates whole documents (for example DOCX and native or scanned PDF) while preserving formatting; supports batch translation from Cloud Storage.	An agent translates a supplier contract and returns a formatted file.
Document AI API	Processors that classify, split and extract structured data from documents (invoices, receipts, forms, IDs); a generative custom extractor; the Layout Parser for RAG.	An accounts-payable agent extracts invoice fields before posting them.
Cloud Vision API	Image analysis: labels, objects, text (OCR), faces, landmarks, logos, explicit content detection.	An insurance agent reads damage photos or extracts text from a scanned image.
Cloud Video Intelligence API	Recognises objects, places and actions in stored and streaming video and extracts metadata at video, shot or frame level.	A media agent tags archive footage to make it searchable.
Natural Language API	Entity analysis, sentiment analysis, content classification and syntax analysis of text.	A feedback agent routes angry reviews to a human.

EXAM TIP

Match the verb in the scenario to the API: **transcribe** = Speech-to-Text; **speak** = Text-to-Speech; **translate text** = Translation; **translate a file keeping its layout** = Document Translation; **extract fields from forms or invoices** = Document AI; **analyse images** = Vision; **analyse video** = Video Intelligence; **sentiment or entities in text** = Natural Language.

COMMON TRAP

Cloud Vision OCR reads text from images, but if the need is to **understand a business document** (which number is the invoice total, which is the VAT) choose **Document AI**. Likewise, Translation API handles strings; whole files with formatting call for **Document Translation**.

2.5.8 The Google Cloud API Library

The **Google Cloud API Library** is the catalogue in the Google Cloud console where teams browse, learn about and **enable** Google APIs (including all the AI APIs above) for a project. Every API must be enabled in a project before an application or agent can call it, and access is then controlled with IAM and monitored with quotas and logs. For a leader, the API Library is the menu of capabilities that developers can add as agent tools without building models, and the control point where platform teams decide which APIs are available.

2.5.9 When to use Agent Studio and Google AI Studio

Both are browser-based studios for experimenting with Gemini models and building prompts, but they sit on different platforms with different terms.

Google AI Studio	"The fastest way to start building with Gemini". A web tool for the **Gemini Developer API**, signed in with a Google Account and using API keys. Ideal for individual developers, start-ups, students and rapid prototyping.
Agent Studio (formerly Vertex AI Studio)	The studio inside **Gemini Enterprise Agent Platform**: "a centralized, collaborative workspace for discovering models, refining system instructions, and optimizing prompts", with grounding (Google Search, Google Maps, RAG Engine, Agent Search, Elasticsearch), safety settings, structured output, low-code agent design, and export to ADK. Requires a Google Cloud project with billing.

Factor	Google AI Studio (Gemini Developer API)	Agent Studio (Agent Platform)
Account and authentication	Google Account; API key	Google Cloud account with terms and billing; IAM and service accounts; VPC options
Terms	Standard terms of service; does not count toward Google Cloud commitments	Enterprise-ready data processing, security and privacy terms; counts toward commitments; custom contracts
Support and SLAs	No enterprise support or SLAs	24/7 enterprise support and availability SLAs
Compliance	No compliance certifications; Google advises regulated customers to use Agent Platform	Supports HIPAA and SOC 2; data residency, CMEK, Access Transparency
Models and platform	Gemini models via the Gemini API	200+ models in Model Garden, tuning, evaluation, RAG Engine, Agent Search, agent deployment and governance
Best for	Learning, prototypes, hackathons, quick proofs of concept, consumer apps that do not need enterprise controls	Production enterprise solutions, sensitive data, regulated industries, integration with Google Cloud services

The two are designed as a path, not as rivals: Google provides a migration guide from Google AI Studio to Agent Platform, and the unified Google Gen AI SDK works with both, so a prototype can move to production with limited change (tuned models created in Google AI Studio must be retrained on Agent Platform).

COMMON TRAP

"Studio" names are easy to confuse. **Google AI Studio** is the developer playground for the Gemini Developer API. **Agent Studio** is on Agent Platform. **CX Agent Studio** builds customer service agents in the Customer Engagement Suite. **Agent Designer** (Workflow Builder) builds no-code agents inside the Gemini Enterprise app. Read the scenario's user (developer, CX team, business user) and data sensitivity before you choose.

LEADER'S LENS

Let teams experiment freely in Google AI Studio with non-sensitive data, but set a clear rule: anything involving customer or confidential data, or heading for production, moves to Agent Studio and Agent Platform where enterprise terms, IAM, compliance and SLAs apply.

2.6 Key terms

Agent Platform	Gemini Enterprise Agent Platform (formerly Vertex AI): Google Cloud's unified platform to build, scale, govern and optimise agents and models.
Agent Search	Formerly Vertex AI Search. Managed Google-quality search, recommendations and out-of-the-box RAG over websites, documents and structured data.
Agent Runtime	Formerly Vertex AI Agent Engine. Managed environment for deploying and scaling agents, including long-running agents.
Agent Garden	Library of prebuilt agents and templates on Agent Platform.
Agent Gallery	The place in Gemini Enterprise where users find Google-made, internal and partner agents.
ADK	Agent Development Kit: Google's open-source, model-agnostic framework for building single and multi-agent systems.
A2A	Agent2Agent protocol: an open protocol for agents from different vendors to communicate, now hosted by the Linux Foundation.
MCP	Model Context Protocol: an open standard for exposing tools and data to agents; supported by Gemini Enterprise (bring-your-own MCP) and governed via Agent Registry and Gateway.
AI Hypercomputer	Google Cloud's integrated AI supercomputing architecture of purpose-built hardware, open software and flexible consumption models.
TPU	Tensor Processing Unit: Google's custom AI accelerator (current generations Ironwood, TPU 8t and TPU 8i).
Model Garden	Library of 200+ first-party, partner and open models with consistent deployment, tuning and evaluation.
AutoML	No-code or low-code training of custom predictive models on your labelled data.
RAG Engine	Managed, configurable RAG framework on Agent Platform that builds corpora and retrieves context for generation.
Grounding	Connecting model output to verifiable sources (Google Search, Agent Search, RAG Engine, Maps and others) to reduce hallucinations and enable citations.

Gemini Enterprise	Formerly Google Agentspace. Company-wide AI front door: permission-aware multimodal search, assistant, prebuilt and custom agents, governance.
Gemini Notebook Enterprise	Formerly NotebookLM Enterprise. Source-grounded research and knowledge tool running in your Google Cloud project, with a Notebook API.
Gem	A saved custom version of Gemini with instructions and files for a repeated task (transitioning to skills).
Customer Engagement Suite	Conversational Agents, Agent Assist, Conversational Insights and CCaaS for customer service; now also marketed as Gemini Enterprise for Customer Experience.
Function calling	The model returns a structured request to call a declared function; the application executes it and returns the result.
Extension	An agent-executed bridge to an external API (the Vertex AI Extensions product is deprecated; the concept remains).
Data store	An indexed knowledge source an agent retrieves from at run time.
Training Restriction	Google Cloud contractual term: Google won't use customer data to train or fine-tune AI/ML models without prior permission or instruction.

2.7 Additional practice questions

These are original scenario-based items written from the official exam guide. Choose the best answer for each.

Practice 2.1 MULTIPLE CHOICE

A pharmaceutical company's legal team is concerned that confidential trial data sent to Gemini models through Gemini Enterprise Agent Platform could be used to improve Google's public models. You need to address the concern accurately. What should you tell them?

A. Google anonymises customer prompts before using them for model training, so no personal data is exposed.
B. Under Google Cloud's Training Restriction, Google won't use their data to train or fine-tune any AI/ML models without their prior permission or instruction.
C. Data is only protected if the company deploys open models such as Gemma on its own servers.
D. Data is protected only when the company uses Google AI Studio with an API key.

Practice 2.2 MULTIPLE CHOICE

A global manufacturer uses Microsoft 365, Jira, ServiceNow and Salesforce. Employees waste hours searching across these systems. Leadership wants one AI assistant that searches all of them, respects each user's existing permissions, and lets business teams create their own no-code agents. Which offering best fits?

A. Gemini for Google Workspace
B. Agent Search with a website widget
C. Gemini Enterprise
D. Google AI Studio

Practice 2.3 MULTIPLE CHOICE

An online retailer's site search returns poor keyword matches, and customers abandon the site. The retailer wants conversational, semantic search with AI-generated summaries and links to source pages, delivered quickly without building search infrastructure. What should you recommend?

A. Agent Search (with commerce-tuned search where product catalogues are central)
B. Agent Platform AutoML trained on product images
C. Conversational Insights
D. Cloud Natural Language API

Practice 2.4 MULTIPLE CHOICE

A telecom operator's contact centre has long handle times. Human representatives struggle to find answers during calls and spend minutes writing notes afterwards. You need to improve representative productivity without replacing them. Which component should you prioritise?

A. Conversational Agents
B. Agent Assist
C. Grounding with Google Search
D. Cloud Video Intelligence API

Practice 2.5 MULTIPLE CHOICE

A bank's head of customer operations wants to understand why call volumes rose 30% last quarter, which topics drive the most negative sentiment, and where quality is slipping, before deciding what to automate. Which offering provides this?

A. Agent Assist
B. Google Cloud Contact Center as a Service routing
C. Conversational Insights (Customer Experience Insights)
D. Model Garden

Practice 2.6 MULTIPLE CHOICE

A product team wants to compare Gemini, an Anthropic Claude model and Gemma on the same summarisation task, then deploy the winner with enterprise governance and the ability to restrict which models other teams may use. Which capability should they use?

A. Gemini app with a Google AI Pro plan
B. Gemini Enterprise Agent Platform Model Garden
C. Agent Platform AutoML
D. Cloud Vision API

Practice 2.7 MULTIPLE CHOICE

A logistics company has three years of labelled shipment records and wants to predict which deliveries will be late. It has no data scientists. You need a solution that business analysts can use with minimal code. What should you recommend?

A. Prompt Gemini in the Gemini app with a sample of records
B. Agent Search with a structured data store
C. Text-to-Speech API
D. Agent Platform AutoML using tabular data

Practice 2.8 MULTIPLE CHOICE

An insurer wants an internal assistant that answers adjusters' questions from 40,000 policy documents, with citations, as fast as possible and with the least engineering effort. Which approach is most appropriate?

A. Use prebuilt RAG with Agent Search over the policy documents
B. Fine-tune a model on all policy documents every month
C. Use Grounding with Google Search
D. Build a custom vector database and retrieval pipeline from scratch

Practice 2.9 MULTIPLE CHOICE

A team already has its own retrieval pipeline but wants to verify, before showing an answer to customers, how well each generated response is supported by the retrieved facts. Which Google Cloud RAG API addresses this need?

A. Ranking API
B. Document AI Layout Parser
C. Check grounding API
D. Speech-to-Text API

Practice 2.10 MULTIPLE CHOICE

A payments company is building an agent that can issue refunds. Compliance requires that the company's own backend validates every refund against business rules and records an audit trail before execution, and the model must never call the payments system directly. Which tool approach fits best?

A. Function calling, where the model proposes the refund call and the application executes it
B. An agent-executed extension calling the payments API directly
C. A data store containing past refunds
D. Grounding with Google Search

Practice 2.11 MULTIPLE CHOICE

An accounts-payable agent receives scanned supplier invoices in many layouts. It must extract supplier name, invoice number, line items and totals as structured fields before posting to the ERP. Which pre-built API should the agent call?

A. Cloud Vision API label detection
B. Natural Language API sentiment analysis
C. Translation API
D. Document AI API

Practice 2.12 MULTIPLE CHOICE

A hospital's innovation team built a promising prototype in Google AI Studio using an API key. They now want to use it with patient data in production. What should you recommend?

A. Keep using Google AI Studio but rotate the API key monthly
B. Rebuild the prototype as a Gem in the Gemini app
C. Move the solution to Agent Studio and Agent Platform for enterprise terms, IAM, compliance support such as HIPAA, and SLAs
D. Move the solution to Gemini Notebook for personal accounts

Practice 2.13 MULTIPLE CHOICE

A consulting firm wants each new client engagement to automatically get a notebook populated with the statement of work and briefing documents, shared with the engagement team, so consultants can ask grounded questions and generate audio briefings. Which capability enables the automation?

A. The Notebook API for Gemini Notebook Enterprise
B. Agent Platform AutoML
C. Cloud Video Intelligence API
D. Google Cloud Contact Center as a Service

Practice 2.14 MULTIPLE CHOICE

A CIO worries that committing to Google Cloud for gen AI will lock the company into a single model vendor and a single cloud. Which statement best reflects Google Cloud's open approach?

A. Agent Platform only supports Gemini models, which avoids integration complexity.
B. Openness means Google Cloud's AI services are free of charge.
C. Google Cloud offers first-party, open (such as Gemma) and third-party models in Model Garden, open-source frameworks such as ADK, open protocols such as A2A and MCP, and works with data in other clouds and Microsoft 365.
D. Openness means customers must self-manage all infrastructure to avoid dependency.

Practice 2.15 MULTIPLE CHOICE

A board member asks why Google Cloud can offer competitive price-performance for training and serving large models. Which answer is most accurate?

A. Google Cloud runs all AI workloads exclusively on CPUs to reduce cost.

B. AI Hypercomputer combines purpose-built hardware such as Google's custom TPUs and NVIDIA GPUs with open software and flexible consumption models, running in energy-efficient data centres.

C. Google Cloud requires customers to buy and host their own TPUs on-premises.

D. Price-performance comes mainly from limiting customers to small models.

Answers and explanations

2.1: B. The Training Restriction in the Service Specific Terms states Google won't use customer data to train or fine-tune AI/ML models without prior permission or instruction, and applies to managed models on Agent Platform. A is false (there is no "anonymise and train" practice). C is unnecessary; managed services are covered. D is wrong: Google AI Studio uses consumer-style terms and is not the enterprise option.

2.2: C. Gemini Enterprise provides permission-aware multimodal search across Google and third-party systems (including Microsoft 365), an assistant, and no-code agent building. A works inside Google's productivity apps but does not unify third-party systems. B is for public website search. D is a developer prototyping tool.

2.3: A. Agent Search delivers Google-quality semantic and conversational search with generative summaries and citations, often via a widget; commerce-tuned search (AI Commerce Search) suits product catalogues. B trains predictive models. C analyses contact-centre conversations. D analyses text but does not provide search.

2.4: B. Agent Assist supports human representatives with real-time knowledge suggestions, transcription and automatic summaries. A would automate conversations rather than help representatives. C adds web knowledge, not call assistance. D analyses video.

2.5: C. Conversational Insights identifies call drivers, sentiment, topics and quality issues for operations managers. A helps representatives during calls. B routes contacts but does not explain drivers. D is a model library.

2.6: B. Model Garden offers first-party, partner and open models in one place, with consistent deployment, evaluation integration and organisation policies to restrict model access. A is a consumer assistant. C trains predictive models. D is an image API.

2.7: D. AutoML trains custom predictive models on labelled tabular data with minimal code. A is not a reliable prediction system. B retrieves information rather than predicting outcomes. C synthesises speech.

2.8: A. Agent Search acts as an out-of-the-box RAG system that handles parsing, chunking, embedding, indexing, retrieval and summarisation with citations. B is costly and does not give citations or freshness. C grounds in public web data, not private policies. D is the high-effort path the scenario wants to avoid.

2.9: C. The check grounding API scores how well a response is supported by given facts. A re-orders retrieved passages by relevance. B parses and chunks documents. D transcribes audio.

2.10: A. With function calling the model only proposes a structured call; the company's application executes it after validation and logging, keeping control. B lets the agent call the API directly, contrary to the requirement. C supplies knowledge, not actions. D is irrelevant.

2.11: D. Document AI extracts structured fields from business documents such as invoices. A detects image labels, not invoice fields. B analyses sentiment. C translates.

2.12: C. Google advises regulated customers to use Agent Platform, which offers enterprise terms, IAM, HIPAA and SOC 2 support, data residency, CMEK and SLAs; Agent Studio is its studio. A does not change the terms. B and D are productivity tools, and D uses personal-account terms.

2.13: A. The Notebook API lets developers create, retrieve, list, share and delete notebooks (and generate podcasts) programmatically in Gemini Notebook Enterprise. B, C and D do not manage notebooks.

2.14: C. Google Cloud's open approach covers model choice (first-party, open and partner models), open frameworks and protocols, and multicloud and hybrid support. A is false (200+ models). B and D misdefine openness.

2.15: B. AI Hypercomputer integrates purpose-built hardware (TPUs, GPUs), open software and flexible consumption models on efficient data centres. A, C and D are incorrect.

Key takeaways

- Google Cloud's strengths are a full stack: AI Hypercomputer and TPUs (Ironwood GA; TPU 8t and 8i announced), Gemini and 200+ models in Model Garden, Agent Platform, and ready-made applications.
- Enterprise readiness means responsible, secure, private, reliable and scalable, backed by the Training Restriction, IAM, CMEK, data residency, Agent Identity and Gateway, Model Armor and SLAs.
- The open approach means model choice (including Gemma and partner models), open frameworks (ADK) and protocols (A2A, MCP), and multicloud and Microsoft 365 support, so customers avoid lock-in.
- For employees: the Gemini app (with Google AI plans and Gems, now becoming skills) for individuals; Gemini for Google Workspace for AI inside Gmail, Docs, Sheets, Meet and more; Gemini Enterprise (formerly Agentspace) for permission-aware multimodal search, agents and governance across all systems; Gemini Notebook Enterprise and its Notebook API for source-grounded research.
- For customers: Agent Search for website and app search; Grounding with Google Search for fresh public knowledge; Customer Engagement Suite components each have one job: Conversational Agents talk to customers, Agent Assist helps representatives, Conversational Insights analyses conversations, CCaaS is the contact-centre platform.
- For developers: Agent Platform (formerly Vertex AI) organised as Build, Scale, Govern, Optimize; Model Garden for model choice; AutoML for no-code predictive models; prebuilt RAG with Agent Search for speed; RAG APIs (Layout Parser, ranking, grounded generation, check grounding) and RAG Engine for control; ADK, Agent Studio, Agent Garden and Agent Runtime for custom agents.
- Agents act through tools: extensions (agent-executed), functions (application-executed), data stores (retrieval), and plugins (packaged connectors such as MCP servers).
- Match verbs to pre-built APIs: transcribe, speak, translate, translate documents, extract from documents, analyse images, analyse video, analyse text.
- Google AI Studio is for fast prototyping with the Gemini Developer API; Agent Studio on Agent Platform is for enterprise and regulated production work.

References (official Google sources)

- Generative AI Leader certification exam guide
- Gemini Enterprise Agent Platform name changes
- Introducing Gemini Enterprise Agent Platform (Google Cloud Blog)
- Agent Platform overview
- Gemini Enterprise Agent Platform product page
- Gemini Enterprise Agent Platform and zero data retention
- Introducing Gemini Enterprise (Google Cloud Blog)

- What's new in Gemini Enterprise (Google Cloud Blog)
- Gemini Enterprise product page
- What is Gemini Enterprise?
- Compare editions of Gemini Enterprise
- Gemini Enterprise Workflow Builder (Agent Designer) overview
- What is Gemini Notebook Enterprise?
- Create and manage notebooks (API)
- NotebookLM is now Gemini Notebook (Google Workspace Updates)
- Gemini subscriptions and Google AI plans
- Gemini Gems and skills overview
- Gemini AI features now included in Google Workspace subscriptions
- AI Hypercomputer
- Tensor Processing Units
- Google Cloud donates A2A to Linux Foundation (Google Developers Blog)
- Agent Search on Gemini Enterprise Agent Platform
- About Agent Search
- Agent Search documentation
- Grounding overview
- RAG Engine overview
- Overview of Model Garden
- AutoML training overview
- Agent Development Kit
- Agent Studio overview
- Migrate from Google AI Studio to Gemini Enterprise Agent Platform
- Gemini Developer API and Gemini platform API
- Google AI Studio
- Vertex AI release notes (Extensions deprecation)
- Gemini Enterprise for Customer Experience documentation
- Gemini Enterprise for Customer Experience product page
- Customer Experience Agent Studio
- Agent Assist
- Google Cloud Contact Center as a Service
- Speech-to-Text
- Text-to-Speech
- Cloud Translation
- Translate documents (Document Translation)
- Document AI
- Vision AI
- Video AI and Video Intelligence
- Cloud Natural Language
- Cloud Run functions

SECTION 3

Techniques to improve gen AI model output

Exam weight: ~20% of the exam

A foundation model is powerful straight out of the box, but it is not automatically accurate, current, fair, or aligned with your business. This section of the exam tests whether you can recognise *why* a model produces a poor answer and choose the *right lever* to fix it: a better prompt, grounding in trusted data, retrieval-augmented generation (RAG), tuning, a human checkpoint, a change of generation settings, or an operational practice such as evaluation, versioning, and drift monitoring.

The questions are scenario based and written for leaders. You will rarely be asked how to call an API. You will often be asked which technique best addresses a stated symptom ("the assistant does not know about last week's price change", "answers vary too much between runs", "the model invents policy clauses"), or which Google Cloud offering fits a stated constraint (minimal engineering effort, need for citations, regulated industry, private documents). Study this chapter by learning the **symptom to technique** mappings in the decision tables, then rehearse them with the practice questions.

