PREPKLOUD FIELD GUIDE SERIES

PrepKloud

Azure AI Fundamentals The Complete AI-901 Field Guide

A practical, exam-ready walkthrough of responsible AI, generative AI concepts, and Microsoft Foundry implementation — built from the active AI-901 blueprint effective April 15, 2026.

1st Edition · 2026 · prepkloud.com

About this book

Azure AI Fundamentals: The Complete AI-901 Field Guide is published by PrepKloud, an independent cloud-certification preparation platform. This guide maps directly to the two graded domains of the active Microsoft AI-901 exam skills outline: Identify AI concepts and capabilities (40-45%) and Implement AI solutions by using Microsoft Foundry (55-60%).

AI-901 replaced the older AI-900 blueprint, with English-language objectives updated April 15, 2026. It is a more implementation-oriented exam: more than half of the questions expect you to reason about a working Microsoft Foundry project, not just define AI terms. This book was written for the current, active blueprint — not the retired AI-900 five-domain outline.

Every chapter pairs plain-English explanations with an original diagram, exam-focused callouts, and practice questions. Basic familiarity with Azure resources and comfort reading simple Python or JSON snippets

is assumed, matching the exam's official audience profile — but no prior AI or machine learning background is required.

How to use this book: Read chapters in order the first time through. Use the glossary and cheat sheet in the appendix for last-minute review. Attempt every practice question before checking the answer, and build the two hands-on labs described in Chapter 9 before your exam date.

Disclaimer: This is an independent study resource. Microsoft, Azure, Microsoft Foundry, and AI-901 are trademarks of Microsoft Corporation. This guide is not endorsed by or affiliated with Microsoft. Service names, model lists, and pricing mentioned reflect general product behavior at time of writing and change quickly in the AI space — always confirm current details on learn.microsoft.com before your exam. This book contains no exam dumps, leaked questions, or recalled content.

Table of Contents

CHAPTER 01

How This Book Is Organized

Welcome to your AI-901 study journey. Microsoft Certified: Azure AI Fundamentals is designed for people at the beginning of a career in AI solution development. You are expected to have conceptual knowledge of AI solutions in Azure and foundational technical skills to work with them — including comfort with Python syntax and programming techniques, familiarity with Azure resources, and familiarity with REST APIs, SDKs, and CLIs. You do not need production machine-learning experience.

AI-901 is **not** the retired AI-900 exam. AI-900 used five older, mostly descriptive domains and retired June 30, 2026. AI-901 is the active exam, with English-language objectives updated April 15, 2026, and it is meaningfully more implementation-focused: it expects you to reason about a real Microsoft Foundry project, not just recite definitions.

Domain	Weight	Chapters
Identify AI concepts and capabilities	40–45%	2–5, 10–13
Implement AI solutions by using Microsoft Foundry	55–60%	6–9, 13–14

> 💡 **Study tip:** Because implementation is worth more than half the exam, do not skip the hands-on labs in Chapters 8 and 9. Reading about Foundry is not the same as having

> clicked through creating a project, deploying a model, and testing an agent at least once.

Each chapter follows a consistent structure: a plain-English explanation, one original diagram, exam-focus and exam-trap callouts, and an end-of-chapter checklist for spaced-repetition review.

> 🎯 **Exam focus:** AI-901 questions are scenario-based. Expect prompts like "A retail company wants to extract line-item totals from scanned invoices — which Foundry capability should they use?" rather than pure definition recall. As you read, practice converting every concept into a one-sentence "when would I use this" statement.

Suggested study plan

Week	Focus

1	Chapters 2–5: AI workloads, responsible AI, generative AI concepts, model selection
2	Chapters 6–8: Foundry foundations, prompt engineering, building a client
3	Chapter 9: agents, evaluation, security, and operations
4	Chapters 10–12: language, speech, vision, and image generation
5	Chapters 13–16: multimodal extraction, cost/ops, full practice set

Chapter 1 checklist

☐ I understand the two exam domains and their approximate weight.

☐ I know AI-901 is the active exam, distinct from the retired AI-900.

☐ I have a five-week study schedule mapped to this book.

AI Workloads & Considerations

Before you can identify which Azure AI capability fits a scenario, you need a shared vocabulary for what "AI" actually covers. Microsoft's exam groups AI into a small set of recognizable workload families, and most AI-901 questions test whether you can match a business scenario to the right family — not whether you can define artificial intelligence philosophically.

2.1 The workload families you must recognize

- **Generative AI** — producing new content (text, images, audio, code) from a prompt, using a large language model or diffusion model.
- **Agentic AI** — a model paired with instructions and tools that can take multi-step action toward a goal, not just answer a single question.
- **Text analysis / NLP** — extracting structure from existing text: keywords, entities, sentiment, summaries.
- **Speech** — converting between spoken audio and text, in both directions.
- **Computer vision** — interpreting or generating images, including object detection and image captioning.

- **Information extraction** — pulling structured fields out of unstructured documents, images, audio, or video (Content Understanding).

