Amazon Bedrock

Workshop

Shamika Ariyawansa (sariyawa@amazon.com)

Senior AI/ML Specialist Solutions Architect
AWS HCLS

© 2023, Amazon Web Services, Inc. or its affiliates.

Agenda

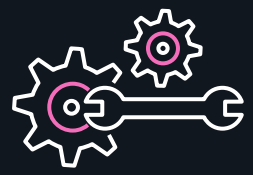
- Introduction Bedrock (Explore UI, notebook based examples)
- GenAI Architecture Patterns
- Text Generation Plotly App
- Plotly Bedrock Chat bot App
- Q & A

© 2023, Amazon Web Services, Inc. or its affiliates.

Amazon Bedrock key benefits

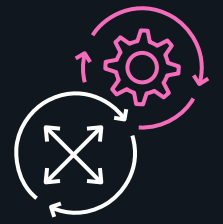
Accelerate development of generative AI applications using FMs through an API, without managing infrastructure

Choose FMs from AI21 Labs, Anthropic, Stability AI, and Amazon to find the right FM for your use case



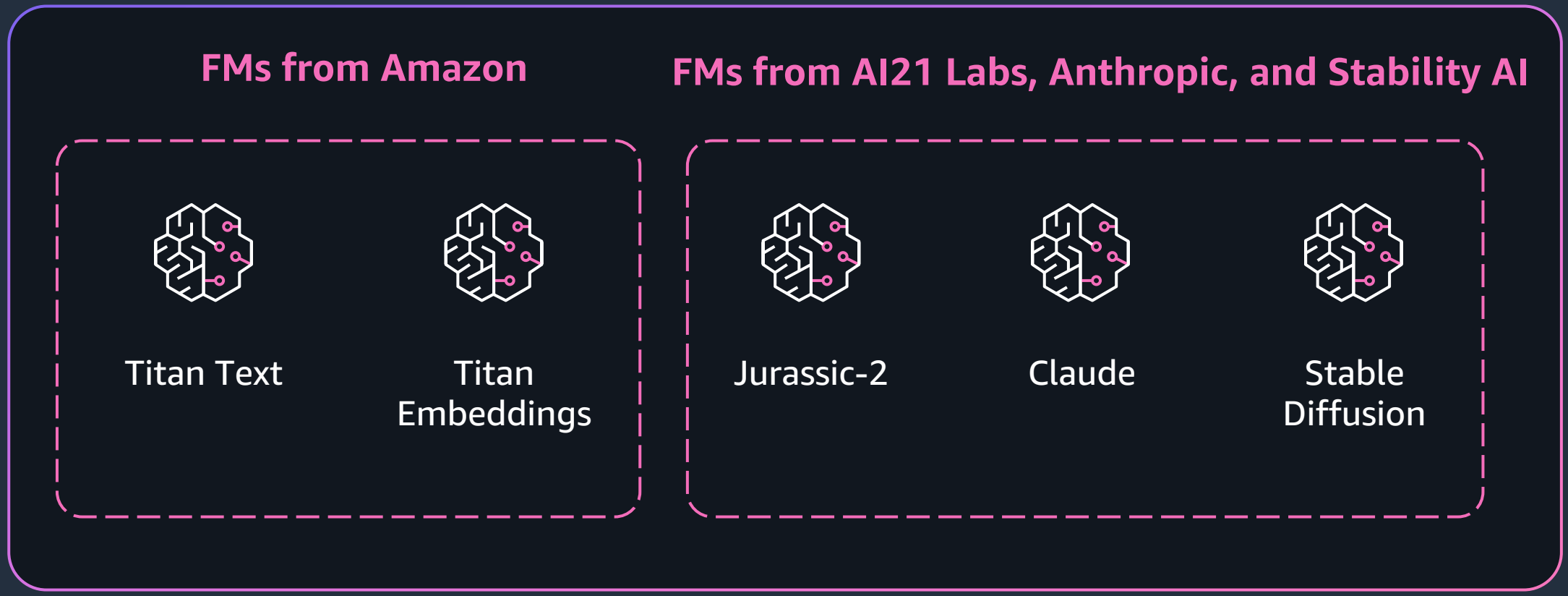
Privately customize FMs using your organization's data

Enhance your data protection using comprehensive AWS security capabilities



Use AWS tools and capabilities that you are familiar with to deploy scalable, reliable, and secure generative AI applications