PRODUCT NAMES USED IN THIS CHAPTER

Google has renamed its AI platform. Vertex AI Platform is now **Gemini Enterprise Agent Platform** (the exam guide shortens this to **Agent Platform**). In this chapter: **Agent Search** (formerly Vertex AI Search), **Agent Studio** (formerly Vertex AI Studio), **RAG Engine** on Gemini Enterprise Agent Platform (formerly Vertex AI RAG Engine), **Vector Search** (formerly Vertex AI Vector Search), **Feature Store on Gemini Enterprise Agent Platform** (formerly Vertex AI Feature Store), **Gemini Enterprise Agent Platform Evals** (formerly the Gen AI evaluation service on Vertex AI), **Model Registry on Gemini Enterprise Agent Platform** (formerly Vertex AI Model Registry), and **Model Monitoring** on the platform (formerly Vertex AI Model Monitoring). Older documentation, blog posts, and API names (for example `aiplatform` and `discoveryengine`) may still use the previous names.

Official objective (exam guide, Section 3)	Covered in
3.1 Identify common limitations of foundation models (data dependency, knowledge cutoff, bias, fairness, hallucinations, edge cases)	3.1.1 – 3.1.7
3.1 Describe Google Cloud-recommended practices to address limitations (grounding, RAG, prompt engineering, fine-tuning, human in the loop)	3.1.8 – 3.1.14
3.1 Recognise Google-recommended practices for continuous monitoring and evaluation (automatic model upgrades, KPIs, security patches and updates, versioning, performance tracking, drift monitoring, Agent Platform Feature Store)	3.1.15 – 3.1.22
3.2 Define prompt engineering and its significance for LLMs	3.2.1 – 3.2.2
3.2 Prompting techniques and use cases (zero-shot, one-shot, few-shot, role prompting, prompt chaining)	3.2.3 – 3.2.7

Official objective (exam guide, Section 3)	Covered in
3.2 Advanced prompting techniques and when to use them (chain-of-thought, ReAct)	3.2.8 – 3.2.10
3.3 Concept of grounding; first-party enterprise data, third-party data, world data	3.3.1 – 3.3.4
3.3 How RAG affects generated output	3.3.5
3.3 Google Cloud grounding offerings: pre-built RAG with Agent Search, RAG APIs, Grounding with Google Search	3.3.6 – 3.3.9
3.3 Sampling parameters and settings (token count, temperature, top-p, safety settings, output length)	3.3.10 – 3.3.16

3.1 Proactively overcoming foundation model limitations

3.1.1 Why foundation models have limitations

Large language models (LLMs) are trained on vast amounts of text and learn the statistical patterns and relationships between units of language. Google's prompt design guidance describes the result plainly: given some text (the prompt), a language model predicts what is likely to come next, "like a sophisticated autocompletion tool". That design explains almost every limitation in this section:

- The model generates the most *plausible* continuation, not a *verified* fact. Plausible but false content is a hallucination.
- The model only "knows" what was in its training data, frozen at a point in time. Anything newer, private, or rare is missing or weakly represented.
- Patterns in the training data, including social biases, are learned and can be reproduced or amplified.

Google's Responsible AI documentation for Gemini Enterprise Agent Platform lists the limitations you can encounter when using generative AI models. They include edge cases; hallucinations, grounding, and factuality; data quality and tuning; bias amplification; uneven language quality; the limited scope of fairness benchmarks; limited domain expertise; and limits on the length and structure of inputs and outputs. The exam guide groups these into six headline limitations, covered one by one below.

3.1.2 Data dependency

Every model output depends on data twice: the data the model was trained (or tuned) on, and the data you supply at run time in the prompt or retrieved context. Google's documentation notes that "the quality, accuracy, and bias of the prompt or data input into a model can have a significant impact on the quality of its responses" and that inaccurate input can lead to "suboptimal performance or false model outputs".

For a leader, data dependency has three practical implications:

- **Garbage in, garbage out still applies.** A RAG system built over outdated, duplicated, or contradictory documents will faithfully retrieve and repeat the problems. Data quality dimensions from Section 1 (completeness, consistency, relevance, availability, format) directly affect output quality.

- **Tuning data must reflect production.** Google's tuning guidance stresses that "high-quality, well-labeled data is crucial for good performance and better than quantity" and that tuning data "should reflect the prompt distribution, format and context the model will encounter in production".
- **Specialised domains need specialised data.** Google lists "limited domain expertise" as a limitation: for specialised, complex use cases, models should be tuned on domain-specific data, with meaningful human supervision where outcomes can materially affect individual rights.

3.1.3 The knowledge cutoff

A model's **knowledge cutoff** is the date after which it has no training data. Anything that happened later, such as new prices, policies, products, news, or regulations, is unknown to the model unless you supply it. A model asked about recent events may say it does not know, or worse, answer with outdated information as if it were current.

Google's Gemini API prompting guidance even recommends adding a clause to system instructions that tells the model its knowledge cutoff date and the current date, so that it handles time-sensitive questions correctly. The structural fix, however, is **grounding**: Google describes Grounding with Google Search as connecting Gemini to real-time web content so that it can "provide more accurate answers and cite verifiable sources beyond its knowledge cutoff". For private and recent internal information, the fix is RAG over your own data.

> **COMMON TRAP**
> Fine-tuning is *not* the right answer to a knowledge cutoff problem when information changes frequently. Tuning bakes knowledge into weights at a point in time and would need repeating at every change. Grounding and RAG retrieve current information at request time.

3.1.4 Bias

Google warns that generative AI models "can inadvertently amplify existing biases in their training data, leading to outputs that can further reinforce societal prejudices and unequal treatment of certain groups". Bias can enter through training data, tuning data, retrieved documents, prompt wording, and few-shot examples (for example, if every example of a "manager" is male, the model learns that pattern).

Mitigations combine several practices covered in this chapter: curating representative data (data dependency), writing neutral instructions and balanced examples (prompt engineering), evaluating outputs across user groups (evaluation), using safety filters and system instructions, and keeping humans accountable for consequential decisions (human in the loop). Responsible AI governance is covered further in Section 4.

3.1.5 Fairness

Fairness is about whether a system performs equitably for different people. Google documents two fairness-related limitations of its generative models. First, **language quality**: most benchmarks, including all fairness evaluations, are in English, and "generative AI models can provide inconsistent service quality to different users", for example performing worse for under-represented dialects or language varieties. Second, **fairness benchmarks and subgroups**: Google Research's fairness analyses focus on particular axes (gender, race, ethnicity, religion) using English data, so they do not provide an exhaustive account of all risks.

The leadership takeaway is that a vendor's fairness testing does not replace your own. You should evaluate the model on *your* users, languages, and tasks, track quality by segment as a KPI, and route high-impact decisions through human review.

3.1.6 Hallucinations

A **hallucination** is model output that is not factual: invented statistics, nonexistent product features, fake citations, or plausible-sounding policy clauses. Google's grounding documentation defines grounding as "the ability to connect model output to verifiable sources of information" and lists its first benefit as reducing "model hallucinations, which are instances where the model generates content that isn't factual".

Hallucinations are especially dangerous because they are fluent and confident. They are most likely when the model is asked about information it never saw (private data, recent events, niche facts), when the prompt is ambiguous, or when the model is pushed to answer rather than allowed to say "I don't know". Effective controls are layered:

- **Grounding and RAG** supply verifiable facts and citations (Section 3.3).
- **Prompt instructions** tell the model to answer only from supplied context and to say when the answer is not present.
- **Lower randomness** (for older models that honour sampling settings) makes factual tasks more predictable.
- **Verification**, such as the check grounding API, scores how well a response is supported by reference text.
- **Human review** for high-stakes content.

3.1.7 Edge cases

Google defines **edge cases** as "unusual, rare, or exceptional situations that are not well-represented in the training data", which can lead to "model overconfidence, misinterpretation of context, or inappropriate outputs". Examples: a customer writes in a mix of two languages; an invoice has an unusual layout; a claim combines two rare circumstances; a user enters an empty or malformed field.

Google's prompt health checklist addresses edge cases directly under "underspecified task": prompts should provide "a clear path for handling edge cases and unexpected inputs" and instructions for handling missing data. Google also notes a related limitation: models have maximum input and output token limits, and performance can suffer when input has an unusual or complex structure. Mitigations include explicit fallback instructions, few-shot examples that include tricky cases, evaluation datasets that deliberately include rare scenarios, tuning on representative examples, and human escalation paths.

3.1.8 Google Cloud-recommended practices: overview

Google Cloud's learning path for this certification describes "Google Cloud strategies to address the limitations of foundation models". The exam guide names five practices. Think of them as a toolkit ordered roughly from lightest to heaviest effort:

Practice	What it changes	Effort and cost	Typical first use
Prompt engineering	The instructions, examples, and context you send	Lowest; no data pipeline or training	Always start here
Grounding	Connects output to verifiable sources (web, maps, your data) with citations	Low to medium; often a configuration toggle	Freshness, factuality, auditability
RAG	Retrieves relevant facts from a knowledge base and adds them to the prompt at request time	Medium; managed options reduce effort	Private, changing enterprise knowledge

Practice	What it changes	Effort and cost	Typical first use
Fine-tuning	Adjusts model weights using labelled examples	Higher; needs curated labelled data and evaluation	Consistent style, format, domain behaviour that prompting cannot achieve
Human in the loop (HITL)	Adds human review, approval, or feedback at defined checkpoints	Ongoing operational cost	High-stakes, subjective, or regulated decisions

These practices are complementary, not mutually exclusive. A production customer-service assistant might use a well-engineered system instruction, RAG over the policy library, a tuned model for the brand's tone, and a human agent for refunds above a threshold.

3.1.9 Grounding (as a mitigation)

Grounding tethers output to specific data sources and "reduces the chances of inventing content". Google lists three benefits: it reduces hallucinations, anchors responses to your data sources, and provides auditability through grounding support (links to sources). It is the primary mitigation for knowledge cutoff and hallucination, and it is detailed in Section 3.3.

3.1.10 Retrieval-augmented generation (as a mitigation)

RAG is the most common technique for implementing grounding on your own data. Google's RAG Engine documentation states the problem it solves: LLMs "don't understand private knowledge, that is, your organization's data". RAG enriches the model's context with relevant private information at request time "so that the model can reduce hallucinations and answer questions more accurately". It addresses data dependency (you control the source), knowledge cutoff (update the index, not the model), and hallucination (answers cite retrieved facts). How RAG changes output, and the Google offerings for it, are covered in 3.3.5 – 3.3.9.

3.1.11 Prompt engineering (as a mitigation)

Clear instructions, relevant context, explicit output formats, few-shot examples, and fallback rules reduce ambiguity, improve consistency, and guide the model's handling of edge cases. Google recommends starting with prompting before considering tuning: "We recommend starting with prompting to find the optimal prompt. Then, move on to fine-tuning (if required) to further boost performances or fix recurrent errors." Section 3.2 covers techniques in depth.

3.1.12 Fine-tuning (as a mitigation)

Model tuning adapts a model to specific tasks by training it on a dataset of examples. On Agent Platform, the supported method for Gemini is **supervised fine-tuning**, which "adapts model behavior with a labeled dataset" by adjusting the model's weights to minimise the difference between its predictions and the labels. Google describes it as "a good option when you have a well-defined task with available labeled data", particularly for domain-specific applications where language or content differs significantly from the model's original training data. Typical tasks include classification, summarisation, extractive question answering, and chat.

Google lists the benefits of tuning over prompt design as deeper customisation and better performance on specific tasks, alignment with custom syntax and domain rules, more consistent results, and lower inference cost and latency because long instructions and few-shot examples can be removed from prompts. Google

also explains two approaches:

- **Parameter-efficient tuning** (also called adapter tuning) updates a relatively small subset of parameters. It is more resource efficient, faster, and works with smaller datasets.
- **Full fine-tuning** updates all parameters. It can achieve higher quality on highly complex tasks but demands more compute for tuning and serving, so it costs more.

Google's guidance on when prompting suffices versus when tuning is warranted:

Choose prompting when...	Choose fine-tuning when...
You have limited labelled data or cannot afford a lengthy tuning process	You have a sizeable labelled dataset (Google suggests thinking of 100 examples or more)
You are rapidly prototyping or establishing a baseline	The task is complex or unique and advanced prompting is not sufficient
Requirements change often	You need highly consistent format, style, or domain behaviour at scale, or want shorter (cheaper) prompts

EXAM TIP

Fine-tuning teaches **behaviour and style** (how to answer). RAG and grounding supply **knowledge** (what is true right now). If a scenario is about facts that change or private documents, prefer RAG or grounding. If it is about a consistent format, tone, terminology, or a specialised task that prompting cannot nail, prefer tuning.

3.1.13 Human in the loop (HITL)

Google Cloud defines human-in-the-loop machine learning as "a collaborative approach that integrates human input and expertise into the lifecycle of machine learning (ML) and artificial intelligence systems". Humans can label training data, evaluate model performance, and provide feedback that helps models improve (for example through active learning or reinforcement learning).

For generative AI agents, Google's Architecture Center describes a **human-in-the-loop pattern**: at a predefined checkpoint the agent pauses and waits for a person to approve a decision, correct an error, or provide input. Google recommends it "for tasks that require human oversight, subjective judgment, or final approval for critical actions", such as approving a large financial transaction, validating the summary of a sensitive document, or giving feedback on creative content. The trade-off: it improves safety and reliability but adds architectural complexity and operational cost.

HITL addresses limitations that technology alone cannot fully remove: residual hallucinations, bias in consequential decisions, unhandled edge cases, and accountability requirements.

LEADER'S LENS

Design HITL deliberately rather than as an afterthought. Decide *which* outputs need review (by risk tier, confidence, or value), *who* reviews them, and *how* reviewer corrections flow back into prompts, evaluation datasets, or tuning data. Measure reviewer workload as a KPI; if humans are rubber-stamping everything, the checkpoint is either unnecessary or not being taken seriously.

3.1.14 Matching limitations to techniques

This decision table is the core of objective 3.1. Learn the primary technique for each symptom, then the supporting techniques.

Limitation or symptom	Primary technique	Supporting techniques	Usually not the answer
Knowledge cutoff: needs recent public information	Grounding with Google Search	Prompt with current date; HITL for critical claims	Fine-tuning (stale again immediately)
Needs private or internal knowledge (policies, product catalogue, contracts)	RAG over enterprise data (Agent Search or RAG Engine)	Prompt instructions to answer only from context; citations	Larger model on its own
Hallucinations: invented facts or citations	Grounding / RAG with citations	"Answer only from the sources; say if unknown" prompts; check grounding API; HITL	Raising temperature
Bias in outputs	Representative data plus evaluation across groups	Neutral prompts and balanced examples; safety settings; HITL for consequential decisions	Assuming vendor testing is enough
Fairness: uneven quality by language or user group	Segment-level evaluation and monitoring	Tuning on under-represented data; human review	Single aggregate accuracy KPI
Edge cases: unusual inputs mishandled	Prompt engineering with explicit fallback rules and examples	Evaluation sets that include rare cases; tuning; escalation to humans	Ignoring because they are rare
Inconsistent format, tone, or domain terminology despite good prompts	Supervised fine-tuning	Few-shot examples; system instructions; structured output	RAG alone
Data dependency: poor or outdated source data	Data quality and governance (fix the source)	Refresh RAG index; data store sync schedules	Prompt tweaks alone
High-stakes or regulated decision	Human in the loop	Grounding for auditability; logging and evaluation	Fully autonomous agent

3.1.15 Continuous monitoring and evaluation: why it matters

Generative AI systems change even when your code does not. Model versions are updated and retired, user behaviour shifts, source documents age, and prompts are edited. Google's migration guidance notes that moving to a newer Gemini version usually needs few changes, but "some applications may require prompt adjustments", that "it's hard to predict these changes without first testing your prompts with the new version", and that "thorough testing is recommended before fully migrating". Continuous evaluation and monitoring is therefore a core operating discipline, not a one-off launch activity. The exam guide lists seven practices, covered in turn.



3.1.16 Automatic model upgrades

Google's Gemini models are offered in different version types. The Gemini API documentation describes them as follows:

Version type	Behaviour	Production suitability
Stable	Points to a specific stable model; stable models usually don't change	Google: "Most production apps should use a specific stable model"
Preview	May be used for production; may have more restrictive rate limits; deprecated with at least 2 weeks' notice	Use with care
Latest (alias)	Points to the latest release of a model variation; "hot-swapped with every new release"; 2 weeks' email notice before breaking changes	Automatic upgrades; convenient but behaviour can change
Experimental	Released to gather feedback; not stable; availability subject to change	Not suitable for production

On Gemini Enterprise Agent Platform, Google publishes a **model versions and lifecycle** table listing each model's release date, retirement date, and recommended replacement. Stable models are available for at least 12 months after release, retirement dates "won't be moved to an earlier date than what is listed", and short-term models get at least 45 days' notice before retirement. When a model approaches retirement, Google's advice is to update the application to the recommended replacement, test all mission-critical features, and deploy as normal.

The leadership trade-off: **automatic upgrades** (a "latest" alias) deliver quality improvements and fixes with no effort, but can change outputs unexpectedly. **Pinning** to a stable version gives predictability but requires a planned migration before retirement. Most enterprises pin in production and run evaluations against the new version before switching.

> **COMMON TRAP**
> "Always use the latest model alias in production so you get improvements automatically" sounds attractive but is not Google's recommendation for most production apps. Upgrades are good; *untested* upgrades are the risk.

3.1.17 Key performance indicators (KPIs)

Google Cloud's guidance on measuring gen AI success groups KPIs into five families. Use a balanced set; improving one can worsen another.

KPI family	Measures	Examples
Model quality	Accuracy and effectiveness of outputs	Computation-based metrics against a reference set; rubric-based quality and groundedness scores
System quality	Operational performance of the AI system	Latency, throughput, uptime, error rates, resource utilisation
Business operational	Impact on business processes	Handle time, resolution rate, cart size, time to complete a task

KPI family	Measures	Examples
Adoption	Whether people actually use the solution	Active users, usage frequency, query volume, feature uptake
Business value	Financial impact and ROI	Cost savings, revenue uplift, productivity value

Google notes that gen AI success "hinges heavily on changes in human behavior and acceptance", which is why adoption KPIs sit alongside technical ones. Measuring impact is also part of Section 4.

3.1.18 Security patches and updates

Gen AI applications have more moving parts than a model: SDKs and libraries, agent frameworks, connectors, data stores, and the hosting environment. Keeping these patched and up to date is part of the shared responsibility for a secure system; Google manages the underlying managed services and model serving, while you manage your application code, configurations, and dependencies. Model updates matter too: Google's safety documentation notes that configurable content filters are versioned with the model, and Google "doesn't update the configurable content filter for a previously released version of a model" but may update it for future versions. Moving off old model versions before retirement is therefore also a way of receiving improved safety behaviour.

Good practice is to treat model upgrades, SDK updates, and security patches as routine change: track release notes and retirement dates, apply updates through a tested pipeline, and re-run evaluations after each change. Secure AI practices, including Google's Secure AI Framework (SAIF), are covered in Section 4.

3.1.19 Versioning

Versioning makes changes traceable and reversible. Version every element that shapes output:

- **Models.** Model Registry on Gemini Enterprise Agent Platform lets you create multiple versions of the same model, view all versions in one place, see "exactly how they performed", and set a *default version* used for predictions. This applies to your own trained and tuned models.
- **Foundation model versions.** Reference a specific stable Gemini version in production (3.1.16).
- **Prompts.** Prompt management on the platform lets teams save and share prompts; treat prompt text and generation settings as versioned assets.
- **Data.** Track the snapshot of the RAG corpus, tuning dataset, and evaluation dataset used for each release.

With versioning, you can answer "what changed?" when quality drops, compare versions side by side, and roll back quickly.

3.1.20 Performance tracking with Gemini Enterprise Agent Platform Evals

Gemini Enterprise Agent Platform Evals (formerly the Gen AI evaluation service) "provides enterprise-grade tools for objective, data-driven assessment of generative AI models" and supports model migrations, prompt editing, and fine-tuning. Key ideas:

- **Adaptive rubrics** (recommended): the service generates a tailored set of pass or fail tests for each prompt, similar to unit tests, then validates the response against each rubric and reports a pass rate with rationales.
- **Static rubrics**: a fixed scoring rubric applied to every prompt.

- **Computation-based metrics**: deterministic metrics such as ROUGE or BLEU when ground truth is available.
- **Custom functions**: your own evaluation logic.

Evaluation datasets can be uploaded, sampled directly from production logs, or generated synthetically. Google's listed use cases map neatly onto this chapter: comparing model versions before a migration, finding the best model for your data, improving prompts with a tight feedback loop, evaluating fine-tuned models, and evaluating agents (including agent traces). Google stresses that this gives insights "which cannot be derived from public leaderboards and general benchmarks".

> **EXAM TIP**
> When a scenario asks how to decide whether a new model version, a new prompt, or a tuned model is *better for your use case*, the answer is to evaluate on your own representative dataset (Agent Platform Evals), not to rely on public benchmark rankings.

3.1.21 Drift monitoring

Drift is a deviation between the data a model sees in production and the data it was built or validated on. Google's Model Monitoring documentation uses a customer lifetime value model as its example: as customer habits change, the factors that predict spending change too, so features used in training "might not be relevant for making inferences today". Model Monitoring runs on demand or on a schedule, and alerts you when metrics exceed thresholds so that you can "re-evaluate or retrain your model". Its monitoring objectives include:

- **Input feature data drift**: changes in the distribution of input values compared with a baseline.
- **Output inference data drift**: changes in the distribution of the model's predictions.
- **Feature attribution drift**: changes in how much each feature contributes to predictions.

Model Monitoring v1 is generally available and is configured on Agent Platform endpoints; Model Monitoring v2 PREVIEW associates monitoring with a model version. Model Monitoring targets tabular predictive models. For generative AI applications, the same principle applies in a different form: watch for shifts in the kinds of questions users ask, the freshness of the knowledge base, groundedness and quality scores from sampled production traffic, safety-filter block rates, and user feedback, then re-evaluate when they move.