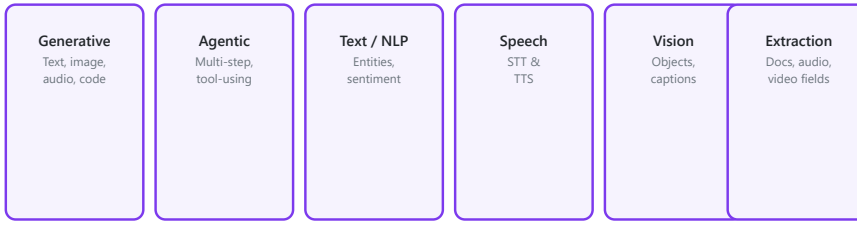

Figure 2.1 — The six AI workload families tested across the AI-901 blueprint.

> ⚠ **Exam trap:** A common distractor swaps **generative** and **agentic** AI. Generative AI answers one prompt with one output. Agentic AI keeps working across multiple steps, may call tools, and can pursue a goal with some autonomy — the exam word to watch for is "autonomously" or "multi-step."

2.2 Key considerations for any AI workload

For every workload family, Microsoft expects you to weigh the same set of practical considerations, regardless of which specific Azure service you pick:

Consideration	What it means in practice
Capability fit	Does the workload actually need generation, or would simple extraction/classification be cheaper and more reliable?
Modality	Text, image, audio, video, or a mix — this narrows which service or model family applies.
Latency	Real-time chat needs low latency; batch document processing can tolerate slower, higher-quality processing.
Accuracy & hallucination risk	Generative outputs can be fluent but wrong — high-stakes scenarios need grounding, citations, or human review.

Cost	Token-based generative pricing scales with input/output size; specialized models are often cheaper per call for narrow tasks.
Deployment & availability	Region availability, quota, and SLA differ by model and service tier.

> 🎯 **Exam focus:** If a scenario says "the company has a small, well-defined extraction task and wants the lowest cost and most predictable output," the expected answer is usually a specialized service (Language, Vision, or Content Understanding) — not a general-purpose generative model.

Scenario

Customer support ticket triage

A support team receives thousands of tickets per day in five languages and wants to automatically detect the customer's sentiment and route angry customers to a senior agent. Is this generative AI?

Analysis: No new content needs to be generated — the task is classification and sentiment detection, which is a text analysis / NLP workload. A generative model could technically do this, but it would be more expensive, harder to make consistent, and unnecessary for a well-defined classification task.

Chapter 2 checklist

☐ I can name the six workload families and give a one-line example of each.

☐ I can distinguish generative from agentic AI.

☐ I can list at least four considerations (capability, modality, latency, cost) that drive workload selection.

Responsible AI

Responsible AI is one of the most consistently tested topics on AI-901, and it is tested through scenarios, not definitions. Microsoft organizes responsible AI around six principles. Memorizing the names is not enough — you need to recognize which principle a scenario violates or protects.

Principle	What it protects against
Fairness	Aggregate accuracy hiding poor performance for a subgroup (e.g., an accent or demographic).
Reliability & safety	Unsafe behavior under edge cases; requires testing, monitoring, and safe failure modes.

Privacy & security	Excess data collection/retention and weak access control around identities, credentials, data, and models.
Inclusiveness	Solutions that only work well for a narrow group of users.
Transparency	Users not understanding that they are interacting with AI, or not understanding a system's limits.
Accountability	No named human owner, escalation path, or governance for AI decisions.

🎯 **Exam focus:** Watch for scenarios where overall accuracy is high (e.g., "95% accurate across all users") but a specific group is underserved — this is a textbook **fairness** question, even though the headline number sounds good.

⚠ **Exam trap:** Do not confuse **reliability & safety** (does the system behave safely, including under failure) with **accountability** (is there a human who owns the outcome).

> A model that fails safely but has no named owner is still an accountability gap.

3.1 Privacy and security in AI systems

Privacy is about purpose limitation, data minimization, retention limits, deletion, and appropriate access to personal data used by or produced by an AI system. Security extends this to protecting identities, credentials, data, deployed models, connected tools, and the underlying infrastructure — for example, using least-privilege access instead of shared API keys embedded in code.

3.2 Transparency and accountability

Transparency means users can tell they are interacting with AI and understand key limitations (for example, that a chatbot might not know about events after its training or knowledge-source cutoff). Accountability means a named person or team owns the system's behavior, has an escalation path for harmful or incorrect outputs, and reviews incidents.

Scenario

Hiring-screening assistant

A company deploys a generative AI assistant to summarize resumes for recruiters. An audit finds the assistant summarizes resumes from one university program noticeably less favorably than others, despite similar underlying qualifications.

Analysis: This is a fairness issue — the aggregate summarization quality may look fine, but a specific subgroup is treated worse. The fix is not just "more testing" generically, but specifically testing and monitoring for subgroup disparities, and defining an

escalation path (accountability) for when they're found.

Q: A company wants users to always know when they are chatting with an AI system rather than a human. Which principle does this satisfy?

- ○ Fairness
- ○ Transparency
- ○ Accountability
- ○ Inclusiveness

Which principle is most directly tested when a system fails unpredictably under unusual or adversarial input rather than failing safely?

Answer: Reliability and safety

Reliability and safety is specifically about designing for safe failure, monitoring, and escalation — not just average-case correctness.

Chapter 3 checklist

- ☐ I can name all six responsible AI principles.
- ☐ I can identify which principle a given scenario violates.
- ☐ I understand the difference between reliability/safety and accountability.