3.1.22 Agent Platform Feature Store

Feature Store on Gemini Enterprise Agent Platform (formerly Vertex AI Feature Store) is "a managed, cloud-native feature store service" that lets you manage feature data in BigQuery tables or views and serve the latest feature values online at low latency. BigQuery acts as the offline store, holding current and historical feature values that can be used for training, so there is no separate offline store to provision.

Why it appears under monitoring and evaluation:

- **Consistency.** A single governed source of features for training and serving reduces training-serving skew, a common cause of degraded performance.
- **Feature monitoring.** Registering features in the Feature Registry lets you monitor feature statistics and detect feature drift and anomalies.
- **Freshness for gen AI.** Low-latency serving of up-to-date features (for example a customer's current tier or recent orders) can supply structured context to personalise or ground generative responses.

LEADER'S LENS

Ask your teams four questions before any gen AI launch: Which model version are we pinned to, and when does it retire? What evaluation dataset and pass threshold gate a release? Which KPIs (quality, system, operational, adoption, value) are on the dashboard? Who is alerted when quality or input patterns drift, and what is the runbook?

3.2 Prompt engineering techniques

3.2.1 What prompt engineering is and why it matters

Google defines a **prompt** as "a natural language request submitted to a language model to receive a response back". Prompts can contain questions, instructions, contextual information, few-shot examples, and partial input for the model to complete. **Prompt design** is "the process of creating prompts that elicit the desired response from language models", and the "iterative process of repeatedly updating prompts and assessing the model's responses is sometimes called **prompt engineering**". Google describes prompt engineering as "a test-driven and iterative process": define the objective and expected outcome, write the prompt, test it, evaluate, and refine.

Why it matters for leaders:

- **It is the cheapest, fastest lever.** It requires no training data, infrastructure, or model change, so it is Google's recommended starting point before tuning.
- **It directly controls quality, safety, and consistency.** Google's safety guidance reports that "in many situations well-crafted system instructions are often more effective than safety filters at generating safe outputs".
- **It affects cost.** Long prompts consume input tokens on every call; a good prompt is complete but not bloated.
- **It is a business skill, not only a technical one.** Domain experts who know what "good" looks like are often the best prompt writers.

Google also notes that modern Gemini models "often perform well without the need for prompt engineering, especially for straightforward tasks", but "for complex tasks, effective prompt engineering still plays an important role".

3.2.2 Anatomy of an effective prompt

Google identifies two aspects that make a prompt effective: **content** (the model has all the information it needs) and **structure** (ordering, labelling, and delimiters that help the model parse that information). Google's prompt components are:

Component	Purpose	Example
Objective (required)	What the model should achieve	"Help students with maths problems without giving them the answer."
Instructions (required)	Step-by-step directions for the task	"1. Understand the problem. 2. Find where the student is stuck. 3. Give a hint."

Component	Purpose	Example
System instructions	Directives that apply across a whole session, sent before user input in a dedicated parameter	Role, tone, safety rules, output format
Persona	Who the model is acting as	"You are a maths tutor."
Constraints	What the model must and must not do (guardrails)	"Do not give the answer directly."
Tone	Style or voice	"Respond in a friendly, plain-English way."
Context	Information the model should use	Lesson plan, policy text, retrieved documents
Few-shot examples	Demonstrations of the desired output	Input and output pairs
Reasoning steps	Ask the model to explain its reasoning	"Explain your reasoning step by step."
Response format	Required output structure	JSON, table, Markdown, bulleted list
Recap	Repeat key constraints and format at the end	"Remember: hints only, Markdown format."

Google's **prompt health checklist** is worth knowing at a conceptual level. It flags writing issues (typos, ambiguity such as "write a brief summary" instead of "three sentences or fewer", undefined jargon, missing key information), instruction issues (conflicting, redundant, or irrelevant instructions; missing output format; missing role definition; and "overt manipulation" such as emotional pressure, which Google says no longer helps and often hurts), and system design issues (an underspecified task without edge-case handling, a task outside model capabilities, or too many tasks in one prompt).

EXAM TIP

If a scenario says the model's answers are vague or inconsistent, look first for an answer that makes the prompt **more specific**: a clear objective, explicit constraints, an output format, and examples. Vague prompts produce vague answers.

3.2.3 Zero-shot prompting

A **zero-shot** prompt gives the model an instruction with *no examples*. The model relies entirely on what it learned during training. Zero-shot works well for common, well-understood tasks (summarise, translate, draft an email) and is the natural first attempt.

Listing 3.1: Zero-shot prompt

```
Summarise the following customer complaint in two sentences
for a service manager. Focus on the problem and what the
customer is asking for.

Complaint: "I was charged twice for my March bill and the
chatbot kept sending me in circles. Please refund the
duplicate charge and confirm by email."
```

Use when: the task is common, the output format is flexible, and you want a quick baseline. **Limitation:** less control over exact format, labels, and style.

3.2.4 One-shot prompting

A **one-shot** prompt includes a *single* example of the desired input and output. One example is often enough to fix a format or style the model would not otherwise guess, such as a specific product description template.

Listing 3.2: One-shot prompt

```
Write a one-line product tagline in our house style.

Example
Product: Trail running shoe, waterproof, 240 g
Tagline: Light on your feet. Dry on every trail.

Product: Insulated steel water bottle, keeps drinks cold 24 h
Tagline:
```

Use when: you need to demonstrate a pattern quickly and have one clear exemplar. **Limitation:** the model may copy surface features of that single example too closely.

3.2.5 Few-shot prompting

Google defines **few-shot** prompts as prompts that contain a few examples, in contrast to zero-shot prompts that contain none. The model "attempts to identify patterns and relationships from the examples and applies them". Few-shot prompts "are often used to regulate the formatting, phrasing, scoping, or general patterning of model responses". The Gemini API guidance goes further: "We recommend to always include few-shot examples in your prompts."

Listing 3.3: Few-shot classification prompt

```
Classify each support ticket into exactly one category:
BILLING, TECHNICAL, ACCOUNT, or OTHER. Reply with the
category only.

<examples>
Ticket: "My invoice shows a charge I don't recognise."
Category: BILLING
Ticket: "The app crashes when I upload a photo."
Category: TECHNICAL
Ticket: "How do I change the email on my profile?"
Category: ACCOUNT
</examples>

Ticket: "I was billed after I cancelled my plan."
Category:
```

Google's best practices for few-shot examples:

- Always pair examples with **clear instructions**; without them, the model may pick up unintended patterns.
- Use **specific and varied** examples, with **consistent formatting**, and mark them up clearly (Google suggests XML-like tags).
- **Experiment with the number** of examples: too few may not change behaviour; too many can cause the model to *overfit* to the examples.

Use when: you need consistent labels, structured output (for example JSON with specific keys), or a particular tone. Few-shot prompting is often the step before fine-tuning; if hundreds of examples are needed to get consistent behaviour, tuning may be more efficient because it removes those examples from every prompt.

3.2.6 Role prompting

Role prompting assigns the model a persona ("You are an experienced insurance claims assessor"). Google's prompt components call this the **persona**: "who or what the model is acting as", noting you can also influence style and tone by specifying a persona. Google recommends placing role definitions in **system instructions** (or at the very beginning of the prompt), and its checklist flags a "missing role definition" as a common problem.

Listing 3.4: Role prompt using a system instruction

```
System instruction:
You are a patient IT help-desk analyst for a hospital.
Explain fixes in plain language for clinical staff who are
not technical. Never ask users to share passwords. If an
issue involves patient data, tell the user to call the
security desk.

User:
My screen says "certificate error" when I open the
rostering site. What should I do?
```

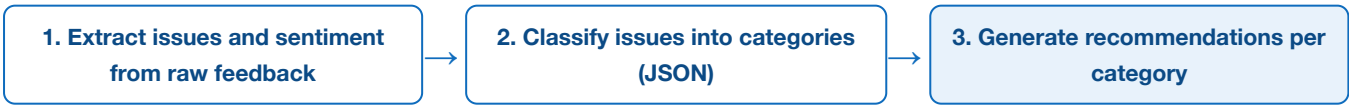
Use when: you need a consistent voice, an expert perspective, an audience-appropriate level of detail, or behavioural guardrails across many interactions (for example a customer-facing assistant). A role shapes style and focus; it does *not* give the model knowledge it lacks. A "senior tax lawyer" persona still needs grounding in current tax rules.

3.2.7 Prompt chaining

For complex tasks, Google recommends breaking prompts into subtasks because "smaller prompts can help you improve controllability, debugging, and accuracy". There are two patterns:

- **Chain prompts**: run subtasks *sequentially*; "the output of one prompt in the sequence becomes the input of the next prompt".
- **Aggregate responses**: run independent subtasks *in parallel* and combine their outputs in a final prompt.

Google's own example is a telecommunications company analysing customer feedback in three chained steps:



Listing 3.5: Prompt chain (two of three steps shown)

```
Prompt 1:
Extract the main issues and sentiment from the customer
feedback below. Return one issue per line.
Input: {CUSTOMER_FEEDBACK}

Prompt 2:
Classify each issue into one of: service reliability,
```

```
pricing, customer support, other. Return JSON with the issue
as key and the category as value.
Input: {OUTPUT_OF_PROMPT_1}
```

Use when: a single prompt is trying to do too many things at once (Google's checklist example is "summarize, extract entities, translate, and draft an email" in one pass); when intermediate results need checking, logging, or human review; or when different steps need different settings or models. **Trade-off:** more calls mean more latency and orchestration, but far easier debugging, because you can see which step failed.

3.2.8 Chain-of-thought (CoT) prompting

Chain-of-thought prompting, introduced by Google Research, "enables models to decompose multi-step problems into intermediate steps". Instead of asking for the answer directly, you ask (or show, through examples) the model to reason step by step before giving a final answer. Google Research showed that this markedly improved performance on arithmetic, commonsense, and symbolic reasoning tasks for sufficiently large models, without changing model weights.

Google's prompting guidance includes the same idea as a best practice: "instruct the model to explain its reasoning", which "can sometimes improve accuracy and nuance, especially for challenging queries". Because reasoning steps are part of the response, Google suggests using XML or other separators so the final answer can be parsed out.

Listing 3.6: Chain-of-thought prompt

```
A customer has 3 annual licences at 1,200 each and adds 2
more halfway through the year at a pro-rated price. We give
a 10% discount on the total if it exceeds 4,000.
What is the total for this year?

Think step by step. Show your working inside
<reasoning> tags, then give only the final amount
inside <answer> tags.
```

Use when: the task needs multi-step reasoning, such as calculations, logic, planning, policy interpretation, or diagnosing a problem; when you want an **auditable rationale** a reviewer can check; or when the model gets the right answer only inconsistently. **Trade-offs:** longer responses (more output tokens, cost, latency), and the reasoning is generated text that can itself contain errors.

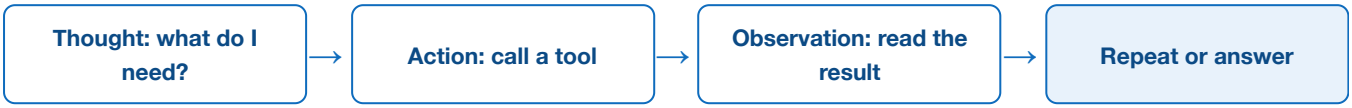
> **NOTE: THINKING MODELS**
> Current Gemini models include built-in **thinking**: they "generate an internal 'thinking' process before returning a response", which helps with "complex multi-step planning, solve mathematical problems, and generate accurate code". Thinking is enabled by default on supported models and can be tuned with a thinking level or budget. Explicit CoT prompting remains useful when you need the reasoning visible, structured, or reviewable.

3.2.9 ReAct (reason and act) prompting

ReAct, also from Google Research, "combines reasoning and acting". Google Research explains the motivation: with chain-of-thought alone, "a model is not grounded in the external world and uses its own internal representations", which limits its ability to update its knowledge. ReAct interleaves reasoning traces

with actions, such as calling a search tool or an API, so the model can "reason to act" and "act to reason". Its best results came from combining ReAct with CoT, using both internal knowledge and externally obtained information.

Google's Architecture Center describes ReAct as an agent design pattern: the agent runs an iterative loop of **thought** (reason about what to do next), **action** (use a tool, or give the final answer if the task is complete), and **observation** (receive and remember the tool output), until it finds an answer, reaches a maximum number of iterations, or hits an error.



Listing 3.7: ReAct-style trace (illustrative)

```
Question: Can we ship 40 units of SKU-2291 to the Lyon store
by Friday?

Thought: I need current stock for SKU-2291.
Action: inventory_lookup(sku="SKU-2291")
Observation: 55 units in the Paris warehouse.
Thought: Stock is sufficient. I need the transit time
Paris to Lyon.
Action: shipping_estimate(from="Paris", to="Lyon")
Observation: 2 business days.
Thought: Today is Tuesday; 2 business days arrives Thursday.
Answer: Yes. 55 units are available in Paris and delivery
takes 2 business days, so they can arrive by Thursday.
```

Use when: the task requires up-to-date or external information, multiple tool calls, or adaptive planning, such as agents that search, query systems of record, or take actions. Google recommends the ReAct pattern "for complex, dynamic tasks that require continuous planning and adaptation". **Trade-offs:** more model calls and latency, and an error in one observation can propagate to the final answer; pair it with tool permissions, iteration limits, and human checkpoints for consequential actions. ReAct underpins how agents use tools, which links to Section 2.5.

3.2.10 Choosing a prompting technique

Scenario signal	Technique	Why
Common task, quick baseline needed	Zero-shot	Fastest; relies on general training
Need to show one specific template or style	One-shot	A single example fixes the pattern
Need consistent labels, JSON keys, or tone across many inputs	Few-shot	Multiple varied examples regulate format and scope
Need a consistent persona, expertise level, or audience fit	Role prompting (in system instructions)	Persona steers voice and focus across a session
Complex workflow with distinct stages; hard to debug	Prompt chaining (or parallel aggregation)	Smaller prompts improve controllability, debugging, and accuracy
Multi-step reasoning, maths, or need an explainable rationale	Chain-of-thought	Decomposes the problem into intermediate steps

Scenario signal	Technique	Why
Needs external tools or live information within the reasoning	ReAct	Interleaves reasoning with actions and observations
Prompting cannot achieve required consistency despite iteration	Consider fine-tuning (3.1.12)	Bakes behaviour into the model

When results are still poor, Google's prompt iteration strategies suggest using different phrasing, switching to an analogous task, and changing the order of prompt content (for example examples, context, then input). Google also offers a prompt optimizer PREVIEW on Agent Platform to help optimise prompts for Google models.

COMMON TRAP

Do not confuse **chain-of-thought** with **prompt chaining**. Chain-of-thought is reasoning step by step *inside one response*. Prompt chaining is splitting a task into *multiple prompts* where each output feeds the next. Likewise, ReAct is distinguished from CoT by **actions with tools**: if the scenario needs the model to look something up or call a system mid-reasoning, the answer is ReAct.

3.3 Grounding techniques and their use cases

3.3.1 The concept of grounding

Google defines grounding as "the ability to connect model output to verifiable sources of information". When you give a model access to specific data sources, grounding "tethers their output to these data and reduces the chances of inventing content", which "is particularly important in situations where accuracy and reliability are significant". The three benefits Google lists are:

- **Fewer hallucinations**: less non-factual content.
- **Anchoring**: responses are tied to your chosen data sources.
- **Auditability**: grounding support provides links to sources, so users and reviewers can check claims.

Grounding is the *goal* (output tied to trusted sources); RAG is the most common *method* for achieving it with your own data; and Google Search grounding is a managed method for world knowledge. On Agent Platform, Google lists these grounding options: Grounding with Google Search, Grounding with Google Maps, Grounding with Agent Search, Grounding with RAG Engine, Grounding with Elasticsearch, Grounding with your own search API, Web Grounding for Enterprise, and Grounding with Parallel web search.

3.3.2 Grounding with first-party enterprise data

First-party enterprise data is data your organisation owns and controls: policies, product catalogues, contracts, knowledge-base articles, support tickets, CRM and ERP records, intranet pages, and data warehouses. Foundation models have never seen it, so grounding on it is the only way to get accurate, organisation-specific answers.

Typical use cases: an HR assistant answering from the current employee handbook; a sales assistant drafting proposals from the product catalogue and pricing rules; a support agent answering from troubleshooting guides; an analyst assistant summarising internal reports.

Google Cloud options: **Grounding with Agent Search** (connect Gemini to website data or document sets in Agent Search data stores; up to 10 data sources, combinable with Google Search grounding), **RAG Engine**, Grounding with Elasticsearch or your own search API if you already have a search system, and the Agent Search RAG building-block APIs. Key leadership concerns are access control (users should only receive answers from documents they are allowed to see), data freshness (sync schedules), and data quality (3.1.2).

3.3.3 Grounding with third-party data

Third-party data is data your organisation does not create but obtains from external providers, typically licensed, curated, specialist datasets such as financial ratings and market data, legal and regulatory content, or company intelligence. It is more authoritative and domain specific than the open web, and often not freely available on it.

Google Cloud has announced work with specialised providers including Moody's, MSCI, Thomson Reuters, and ZoomInfo to enable grounding on their datasets. Third-party sources can also be connected through Agent Search connectors, Grounding with Elasticsearch, or Grounding with your own search API.

Use cases: a wealth-management assistant grounded on licensed market and ratings data; a legal research assistant grounded on a licensed case-law service; a sales-prospecting agent grounded on company intelligence data. Leadership considerations include licensing terms (what the data may be used for, and whether outputs can be shown to customers), cost, provenance, and update frequency.

3.3.4 Grounding with world data

World data is broad, public, general knowledge, especially current information on the internet. **Grounding with Google Search** connects the model "with world knowledge, a wide possible range of topics, or up-to-date information on the Internet". It requires no index management: Google Search does the retrieval. **Grounding with Google Maps** adds geospatial context. For highly regulated industries, **Web Grounding for Enterprise** uses a web index that is "a subset of what's available on Google Search", does not log customer data, and supports VPC Service Controls; Google advises using standard Google Search grounding if you do not need those compliance controls, because it offers a broader and more up-to-date index.

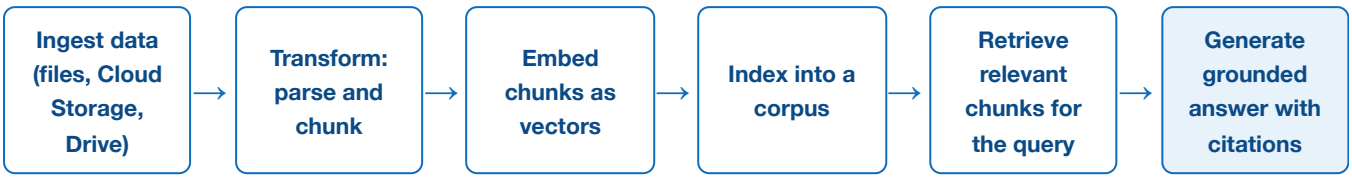
Data type	Examples	Solves	Typical Google Cloud option
First-party enterprise	Policies, catalogues, contracts, tickets, intranet	Private knowledge; organisation-specific accuracy	Agent Search (prebuilt RAG); RAG Engine; RAG APIs; your search API or Elasticsearch
Third-party	Licensed financial, legal, or business intelligence datasets	Authoritative specialist facts not in your data or on the open web	Partner datasets; connectors; your search API
World	Public web, news, maps, current events	Knowledge cutoff; general facts; freshness	Grounding with Google Search; Google Maps; Web Grounding for Enterprise

EXAM TIP

Read the scenario for **where the truth lives**. "Our internal documents" points to enterprise data (Agent Search or RAG Engine). "Recent public events" or "information beyond the model's training" points to Google Search. A combination ("our product docs plus current public news") can be met by combining Agent Search grounding with Google Search grounding, which Google documents as supported.

3.3.5 How RAG affects generated output

RAG has two stages, as Google's RAG API documentation explains: **retrieval** (quickly finding the most relevant facts) and **generation** (the model uses the retrieved facts to produce a grounded response). Google's RAG Engine documentation details the full pipeline:



Chunking	Splitting documents into smaller passages so that retrieval can return just the relevant parts.
Embedding	A numerical vector that captures the semantic meaning of text; similar meanings are close together in vector space, enabling semantic search.
Index / corpus	The searchable structure over your knowledge base; Google likens it to "a detailed table of contents for a massive reference book".
Retrieval	Finding the passages most relevant to the user's query, by keyword, semantic (vector), or hybrid search.
Re-ranking	Re-ordering retrieved candidates by how well each actually answers the query, before generation.

How RAG changes the output, in business terms:

Effect	Explanation
More factual and relevant	The model answers from retrieved facts instead of guessing; Google states that the improved context "enhances the quality of the LLM's response".
Current without retraining	Update or re-sync the index and answers change immediately; no model retraining.
Organisation-specific	Answers reflect your policies, products, and terminology.
Verifiable	Responses can include citations to source passages, supporting trust and audit.
Bounded by retrieval quality	If retrieval misses the right passage, or the source is wrong or outdated, the answer will be incomplete or wrong. RAG moves the quality problem to data and retrieval.
Cost and latency	Retrieved context adds input tokens and a retrieval step to every request.

COMMON TRAP

RAG does not make a model infallible. Poor chunking, stale documents, conflicting sources, or missing access controls produce poor or inappropriate answers. Combine RAG with prompt instructions ("answer only from the provided sources; if the answer is not there, say so"), re-ranking, grounding checks, evaluation, and data governance.

3.3.6 Pre-built RAG with Agent Search

Agent Search on Gemini Enterprise Agent Platform (formerly Vertex AI Search) is "a Google Search quality information retrieval and answer generation system that can be a component of any generative AI application that uses your enterprise data". Google positions it for two opportunities: improving search experiences across intranets and customer-facing websites, and grounding generative AI applications in enterprise data, where it "serves as an out-of-the-box system for retrieval augmented generation or RAG".

What "pre-built" means: Google states that Agent Search simplifies the end-to-end process of "managing ETL, OCR, chunking, embedding, indexing, storing, input cleaning, schema adjustments, information retrieval, and summarization to just a few clicks". It provides semantic and keyword search, processes many document types, and offers connectors to source systems including BigQuery and many third-party systems. It also has industry offerings (for example for media and healthcare data).

Ways to use it for grounding:

- Turn on **Grounding with Agent Search** (the "Your data" grounding option in Agent Studio or the API) so Gemini answers from your website or document data stores.
- Use Agent Search's own answer and summarisation features for search applications with generative answers and follow-ups.
- Add a search widget to a website for a conversational search experience.

Choose Agent Search when you want the fastest route to high-quality RAG over enterprise content with minimal engineering effort and no need to design the retrieval pipeline yourself.

3.3.7 RAG APIs: RAG Engine and the building-block APIs

When teams need more control than an out-of-the-box system, Google offers APIs at two levels.

RAG Engine on Gemini Enterprise Agent Platform is a component of the platform that facilitates RAG; Google describes it as "a data framework for developing context-augmented large language model (LLM) applications" and as "a fully-managed runtime for RAG orchestration" for production and enterprise contexts. It handles ingestion (for example from local files, Cloud Storage, and Google Drive), transformation and chunking, embedding, indexing into a corpus, retrieval, and grounded generation, while letting developers configure choices such as chunking and the vector database. It suits teams that want a managed pipeline with more configurability than Agent Search.

Building-block APIs let teams "build your own" RAG with full flexibility while relying on Google-managed components:

API or component	Role in RAG
Document AI Layout Parser	Transforms documents into structured representations (paragraphs, tables, lists, headings) and creates context-aware chunks
Embeddings API	Creates text or multimodal embeddings that capture meaning for semantic search
Vector Search	Retrieval engine that searches billions of semantically similar items with high QPS, high recall, and low latency; supports hybrid search
Ranking API	Re-ranks a list of documents by relevance to a query; gives more precise relevance scores than embedding similarity alone; stateless, so no indexing needed

API or component	Role in RAG
Grounded generation API	Generates well-grounded answers using specialised Gemini models; grounding sources can be Agent Search data stores, custom data you provide, or Google Search
Check grounding API	Determines how grounded a piece of text is in a set of reference texts, returns supporting citations, and can be used to assess the groundedness of RAG responses; contradicting citations are an experimental feature

Google's documented workflow for unstructured data is: import documents to Cloud Storage, parse and chunk with the layout parser, create embeddings, index and retrieve with Vector Search, rank chunks with the ranking API, and generate grounded answers with the grounded generation API or the generate content API. If answers come from a non-Google model, check their grounding with the check grounding API.

LEADER'S LENS

The RAG options form a build-versus-buy spectrum. Agent Search minimises time to value and engineering effort. RAG Engine offers a managed but configurable pipeline. Building-block APIs maximise control for teams with specialised requirements and the skills to own a custom pipeline. Choose the least complex option that meets your requirements for quality, control, data residency, and integration.

3.3.8 Grounding with Google Search

Use **Grounding with Google Search** "if you want to connect your model with world knowledge, a wide possible range of topics, or up-to-date information on the Internet". Gemini uses Google Search and generates output grounded in the relevant results; the response includes grounding metadata such as the search queries used and source links. Google's Gemini API documentation lists three benefits: increased factual accuracy, access to real-time information, and citations that build user trust.

Points worth knowing at a leadership level:

- It is a managed retrieval method: there is no index to build or maintain.
- Google's documentation states that to use Grounding with Google Search you must enable **Google Search Suggestions**, which are included in grounded responses.
- You can exclude specific domains and tailor results to a user's location.
- Google recommends a temperature of 1.0 for ideal results.
- There is a documented daily query limit (one million queries per day at the time of writing; contact Google Cloud support for more).
- It can be combined with Grounding with Agent Search, but Google Search cannot be combined with non-search tools (such as function calling or the RAG Engine retrieval tool) in the same request.

The Gemini API guidance also recommends enabling Google Search grounding "whenever the model may need to know obscure or recent facts".

3.3.9 Choosing a Google Cloud grounding offering

Requirement in the scenario	Best fit	Why
Answers about current public events or facts beyond the knowledge cutoff, with citations	Grounding with Google Search	World knowledge, real-time, no index to manage
Regulated industry needs web grounding with compliance controls (no logging of customer data, VPC Service Controls)	Web Grounding for Enterprise	Compliance-oriented web index
Answers from company documents or websites, fast, minimal engineering effort	Pre-built RAG with Agent Search	Out-of-the-box ingestion, chunking, embedding, retrieval, and answers
Managed RAG pipeline, but with control over chunking, embedding, and vector store choices	RAG Engine	Managed orchestration with configurability
Fully custom retrieval pipeline built by an experienced team	Building-block RAG APIs (layout parser, embeddings, Vector Search, ranking, grounded generation)	Maximum flexibility using managed components
Improve relevance of retrieved passages from an existing search or vector database	Ranking API	Re-ranks candidates by how well they answer the query
Verify how well an existing answer is supported by source documents	Check grounding API	Scores groundedness and returns supporting citations
Already have Elasticsearch or another search system	Grounding with Elasticsearch or with your search API	Reuse existing retrieval investment

3.3.10 Controlling model behaviour: tokens and token count

Every request includes parameter values that control how the model generates a response. Google groups them into token sampling parameters (temperature, top-p, top-k), stopping parameters (maximum output tokens, stop sequences), penalisation parameters, response format parameters, and safety settings. Before looking at each, understand **tokens**.

A **token** is the unit of text a model reads and writes. Google's rule of thumb: "A token is approximately four characters. 100 tokens correspond to roughly 60-80 words." Token counts matter for three reasons:

- **Cost**: pricing is typically based on input and output tokens.
- **Limits**: each model has a maximum context window (input) and output length; for example, current Gemini models on Agent Platform list a total context window of 1,048,576 tokens. Google also notes that if input or output exceeds the token limit, safety classifiers are not applied, which can lead to poor model performance.
- **Latency**: more tokens to read and generate means slower responses.

Long few-shot prompts, verbose system instructions, and large RAG contexts all increase input token count. This is one reason tuning can reduce inference cost: behaviour learned in the weights no longer needs to be repeated in every prompt.

3.3.11 Temperature

Temperature "controls the degree of randomness in token selection". Lower temperatures suit prompts that need "a less open-ended or creative response"; higher temperatures "can lead to more diverse or creative results". A temperature of 0 means the highest-probability tokens are always selected, so responses are "mostly deterministic, but a small amount of variation is still possible". Gemini models support values from 0.0 to 2.0 with a default of 1.0, and Google states that "1.0 is the recommended starting value". Google's troubleshooting tips: if responses are too generic, too short, or a fallback response, try increasing temperature; if the model gets stuck in repetitive or infinite generation, raising temperature to at least 0.1 may help.

3.3.12 Top-p (nucleus sampling)

Top-p changes how the model selects tokens: "Tokens are selected from the most probable to least probable until the sum of their probabilities equals the top-P value." Google's example: if tokens A, B, and C have probabilities 0.3, 0.2, and 0.1 and top-p is 0.5, the model chooses between A and B (using temperature) and excludes C. A lower value gives less random responses; a higher value gives more random responses. The Gemini API documentation gives a default top-p of 0.95.

3.3.13 Top-k

Top-k limits selection to the k most probable tokens. A top-k of 1 means the most probable token is always selected (also called **greedy decoding**); a top-k of 3 means selection from the three most probable tokens using temperature. Google describes the order of operations: the top-k tokens are sampled first, then further filtered by top-p, and the final token is selected using temperature sampling. Lower values give less random responses.

3.3.14 Output length: maximum output tokens and stop sequences

Maximum output tokens sets the maximum number of tokens that can be generated. "Specify a lower value for shorter responses and a higher value for potentially longer responses." It is a hard cap, not a target: if the cap is hit, generation stops (the finish reason is `MAX_TOKENS`) and the answer may be cut off mid-sentence. To get concise answers, combine a sensible cap with a prompt instruction such as "answer in three bullet points".

Stop sequences tell the model to stop generating when a specified string appears; the response is truncated where the string is first encountered, and strings are case-sensitive. Google advises choosing a sequence that is unlikely to appear in normal content. Stop sequences are useful for structured outputs, for example stopping after one record or one dialogue turn.

3.3.15 Safety settings

Google's models are designed to prioritise safety, but "they can still generate harmful responses, especially when they're explicitly prompted". Safety settings let you configure content filters to block potentially harmful responses. Google distinguishes three kinds of filter:

- **Non-configurable safety filters**, which always block child sexual abuse material (CSAM) and sensitive personally identifiable information (SPII).
- **Configurable content filters**, which score content in four harm categories (hate speech, harassment, sexually explicit, dangerous content) for both *probability* and *severity*, and block according to thresholds you set.
- **Citation filters**, which provide citations for source material and can stop generation for potential recitation.

Configurable thresholds include `BLOCK_LOW_AND_ABOVE` (strictest), `BLOCK_MEDIUM_AND_ABOVE`, `BLOCK_ONLY_HIGH`, `BLOCK_NONE` (no automated blocking; scores are returned so you can apply your own guidelines), and `OFF`. Google notes that `OFF` is the default for `gemini-3.5-flash` and later models, so organisations should deliberately decide their thresholds rather than assume strict blocking is on. When a filter blocks a response, the response content is cleared and the finish reason indicates why (for example `SAFETY`).

Google also emphasises that filters "act as a barrier" but "don't directly influence the model's behavior". **System instructions for safety** steer behaviour directly, and Google's testing shows that "in many situations well-crafted system instructions are often more effective than safety filters". The recommended approach is both: instructions to steer, filters as a backstop, plus safety testing, user feedback, and monitoring.

EXAM TIP

Safety settings match strictness to audience. A children's education app warrants strict thresholds; a platform for security researchers or clinicians may need less aggressive blocking so legitimate content is not refused. Safety settings are *not* a fix for hallucination: they target harmful content categories, not factual accuracy.

3.3.16 Parameter effects and recommended settings by use case

Setting	Lower value	Higher value	Business lever
Temperature	More focused, predictable, consistent	More diverse, creative, surprising	Consistency versus creativity
Top-p	Only the most likely tokens considered; less random	Wider pool of tokens; more varied	Narrow versus broad word choice
Top-k	Few candidates (1 = greedy); less random	More candidates; more varied	Narrow versus broad word choice
Max output tokens	Shorter, cheaper, may truncate	Allows long answers; more cost and latency	Length, cost, latency
Stop sequences	End output at a defined marker		Structure and predictable length
Safety thresholds	Block more (stricter)	Block less (more permissive)	Risk tolerance for the audience

Directional starting points derived from Google's guidance (always validate with evaluation on your own data):

Use case	Randomness (temperature, top-p, top-k)	Output length	Safety settings
Data extraction, classification, structured JSON	Low	Short cap; stop sequence after the record	Standard
Factual Q&A over policies (RAG)	Low to moderate; rely on grounding for accuracy	Moderate; ask for concise answers with citations	Standard to strict for public users
Grounding with Google Search	Google recommends temperature 1.0	Moderate	Standard

Use case	Randomness (temperature, top-p, top-k)	Output length	Safety settings
Marketing copy, brainstorming, product name ideas	Higher	Moderate	Standard
Long-form drafting (reports, articles)	Moderate	High cap	Standard
Customer-facing chatbot for a general or young audience	Low to moderate	Short to moderate	Strict, plus safety system instructions

NOTE: NEWER GEMINI MODELS

Google's current documentation states that for Gemini 3.6 Flash and later models (and Gemini 4), custom values for temperature, top-p, and top-k are not supported and are ignored, and that for earlier Gemini 3 models these should be kept at their defaults to avoid degraded reasoning. On these models, control output mainly through prompt design, system instructions, thinking level, output length, structured output, and grounding. The concepts above remain examinable and still apply to many other models.

COMMON TRAP

Lowering temperature makes a model more *consistent*, not more *correct*. A model with temperature 0 can repeat the same hallucination every time. Accuracy problems are solved with grounding, RAG, and better prompts; temperature controls variability.

3.4 Additional practice questions

These are original scenario-based questions written from the official exam guide. Choose the single best answer.

Practice 3.1 MULTIPLE CHOICE

A bank's internal research assistant, built on Gemini, is asked by analysts about a central bank interest rate decision announced yesterday. The assistant replies with the previous rate and presents it as current. You need to make the assistant answer accurately about recent public events and show sources, with minimal ongoing maintenance. What should you do?

A. Fine-tune the model each week on recent financial news articles.
B. Lower the temperature to 0 so the model answers more precisely.
C. Enable Grounding with Google Search for the assistant's requests.
D. Add a role prompt instructing the model to act as an expert economist.

Practice 3.2 MULTIPLE CHOICE

An HR team's chatbot confidently quotes leave entitlements that do not exist in the company's handbook. The handbook is a set of PDFs and intranet pages updated several times a year. The HR director wants accurate answers with links to the source pages, delivered quickly and with minimal engineering effort. Which approach best meets these requirements?

A. Use Agent Search as a pre-built RAG system over the handbook and ground the chatbot's responses in it.
B. Build a custom pipeline using the layout parser, embeddings, Vector Search, and the ranking API.
C. Fine-tune Gemini on the handbook text so it memorises the policies.
D. Increase the maximum output tokens so the model can give fuller explanations.

Practice 3.3 MULTIPLE CHOICE

An insurer uses Gemini to draft claim summaries. Despite detailed system instructions and several few-shot examples, summaries still drift from the insurer's mandatory structure and specialised terminology, and the long prompts are increasing cost. The team has 3,000 high-quality, reviewed example summaries. You need more consistent output and shorter prompts. What should you recommend?

A. Add more few-shot examples to every prompt.
B. Ground responses with Google Search.
C. Raise the temperature so the model explores better phrasings.
D. Use supervised fine-tuning with the labelled example summaries, then evaluate the tuned model.

Practice 3.4 MULTIPLE CHOICE

A lender uses a gen AI agent to draft loan decision letters. Regulators require that a qualified person is accountable for every adverse decision communicated to an applicant. Which practice most directly addresses this requirement?

A. Configure the strictest safety filter threshold for all harm categories.
B. Add a human-in-the-loop checkpoint where a credit officer reviews and approves adverse decision letters before they are sent.
C. Use chain-of-thought prompting so the letter includes the model's reasoning.
D. Pin the agent to a stable model version.

Practice 3.5 MULTIPLE CHOICE

A retailer's product-feed generator references a "latest" model alias. After a new model release, the output format changed slightly and broke a downstream parser overnight. Leadership wants to keep benefiting from model improvements without surprise production changes. What is the best practice?

A. Keep the "latest" alias and add more retries to the parser.
B. Pin production to a specific stable model version, and evaluate new versions against a representative dataset with Agent Platform Evals before switching.
C. Switch to an experimental model to get changes earlier.
D. Stop upgrading models and stay on the current version indefinitely.

Practice 3.6 MULTIPLE CHOICE

A marketing team uses a model that honours sampling parameters to brainstorm campaign slogans. Every run returns near-identical, generic slogans. The prompt is clear and includes the brand guidelines. Which change is most likely to produce more varied ideas?

A. Reduce maximum output tokens.
B. Set top-k to 1.
C. Increase the temperature.
D. Add a stop sequence after the first slogan.

Practice 3.7 MULTIPLE CHOICE

An operations team wants Gemini to route incoming emails into exactly one of six queue names, returned in a fixed format that a workflow tool can read. Zero-shot prompts sometimes invent new queue names or add explanations. What is the most effective next step?

A. Use a few-shot prompt with clear instructions and several consistently formatted examples covering the six queues.
B. Use ReAct prompting so the model can search the web for queue names.
C. Ground the model with Google Search.
D. Increase top-p to 1.0.

Practice 3.8 MULTIPLE CHOICE

A product team uses one large prompt that asks the model to summarise 500 survey responses, extract feature requests, categorise them, estimate priority, and draft a roadmap memo. Results are inconsistent and it is hard to tell which part is failing. What should you recommend?

A. Increase the temperature to encourage more complete answers.
B. Add a persona: "You are a world-class product manager."
C. Fine-tune the model on previous roadmap memos.
D. Break the task into a chain of prompts where each step's output becomes the next step's input.

Practice 3.9 MULTIPLE CHOICE

A logistics company is building an agent that answers "Can we fulfil this order by Friday?" To answer, it must check live inventory, query a shipping-time service, reason about the results, and possibly check a second warehouse. Which prompting approach best fits?

A. Zero-shot prompting.
B. ReAct prompting, interleaving reasoning with tool calls and observations.
C. Chain-of-thought prompting without tools.
D. One-shot prompting with an example order.

Practice 3.10 MULTIPLE CHOICE

A finance team asks Gemini to calculate pro-rated contract renewals with tiered discounts. All required figures are in the prompt, but the final totals are sometimes wrong, and auditors want to see how each total was derived. Which technique best addresses both needs?

A. Grounding with Agent Search.
B. Role prompting as a chartered accountant.
C. Chain-of-thought prompting that asks for step-by-step working in a separate section, followed by the final total.
D. Increasing maximum output tokens.

Practice 3.11 MULTIPLE CHOICE

A company's data science team built its own RAG pipeline using a third-party model for generation. Compliance wants an automated way to measure how well each generated answer is supported by the retrieved source passages, with citations showing which passages support which claims. Which Google Cloud capability fits best?

A. The ranking API.
B. Feature Store on Gemini Enterprise Agent Platform.
C. Grounding with Google Search.
D. The check grounding API.

Practice 3.12 MULTIPLE CHOICE

A subscription business runs a tabular churn-prediction model on Agent Platform whose scores feed a gen AI retention-offer generator. Over six months, customer behaviour has shifted and offer acceptance has fallen. You need to be alerted automatically when the distribution of the model's input features moves away from the training baseline. What should you use?

A. Model Monitoring with input feature drift objectives, thresholds, and alerts.
B. A lower temperature on the retention-offer generator.
C. Grounding with Google Search on the churn model.
D. A stop sequence on the offer generator's output.

Answers and explanations

3.1: C. The problem is the knowledge cutoff for recent public information. Grounding with Google Search connects Gemini to up-to-date web content and returns sources, with no index to maintain. Weekly fine-tuning (A) is costly and always stale. Temperature (B) affects variability, not knowledge. A persona (D) changes style, not facts.

3.2: A. Private, periodically updated documents plus "quick, minimal engineering, with sources" points to pre-built RAG with Agent Search, which handles ingestion, chunking, embedding, retrieval, and grounded answers. A custom pipeline (B) is possible but contradicts minimal effort. Fine-tuning (C) does not reliably provide citable, current facts. Output length (D) does not address hallucination.

3.3: D. Prompting has been tried and is insufficient, a large labelled dataset exists, and the goals are consistent structure and terminology plus shorter prompts. Google lists exactly these benefits for supervised fine-tuning, and recommends evaluating the tuned model. More examples (A) increase cost further. Search grounding (B) adds world facts, not house style. Higher temperature (C) increases variability.

3.4: B. Accountability for consequential decisions is the classic use of human in the loop: the agent pauses at a checkpoint for a person to approve. Safety filters (A) target harmful content categories. Chain-of-thought (C) adds explanation but not human accountability. Pinning a version (D) improves predictability only.

3.5: B. Google advises that most production apps use a specific stable model, and its migration guidance recommends thorough testing before moving to a new version. Evaluating candidates with Agent Platform Evals on your own data, then switching deliberately, keeps improvements without surprises. Retries (A) hide the problem. Experimental models (C) are not suitable for production. Never upgrading (D) fails at retirement dates and forgoes improvements.

3.6: C. Higher temperature increases randomness and diversity, which suits creative brainstorming, and Google suggests increasing temperature when responses are too generic. Top-k of 1 (B) is greedy decoding, the least varied. Output length (A) and stop sequences (D) affect length, not diversity.

3.7: A. Few-shot prompts are recommended for regulating format, phrasing, and scope; combined with clear instructions and consistent examples, they constrain the model to the six labels. ReAct (B) and Google Search (C) add external information that is not needed. Higher top-p (D) makes output more random.

3.8: D. Google recommends breaking complex tasks into subtasks; prompt chaining improves controllability, debugging, and accuracy because each step can be inspected. Its checklist explicitly flags "too many tasks" in one prompt. Temperature (A) and personas (B) do not fix an overloaded prompt. Fine-tuning (C) is premature and does not make failures easier to diagnose.

3.9: B. The task needs live information from multiple tools and adaptive reasoning about the results: the thought, action, observation loop of ReAct. Chain-of-thought alone (C) cannot fetch live data. Zero-shot (A) and one-shot (D) do not provide a tool-use loop.

3.10: C. The data is already in the prompt, so this is a multi-step reasoning problem, not a knowledge problem. Chain-of-thought decomposes it into intermediate steps and produces an auditable rationale; separating the working from the final answer makes it easy to parse. Grounding (A) adds data the model already has. A persona (B) does not ensure correct arithmetic. Output length (D) is irrelevant.

3.11: D. The check grounding API determines how grounded a piece of text is in a set of reference texts and returns supporting citations; Google specifically mentions using it to check answers generated by non-Google models. The ranking API (A) re-orders retrieved documents before generation. Feature Store (B) manages ML features. Google Search grounding (C) is a generation-time source, not a verifier for a custom pipeline.