How Generative AI & Language Models Work

You do not need to derive the mathematics behind a transformer model for AI-901, but you do need working conceptual fluency in how large language models (LLMs) process input and why they sometimes produce confident, fluent, and wrong answers.

4.1 Tokens and context

Models do not read raw text — text is broken into **tokens** (roughly word-pieces) before being processed. The model's **context window** is the

maximum number of tokens (input plus output) it can consider at once. A longer conversation or a large document pasted into a prompt consumes context budget, which is why very long inputs can cause a model to "forget" earlier instructions.

4.2 Probabilistic generation and hallucination

An LLM generates output by predicting the most probable next token given everything before it — it is not looking up facts in a database unless it has been explicitly connected to one (grounding). This is why models can produce **hallucinations**: fluent, confident-sounding statements that are factually incorrect, because the model is optimizing for plausible continuation, not verified truth.

> ⚠ **Exam trap:** A frequent exam trap treats "the model sounded confident" as evidence of correctness. Confidence in tone is not evidence of accuracy — grounding (connecting the model to verified source data) and evaluation are the actual mitigations.

Figure 4.1 — Why hallucination risk exists, and where grounding intervenes.

4.3 Generation parameters

Two parameters show up repeatedly in scenario questions:

- **Temperature** — controls randomness. Lower temperature produces more deterministic, repetitive output; higher temperature produces

more varied, creative output. Temperature is a control on style/variety, not a guarantee of factual accuracy.

- **Output/token limits** — cap how long a response can be, affecting both cost and whether a full answer fits.

> 🎯 **Exam focus:** If a scenario needs consistent, repeatable output (e.g., structured data extraction), the expected answer is **lower temperature**. If a scenario wants varied marketing copy ideas, the expected answer is **higher temperature**.

Does raising the temperature parameter reduce hallucination risk?

Answer: No

Temperature only affects output variety/randomness, not factual grounding. Reducing hallucination requires grounding, retrieval, or evaluation — not a temperature change.

Chapter 4 checklist

- ☐ I can explain tokens and context window in plain language.
- ☐ I can explain why LLMs hallucinate and what grounding does about it.
- ☐ I know temperature affects variety, not truthfulness.

Choosing the Right Model

AI-901 deliberately avoids testing one fixed, memorized list of model names, because the available model catalog changes frequently. Instead, it tests whether you can reason about model selection using stable criteria.

5.1 Selection criteria

Criterion	Question to ask
Capability	Does it support the task (text, reasoning, tool-calling, vision, audio)?
Modality	Does it accept/produce the input/output types the scenario needs?

Latency	Is a smaller/faster model acceptable, or does the scenario need maximum quality regardless of speed?
Safety	Does it meet the content-safety and responsible-AI requirements for the use case?
Availability & deployment	Is it available in the required region, and can it be deployed at the needed scale?
Cost	What is the per-token or per-call cost relative to the value of the task?

> 💡 **Study tip:** Practice comparing candidate models with the *same* representative prompts and the *same* acceptance criteria before deploying — this exact workflow (compare, then deploy) is explicitly part of the Foundry implementation domain, not just a "nice to have."

Scenario

Real-time voice assistant vs. overnight report generator

Company A needs a voice assistant that must respond within one second. Company B needs a detailed nightly summary report and has no strict time pressure.

Analysis: Company A should prioritize latency, likely choosing a smaller or faster-deployed model even if it sacrifices some output depth. Company B can prioritize quality/capability over latency, since a slower but more thorough model is acceptable overnight.

> ⚠ **Exam trap:** A common wrong-answer pattern picks "the biggest, most capable model" as always correct. The exam expects you to weigh **fit for the scenario**, including cost and latency — not "always pick maximum capability."

Chapter 5 checklist

- ☐ I can list six model-selection criteria without needing a memorized model-name list.
- ☐ I can pick the right tradeoff (latency vs. capability vs. cost) for a given scenario.
- ☐ I understand comparing models before deployment is an expected Foundry workflow step.

Microsoft Foundry Foundations

Microsoft Foundry is the unified platform for building, deploying, and operating AI solutions on Azure. From this chapter forward, the exam shifts from "what is AI" to "can you build with it" — this is where the 55-60% implementation domain lives, so read slowly and try each concept in a real Foundry project if you can.

6.1 Core building blocks

Concept	What it is

Foundry project	The top-level workspace that groups your models, deployments, agents, and connected data/tools.
Model catalog	A browsable list of available foundation models you can deploy into your project.
Deployment	A specific model made callable at an endpoint, with its own quota, region, and version.
Endpoint	The URL/address your application calls to send prompts and receive completions.
Connections	Configured links to external resources (data sources, search indexes, other Azure services) a project can use.

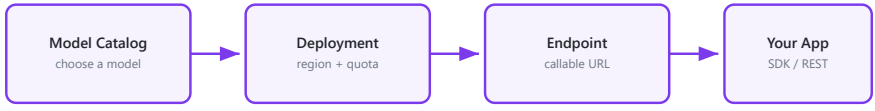

Figure 6.1 — Path from model catalog to a callable endpoint your app uses.

6.2 Authentication

Foundry projects support key-based authentication for quick testing and Microsoft Entra ID (identity-based) authentication for production. Identity-based auth is the recommended pattern for real deployments because it avoids long-lived secrets in application code and supports role-based access control per identity.

> 🎯 **Exam focus:** If a scenario emphasizes "production," "enterprise," or "avoid storing secrets in code," the expected answer is Microsoft Entra ID / identity-based authentication, not a raw API key.

> 💡 **Study tip:** Create a project, browse the model catalog, and deploy at least one lightweight text model before your exam. Muscle memory from actually clicking through project creation → model selection → deployment →

> endpoint testing will make several exam scenarios feel familiar instead of abstract.

Chapter 6 checklist

- ☐ I can define project, model catalog, deployment, endpoint, and connection.
- ☐ I know when identity-based auth is expected over key-based auth.
- ☐ I have deployed at least one model in a Foundry project.

Prompt Engineering

Prompt engineering is the craft of writing instructions that reliably steer a model's output. AI-901 expects you to recognize good prompt structure and common techniques by name and by effect, not to write essays about prompting theory.

7.1 Prompt structure

- **System message** — sets persistent role, tone, and constraints for the whole conversation (e.g., "You are a support assistant. Only answer questions about our product.").

- **User message** — the specific request or question for this turn.
- **Context / grounding data** — reference material supplied so the model answers from real information instead of guessing.
- **Few-shot examples** — sample input/output pairs included in the prompt to demonstrate the desired format.

1. Start with a clear system message defining role and boundaries.
2. State the task explicitly — what output format, what length, what tone.
3. Provide grounding context or examples when consistency matters.
4. Test with edge cases, not just the happy path.

5. Iterate based on evaluation results, not a single manual check.

7.2 Common techniques

Technique	Effect
Zero-shot prompting	Ask directly with no examples — fastest, works for simple/common tasks.
Few-shot prompting	Include 2–5 examples to steer format/style consistency.
Chain-of-thought style prompting	Ask the model to reason step by step before the final answer, improving multi-step accuracy.
Retrieval-augmented grounding	Inject retrieved, verified content into the prompt so answers cite real data instead of relying purely on parametric knowledge.

> ⚠ **Exam trap:** A recurring exam trap: assuming a longer prompt is automatically a better prompt. Precision,

structure, and relevant grounding matter far more than raw prompt length.

A company wants a support bot to only answer questions using their internal knowledge base, and never invent policy details. Which technique best supports this?

Answer: Retrieval-augmented grounding

Grounding injects verified source content into the prompt, directly reducing the chance of the model inventing (hallucinating) unsupported policy details.

Chapter 7 checklist

- ☐ I can name the four parts of prompt structure.
- ☐ I can match zero-shot / few-shot / chain-of-thought / grounding to the right scenario.

☐ I understand why prompt length alone doesn't determine quality.

Building Your First Foundry Client

AI-901 expects you to recognize the shape of client code that calls a deployed model — you will not be asked to write a full application from scratch, but you should be able to read a short SDK or REST snippet and identify what it does or what's missing.

8.1 Two ways to call a deployment

- **SDK** — a language-specific library (Python, .NET, JavaScript, etc.) that wraps authentication and request formatting for you.

- **REST API** — direct HTTP calls to the endpoint, useful when no SDK exists for your language or you need full control over the request.

Code pattern

Minimal chat completion request (conceptual)

A typical request body sent to a deployed chat model includes: the deployment name, a list of messages (system + user), and generation parameters such as temperature and max tokens. The response includes the generated message plus usage/token counts, which is what drives per-call cost.

8.2 Error handling you're expected to recognize

Situation	Expected handling
Rate limiting / throttling	Retry with backoff; do not hammer the endpoint immediately.
Content filtered response	Handle gracefully — inform the user rather than treating it as a generic crash.
Authentication failure	Check credentials/identity configuration before assuming the model itself is broken.
Token/context limit exceeded	Trim or summarize input rather than sending an oversized request.

> 💡 **Study tip:** Try both an SDK call and a raw REST call against the same deployment. Seeing the same request expressed two ways builds the pattern recognition the exam actually tests.

Chapter 8 checklist

☐ I can describe the difference between SDK and REST access to a deployment.

☐ I know the fields a typical chat completion request/response contains.

☐ I can match common failure modes to the correct handling approach.

AI Agents in Foundry

An AI agent pairs a model with instructions, tools, and (often) memory so it can pursue a goal across multiple steps rather than answering one isolated prompt. This is one of the most heavily implementation-focused topics on AI-901.

9.1 Anatomy of an agent

- **Instructions** — the persistent goal/role definition, similar to a system message but scoped to the agent's whole task.

- **Tools** — functions, APIs, or connected data sources the agent may call to gather information or take action.
- **Planning/reasoning loop** — the agent decides which tool to call next based on progress toward its goal.
- **Memory/state** — information carried between steps or sessions.

> 🎯 **Exam focus:** If a scenario describes a system that "decides which tool to use," "takes multiple steps," or "checks a calendar and then books a follow-up," that is an **agent** scenario, not a single-turn generative AI scenario.