3.12: A. Model Monitoring tracks input feature drift (and output and attribution drift) against a baseline and alerts when thresholds are exceeded, so you can re-evaluate or retrain. The other options change generation behaviour of a different component and do not detect drift.

Key takeaways

- Foundation models predict plausible text from training data. That causes the six headline limitations: data dependency, knowledge cutoff, bias, fairness, hallucinations, and edge cases.
- Match the fix to the cause: grounding and RAG for knowledge and factuality; prompt engineering first for clarity, format, and edge cases; supervised fine-tuning for consistent behaviour, style, and domain tasks when prompting is insufficient; human in the loop for high-stakes, subjective, or regulated decisions.
- Fine-tuning teaches *how* to answer; RAG and grounding supply *what* is true now. Do not use tuning to keep up with changing facts.
- Operate gen AI continuously: pin stable model versions in production and track retirement dates; evaluate new models, prompts, and tuned models on your own data with Agent Platform Evals; track a balanced KPI set (model quality, system quality, business operational, adoption, business value); patch and update dependencies; version models, prompts, and data; monitor drift with Model Monitoring; use Feature Store for consistent, monitored features.
- Prompt techniques: zero-shot for quick baselines; one-shot to show a single pattern; few-shot (recommended by Google) to regulate format and scope; role prompting via system instructions for persona and tone; prompt chaining to split complex workflows; chain-of-thought for multi-step reasoning and auditable rationale; ReAct for reasoning combined with tool use.
- Grounding connects output to verifiable sources, reducing hallucinations and adding auditability. Distinguish first-party enterprise data, third-party licensed data, and world data.
- Google Cloud grounding offerings: pre-built RAG with Agent Search (fastest, least effort); RAG Engine (managed, configurable); building-block RAG APIs (layout parser, embeddings, Vector Search, ranking, grounded generation, check grounding); Grounding with Google Search for current world knowledge; Web Grounding for Enterprise for regulated industries.
- Generation settings: temperature, top-p, and top-k control randomness; maximum output tokens and stop sequences control length; safety settings set blocking thresholds for harm categories. Lower randomness means more consistency, not more truth. On the newest Gemini models, custom sampling values are ignored, so prompts, system instructions, and grounding are the main levers.

Official Google references

- Generative AI Leader certification exam guide
- Gemini Enterprise Agent Platform name changes
- Google Skills: Generative AI Leader Certification learning path
- Gen AI: Unlock Foundational Concepts (course)
- Gen AI Apps: Transform Your Work (course)
- Responsible AI (model limitations and recommended practices)
- Introduction to prompting
- Overview of prompting strategies (components, template, prompt health checklist)
- Include few-shot examples
- Break down complex tasks (chain prompts, aggregate responses)
- Instruct the model to explain its reasoning
- Use system instructions
- Experiment with parameter values
- Thinking
- Gemini API: Prompt design strategies

- Gemini API: Models (model version name patterns)
- Gemini API: Grounding with Google Search
- Google whitepaper: Prompt Engineering (Kaggle)
- Google whitepaper: Agents (Kaggle)
- Google Research: Language models perform reasoning via chain of thought
- Google Research: ReAct, synergizing reasoning and acting in language models
- Architecture Center: Choose a design pattern for your agentic AI system (ReAct, human-in-the-loop)
- What is human-in-the-loop?
- Grounding overview
- Grounding with Google Search
- Grounding with Agent Search
- Web Grounding for Enterprise
- RAG Engine overview
- Agent Search on Gemini Enterprise Agent Platform (product page)
- Overview of Agent Search
- APIs for building search and RAG experiences
- Improve search and RAG quality with the ranking API
- Check grounding with RAG
- Google Cloud blog: RAG and grounding (third-party datasets announcement)
- Content generation parameters
- Safety and content filters
- System instructions for safety
- Introduction to tuning
- About supervised fine-tuning for Gemini models
- Gen AI evaluation service overview (Agent Platform Evals)
- Model versions and lifecycle
- Migrate to the latest Gemini models
- Model versioning with Model Registry
- Introduction to Model Monitoring
- About Feature Store on Gemini Enterprise Agent Platform
- KPIs for gen AI: Measuring your AI success

SECTION 4

Business strategies for a successful gen AI solution

~15% of the exam

Sections 1 to 3 explain what generative AI is, which Google Cloud offerings deliver it and how to improve model output. Section 4 asks the question a board, a chief financial officer or a chief information security officer will ask next: *how do we turn this into a solution that creates measurable value, stays secure and earns trust?* It is the most "leadership" part of the exam. Questions are scenario based and usually ask you to pick the most appropriate strategic action, framework, metric or control rather than a technical configuration.

The section has three objective groups. **4.1** covers the Google Cloud-recommended path from idea to transformational solution: recognising solution types, understanding what drives gen AI needs, choosing between buying, customising and building, integrating gen AI into the organisation and measuring impact. **4.2** covers secure AI: security across the machine learning (ML) lifecycle, Google's Secure AI Framework (SAIF) and the Google Cloud security tools that implement it. **4.3** covers responsible AI: transparency, privacy (including anonymisation and pseudonymisation), data quality, bias and fairness, accountability and explainability.

How to study: learn the **frameworks and their vocabulary** (the AI Adoption Framework themes, the value versus feasibility prioritisation matrix, the five KPI areas, the six SAIF core elements, the three Google AI Principles) and practise mapping a business problem to the right one. Then learn the **purpose** of each security tool in one sentence. The exam rarely asks "how do you configure X"; it often asks "which tool or approach addresses this risk".

PRODUCT NAMES USED IN THIS CHAPTER

Google renamed the Vertex AI family. Following the official name-change table, this chapter uses: **Gemini Enterprise Agent Platform** (formerly Vertex AI Platform; shortened to **Agent Platform** as in the exam guide), **Agent Studio** (formerly Vertex AI Studio), **Agent Search** (formerly Vertex AI Search), **Agent Runtime** (formerly Vertex AI Agent Engine), **Gemini Enterprise Agent Platform Model Garden** (formerly Vertex AI Model Garden), **Feature Store on Gemini Enterprise Agent Platform** (formerly Vertex AI Feature Store), **Gemini Enterprise Agent Platform Evals** (formerly the Gen AI evaluation service) and **Model Monitoring on Gemini Enterprise Agent Platform** (formerly Vertex AI Model Monitoring). RAG Engine keeps its name. Agentspace is now **Gemini Enterprise**. Explainable AI documentation is still titled "Vertex Explainable AI" at the time of writing; this chapter calls it **Explainable AI on Agent Platform**.

Official objective (exam guide, Section 4)	Covered in
4.1 Describe the Google Cloud-recommended steps to successfully implement a transformational gen AI solution	4.1
Recognising the different types of gen AI solutions (text generation, image generation, code generation, personalised user needs)	4.1.1

Official objective (exam guide, Section 4)	Covered in
Identifying the key factors that influence gen AI needs (business requirements, technical constraints)	4.1.2
Describing how to choose the right gen AI solution for a specific business need	4.1.3
Identifying the steps to integrate gen AI into an organisation	4.1.4
Identifying techniques to measure the impact of gen AI initiatives	4.1.5
4.2 Define secure AI and its importance in protecting AI systems from malicious attacks and misuse	4.2
Explaining security throughout the ML lifecycle	4.2.1
Identifying the purpose and benefits of Google's Secure AI Framework (SAIF)	4.2.2, 4.2.3
Recognising Google Cloud security tools and their purpose (secure-by-design infrastructure, IAM, Security Command Center, workload monitoring tools)	4.2.4
4.3 Describe the importance of responsible AI in business	4.3
Explaining the importance of responsible AI and transparency	4.3.1
Describing privacy considerations (privacy risks, data anonymisation and pseudonymisation)	4.3.2
Describing the implications of data quality, bias and fairness	4.3.3
Describing the importance of accountability and explainability in AI systems	4.3.4

4.1 Implementing a transformational gen AI solution

Google Cloud's guidance for leaders is consistent across its executive content: anchor AI in a clear vision connected to corporate strategy, focus implementation on the right use cases, and consistently track results. A transformational solution is not a single clever demo; Google Cloud notes that a single AI implementation rarely moves the financial needle on its own and that the most significant impact typically comes from several use cases working together to reimagine an entire value chain, such as a customer journey or a contact centre workflow. The five topics below follow the order of the exam objectives and, roughly, the order in which a leader makes the decisions.

4.1.1 Types of gen AI solutions

The exam guide lists four illustrative solution types: **text generation**, **image generation**, **code generation** and **personalised user needs**. Section 1 of the guide adds video generation and data analysis, and frames the business value of gen AI as the ability to **create, summarise, discover and automate**. A leader should be able to recognise each type from a business description and name a typical value lever.

Solution type	What it produces	Typical business examples	Primary value lever
Text generation (including summarisation and question answering)	Drafts, summaries, answers, translations, classifications expressed in natural language	Drafting marketing copy or customer emails; summarising long contracts or call transcripts; answering employee questions over policy documents; producing interaction summaries for contact centre agents	Time saved, faster decisions, consistency of tone
Image generation (and video generation)	New or edited images and video from text or image prompts	Product imagery variants, campaign concepts, storyboards, on-brand social assets	Creative throughput, lower production cost, faster campaign iteration
Code generation	Source code, tests, explanations of code, migrations	Developer assistance, test generation, modernising legacy code, generating queries for analysts	Developer productivity, faster time to market, code quality
Personalised user needs	Content, recommendations or experiences tailored to an individual's context	Personalised product discovery and recommendations, tailored learning content, individualised offers, conversational shopping assistants	Conversion, revenue per visit, customer satisfaction, loyalty
Discovery and analysis (search, data analysis)	Grounded answers across many sources; insights from structured and unstructured data	Enterprise search over internal knowledge, research synthesis, natural-language questions over business data	Closing "information deficits", better decisions
Automation and agents	Multi-step actions performed with tools on a user's behalf	Document processing pipelines, claims triage, self-service customer agents	Cost per transaction, capacity, containment

Google Cloud's advice on where to start is pragmatic. Two patterns that work well as early use cases are **information deficits** (people spend time and energy digging for information spread across many places; enterprise search and question answering close that gap) and **drudgery** (repetitive, procedural work such as moving information between formats, writing first drafts or repeating context to customers). Whatever the use case, leaders are advised to **start with a specific persona and task**: a confined, scalable project of clear value that builds practices which transfer to more ambitious projects, rather than an attempt to transform everything overnight.

EXAM TIP

When a scenario describes a goal in business words, translate it to a solution type first. "Employees cannot find answers in thousands of documents" is discovery (search and grounded question answering), not fine-tuning a model. "We want every shopper to see offers relevant to them" is personalisation. "Developers spend days writing tests" is code generation. Picking the right type usually eliminates two of the four options.

4.1.2 Key factors that influence gen AI needs

The exam guide names two families of factors: **business requirements** and **technical constraints**. Google Cloud's use-case prioritisation questions (see 4.1.4) and its model-selection guidance give a practical checklist for each. In practice the two families constantly interact: an ambitious business requirement (for example, real-time multilingual voice support) can be ruled out or reshaped by a technical constraint (latency budget, data residency, skills).

Business requirements	Questions a leader asks
Strategic alignment and impact	Does this move the needle for customers or employees? Which top business objective does it support? Can it be reused or extended for future use cases?
Required quality and risk tolerance	How accurate must the output be to be useful? What happens if it is wrong or misused? Is a human reviewer required?
Users and adoption	Who is the persona? How easily does the solution fit existing processes and tools?
Speed and time to value	How quickly must value be shown to keep sponsorship and funding?
Budget and ROI expectations	What is the cost envelope (licences, usage, build, run, change management) and what return justifies it?
Compliance and governance	Which regulations, contracts and internal policies apply (privacy, sector rules, intellectual property)?

Technical constraints	Questions a leader asks
Data readiness and accessibility	Is the data available, trustworthy, complete, relevant and in a usable format? Who owns it and can it legally be used?
Modality and context	Text only, or images, audio, video and code? How much context must the model consider at once (context window)?
Latency, scale and reliability	Interactive (sub-second) or batch? Expected volume and peak load? Availability requirements?
Integration	Which upstream and downstream systems and APIs must it connect to? Will future model versions break the integration?
Security, privacy and residency	Where may data be processed and stored? What access controls and perimeters are needed?
Skills and operating capacity	Does the organisation have people to build, evaluate and operate a custom solution, or should it consume a managed service?

Google Cloud's guidance on model choice adds a further dimension: performance, privacy, complexity and cost differ between large foundation models, domain-specific or task-specific models, lightweight on-device models and open versus proprietary models. Open models give more freedom to customise but, as Google Cloud notes, can require additional work for privacy, compliance, licensing and guardrails, whereas proprietary models offered as managed services typically come with enterprise security features such as safety filters and, for Google's generative AI services, indemnification commitments.

LEADER'S LENS

Write the requirements down before any vendor or model discussion. A one-page "use-case card" listing persona, task, success metric, required accuracy, risk if wrong, data sources, latency and compliance constraints makes the later buy-versus-build decision almost mechanical, and it becomes the baseline for measuring impact.

4.1.3 Choosing the right gen AI solution for a business need

Google Cloud's portfolio can be read as a **spectrum from buy to build**. At one end are ready-to-use applications that employees switch on; in the middle are platforms that let teams customise models and build agents grounded in company data; at the other end is full control over training and serving custom or open models. Moving right increases control and differentiation but also cost, time, skills required and operational responsibility.

Use prebuilt apps (Gemini for Google Workspace, Gemini Enterprise) → **Configure and ground (Agent Search, low-code agents in Agent Studio)** → **Customise on Agent Platform (prompting, RAG, tuning, ADK agents)** → **Build or train custom and open models (Model Garden, custom training)**

Option on the spectrum	Choose it when	Examples on Google Cloud	Trade-offs
Buy: prebuilt gen AI applications	The need is common productivity (writing, summarising, meeting notes, research over company content); speed matters; there is no differentiating data or process; little or no development capacity	Gemini for Google Workspace inside Gmail, Docs, Sheets, Meet; the Gemini app; Gemini Enterprise as an agentic front door across company data and business applications, running under each user's existing permissions	Fastest time to value and lowest build effort; limited differentiation; value depends on adoption and change management
Configure: managed search and agent services	The need is grounded answers or customer self-service over your own content; you want enterprise quality without assembling the components	Agent Search on Gemini Enterprise Agent Platform for out-of-the-box retrieval-augmented generation (RAG) and search; Customer Engagement Suite (Conversational Agents, Agent Assist) for contact centres	Quick to deploy, grounded in your data; customisation within the service's boundaries
Customise: build on Agent Platform	The use case is differentiating, must integrate with internal systems or workflows, or needs specific behaviour, tone or tools	Gemini and other models from Model Garden; prompt design in Agent Studio; RAG Engine and Vector Search; tuning; Agent Development Kit (ADK) agents deployed on Agent Runtime; Evals for evaluation	More control and differentiation; requires product, data and engineering skills plus an evaluation and operations practice
Build: custom or open models	No available model meets a specialised domain need; you hold unique data at scale; you require full control of weights or deployment location	Open models such as Gemma from Model Garden; Managed Training on Agent Platform; AI-optimised infrastructure (TPUs, GPUs)	Maximum control; highest cost, time and risk; you own guardrails, licensing checks, security and lifecycle management

Three principles from Google Cloud guidance help choose a point on the spectrum:

- **Start with the simplest option that meets the requirement.** Many customers begin with a large proprietary foundation model through a managed platform to learn its inputs, outputs and limits, then move to smaller, cheaper or more specialised models once they understand what the business actually needs (lower cost, faster responses or deeper domain knowledge).
- **Invest in a platform, not just a model.** Choosing a model is rarely a one-off decision: models are upgraded, versions change and some use cases combine several models to balance performance and cost. A platform such as Agent Platform provides model choice, evaluation, tuning, grounding, deployment, monitoring and governance in one place.
- **Prefer augmentation before alteration.** To improve responses, Google Cloud lists prompt optimisation, retrieval-augmented generation and fine-tuning. Grounding and RAG keep answers current and specific to your data without changing the model; tuning is justified when you need consistent style, format or specialised behaviour that prompting and grounding cannot achieve.

COMMON TRAP

Do not jump to "train a custom model" because a use case sounds important. If the need is answering questions about internal documents, the right answer is grounding or RAG (for example Agent Search or RAG Engine), not training. If the need is general office productivity, the right answer is a prebuilt application, not a development project. Custom training is the exception, reserved for unique data and needs that existing models cannot meet.

4.1.4 Steps to integrate gen AI into an organisation

Google Cloud describes integration as an organisational journey, not a technology installation. Two official sources structure it. The **Google Cloud AI Adoption Framework** (a whitepaper written with input from the Office of the CTO and other Google teams) describes the capabilities an organisation needs, and Google Cloud's executive guidance on AI strategy and on moving from prototype to production describes the sequence of steps.

The AI Adoption Framework: four areas, six themes, three maturity phases

The framework builds a structure on four areas, **people, process, technology and data**. The interplay between them highlights six themes that are critical for success:

Theme	What it means (Google Cloud definition, summarised)	Typical gen AI action
Lead	The extent to which leadership supports and encourages teams to apply ML to business use cases, and how cross-functional, collaborative and self-motivated those teams are	Executive sponsorship across the C-suite, an AI vision tied to corporate strategy, funding for experimentation
Learn	The quality and scale of learning programmes to upskill staff, hiring talent and augmenting teams with experienced partners	AI fluency training for all staff, prompt and evaluation skills for practitioners, partner support
Access	Recognising data management as key to AI, and the degree to which teams can share, discover and reuse data and ML artefacts	Governed, discoverable enterprise data for grounding; reusable prompts, agents and components
Scale	Using cloud-native services to scale with big datasets and many jobs while reducing operational overhead	Managed platforms (Agent Platform, Agent Runtime) instead of bespoke infrastructure

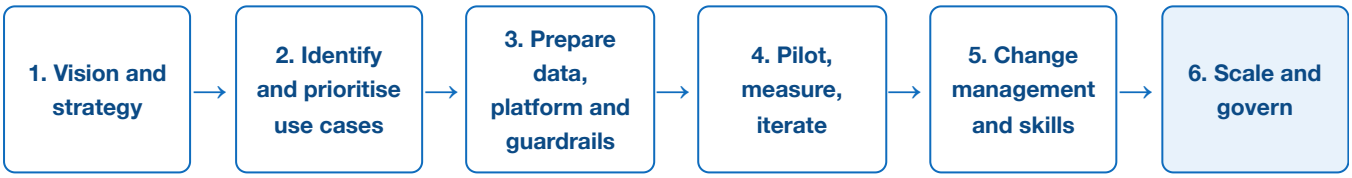
Theme	What it means (Google Cloud definition, summarised)	Typical gen AI action
Secure	Classifying and protecting sensitive data, and implementing responsible and explainable AI practices	SAIF-aligned controls, data protection, responsible AI reviews (see 4.2 and 4.3)
Automate	Deploying, executing and operating data and ML pipelines in production efficiently, frequently and reliably	Automated evaluation, deployment and monitoring pipelines

An organisation's practices in each theme fall into one of three maturity phases: **tactical** (isolated, short-term projects), **strategic** (AI tied to business goals with repeatable practices) or **transformational** (AI embedded across the organisation and continuously improving). Knowing the current phase per theme shows where the organisation is and where to invest next.

EXAM TIP

Memorise the six themes as "Lead, Learn, Access, Scale, Secure, Automate" and the three phases as "tactical, strategic, transformational". A scenario about data scientists who cannot find or reuse datasets maps to **Access**; one about staff lacking skills maps to **Learn**; one about no executive backing maps to **Lead**.

The integration steps



Step 1: Set a clear vision and strategy. Google Cloud recommends treating AI initiatives as part of the overall corporate strategy rather than standalone projects, with IT and business working closely together. It uses a combined **top-down and bottom-up** approach. Top-down, leaders connect strategic priorities to **AI domains**: a department, a core product or an end-to-end process such as a customer journey. Bottom-up, leaders crowdsource ideas from teams on the ground (submission forms, hackathons, briefings) because big-picture strategy tends to overlook the operational detail that reveals the best opportunities. Google Cloud's research also stresses sponsorship: the National Research Group study commissioned by Google Cloud found only one in four organisations had C-level sponsorship for AI initiatives, while those with shared C-suite support brought ideas to market faster.

Step 2: Identify and prioritise use cases. Google Cloud uses a simple matrix that plots each candidate use case by **expected business value** against **actionability and feasibility**, with guiding questions for each dimension:

Dimension	Criterion	Guiding question
Business value	Impact	Will this move the needle for our customer and employee needs?
	Alignment	How does this support our top business objectives?
	Reuse and extensibility	Can we reuse and build on it for future use cases?

Dimension	Criterion	Guiding question
Actionability	Usability	How accurate does AI have to be to make it truly useful?
	Adoption	How easily can we adapt this into our existing processes?
	Speed	How quickly can we drive value for our organisation?
Feasibility	Technical fit	Is AI actually the right fit for this use case?
	Data readiness	How accessible and trustworthy is the data?
	Risk tolerance	Are there implications of inaccuracy or misuse?

The matrix produces four practical categories. The labels below are a study aid; the logic is Google Cloud's.

	Higher feasibility and actionability	Lower feasibility and actionability
Higher business value	**Prioritise now.** Google Cloud calls high-impact use cases that can deliver measurable value quickly "Best Bets". Example from Google Cloud: virtual agents answering common questions in a large bank's contact centre, high value through lower operating cost and highly feasible.	**Plan as strategic bets.** Valuable but needing more data integration, workflow change, risk management and stakeholder alignment. Example from the same bank: personalised financial advice with safeguards. Sequence after foundations are in place.
Lower business value	**Quick wins.** Useful to build skills, confidence and funding, especially for organisations without strong foundations yet; avoid letting them crowd out higher-value work.	**Deprioritise.** Revisit if data, technology or business priorities change.