9.2 Testing, evaluation, and security boundaries

Because agents can take action (not just produce text), Foundry emphasizes evaluating agents before production release and constraining what tools an agent may call. Key practices:

Practice	Why it matters
Scoped tool access	Only grant the specific tools/data an agent needs — least privilege reduces blast radius of a mistake.
Evaluation before release	Test the agent against representative scenarios and edge cases, not just happy-path demos.
Human-in-the-loop for high-risk actions	Require approval before an agent takes an irreversible or high-impact action (e.g., sending a payment).
Monitoring in production	Log agent decisions/tool calls so failures and drift can be caught after launch.

Scenario

Expense-approval agent

A finance team builds an agent that reviews expense reports and can approve reimbursements under $50 automatically, but must flag anything above that for a human.

Analysis: This demonstrates scoped tool access (the agent's "approve" tool is capability-limited by amount) plus human-in-the-loop for higher-risk, higher-value actions — the correct combination the exam expects for agent safety design.

Q: Which practice most directly reduces the impact of an agent malfunctioning or being manipulated into an unintended action?

- ○ Increasing the model's temperature
- ○ Scoped, least-privilege tool access
- ○ Using a larger context window
- ○ Disabling monitoring to reduce log noise

Chapter 9 checklist

- ☐ I can describe an agent's four core parts: instructions, tools, planning loop, memory.
- ☐ I can distinguish an agent scenario from a single-turn generative scenario.
- ☐ I know why scoped tool access and human-in-the-loop matter for agent security.

Natural Language Processing Workloads

Text/NLP workloads extract structure and meaning from existing text, rather than generating new content. Recognizing these tasks by name is a frequent scenario-matching pattern on the exam.

Capability	What it produces
Key phrase extraction	The main topics/phrases in a document.
Named entity recognition (NER)	People, places, organizations, dates, and other structured entities.
Sentiment analysis	Positive/negative/neutral/mixed scoring, often per sentence.

Language detection	Identifies the language(s) present in a text sample.
Summarization	A shorter version of a longer document, either extractive (pulled sentences) or abstractive (rewritten).
PII detection/redaction	Identifies and can mask personally identifiable information in text.

> 🎯 **Exam focus:** "Extract the names of companies mentioned in these press releases" is an NER question. "Tell me how customers feel about our product" is a sentiment analysis question. Match the verb in the scenario to the capability name.

Scenario

Legal document review

A law firm wants to automatically flag any document that contains a person's social security number before it's shared externally.

Analysis: This is a PII detection scenario — the goal is identifying sensitive entities for compliance, not generating new content or general sentiment.

Chapter 10 checklist

☐ I can name six core NLP capabilities and match each to an example scenario.

☐ I can distinguish extractive vs. abstractive summarization.

Speech Workloads

Speech workloads convert between spoken audio and text, and the exam distinguishes carefully between the two directions and their qualifiers.

Capability	Direction	Typical use
Speech-to-text (STT)	Audio → text	Transcribing meetings, captions, voice commands.
Text-to-speech (TTS)	Text → audio	Voice assistants, accessibility read-aloud, IVR prompts.
Speech translation	Audio → translated text/audio	Real-time multilingual conversation support.

Speaker recognition	Audio → identity	Verifying or identifying who is speaking.

> ⚠ **Exam trap:** Speech translation is not the same as running speech-to-text and then a separate translation step manually — it refers to the integrated capability designed for that combined workload, which the exam may reference directly by name.

A call center wants to automatically produce written transcripts of every support call for quality review.

Answer: Speech-to-text

The goal is converting existing audio into text; no synthesis of new speech or translation is required.

Chapter 11 checklist

- [ ] I can name four speech capabilities and their input/output direction.
- [ ] I can match a scenario's verbs (transcribe, read aloud, translate, verify) to the right capability.

Computer Vision & Image Generation

Vision workloads split into two directions: understanding existing images, and generating new ones.

12.1 Understanding images

Capability	What it produces
Image classification	A label describing the whole image.
Object detection	Bounding boxes plus labels for individual objects in an image.

Image captioning	A natural-language description of the image's content.
Optical character recognition (OCR)	Text extracted from an image (signs, scanned pages, receipts).
Facial detection	Locates faces in an image (distinct from identifying who a specific person is).

> ⚠ **Exam trap:** Object detection returns bounding boxes for multiple distinct items; image classification returns one label for the whole image. A scenario asking to "count how many of each product appear on a shelf" needs object detection, not classification.

12.2 Generating images

Generative image models produce new images from a text prompt (text-to-image) or transform an existing image based on instructions (image

editing/inpainting). As with text generation, output is probabilistic — the same prompt can yield different results, and content-safety filtering applies to both prompts and generated images.

Scenario

Retail shelf audit

A retailer wants to verify that promotional products are actually displayed on store shelves, using photos taken by store staff.

Analysis: This requires locating and counting specific products within a photo — object detection is the correct capability, not simple image classification or captioning.

Chapter 12 checklist

☐ I can distinguish classification, object detection, captioning, OCR, and facial detection.

☐ I understand image generation is probabilistic and subject to content-safety filtering.

Multimodal Extraction with Content Understanding

Content Understanding is the capability family for pulling structured, schema-defined fields out of unstructured, multimodal input — documents, images, audio, and video — in one workflow rather than stitching together separate single-modality services yourself.