Google Cloud also uses a variant that plots expected revenue generation against expected time to value. Some organisations should lead with quick wins to earn funding and cross-functional backing; others with the right foundations can pursue high-growth use cases first. Growth-oriented goals (customer experience, new products) distinguished the "Gen AI Leaders" in Google Cloud's ROI research from organisations focused only on incremental efficiency.

Step 3: Prepare data, platform and guardrails. Gen AI is only as good as the context it is given. Google Cloud's internal lessons emphasise grounding models in "enterprise truth" through curated, dynamic data rather than static file dumps, and observe that smaller, targeted datasets of relevant information outperform massive, comprehensive ones. In parallel, choose a platform (see 4.1.3), define the evaluation approach and KPIs, and design security and responsible AI controls from the start (see 4.2 and 4.3). Google Cloud is explicit that responsible AI is "not a step along the path" but a practice upheld throughout.

Step 4: Pilot, measure and iterate. Start with a confined, scalable pilot for a specific persona and task. Gen AI is non-deterministic: the same prompt can produce different outputs, so an evaluation framework with metrics must exist from the beginning. Google Cloud's account of its own go-to-market transformation recommends **launch and iterate** over waiting for perfection: reduce initial scope, accept manual workarounds, put tools in users' hands quickly, collect feedback (star ratings, forms, interviews) and refine prompts and code in days. Design the experience around task completion: guided inputs and a button that runs an expert-crafted prompt often drive more adoption than an open-ended chat box.

Step 5: Manage change and build skills. Adoption "doesn't happen overnight"; new AI-powered behaviours require patience and persistence. Practical levers include executive sponsorship that creates psychological safety for experimentation, AI champions and expert reviewers who set evaluation standards, business

subject-matter experts who design workflows based on how people actually work, and upskilling (the **Learn** theme). Google Cloud's ROI research lists aligning business and IT, upskilling and hiring talent, and improving data quality and knowledge management as top investment priorities for organisations that get value from gen AI.

Step 6: Scale and govern. As adoption grows, teams build overlapping tools ("AI sprawl" or "agent sprawl"). Google Cloud's lesson is neither to be a strict gatekeeper (which stifles innovation) nor to allow uncontrolled duplication, but to build interoperable, reusable agents (for example with ADK and the Agent2Agent protocol), let central teams standardise tools that prove their value, align roadmaps regularly and keep everything under shared responsible AI principles and governance. Scaling also means moving pilots onto managed, monitored production platforms with defined owners, budgets and controls.

LEADER'S LENS

Every step has a people component. In exam scenarios where a technically successful pilot fails to spread, the best answer is usually about sponsorship, change management, user experience or skills, not about switching to a bigger model.

4.1.5 Measuring the impact of gen AI initiatives

Google Cloud's guidance is summarised in the phrase "you can't manage what you don't measure". KPIs for gen AI help objectively assess performance, align initiatives with business goals, enable data-driven adjustments, communicate with stakeholders and demonstrate return on investment (ROI). Google Cloud observes that many organisations measure only computation-based model quality and confuse operational efficiency gains with end-state business goals. Its recommended approach measures AI across **five key areas**:

KPI area	What it tells you	Example metrics from Google Cloud guidance
1. Model quality	Whether outputs are accurate, factual, safe and useful	Error rate, accuracy thresholds, safety score; for unbounded outputs, model-based (autorater) metrics such as coherence, fluency, safety, groundedness, instruction following, verbosity; pointwise scores and pairwise win rates against a baseline; human evaluation
2. System quality	Whether the end-to-end system runs efficiently, reliably and at scale	Uptime, error rate, model latency and retrieval latency, request and token throughput, accelerator utilisation, number of deployed models, time to deployment, percentage of automated pipelines, percentage of models with monitoring, data relevance
3. Adoption	Whether people actually use it and like it	Adoption rate (active users divided by intended audience), frequency of use, session length, queries per session, query length, abandonment rate, thumbs up or down feedback, user satisfaction and Net Promoter Score (NPS)
4. Business operational	Whether business processes improve	Contact centre: call and chat containment, average handle time, customer satisfaction (CSAT), churn, human agent satisfaction. Discovery: click-through rate, revenue per visit, time on site. Documents: processing time, process capacity, knowledge extensibility
5. Business value (financial impact)	Whether strategic and financial goals are met	Productivity value, cost savings, innovation and growth (new products, revenue uplift), customer experience (churn reduction, revenue), resilience and security improvements

Several practices turn these metrics into credible evidence of impact:

- **Plan measurement during design.** Google Cloud calls it a best practice to develop the measurement plan in the design phase of a use case, not bolt it on during implementation.
- **Establish a baseline.** Measure the current process before launch, or A/B test the AI-enabled process against the same process without AI, so improvements can be attributed.
- **Translate operations into finance.** Work with the finance team to convert operational and adoption KPIs into financial impact, and include the full cost of building and running the solution (data size and complexity, usage volume, number and size of models, people and resources) to calculate ROI. In simple terms, ROI compares quantified benefits with total investment over a defined period.
- **Interpret trade-offs with context.** Improving one KPI can move another the wrong way; Google Cloud's retail example notes that a more engaging assistant may increase cart size but also time to cart. Business stakeholders, data scientists and developers should interpret results together.
- **Read adoption trends.** Consistently low adoption suggests a lack of awareness; high adoption that drops suggests a performance or usefulness problem.
- **Measure outcomes, not just activity.** Google Cloud's internal approach combines adoption tracking (which features, by whom, for what), sentiment (ratings, focus groups) and impact (tying usage to accounts or opportunities). Where direct value is hard to prove early, adoption volume and sentiment serve as proxies while richer impact analytics are built.

Google Cloud's research gives context for expectations. In the National Research Group study of more than 2,500 senior leaders, 74% of enterprises using gen AI reported already seeing ROI, and organisations with a comprehensive AI strategy were more likely to see it. A later Google Cloud ROI of AI report found 52% of executives reporting their organisations actively use AI agents. Treat such figures as context, not as targets; your own baseline and KPIs matter more.

COMMON TRAP

Model accuracy alone is not a measure of business impact, and "number of users" alone is not ROI. Exam answers that measure only technical metrics, or only usage, are usually incomplete. Look for the option that links model, system, adoption, operational and financial measures, compared against a baseline.

AI Adoption Framework	Google Cloud whitepaper that structures AI capability around people, process, technology and data, six themes (Lead, Learn, Access, Scale, Secure, Automate) and three maturity phases (tactical, strategic, transformational).
AI domain	A focus area for AI investment linked to strategy, such as a department, product or end-to-end process.
Best Bets	Google Cloud's term for higher-impact use cases that can deliver measurable value quickly.
Baseline	Measurement of a process before AI is introduced, used to attribute improvements.
Adoption rate	Active users of an AI tool as a percentage of its intended audience.
Containment rate	Share of customer calls or chats handled and resolved by AI without a human agent.
Return on investment (ROI)	Quantified benefits of an initiative compared with its total cost over a defined period.

4.2 Secure AI: protecting AI systems from attacks and misuse

Secure AI means protecting AI systems, including their data, models, infrastructure, applications and agents, against malicious attack and misuse throughout their life, so that they behave as intended, protect confidential information and cannot be turned against the organisation or its users. AI inherits every traditional security risk (stolen credentials, misconfiguration, vulnerable software) and adds new ones. Google Cloud highlights examples such as **prompt injection and jailbreaking** (malicious instructions that manipulate a model), **training data poisoning** (contaminated data that compromises model integrity), **data leakage** (a model revealing sensitive information) and **model theft** (copying or reconstructing a valuable model). Agents raise the stakes because they can take actions with real permissions.

The business case is direct: a survey cited by Google Cloud found sensitive data leaks were the top gen AI concern for 80% of business leaders and 82% of cybersecurity professionals. Leaders who cannot answer "how is this secured?" will struggle to get approval to move beyond pilots.

4.2.1 Security throughout the ML lifecycle

Google Cloud's Well-Architected Framework (AI and ML perspective) calls for a holistic security approach that "starts from the initial conceptualisation of your AI and ML solutions and extends to development, deployment, and ongoing operations". It is easier and cheaper to build controls in early ("shift left") than to add them after development. Its six security principles for AI and ML are:

1. **Define clear goals and requirements**: translate business goals into security requirements, assess risk, balance security against business needs and adopt a secure-by-design approach.
2. **Keep data secure and prevent loss or mishandling**: data minimisation, de-identification, governance and lineage, least-privilege access, encryption and controls on data movement.
3. **Keep AI pipelines secure and robust against tampering**: protect code, artefacts and pipelines that produce models.
4. **Deploy on secure systems with secure tools and artefacts**: hardened, trusted infrastructure and components.
5. **Verify and protect inputs**: screen prompts and other inputs for malicious content.
6. **Monitor, evaluate and prepare to respond to outputs**: check outputs, monitor behaviour and have an incident response plan.

SAIF's risk map offers a complementary view. It divides AI development into four areas of components, **Data**, **Infrastructure**, **Model** and **Application**, and shows for each risk where it is *introduced*, where it is *exposed* and where it can be *mitigated*. SAIF notes that in AI development "data takes on aspects of the role previously played by code", so compromising training data or model weights can be as damaging as tampering with application code.

Lifecycle stage (SAIF area)	What is at risk	Example risks (SAIF names)	Example protective practices
Data ingestion and preparation (Data: sources, filtering and processing, training data)	Integrity, lawfulness and confidentiality of data	Data Poisoning; Unauthorized Training Data; Excessive Data Handling	Data provenance and inventory, sanitisation, authorised-use checks, de-identification, access control, lineage and logging

Lifecycle stage (SAIF area)	What is at risk	Example risks (SAIF names)	Example protective practices
Model training and tuning (Infrastructure: frameworks and code; training, tuning and evaluation; data and model storage)	Model code, weights, dependencies, checkpoints	Model Source Tampering; Model Exfiltration	Integrity protection and inventory of models and code, least-privilege access to storage, secure-by-default tooling, supply-chain controls
Deployment and serving (Infrastructure: model serving)	Production model and serving stack	Model Deployment Tampering; Model Exfiltration	Hardened serving infrastructure, encryption, perimeters, vulnerability management
Use: inputs and outputs (Model and Application)	Model behaviour, data exposed through responses, availability	Prompt Injection; Model Evasion; Sensitive Data Disclosure; Inferred Sensitive Data; Insecure Model Output; Denial of ML Service; Model Reverse Engineering	Input and output filtering, rate limiting, access management, safety filters, user transparency
Agents and integrations (Application: agents and plugins)	Actions taken on behalf of users, connected systems	Insecure Integrated Component; Rogue Actions	Least-privilege agent permissions, user approval for consequential actions, agent observability and audit logs
Ongoing operations and management (all areas)	Whole system over time	All of the above, plus drift and new threats	Threat detection, red teaming, vulnerability management, incident response, governance

EXAM TIP

If a question asks *when* to think about security for a gen AI initiative, the answer is "from the start and at every stage", not "after the pilot" or "before go-live only". Security, like responsible AI, is continuous across the lifecycle.

4.2.2 Google's Secure AI Framework (SAIF): purpose and core elements

Google introduced **SAIF** as "a conceptual framework to secure AI systems". Its purpose is to provide industry security standards for building and deploying AI responsibly, addressing concerns such as AI and ML model risk management, security and privacy, and helping ensure that AI models are **secure by default** when implemented. SAIF is distilled from Google's long experience mapping security best practices to AI and reflects Google's own approach to building ML and gen AI-powered applications. It is not a product; it is a framework you apply with whatever tools you use.

SAIF has **six core elements**:

#	Core element	What it means in practice
1	**Expand strong security foundations to the AI ecosystem**	Reuse proven, secure-by-default infrastructure protections and security expertise (identity, access control, encryption, supply chain) for AI systems, adapting them to AI-specific threats.

#	Core element	What it means in practice
2	**Extend detection and response to bring AI into an organisation's threat universe**	Monitor AI systems, including the inputs and outputs of gen AI, for anomalies and attacks; use threat intelligence; include AI in security operations and incident response.
3	**Automate defences to keep pace with existing and new threats**	Use AI itself to improve the scale and speed of defence, because attackers will also use AI.
4	**Harmonise platform-level controls to ensure consistent security across the organisation**	Apply consistent controls and frameworks across platforms and tools, so every AI application is protected the same way rather than team by team.
5	**Adapt controls to adjust mitigations and create faster feedback loops for AI deployment**	Continuously test and evolve protections, for example through red teaming and learning from incidents and feedback.
6	**Contextualise AI system risks in surrounding business processes**	Assess AI risk end to end in the context of how the business uses the system (data, decisions, users, regulation), so risk management is proportionate and integrated.

Google's quick guide to implementing SAIF has four steps: **(1) understand the use** (the business problem and the data needed drive the policies and controls); **(2) assemble the team** (a cross-functional team so security, privacy, risk and compliance are included from the start); **(3) level set with an AI primer** (everyone understands the model development lifecycle, capabilities and limitations); **(4) apply the six core elements**, which Google notes are not intended to be applied in chronological order.

4.2.3 SAIF benefits, risk map and controls

For a business leader, SAIF's benefits are:

- **A common language** for business, security, data science and legal teams to discuss AI risk.
- **Coverage of AI-specific risks** that traditional frameworks miss, such as data poisoning, prompt injection, model exfiltration and rogue agent actions.
- **Secure by default** deployment, reusing existing security investments rather than starting from scratch.
- **Lifecycle and shared-responsibility clarity**: for each risk, SAIF states whether **model creators**, **model consumers** or both can mitigate it. Most enterprises are model consumers of foundation models, so they focus on application, agent and data controls.
- **Ecosystem alignment**: Google works with governments and standards bodies (for example NIST) and formed the Coalition for Secure AI (CoSAI) with industry partners to advance secure AI practices.

The resource hub **saif.google** provides the SAIF Map, a catalogue of **15 AI risks** and the controls that address them, a focus area on **AI agent security**, and a risk self-assessment that suggests relevant controls. SAIF groups controls into Data, Infrastructure, Model and Application controls, plus **Assurance** (red teaming, vulnerability management, threat detection, incident response) and **Governance** (user and internal policies and education, product governance, risk governance), which apply to all risks.

Threat (SAIF risk)	Business example	SAIF control(s)	Google Cloud capability that helps
Prompt Injection (direct or indirect, including jailbreaks)	A customer pastes "ignore your instructions" into a support chatbot; a web page read by an agent contains hidden instructions	Input validation and sanitisation; output filtering; red teaming	Model Armor screens prompts and responses for prompt injection and jailbreak attempts and malicious URLs
Sensitive Data Disclosure	A chatbot reveals another customer's account details or confidential system instructions	Privacy enhancing technologies; user data management; agent permissions	Model Armor with Sensitive Data Protection; Sensitive Data Protection de-identification; IAM least privilege
Data Poisoning	An insider alters tuning examples to bias a pricing assistant	Training data sanitisation; model and data inventory, access and integrity management	IAM, Cloud Audit Logs, data governance and lineage, VPC Service Controls
Unauthorized Training Data	Tuning on customer data collected without consent	Training data management	Sensitive Data Protection discovery and profiling; governance processes
Model Exfiltration and Model Source Tampering	Theft of a proprietary fine-tuned model or tampering with its weights	Model and data access controls; integrity management; secure-by-default ML tooling	IAM, VPC Service Controls, customer-managed encryption keys, Security Command Center AI Protection threat detection
Denial of ML Service and Model Reverse Engineering	Flooding a model API, or mass querying it to clone behaviour	Application access management (rate limiting, authentication)	API management and quotas, IAM, monitoring and alerting
Insecure Model Output	Generated marketing email contains a malicious link	Output validation; content safety	Model Armor malicious URL and content safety filters; safety filters on Gemini models
Rogue Actions and Insecure Integrated Component	An agent emails the wrong person or executes an injected instruction through a plugin	Agent permissions (least privilege); agent user control (human approval for consequential actions); agent observability	Agent Identity and Agent Gateway with Model Armor on Agent Platform; Security Command Center detection of over-privileged agents; Cloud Logging

COMMON TRAP

SAIF is a **framework**, not a product you buy or switch on. If an option says "enable SAIF in the console", it is wrong. SAIF is implemented through policies, people and tools such as IAM, Security Command Center and Model Armor. Also do not confuse SAIF (security) with Google's AI Principles (responsible AI values); they complement each other.

4.2.4 Google Cloud security tools and their purpose

The exam guide names **secure-by-design infrastructure**, **Identity and Access Management (IAM)**, **Security Command Center** and **workload monitoring tools**. Several other services appear frequently in Google Cloud's AI security guidance and are worth recognising.

Secure-by-design infrastructure. Google designs security into its technical infrastructure in progressive layers: low-level infrastructure (physical data centre security, Google-designed server boards and the Titan hardware security chip as a hardware root of trust, secure boot and machine identity), service deployment, data storage (including encryption), internet communication and operations. Communication between services is authenticated and encrypted (for example with Application Layer Transport Security). Google Cloud AI services such as Agent Platform and Gemini Enterprise run on this foundation, which is the practical expression of SAIF element 1. Google Cloud also states that customer data used with Gemini for business, including prompts, outputs and tuning, is not used to train Google models or models for other customers.

Identity and Access Management (IAM). IAM controls *who* (principal: a user, group, service account or other identity) can do *what* (role: a collection of permissions) on *which resource*. Roles are granted through allow policies that inherit down the resource hierarchy (organisation, folders, projects). Google recommends predefined or custom roles and the **principle of least privilege**; highly permissive basic roles should not be used in production. For AI this means separate, minimal access to training data, models, endpoints and agents, and service identities for workloads and agents.

VPC Service Controls. Creates **service perimeters** around Google Cloud resources (for example Cloud Storage, BigQuery and AI services) to reduce data exfiltration risk. It blocks copying data to resources outside the perimeter and denies access from unauthorised networks even if credentials are stolen or an IAM policy is misconfigured. IAM gives identity-based access control; VPC Service Controls gives context-based perimeter security. Google recommends using both for defence in depth.

Security Command Center (SCC). Google Cloud's risk management platform for preventing, detecting and responding to security issues: vulnerability and misconfiguration detection, threat detection, posture management, compliance monitoring and attack path analysis. Its **AI Protection** capability gives a view of AI security across the environment: an inventory of AI assets, identification of vulnerabilities and risks in agentic workloads, detection of over-privileged agents, compliance frameworks for AI (such as Google Recommended AI Essentials for Gemini Enterprise Agent Platform) and threat detection for AI assets (for example anomalous service account activity or new AI API usage). Capabilities depend on the service tier (Standard, Premium or Enterprise). Agent Anomaly Detection for agents on Agent Runtime is an integrated source PREVIEW.

Model Armor. Runtime security for gen AI and agentic AI that screens prompts, responses and agent interactions. It detects and blocks prompt injection and jailbreaking, malicious URLs and malware, applies configurable content safety filters (for example hate speech, harassment, sexually explicit and dangerous content) and, through integration with Sensitive Data Protection, helps prevent leakage of personal information, financial data and credentials. It is model-agnostic (it can protect Gemini and other providers' models) and integrates in-line with Google Cloud services, acting as an "AI firewall".

Sensitive Data Protection (the service that includes Cloud Data Loss Prevention). Discovers, classifies and profiles sensitive data, and de-identifies it with techniques such as masking, redaction, tokenisation and generalisation. It underpins privacy controls covered in 4.3.2.

Workload monitoring tools. Cloud Logging is a real-time log management system that collects logs from Google Cloud resources and applications, supports search and analysis, and can trigger alerts. **Cloud Monitoring** tracks metrics, dashboards and alerting policies. **Cloud Audit Logs** record administrative and data access activity (who did what, where and when). For AI they support detection and response (SAIF

element 2), accountability and agent observability. On Agent Platform, **Model Monitoring** watches deployed models for input skew and drift. Google's guidance also warns not to log personal or confidential data unnecessarily.

Encryption and key management. Data is encrypted at rest and in transit by default; customer-managed encryption keys (CMEK) in Cloud Key Management Service give additional control and are supported by Agent Platform, Cloud Storage and BigQuery.

If the requirement is...	Primary tool	Purpose in one sentence
Trust the underlying platform for hardware, network and physical security	Secure-by-design infrastructure	Layered security built into Google's infrastructure from hardware to operations.
Give only the right people and agents the minimum access needed	IAM	Who can do what on which resource, following least privilege.
Stop data leaving a trusted boundary, even with stolen credentials	VPC Service Controls	Context-based service perimeters that reduce exfiltration risk.
See and fix misconfigurations, vulnerabilities and threats across cloud and AI assets	Security Command Center (AI Protection)	Centralised posture, risk and threat management, including AI inventory and agent risks.
Block prompt injection, jailbreaks, harmful content and data leakage at runtime	Model Armor	Screens prompts and responses of gen AI applications and agents.
Find and de-identify personal or confidential data	Sensitive Data Protection	Discovery, classification and de-identification of sensitive data.
Know what happened, alert on anomalies, support audits	Cloud Logging, Cloud Monitoring, Cloud Audit Logs	Observability and audit trails for workloads, models and agents.

LEADER'S LENS

No single tool secures AI. A credible answer to a board question combines layers: secure infrastructure, least-privilege identity, perimeters around sensitive data, runtime screening of prompts and outputs, continuous posture and threat monitoring, and a tested incident response plan, all mapped to SAIF so nothing is missed.