13.1 What makes it different from single-modality services

Where OCR extracts raw text and speech-to-text extracts a transcript, Content Understanding is built

to return a structured, schema-defined output (e.g., "invoice number," "total amount," "vendor name") from a document, and analogous structured fields from images, audio, or video — regardless of the source modality.

Figure 13.1 — Multiple input modalities converge into one schema-defined output.

13.2 Defining a schema

You define the fields you want extracted (a schema) rather than writing custom parsing logic per document type. This is why Content Understanding scenarios often mention "consistent structured

output regardless of document layout" — the schema, not hand-written rules, drives consistency.

> 🎯 **Exam focus:** If a scenario needs one workflow that pulls specific named fields from mixed inputs (say, both scanned PDFs and photos of receipts) into a consistent structured format, that's Content Understanding — plain OCR alone would return raw text, not schema-mapped fields.

Scenario

Insurance claims intake

An insurer receives claim submissions as PDFs, photos of damage, and voicemail descriptions, and wants a consistent set of fields (claimant name, incident date, estimated damage) regardless of submission format.

Analysis: The multimodal input plus the need for consistent, named structured fields is the signature pattern for Content Understanding, versus using three separate single-purpose services with custom glue code.

Chapter 13 checklist

- [ ] I can explain what Content Understanding does differently from single-modality OCR/speech services.
- [ ] I understand the role of a defined schema in producing consistent structured output.
- [ ] I can recognize multimodal extraction scenarios in exam wording.

Cost, Monitoring & Operations

Running an AI solution responsibly means understanding what drives cost and how to detect problems after launch — both are implementation-domain topics.

14.1 What drives generative AI cost

- **Input tokens** — the prompt, system message, and any grounding context you send.
- **Output tokens** — the length of the generated response.

- **Model tier** — larger/more capable models typically cost more per token than smaller ones.
- **Call volume** — total requests per day/month; caching repeated queries can reduce this.

> 💡 **Study tip:** To reduce cost without switching models entirely: trim unnecessary context, cap max output tokens, cache repeated/common queries, and route simple tasks to a smaller model while reserving the largest model for genuinely complex requests.

14.2 Monitoring in production

Signal	Why you track it
Latency	Detects performance regressions or throttling before users complain.
Token usage/cost trend	Detects runaway cost from a bug, abuse, or an unexpectedly popular

	feature.
Content-safety filter triggers	Detects attempted misuse or prompts drifting toward unsafe topics.
Evaluation/quality scores	Detects quality drift as models, prompts, or user behavior change over time.

> 🎯 **Exam focus:** A scenario describing "the AI feature's cost grew unexpectedly overnight" is testing whether you'd check token usage/volume monitoring and caching — not whether you'd immediately downgrade the model, which may not address the actual root cause.

Chapter 14 checklist

- ☐ I can list four drivers of generative AI cost.
- ☐ I can list four production monitoring signals and why each matters.

5-Week Study Planner

This planner assumes roughly 45–60 minutes of daily study. Adjust the pace to fit your schedule, but keep the order — each week builds on groundwork from the one before it.

Week	Days 1–3	Days 4–5	Days 6–7
1	Ch. 1–2: exam map, workload families	Ch. 3: responsible AI	Review + flashcards
2	Ch. 4–5: generative concepts, model selection	Ch. 6: Foundry foundations	Deploy a model hands-on

3	Ch. 7–8: prompting, building a client	Ch. 9: agents	Build a small agent hands-on
4	Ch. 10–11: text, speech	Ch. 12–13: vision, Content Understanding	Review weak areas
5	Ch. 14: cost/ops	Ch. 16: full practice set	Cheat sheet review + rest day before exam

> 💡 **Study tip:** Retake the practice set in Chapter 16 twice: once untimed to check understanding, once timed to build exam-day pacing. Aim to average under 90 seconds per question on the timed pass.

Chapter 15 checklist

- ☐ I have a written date for my exam attempt.

☐ I have blocked calendar time for the two hands-on labs (deploy a model, build an agent).

CHAPTER 16

40 Practice Questions with Explanations

These questions are original and written to mirror the style and scenario-based format of AI-901, grouped by chapter for targeted review. They are not recalled or leaked exam content.

Domain 1: AI Concepts (Q1–Q22)

1. A system autonomously books a multi-leg trip, checking flight and hotel availability across several steps. What kind of AI is this?

Answer: Agentic AI

Multi-step, tool-using, goal-directed behavior is the signature of agentic AI, not single-turn generative AI.

2. A model produces a well-written but factually incorrect summary of a court case it was never given source text for. What is this called?

Answer: Hallucination

Fluent, confident, but ungrounded and incorrect output is a hallucination.

3. Which responsible AI principle is most directly about a named owner and escalation path for AI incidents?

Answer: Accountability

Accountability is about ownership and governance, distinct from safe failure behavior (reliability & safety).

4. A voice interface performs equally well for adult speakers but poorly for children's voices, which were underrepresented in training. Which principle is violated?

Answer: Fairness

Uneven performance across a subgroup is a fairness issue, even if overall average accuracy looks acceptable.

5. What does temperature control in a generative model?

Answer: Output randomness/variety

Not factual accuracy — grounding and evaluation address accuracy, not temperature.

6. A company needs to detect the language of incoming support tickets before routing them. Which capability applies?

Answer: Language detection

A dedicated NLP capability distinct from translation or sentiment analysis.

7. Extracting "total amount due" and "due date" from scanned utility bills of varying formats is best solved by...?

Answer: Content Understanding (schema-based extraction)

Consistent structured fields across varying document layouts is the signature Content Understanding use case.

8. Which vision capability returns bounding boxes for each item found in an image?

Answer: Object detection

Classification returns one label for the whole image; object detection locates multiple items.

9. Converting a recorded lecture into a written transcript is which speech capability?

Answer: Speech-to-text

Audio-to-text direction.

10. A screen-reader style feature reads website text aloud to users. Which capability is this?

Answer: Text-to-speech

Text-to-audio direction.

11. Which principle addresses whether users know they're talking to an AI system?

Answer: Transparency

Distinct from fairness or privacy.

12. A model's context window is exceeded mid-conversation. What is the likely symptom?

Answer: The model appears to "forget" earlier instructions

Earlier tokens fall outside the window once the budget is exceeded.

13. Which is the correct pairing: token = ?

Answer: A word-piece unit the model processes text as

Not a whole word necessarily, and not a security credential.

14. A retailer wants product photos automatically labeled with a single overall category (e.g., "footwear"). Which capability?

Answer: Image classification

One label for the whole image, not per-object boxes.

15. Detecting and masking credit card numbers found in free-text chat logs is an example of...?

Answer: PII detection/redaction

A specific NLP capability distinct from generic entity recognition, though related.

16. Which is NOT a responsible AI principle: Fairness, Inclusiveness, Popularity, Transparency?

Answer: Popularity

The six principles are fairness, reliability & safety, privacy & security, inclusiveness, transparency, accountability.

17. A company wants concise one-sentence summaries of long articles preserving original wording where possible. This best fits...?

Answer: Extractive summarization

Extractive pulls existing sentences; abstractive rewrites in new words.

18. Generating a brand-new marketing illustration from a text description uses which capability?

Answer: Text-to-image generation

A generative image capability, distinct from image classification or captioning.

19. A meeting-notes tool distinguishes each speaker's turns in an audio recording. Which capability is this closest to?

Answer: Speaker recognition

Identifying/distinguishing who is speaking, separate from transcription itself.

20. Which factor is NOT one of the standard model-selection criteria discussed in this book: capability, modality, popularity on social media, cost?

Answer: Popularity on social media

Selection criteria are capability, modality, latency, safety, availability, and cost.

21. A high-stakes medical-summary generator should prioritize which mitigation for hallucination risk?

Answer: Grounding with verified source data

Grounding reduces (not eliminates) hallucination by anchoring output to real content.

22. Which workload family best fits "summarize the top three customer complaints from this week's reviews"?

Answer: Text analysis / NLP (summarization)

Extracting insight from existing text, not generating new creative content.

Domain 2: Implement with Microsoft Foundry (Q23–Q40)

23. What does a Foundry deployment represent?

Answer: A specific model made callable at an endpoint, with its own region/quota/version

Distinct from the model catalog, which is just the browsable list of available models.

24. Which authentication approach is recommended for production Foundry applications?

Answer: Microsoft Entra ID / identity-based authentication

Avoids long-lived secrets embedded in code; supports role-based access control.

25. What are the two primary ways application code calls a deployed model?

Answer: SDK and REST API

SDKs wrap auth/formatting; REST gives direct HTTP control.

26. A request to a deployed model is being throttled. What is the expected handling?

Answer: Retry with backoff

Not immediately treating it as a hard failure or re-sending instantly.

27. Which prompt-engineering technique includes 2–5 sample input/output pairs to steer format consistency?

Answer: Few-shot prompting

Zero-shot has no examples; chain-of-thought asks for step-by-step reasoning.

28. Which technique most directly reduces hallucination by injecting verified content into the prompt?

Answer: Retrieval-augmented grounding

Anchors generation to real source data rather than relying purely on parametric knowledge.

29. What are the four core parts of an AI agent?

Answer: Instructions, tools, planning/reasoning loop, memory/state

These distinguish an agent from a single-turn generative call.

30. Which practice reduces the blast radius if an agent is manipulated into an unintended action?

Answer: Scoped, least-privilege tool access

Limiting what an agent's tools can do, independent of model quality.

31. Which practice requires a human to approve high-impact or irreversible agent actions?

Answer: Human-in-the-loop

Common for financial transactions, deletions, or other hard-to-reverse actions.

32. What is the primary purpose of evaluating an agent before production release?

Answer: Testing against representative scenarios and edge cases, not just happy-path demos

Prevents surprises from untested inputs once deployed.

33. A voice assistant needs sub-one-second responses. Which selection criterion should be weighted most heavily?

Answer: Latency

May require trading off some capability/quality for a faster model.

34. An overnight batch report has no strict time pressure. Which criterion can be prioritized instead?

Answer: Capability/quality

Latency matters less when there's no real-time constraint.

35. What are the two main drivers of per-call generative AI cost?

Answer: Input tokens and output tokens

Both count toward token-based pricing; model tier and volume also matter.