Secure AI	Protecting AI data, models, infrastructure, applications and agents against attack and misuse across the lifecycle.
SAIF	Google's Secure AI Framework: a conceptual framework with six core elements, a risk map of 15 risks and associated controls.
Prompt injection	Causing a model to execute instructions hidden in its input; jailbreaks are one type.
Data poisoning	Altering training or tuning data to degrade a model, skew results or plant backdoors.
Model exfiltration	Unauthorised copying of a model or its weights.
Rogue actions	Unintended actions executed by an AI agent, accidental or malicious.
Least privilege	Granting only the minimum permissions needed for a task.

Service perimeter	A VPC Service Controls boundary that restricts data movement and access to protected services.
Red teaming	Self-directed adversarial testing to find security and safety weaknesses before attackers do.

4.3 Responsible AI in business

Security asks "can someone attack or misuse this system?". Responsible AI asks "even when it works as designed, is it fair, transparent, private, safe and accountable?". Google Cloud frames responsible AI as good for business: evaluating systems both when they perform as intended and when they do not leads to **safer and more accountable products**; a responsible approach helps **earn and keep customer trust**, since lack of trust is a growing barrier to enterprise adoption; and empowering decision-makers to consider ethics fosters a **culture of responsible innovation**.

4.3.1 The importance of responsible AI and transparency

Google's **AI Principles** (ai.google/principles) are the foundation of its approach. The current version is organised into three principles:

Principle	Key commitments (summarised)
1. Bold innovation	Develop AI that assists, empowers and inspires people and drives economic and scientific progress; deploy models and applications where the likely overall benefits substantially outweigh the foreseeable risks; focus on real-world problems and measure tangible outcomes.
2. Responsible development and deployment	Pursue AI responsibly across the whole lifecycle, from design to testing to deployment to iteration: appropriate human oversight, due diligence and feedback mechanisms; investment in safety and security research; rigorous design, testing, monitoring and safeguards to mitigate harmful outcomes and avoid unfair bias; promoting privacy and security and respecting intellectual property rights.
3. Collaborative progress, together	Make tools that empower others; collaborate with researchers, governments and civil society on challenges no single stakeholder can solve; foster a vibrant ecosystem.

Google states that these principles are operationalised through multi-layered governance spanning the model lifecycle, from development to post-launch monitoring and remediation. At Google Cloud, review bodies carry out ethical analyses and risk and opportunity assessments for products and for early-stage deals involving custom work. Agent Platform's generative AI documentation reminds customers that, although the APIs are designed with the AI Principles in mind and include safety filters with configurable thresholds, customers remain responsible for testing their own use case, following the Acceptable Use Policy and the Generative AI Prohibited Use Policy, and applying fairness, interpretability, privacy and security practices. Google recommends four steps: assess the application's security risks, perform safety testing appropriate to the use case, configure safety filters if required, and solicit user feedback and monitor content.

Transparency means people understand when they are interacting with AI, what it can and cannot do, which data it uses and how outputs were produced. In business terms transparency includes:

- **Disclosure**: telling customers and employees when content or answers are AI-generated and what the known limitations are. SAIF's "User Transparency and Controls" control asks organisations to inform users of relevant AI risks and give them control over use of their data.

- **Documentation**: structured model information (Google publishes model cards) and internal records of data sources, intended use, evaluations and known limitations.
- **Traceable answers**: grounding and citations that let users verify a response against source documents.
- **Clear policies and education**: SAIF's governance controls call for easy-to-understand AI policies for users and comprehensive policies and training for employees.

EXAM TIP
Responsible AI is not a final checkpoint. Google Cloud's prototype-to-production guidance says governance, safety and fairness "are not a step along the path" but core practices upheld throughout. Options that defer responsible AI review to "after launch" are almost always wrong.

4.3.2 Privacy considerations

Gen AI creates privacy risks at every stage. SAIF names several directly:

- **Sensitive Data Disclosure**: models leaking memorised training data, user chat history or confidential prompt content; agents with access to email or files leaking data through their tools.
- **Inferred Sensitive Data**: models inferring sensitive attributes about people (for example political affiliation or health) that were never provided. If the inference is true, SAIF notes it can be a privacy incident even if it arose from a hallucination.
- **Excessive Data Handling**: collecting, retaining, processing or sharing user data (queries, interactions, preferences) beyond what policies allow.
- **Unauthorized Training Data**: training or tuning on data without consent, licence or legal basis.

Other common sources of exposure are prompts containing personal data sent to tools that are not approved, logs that capture full interactions, and test environments filled with copies of production data. Google Cloud's privacy controls start with **data minimisation**: do not collect, keep or use data that is not strictly necessary, and where possible use synthetic or fully anonymised data. Then apply de-identification with Sensitive Data Protection, least-privilege access, retention limits and logging that avoids personal data.

De-identification is the umbrella term for removing identifying information from data. Sensitive Data Protection supports these transformation types: **redaction** (delete a value), **replacement** (substitute a fixed surrogate value), **masking** (replace characters with a symbol such as # or *), **crypto-based tokenisation** (encrypt a value with a key; some methods are reversible), **bucketing** (generalise a value into a range, such as an age band), **date shifting** (shift dates by a random amount) and **time extraction** (keep only part of a date or time). Two outcomes of de-identification are tested on the exam: anonymisation and pseudonymisation.

Anonymisation, as Google describes it, removes or modifies personally identifiable information so that the result "cannot be associated with any one individual". Google's techniques include **generalisation** (removing part of a value or replacing it with a common value) to achieve **k-anonymity** (each individual is indistinguishable from at least k-1 others sharing the same properties), **l-diversity** (ensuring diversity of sensitive values within such groups) and **differential privacy** (adding mathematical noise so the output looks essentially the same whether or not any individual is included). Adding noise can make data less useful.

Pseudonymisation, in Sensitive Data Protection's definition, replaces sensitive values with cryptographically generated **tokens** (also called tokenisation or surrogate replacement). Tokens can be **two-way** (reversible with the key, for example deterministic AES-SIV encryption or format-preserving encryption) or **one-way**

(cryptographic hashing with HMAC-SHA-256, which cannot be reversed). All three methods preserve **referential integrity**: the same input always maps to the same token, so analysts can still join and aggregate tables without seeing raw identifiers.

	Anonymisation	Pseudonymisation
Goal	Data can no longer be associated with any individual	Direct identifiers are hidden but records stay consistent and linkable
Typical techniques	Generalisation and bucketing, suppression or redaction, aggregation, k-anonymity, l-diversity, differential privacy	Crypto-based tokenisation: deterministic encryption (AES-SIV), format-preserving encryption, cryptographic hashing (HMAC-SHA-256)
Reversible?	No, by design	Two-way tokens: yes, with the key. One-way hashes: no, but the consistent token still allows records to be correlated
Data utility	Lower for individual-level analysis; good for aggregate statistics and sharing	High: joins, de-duplication, longitudinal analysis and, if authorised, re-identification
Residual risk	Re-identification through combination with other data if done poorly; measure it with re-identification risk analysis (Sensitive Data Protection computes k-anonymity, l-diversity, k-map and delta-presence)	Remains sensitive: keys must be protected (for example in Cloud Key Management Service) and token tables controlled; linkage with other datasets can reveal identity
Choose when	Publishing or sharing data externally, building aggregate analytics or training data where no one ever needs to know who an individual is	Internal analytics, model training on customer histories, test and development environments, or workflows where an authorised team must occasionally re-identify (for example fraud investigation)

COMMON TRAP

Pseudonymised data is **not** anonymous. If a key exists, or tokens are consistent across datasets, individuals can be re-identified or linked. If a scenario requires that data "can never be traced back to an individual", choose anonymisation. If it requires joining records or authorised re-identification, choose pseudonymisation and protect the keys.

Google Cloud also reduces privacy exposure through product commitments that leaders should know. For Gemini for business (Gemini Enterprise), Google Cloud states that customers own their data, that prompts, outputs and tuning data are not used to train Google models or other customers' models, that customer data is not sold and is not used for advertising, and that access follows each user's existing permissions.

4.3.3 Data quality, bias and fairness

Models inherit the strengths and flaws of their data. Google Cloud warns that without proper data governance, models can be trained on low-quality, biased or irrelevant data, increasing the chance of hallucinations or problematic outputs, and recommends investing in labelling, organising and monitoring data. Data quality dimensions from Section 1 of the exam guide apply directly: **completeness**, **consistency**, **relevance**, **availability**, **cost** and **format**. Google Cloud's own experience is that the right data matters more than the most data: smaller, targeted, relevant datasets outperform large unfiltered dumps, and irrelevant data can introduce bias and inefficiency.

Agent Platform documentation lists limitations that link data, bias and fairness:

- **Data quality and tuning**: the quality, accuracy and bias of prompts or input data significantly affect response quality.
- **Bias amplification**: models can amplify existing biases in training data, reinforcing societal prejudices and unequal treatment of groups.
- **Language quality**: performance can be weaker for under-represented languages, dialects and language varieties, so service quality can be inconsistent across users.
- **Fairness benchmarks and subgroups**: published fairness analyses do not cover every risk, so organisations must evaluate fairness for their own users and context.
- **Edge cases** and **limited domain expertise**: rare situations and specialised topics can produce overconfident or wrong answers; Google advises domain-specific tuning and meaningful human supervision where outputs could materially affect individual rights.

The business implications are concrete: unfair outcomes harm customers and employees, create legal and regulatory exposure, damage brand reputation and erode trust that is hard to rebuild. Bias also reduces value; a model that works well only for some customer segments under-delivers on revenue and satisfaction targets.

Source of unfairness	Example	Mitigation a leader should require
Historical bias in data	A screening assistant tuned on past hiring decisions reproduces past preferences	Review data representativeness and labels; evaluate outcomes by subgroup; remove or reconsider proxies; human review of decisions
Under-representation	A support assistant performs poorly for some languages or regions	Add representative data and evaluation sets; test per language and segment before launch
Irrelevant or excessive data	Including unnecessary personal attributes in prompts or grounding data	Data minimisation; relevance checks; governance of data sources
Feedback loops after launch	Errors reinforced over time as outputs feed future data	Continuous monitoring, user feedback, drift monitoring, periodic fairness re-evaluation

LEADER'S LENS

Make fairness measurable. Ask for evaluation results broken down by relevant user groups, not just an overall accuracy score, and set a threshold for acceptable differences before launch. An average can hide a group for whom the system fails.

4.3.4 Accountability and explainability

Accountability means that named people and processes are answerable for an AI system's outcomes: who approved it, who monitors it, who can stop it and who remediates harm. Google's AI Principles call for appropriate human oversight, due diligence and feedback mechanisms. In practice accountability requires:

- **Clear ownership** for each use case (business owner, technical owner, risk owner) and executive sponsorship, which Google Cloud notes also drives faster decision-making.
- **Governance processes**: review before launch, approval criteria, and SAIF's product governance (validate that models and products meet security and privacy requirements) and risk governance (inventory, measure and monitor residual risk).

- **Human in the loop** for consequential decisions, and user approval for agent actions that alter data or act on a user's behalf (SAIF's agent user control).
- **Audit trails**: logs of who accessed what and which actions agents took (Cloud Audit Logs, Cloud Logging, SAIF's agent observability), so decisions can be reconstructed and investigated.
- **Feedback and redress**: channels for users to report problems and processes to correct them.

Explainability is the ability to understand and communicate why a model produced a particular output. Google Cloud notes that ML models are often "black boxes" in which even their designers cannot explain a specific inference. Explainability builds confidence in outputs, helps teams debug and improve models, supports detection of bias and helps satisfy regulators and customers who are entitled to understand decisions that affect them. Google Cloud's **Explainable AI on Agent Platform** (documented as Vertex Explainable AI) offers two approaches for predictive models:

Approach	How it works	Business use
Feature-based explanations (feature attributions)	Shows how much each input feature contributed to a specific prediction	Explaining why a loan application or claim received a particular score; spotting reliance on inappropriate features; debugging
Example-based explanations	Uses nearest neighbour search to return training examples most similar to the input	Understanding mistakes and data gaps (for example a misclassification caused by missing examples), detecting anomalies, choosing which new data to label

For generative models, explainability is achieved more through **grounding with citations** (showing which sources supported an answer), **evaluation** with transparent criteria (for example groundedness and instruction-following metrics in Gemini Enterprise Agent Platform Evals), documentation of model limitations, and **observable reasoning and tool use** for agents through logging.

COMMON TRAP

Explainability does not transfer responsibility to the model or the vendor. Even with Google's safety filters and explanations, the organisation deploying the system remains accountable for its use. Options that suggest "the provider is responsible for outcomes" are wrong.

Responsible AI	Designing, building and using AI so it is fair, safe, private, transparent and accountable throughout its lifecycle.
AI Principles	Google's three principles: bold innovation; responsible development and deployment; collaborative progress, together.
Transparency	Making clear when AI is used, how it works, its limitations and the sources behind its outputs.
De-identification	Removing identifying information from data using transformations such as redaction, masking, tokenisation or bucketing.
Anonymisation	Irreversibly transforming data so it cannot be associated with any individual.
Pseudonymisation	Replacing identifiers with consistent tokens that may be reversible with a key; the data remains sensitive.
k-anonymity	Property where every individual shares identifying attributes with at least k-1 others in the dataset.
Differential privacy	Adding mathematical noise so results do not reveal whether any individual is in the data.

Bias amplification	A model reproducing and strengthening biases present in its training data.
Explainability	The ability to understand and communicate why a model produced an output.
Accountability	Clear human ownership of, and answerability for, an AI system's outcomes.

4.4 Additional practice questions

These original questions are written from the official exam guide objectives for Section 4. Answers and explanations follow the questions.

Practice 4.1 MULTIPLE CHOICE

A national retailer has collected 40 gen AI ideas from a company-wide submission form. The chief executive wants a first initiative that shows measurable value within one quarter and builds momentum for a wider programme. You need to recommend how to select it. What should you do?

A. Select the idea with the highest estimated revenue, regardless of data readiness, because transformational value justifies the effort.
B. Plot the ideas by expected business value against actionability and feasibility, and choose a high-value use case that is also highly feasible.
C. Select the idea requested by the most employees, because adoption is the most important KPI.
D. Ask the data science team to fine-tune a model on company data first, then decide which use case it fits best.

Practice 4.2 MULTIPLE CHOICE

A 300-person professional services firm already uses Google Workspace. Partners want help drafting client emails, summarising long documents and capturing meeting notes. The firm has no software developers and wants results within weeks. You need to recommend the most appropriate approach. What should you recommend?

A. Adopt Gemini for Google Workspace and invest in training and change management to drive adoption.
B. Build a custom writing assistant with the Agent Development Kit and deploy it on Agent Runtime.
C. Train a custom language model on the firm's historical documents using Managed Training.
D. Download an open model from Model Garden and host it on the firm's own infrastructure.

Practice 4.3 MULTIPLE CHOICE

An insurer wants a claims assistant that answers adjusters' questions using the latest internal policy manuals, cites its sources and can create follow-up tasks in the claims management system. Policy manuals change monthly. You need to choose an approach that keeps answers current with the least unnecessary effort. What should you do?

A. Fine-tune a foundation model every month on the updated manuals.
B. Train a new custom model from scratch on all historical claims.
C. Rely on a general-purpose chatbot without access to internal data, and ask adjusters to verify every answer manually.
D. Build an agent on Agent Platform that is grounded in the manuals through retrieval (for example Agent Search or RAG Engine) and uses a tool to create tasks in the claims system.

Practice 4.4 MULTIPLE CHOICE

A bank plans to pilot gen AI-generated call summaries for contact centre agents. The chief financial officer says she will fund a wider rollout only if the pilot proves a reduction in handling time. You need to make sure the pilot can demonstrate this. What should you do first?

A. Launch the pilot to all agents and track the number of summaries generated per day.
B. Measure the model's summarisation quality score and report it as the ROI.
C. Define the KPIs during design and capture a baseline of average handling time, or run an A/B comparison with agents not using the tool.
D. Wait until the pilot ends, then ask agents in a survey whether they feel faster.

Practice 4.5 MULTIPLE CHOICE

Three months after launching an internal knowledge assistant, the adoption rate rose quickly to 70% of the intended audience and then fell to 20%. Awareness campaigns are still running. You need to identify the most likely cause to investigate first. What is it?

A. Employees are not aware that the assistant exists.
B. The assistant's usefulness or performance, such as answer quality or relevance, did not meet users' needs.
C. The licence cost per user is too high.
D. The organisation lacks a service perimeter around the assistant.

Practice 4.6 MULTIPLE CHOICE

A manufacturer is assessing its AI maturity with the Google Cloud AI Adoption Framework. Executives are supportive and staff are trained, but data scientists report they cannot find, share or reuse datasets and models created by other teams. You need to identify which theme needs investment. Which theme is it?

A. Lead
B. Learn
C. Access
D. Automate

Practice 4.7 MULTIPLE CHOICE

A telecoms company's security operations centre monitors network and endpoint alerts but has no visibility of what users send to, or receive from, its customer-facing gen AI assistant. The CISO wants AI included in existing monitoring and incident response. You need to identify which SAIF core element this most directly addresses. Which is it?

A. Extend detection and response to bring AI into an organisation's threat universe.
B. Automate defences to keep pace with existing and new threats.
C. Harmonise platform-level controls to ensure consistent security across the organisation.
D. Contextualise AI system risks in surrounding business processes.

Practice 4.8 MULTIPLE CHOICE

A travel company's public chatbot has been manipulated by users typing instructions that make it ignore its rules, and testers found it occasionally repeats card numbers that customers pasted into the chat. You need a runtime control that screens prompts and responses for both issues. What should you use?

A. VPC Service Controls
B. Cloud Monitoring dashboards
C. IAM custom roles
D. Model Armor

Practice 4.9 MULTIPLE CHOICE

A pharmaceutical company stores tuning data for a proprietary model in Cloud Storage and BigQuery. The security team worries that an attacker with stolen credentials, or a misconfigured access policy, could copy the data to an external project. You need a control that blocks data from leaving a defined trusted boundary. What should you recommend?

A. Grant the basic Owner role only to senior engineers.
B. Create a VPC Service Controls perimeter around the projects and services, used together with least-privilege IAM.
C. Enable Explainable AI on the model.
D. Apply bucketing to all numeric fields in the dataset.

Practice 4.10 MULTIPLE CHOICE

A health insurer wants analysts to train a churn model on member histories joined across three tables by member ID. Analysts must not see real member IDs, but the fraud team, under a documented legal process, must occasionally recover the original ID for flagged records. You need to choose a privacy approach. What should you do?

A. Anonymise the data with differential privacy so no individual can ever be identified.
B. Redact member IDs from all three tables.
C. Pseudonymise member IDs with reversible, deterministic tokenisation in Sensitive Data Protection, and keep the key in Cloud Key Management Service with restricted access.
D. Replace each member ID with a random value generated independently in each table.

Practice 4.11 MULTIPLE CHOICE

An HR department proposes a gen AI assistant that ranks job applicants, tuned on ten years of past hiring decisions. In early tests the assistant ranks candidates from some universities and regions consistently lower. You need to recommend the most responsible next step. What should you recommend?

A. Pause rollout, review the training data for historical bias and representativeness, evaluate outcomes by relevant subgroups and keep human reviewers accountable for decisions.
B. Launch as planned, because the model reflects the company's actual historical preferences.
C. Increase the model temperature so rankings vary more between runs.
D. Remove all logging so that applicants' data cannot be exposed.

Practice 4.12 MULTIPLE CHOICE

A lender uses a predictive model on Agent Platform to score small-business loan applications. A regulator asks the lender to explain why a specific applicant was declined, and the board asks who is answerable for the model's decisions. You need to address both requests. What should you do?

A. State that the cloud provider is responsible for model outcomes because it hosts the model.
B. Replace the model with a larger generative model, which is more accurate and therefore needs no explanation.
C. Publish the model's overall accuracy figure to the regulator.
D. Use feature-based explanations from Explainable AI on Agent Platform for the individual decision, and assign named business and risk owners with documented review, monitoring and audit logs.

Answers and explanations

4.1: B. Google Cloud's prioritisation matrix plots use cases by business value against actionability and feasibility; high-value, highly feasible "Best Bets" deliver measurable value quickly. A ignores feasibility and data readiness. C confuses popularity with value. D puts technology before the business problem.

4.2: A. A common productivity need, existing Workspace usage, no developers and urgency all point to the "buy" end of the spectrum. Value then depends on adoption, so training and change management matter. B, C and D add cost, time and skills the firm does not have.

4.3: D. Grounding through retrieval keeps answers current as manuals change and supports citations, while a tool lets the agent act in the claims system. Monthly fine-tuning (A) or training from scratch (B) is costly and still lags behind changes. C does not use internal knowledge and shifts all risk to users.

4.4: C. Google Cloud recommends defining the measurement plan during design and establishing a baseline or A/B test so improvements can be attributed. A measures activity, not outcome. B is a model quality metric, not ROI. D is subjective and has no baseline.

4.5: B. Google Cloud's guidance is that consistently low adoption suggests lack of awareness, while high adoption that drops suggests a lack of performance or usefulness. Awareness is already high (A). C and D do not explain the pattern.

4.6: C. The Access theme covers data management and the ability to share, discover and reuse data and ML artefacts. Lead (A) and Learn (B) are already strong; Automate (D) concerns operating pipelines in production.

4.7: A. This SAIF element is about monitoring AI systems, including gen AI inputs and outputs, and bringing them into detection and incident response. B is about using AI to speed up defence, C about consistent controls across platforms, D about assessing risk in business context.

4.8: D. Model Armor screens prompts and responses for prompt injection and jailbreaks and, with Sensitive Data Protection, for sensitive data such as card numbers. VPC Service Controls (A) protects data perimeters, monitoring (B) observes but does not block, and IAM (C) controls resource access, not prompt content.

4.9: B. VPC Service Controls creates perimeters that block copying data outside the boundary and deny access from unauthorised networks even with stolen credentials or misconfigured IAM; Google recommends combining it with IAM. A uses a basic role, which should not be used in production. C and D do not prevent exfiltration.

4.10: C. Deterministic tokenisation preserves referential integrity for joins and is reversible with the key for authorised re-identification; protecting the key limits who can do so. A and B prevent the required re-identification (and B breaks the joins). D breaks referential integrity across tables.

4.11: A. Models can amplify bias in historical data. Responsible practice is to review data, evaluate fairness by subgroup and keep humans accountable for consequential decisions before rollout. B entrenches unfairness. C does not address bias. D undermines accountability and auditability.

4.12: D. Feature attributions explain how each input contributed to a specific prediction, and accountability requires named owners, governance and audit trails. A is wrong because the deploying organisation remains accountable. B removes explainability rather than providing it. C does not explain an individual decision.

Key takeaways

- Recognise solution types (text, image, code, personalisation, discovery, automation) and start with a specific persona and task, often targeting information deficits or drudgery.
- Gen AI needs are shaped by business requirements (impact, alignment, accuracy, risk tolerance, adoption, speed, budget, compliance) and technical constraints (data readiness, modality, latency, integration, security, skills).
- Choose along the buy-to-build spectrum: prebuilt apps (Gemini for Google Workspace, Gemini Enterprise), managed search and agent services (Agent Search), customisation on Gemini Enterprise Agent Platform, and custom or open models only when necessary. Invest in a platform, not just a model.
- The AI Adoption Framework has four areas (people, process, technology, data), six themes (Lead, Learn, Access, Scale, Secure, Automate) and three phases (tactical, strategic, transformational).
- Integration steps: vision and strategy (top-down and bottom-up), prioritisation by value versus feasibility, data and guardrails, pilot and iterate, change management and skills, scale and govern.
- Measure across model quality, system quality, adoption, business operations and business value; plan measurement at design time and compare against a baseline.
- Secure AI spans the whole lifecycle; SAIF's risk map covers Data, Infrastructure, Model and Application components and 15 risks.
- SAIF's six core elements: expand foundations, extend detection and response, automate defences, harmonise platform controls, adapt controls, contextualise risks. SAIF is a framework, not a product.
- Tools: secure-by-design infrastructure, IAM (least privilege), VPC Service Controls (perimeters), Security Command Center with AI Protection, Model Armor (prompt and response screening), Sensitive Data Protection, Cloud Logging, Monitoring and Audit Logs.
- Google's AI Principles: bold innovation; responsible development and deployment; collaborative progress, together. Responsible AI is continuous, not a final gate.
- Anonymisation is irreversible and unlinkable; pseudonymisation keeps consistent, sometimes reversible tokens and remains sensitive.
- Data quality drives bias and fairness; evaluate by subgroup and keep humans in the loop for consequential decisions. Accountability needs named owners and audit trails; explainability uses feature-based and example-based explanations, plus citations for gen AI.

References

- Generative AI Leader certification exam guide
- Google Cloud: Generative AI Leader certification
- Google Skills: Generative AI Leader Certification learning path
- Google Cloud's AI Adoption Framework (blog)
- How to build an effective AI strategy
- The Prompt: choosing your first custom generative AI use cases
- Generating value from generative AI: ROI study
- Gen AI: going from prototype to production

- Beyond the pilot: five lessons from Google Cloud's AI transformation strategy
- Choosing the right gen AI model: trade-offs and benefits
- KPIs for gen AI: why measuring your new AI is essential
- Gen AI KPIs: measuring AI success, a deep dive
- Five gen AI security terms business leaders should know
- Gemini Enterprise Agent Platform product page
- Agent Platform overview
- Name changes from Vertex AI
- Gemini Enterprise (Gemini for business)
- Google's Secure AI Framework (SAIF), Google Safety Center
- SAIF.Google resource hub
- SAIF: risks
- SAIF: map
- SAIF: components
- SAIF: controls
- Well-Architected Framework, AI and ML perspective: Security
- Google infrastructure security design overview
- IAM overview
- VPC Service Controls overview
- Security Command Center overview
- AI Protection overview
- Model Armor
- Cloud Logging overview
- Sensitive Data Protection: de-identifying sensitive data
- Sensitive Data Protection: pseudonymization
- Sensitive Data Protection: transformation reference
- How Google anonymizes data
- Google AI Principles
- Google Cloud: Responsible AI
- Responsible AI and model limitations on Agent Platform
- Explainable AI overview

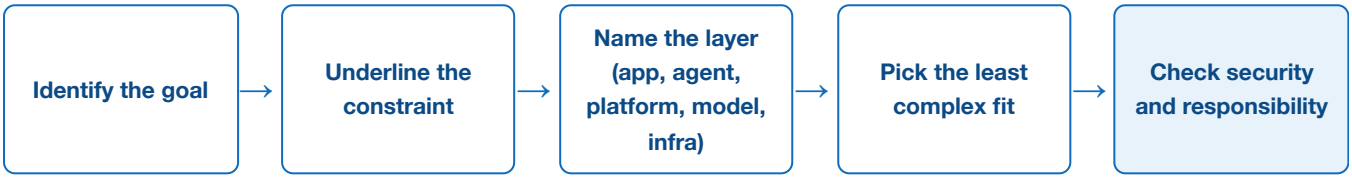
CROSS-SECTION PRACTICE

Case study practice

Cymbal Retail's gen AI journey

Leader-level questions rarely test one fact in isolation. They describe an organisation with goals and constraints and ask for the best next step. This chapter walks one fictional company through all four sections of the exam.

C.1 A method for scenario questions



1. **Goal.** Create, summarise, discover, or automate? Internal productivity or customer experience?
2. **Constraint.** Speed to value, limited technical skills, data residency, privacy, cost, accuracy, freshness.
3. **Layer.** Prebuilt application first, then platform customisation, then custom models. Google's guidance favours the simplest solution that meets the need.
4. **Technique.** If output quality is the problem, match the limitation to the fix: knowledge cutoff to grounding, company facts to RAG, format or tone to prompting or tuning, high-stakes decisions to human in the loop.
5. **Responsibility.** Any option that ignores privacy, fairness, transparency, or security is rarely the best answer.

C.2 Case study: Cymbal Retail

Overview

Cymbal Retail sells home goods in 12 countries through 300 stores and an e-commerce site with 80,000 products. The CEO wants measurable gains from gen AI within two quarters. The company already uses Google Workspace and BigQuery.

Situation

- Store managers spend hours writing weekly reports and searching policies spread across Drive, an intranet, and a ticketing system.
- Online shoppers abandon searches because the site's keyword search fails on natural-language queries such as "durable rug for a house with dogs".
- The contact centre handles 50,000 calls a month; average handle time is rising and quality reviews cover under 2% of calls.
- Marketing wants on-brand product images and short videos for seasonal campaigns.
- A pilot chatbot answered questions about a returns policy that changed last month with the old policy.

- The data team has limited ML experience. Legal requires that customer data is not used to train third-party models and that personal data in transcripts is protected.

Mapping needs to approaches

Need	Best-fit approach	Section
Store managers' reports and policy search	Gemini for Google Workspace for drafting; Gemini Enterprise for search and agents across Drive, intranet, and ticketing	2
Natural-language product search	Agent Search over the product catalogue	2
Contact-centre efficiency and quality	Agent Assist for live agents; Conversational Insights for analysing all calls; Conversational Agents for self-service	2
On-brand images and video	Imagen and Veo on Gemini Enterprise Agent Platform, with brand guidelines in prompts and human review	1, 2
Outdated returns policy answers	Grounding in first-party data through RAG (for example Agent Search over current policy documents) instead of relying on model knowledge	3
Limited ML skills	Prebuilt and low-code offerings first (Workspace, Gemini Enterprise, Agent Search, Agent Studio)	2, 4
Data protection	Google Cloud's enterprise data commitments, IAM least privilege, Sensitive Data Protection de-identification of transcripts, SAIF-aligned controls	4
Measurable gains in two quarters	Prioritise high-value, high-feasibility use cases; define KPIs (time saved per manager, search conversion, handle time, CSAT) before pilots	4

Practice items

Practice C.1 SINGLE ANSWER

Cymbal Retail wants the fastest path to help store managers draft weekly reports in the tools they already use. What should it adopt?

A. A custom model trained on past reports
B. Gemini for Google Workspace
C. Conversational Insights
D. Gemma deployed on-premises

Practice C.2 SINGLE ANSWER

Shoppers' natural-language searches fail on the website. Which offering best addresses this?

A. Agent Search over the product catalogue
B. Google AI Studio
C. Cloud Vision API
D. Gemini app with a Gem

Practice C.3 SINGLE ANSWER

The pilot chatbot gave last month's returns policy. What is the most likely root cause, and what is the recommended fix?

A. Temperature too low; increase it
B. The model relies on knowledge that is outdated for this question; ground responses in the current policy documents with RAG
C. The context window is too small; switch to a larger model
D. Bias in training data; fine-tune on fairer data

Practice C.4 MULTIPLE SELECT (CHOOSE TWO)

The contact centre wants to reduce handle time and review far more calls for quality. Which two components of Customer Engagement Suite should it use?

A. Agent Assist
B. Conversational Insights
C. Model Garden
D. Gemini for Google Workspace
E. Document AI

Practice C.5 SINGLE ANSWER

Legal requires that personal data in call transcripts is protected before analysts use them, while preserving the ability to link records for a single customer when authorised. Which technique fits?

A. Anonymisation by deleting all identifiers irreversibly
B. Pseudonymisation that replaces identifiers with tokens that authorised processes can re-identify
C. Increasing model temperature
D. Publishing transcripts with names masked by hand

Practice C.6 SINGLE ANSWER

The CEO asks how to choose the first gen AI project. What should leadership do?

A. Start with the most technically ambitious idea
B. Score candidate use cases on business value and feasibility (data readiness, risk, skills) and pick a high-value, high-feasibility one with clear KPIs
C. Wait until a custom foundation model can be trained
D. Let each department choose independently with no shared criteria

Answers and explanations

C.1: B. Gemini for Google Workspace brings drafting and summarising into Docs, Gmail, and Sheets with no build effort, matching speed and limited skills.

C.2: A. Agent Search provides Google-quality semantic search over a site or catalogue. AI Studio is a developer playground; Vision API classifies images; the Gemini app is for individual productivity.

C.3: B. A foundation model's knowledge is fixed at its cutoff and does not include your latest policies. Grounding in current first-party documents fixes freshness and accuracy.

C.4: A and B. Agent Assist supports live agents in real time; Conversational Insights analyses conversations at scale for topics, sentiment, and quality.

C.5: B. Pseudonymisation replaces identifiers but allows controlled re-identification; anonymisation is irreversible and would prevent authorised linking.

C.6: B. Prioritising by value and feasibility with KPIs defined up front is the recommended way to deliver measurable results quickly.

EXAM DAY

Exam technique and signal words

Turning knowledge into points

E.1 Signal words that map to answers

If the scenario says...	Think first of...
Labelled examples, predict a known category	Supervised learning
Find groups or patterns in unlabelled data	Unsupervised learning (clustering)
Learn by trial, reward, and penalty	Reinforcement learning
Generate images by gradually removing noise	Diffusion model
One model, many tasks, adaptable	Foundation model
Text, images, audio, video in one model	Multimodal model (Gemini)
Open, lightweight, run locally or fine-tune on own hardware	Gemma
Text-to-image or image editing	Imagen
Text- or image-to-video	Veo
Analyse very long documents or many files at once	Gemini with a long context window
Hardware for training and serving at scale	Infrastructure layer: TPUs, GPUs, AI Hypercomputer
Choose among many models, avoid lock-in	Model Garden; Google's open approach
Reusable custom assistant for an individual	Gems in the Gemini app
Employees search and act across company systems	Gemini Enterprise
Gen AI in Gmail, Docs, Sheets, Meet	Gemini for Google Workspace
Customer-facing semantic search	Agent Search
Virtual agent / chatbot for customers	Conversational Agents
Real-time help for human contact-centre agents	Agent Assist
Analyse trends and sentiment across conversations	Conversational Insights
End-to-end cloud contact centre	Google Cloud Contact Center as a Service
Custom model with little ML expertise on tabular, image, or text data	AutoML on Gemini Enterprise Agent Platform
Manage model versions and lineage	Model Registry

If the scenario says...	Think first of...
Consistent, reusable features for training and serving	Feature Store
Agent must act in other systems	Tools: functions, extensions, data stores, plugins, APIs
Extract fields from scanned forms	Document AI
Quick prototyping with the Gemini API	Google AI Studio
Enterprise prototyping with governance in Google Cloud	Agent Studio
Knowledge cutoff, outdated facts	Grounding (Google Search for world data; RAG for company data)
Answers must use internal documents	RAG / grounding in first-party data (Agent Search, RAG Engine)
Consistent style, format, or domain behaviour	Few-shot prompting, then fine-tuning if needed
Step-by-step reasoning	Chain-of-thought prompting
Reason and use tools iteratively	ReAct prompting
Break a complex task into stages	Prompt chaining
More predictable, factual output	Lower temperature (and narrower top-p)
More creative, varied output	Higher temperature
High-stakes decisions (loans, medical, legal)	Human in the loop
Model quality declines over time	Monitoring for drift; evaluation; versioning
Protect AI systems from attacks	SAIF; IAM; Security Command Center
Remove identity irreversibly	Anonymisation
Replace identity, allow controlled re-linking	Pseudonymisation
Who is responsible; can we explain decisions	Accountability and explainability

E.2 Distractor patterns

Pattern	Example
Over-engineering	Training a custom model when a prebuilt product or prompt solves the need.
Wrong audience	Consumer Gemini app for an enterprise-wide governed deployment; Workspace for customer-facing search.
Wrong fix for the limitation	Fine-tuning to fix freshness; raising temperature to fix accuracy.
Ignoring responsibility	Fully automating a high-stakes decision with no human review or transparency.
Studio confusion	Google AI Studio for governed enterprise work; Agent Studio for a quick personal API experiment.

E.3 Time management

- Ninety minutes for 50 to 60 questions is about 90 seconds each; leave ten minutes to review flagged items.
- Read the last sentence first to know what is asked, then read the scenario for the constraint.
- When two answers seem right, choose the one that is simpler, more responsible, and more directly tied to the stated business goal.

APPENDIX A

One-page cheat sheet

The decisions the exam asks most often

Topic	Remember
AI family	AI ⊃ ML ⊃ deep learning ⊃ gen AI. NLP is the field of understanding and generating language. LLMs are foundation models specialised in language.
ML approaches	Supervised (labelled, predict), unsupervised (unlabelled, find structure), reinforcement (reward-driven actions).
ML lifecycle	Data ingestion, data preparation, model training, model deployment, model management (monitoring, versioning, retraining).
Data quality	Completeness, consistency, relevance, availability, cost, format. Structured (tables) vs unstructured (text, images, audio, video). Labelled vs unlabelled.
Choosing a model	Modality, context window, security, availability and reliability, cost, performance, fine-tuning and customisation.
Google models	**Gemini**: multimodal, long context, reasoning. **Gemma**: open, lightweight, self-hostable. **Imagen**: images. **Veo**: video.
Layers	Infrastructure, models, platform, agents, applications.
Strengths of Google Cloud	AI-first research heritage; enterprise-ready (responsible, secure, private, reliable, scalable); integrated ecosystem; open approach; AI-optimised infrastructure; customer control of data; democratised tools.
Prebuilt apps	Gemini app and Gems; Gemini Enterprise; Gemini for Google Workspace.
Customer experience	Agent Search; Customer Engagement Suite: Conversational Agents, Agent Assist, Conversational Insights, Contact Center as a Service.
Builders	Gemini Enterprise Agent Platform: Model Garden, Agent Studio, Agent Search, AutoML, RAG Engine, ADK, Agent Runtime.
Agent tools	Extensions, functions, data stores, plugins; Google Cloud services and pre-built APIs as tools.
Limitations	Data dependency, knowledge cutoff, bias, fairness, hallucinations, edge cases.
Fixes	Grounding, RAG, prompt engineering, fine-tuning, human in the loop; then monitor and evaluate continuously.
Prompting	Zero-shot, one-shot, few-shot, role, prompt chaining; chain-of-thought; ReAct.
Grounding	First-party (company), third-party (licensed or partner), world (Google Search). Offerings: prebuilt RAG with Agent Search, RAG APIs, Grounding with Google Search.
Parameters	Temperature (randomness), top-p (nucleus sampling), output tokens (length), safety settings (block thresholds).
Adoption	Define goals, prioritise use cases by value and feasibility, pilot, measure KPIs, scale with governance and change management.

Topic	Remember
Secure AI	Security across the ML lifecycle; SAIF; IAM, Security Command Center, secure-by-design infrastructure, workload monitoring.
Responsible AI	Transparency, privacy (anonymisation vs pseudonymisation), data quality, bias and fairness, accountability, explainability.

APPENDIX B

Glossary

Terms used across the sections

Agent	A system that uses a model to reason, plan, and take actions with tools to achieve a goal, observing results in a loop.
Context window	The amount of input (in tokens) a model can consider at once.
Diffusion model	Generative model that learns to create data, typically images or video, by reversing a gradual noising process.
Drift	Change in data or model behaviour over time that degrades quality.
Embedding	Numeric vector representing meaning, used for semantic search and RAG.
Fine-tuning	Further training a pre-trained model on task-specific examples to adapt behaviour.
Foundation model	Large model trained on broad data that can be adapted to many tasks.
Gem	A customised version of Gemini in the Gemini app with saved instructions for repeated tasks.
Grounding	Connecting model output to verifiable sources of information.
Hallucination	Plausible-sounding output that is false or unsupported.
Human in the loop	A person reviews, corrects, or approves AI outputs before they take effect.
Knowledge cutoff	The date after which a model has no training knowledge.
Multimodal	Able to process or generate more than one type of data (text, images, audio, video).
Prompt engineering	Designing inputs to guide a model toward the desired output.
Prompt tuning	Parameter-efficient adaptation that learns soft prompt vectors while keeping model weights frozen.
RAG	Retrieval-augmented generation: retrieve relevant information and include it in the prompt so answers are grounded.
ReAct	Prompting pattern that interleaves reasoning and actions (tool use) with observations.
SAIF	Google's Secure AI Framework for securing AI systems.
Temperature	Sampling parameter controlling randomness; lower is more deterministic.
Token	Unit of text (part of a word) processed by a model; used for limits and pricing.
Top-p	Nucleus sampling: choose from the smallest set of tokens whose cumulative probability reaches p.
TPU	Tensor Processing Unit: Google's custom accelerator for ML.

APPENDIX C

Hands-on exploration plan

No code required

#	Activity	What to observe	Section
1	Gemini app and a Gem	How saved instructions make repeated tasks consistent.	2
2	Long-document analysis	Upload a long report; ask for a summary with section references; note the effect of context length.	1
3	Agent Studio prompt comparison	Compare two models and two temperatures on the same prompt.	2, 3
4	Prompting techniques	Zero-shot vs few-shot classification; chain-of-thought on a multi-step problem; a two-step prompt chain.	3
5	Grounding	Ask about a recent event with and without Grounding with Google Search; inspect citations.	3
6	Model Garden tour	Find Gemini, Gemma, Imagen, Veo, and a partner model; read their model cards.	1, 2
7	Product pages	Gemini Enterprise, Workspace, Customer Engagement Suite, Agent Search: list one use case and one business metric for each.	2
8	SAIF and AI Principles	Map each SAIF element to one control your organisation already has.	4
9	Use-case scoring	Score five ideas from your organisation on value and feasibility; define KPIs for the top one.	4

APPENDIX D

Final readiness checklist

Tick every line before booking

☐	I can...
☐	Define AI, ML, NLP, gen AI, foundation, multimodal, and diffusion models, LLMs, prompt engineering, and prompt tuning.
☐	Distinguish supervised, unsupervised, and reinforcement learning with examples.
☐	Name the ML lifecycle stages and a Google Cloud tool for each.
☐	Choose a foundation model using modality, context window, security, availability, cost, performance, and customisation.
☐	Explain data quality dimensions, structured vs unstructured, and labelled vs unlabelled data.
☐	Place offerings in the five layers and describe Gemini, Gemma, Imagen, and Veo.
☐	Explain Google Cloud's strengths, including the open approach and AI-optimised infrastructure.
☐	Match needs to the Gemini app, Gemini Enterprise, Workspace, Agent Search, and Customer Engagement Suite components.
☐	Describe Agent Platform capabilities: Model Garden, Agent Search, AutoML, RAG offerings, custom agents.
☐	Explain agent tools and when to use Agent Studio vs Google AI Studio.
☐	Match each model limitation to a mitigation and describe continuous monitoring practices.
☐	Use and recognise zero-, one-, few-shot, role, chaining, chain-of-thought, and ReAct prompting.
☐	Explain grounding types and offerings, and the effect of temperature, top-p, output length, and safety settings.
☐	Describe adoption steps, solution selection, and impact measurement.
☐	Explain SAIF, security across the ML lifecycle, and Google Cloud security tools.
☐	Explain responsible AI: transparency, privacy techniques, bias, fairness, accountability, explainability.
☐	Score 85% or better on every chapter's practice questions and complete Google's official sample questions.

APPENDIX E

Official resources

Where to go next

- Generative AI Leader certification page
- Official exam guide (PDF)
- Google Cloud Skills Boost (Generative AI Leader learning path)
- Gemini Enterprise Agent Platform name changes
- Gemini Enterprise
- Gemini for Google Workspace
- Secure AI Framework (SAIF)
- Google AI Principles

Section chapters list the specific Google sources used for each topic.