36. Which mitigation reduces cost without necessarily changing the model?

Answer: Caching repeated queries and capping max output tokens

Trims unnecessary spend independent of model choice.

37. Which production monitoring signal would first reveal a sudden, unexplained spend increase?

Answer: Token usage/cost trend monitoring

Directly tracks the driver of unexpected billing changes.

38. A content-safety filter blocks a generated response. What is the correct handling in application code?

Answer: Handle gracefully and inform the user, rather than treating it as a generic crash

Filtered responses are an expected condition to design for, not an application bug.

39. What does a Foundry "connection" represent?

Answer: A configured link to an external resource (data source, search index, other Azure service) usable by a project

Distinct from a deployment, which makes a model callable.

40. Before deploying a new model into production, what workflow step does Foundry emphasize?

Answer: Comparing candidate models with the same representative prompts and acceptance criteria

Structured comparison before committing to a deployment choice.

Chapter 16 checklist

☐ I answered all 40 questions before checking explanations.

☐ I reviewed every explanation for questions I got wrong.

☐ I can re-derive the reasoning, not just recall the letter/answer.

CHAPTER 17

Glossary of Key Terms

Agent

A model paired with instructions, tools, and often memory that can take multi-step action toward a goal.

Agentic AI

AI that acts autonomously across multiple steps, as opposed to answering a single prompt.

Content Understanding

Schema-based extraction of structured fields from multimodal input (documents, images, audio, video).

Context window

The maximum number of tokens (input + output) a model can consider at once.

Deployment

A specific model made callable at an endpoint, with its own region, quota, and version.

Endpoint

The URL an application calls to send prompts to a deployed model and receive a response.

Few-shot prompting

Including example input/output pairs in a prompt to steer format and style.

Foundry project

The top-level Foundry workspace grouping models, deployments, agents, and connections.

Grounding

Supplying verified reference data to a model so its output is anchored to real information.

Hallucination

Fluent, confident model output that is factually incorrect or unsupported.

Human-in-the-loop

A design pattern requiring human approval before a high-impact or irreversible AI action proceeds.

Microsoft Entra ID authentication

Identity-based authentication recommended for production Foundry apps, avoiding long-lived embedded secrets.

Model catalog

The browsable list of foundation models available to deploy within a Foundry project.

Named entity recognition (NER)

Identifying structured entities (people, places, organizations, dates) within text.

Object detection

Locating and labeling multiple distinct objects within an image using bounding boxes.

Prompt engineering

The practice of writing instructions/prompts that reliably steer a model's output.

Responsible AI

The six-principle framework — fairness, reliability & safety, privacy & security, inclusiveness, transparency, accountability.

Temperature

A generation parameter controlling output randomness/variety, not factual accuracy.

Token

A word-piece unit of text that a language model processes; the basis for context window and cost calculations.

CHAPTER 18

Appendix: Quick-Reference Cheat Sheet

Six Responsible AI Principles

- Fairness
- Reliability & safety
- Privacy & security
- Inclusiveness
- Transparency
- Accountability

Prompting Techniques

- Zero-shot
- Few-shot
- Chain-of-thought style
- Retrieval-augmented grounding

Model Selection Criteria

- Capability
- Modality
- Latency
- Safety
- Availability/deployment
- Cost

Agent Safety Practices

- Scoped, least-privilege tools
- Evaluation before release
- Human-in-the-loop for high-risk actions
- Production monitoring

Domain Weight Reminder

Domain	Weight
Identify AI concepts and capabilities	40–45%

Implement AI solutions by using Microsoft Foundry	55–60%

CHAPTER 19

Exam-Day Strategy & FAQ

Exam-day strategy

- Read the full scenario before looking at answer options — many wrong answers are technically true statements that don't fit the specific scenario.
- Flag and skip questions you're unsure of; return after finishing the rest so uncertainty on one item doesn't cost you time on easier ones.
- When two options both sound plausible, look for the more *specific* and scenario-fitting answer rather than the more generally-true one.

- Watch for the words "autonomously," "multi-step," and "decides" — they usually signal agentic AI, not single-turn generative AI.

Is AI-901 the same as AI-900?

No. AI-900 is the older, five-domain, more descriptive exam that is retiring June 30, 2026. AI-901 is the active exam with updated objectives (English version updated April 15, 2026) and a narrower, more implementation-focused two-domain blueprint centered on Microsoft Foundry.

Do I need prior machine learning experience?

No. The official audience profile expects comfort with Python syntax and general Azure/REST/SDK familiarity, not production ML experience.

What is the passing score?

700 on Microsoft's standard 1000-point scale, as published on the official exam page at the time of writing. Always confirm current scoring on learn.microsoft.com before your exam, since Microsoft can update this.

How important are the hands-on labs really?

Very. Because the implementation domain is worth 55-60% of the exam, questions frequently assume familiarity with what deploying a model or building an agent actually looks like in the Foundry portal — reading alone is a weaker preparation strategy than reading plus practicing.

Where can I find more free study material?

PrepKloud publishes a full AI-901 roadmap, flashcards, and practice questions at

prepkloud.com — search "AI-901" from the homepage catalog.

Thank you for studying with PrepKloud. Good luck on your exam — and once you pass, consider sharing your experience to help the next learner. Visit prepkloud.com for more free certification guides.