Bedrock supports a wide range of foundation models



… with more to come

Amazon Titan

INNOVATE RESPONSIBLY WITH HIGH-PERFORMING FMs FROM AMAZON

Titan Text
focused on NLP tasks

Titan Embeddings
for enterprise tasks such as search and personalization

Benefits

- Built with 20+ years of Amazon ML experience
- Automate language tasks such as summarization and text generation with Amazon Titan Text FM
- Enhance search accuracy and improve personalized recommendations with Amazon Titan Embeddings FM
- Support responsible use of AI by reducing inappropriate or harmful content

Foundation models from top AI startups

Jurassic-2

Multilingual LLMs for text generation in Spanish, French, German, Portuguese, Italian, and Dutch

Claude

LLM for conversations, question answering, and workflow automation based on research into training honest and responsible AI systems

Stable Diffusion

Generation of unique, realistic, high-quality images, art, logos, and designs

Current models within Bedrock

Models	Claude	Claude Instant	J2 Grande Instruct	J2 Jumbo Instruct	Stable Diffusion XL	Titan Text Embeddings	Titan Text Large
Model attributes	Text generation, Conversational	Text generation, Conversational	Text, Classification, Insert/edit, Math	Text, Classification, Insert/edit	Text-to-image	Output vector = 4096	Text generation, Code generation, Instruction following
Languages	English	English	English, Spanish, French, German, Portuguese, Italian, Dutch	English, Spanish, French, German, Portuguese, Italian, Dutch	English	English	English
Max Tokens	12000	9000	8191	8191	8192	512	4096

aws

© 2023, Amazon Web Services, Inc. or its affiliates.

Current models within Bedrock

Models	Claude	Claude Instant	J2 Grande Instruct	J2 Jumbo Instruct	Stable Diffusion XL	Titan Text Embeddings	Titan Text Large
Description	Anthropic's most powerful model, which excels at a wide range of tasks from sophisticated dialogue and creative content generation to detailed instruction following.	A faster and cheaper yet still very capable model, which can handle a range of tasks including casual dialogue, text analysis, summarization, and document question-answering.	Jurassic-2 Grande Instruct is the best-in-class large language model (LLM) by AI21 Labs that can be applied to any language comprehension or generation task. It is optimized to follow natural	Jurassic-2 Jumbo Instruct is a top-of-the-line large language model (LLM) by AI21 Labs that can be applied to any language comprehension or generation task. It is optimized to follow natural	SDXL produces more detailed imagery and composition than its predecessor Stable Diffusion 2.1 and represents an important step forward in the lineage of Stability's image generation models.	Fast and cost-effective. Designed for fast responses, this model is an ideal choice for language processing tasks that require maximum affordability and less processing power.	Titan Text is a generative large language model (LLM) for tasks such as summarization, text generation (for example, creating a blog post), classification, open-ended Q&A, and information extraction.
Supported use cases	Question answering, information extraction, removing PII, content generation, multiple choice classification, Roleplay, comparing text, summarization,	Question answering, information extraction, removing PII, content generation, multiple choice classification, Roleplay, comparing text, summarization,	Open book question answering, summarization, draft generation, information extraction, ideation	Open book question answering, summarization, draft generation, information extraction, ideation	image generation, image editing	text retrieval, semantic similarity, clustering	Open ended text generation, brainstorming, summarization, code generation, table creation, data formatting, paraphrasing, chain of though, rewrite, extraction, Q&A, chat

Tuning parameters

© 2022, Amazon Web Services, Inc. or its affiliates.

Temperature

Prompt : Explain the human sleep cycle in two sentences

Temperature : 0.1

The human sleep cycle consists of two main stages: Non-Rapid Eye Movement (NREM) and Rapid Eye Movement (REM) sleep. NREM sleep includes four stages from light sleep to deep sleep, while REM sleep is associated with dreaming and memory consolidation.

Temperature : 1

Our nightly rest dances between two key acts: the deep, restorative slumber of Non-Rapid Eye Movement (NREM) sleep, and the dream-filled flicker of Rapid Eye Movement (REM) sleep. This sleep symphony repeats throughout the night, renewing our bodies and minds in each cycle.

Top K

Prompt : Explain the human sleep cycle in two sentences

topk : 1 (very deterministic)

The human sleep cycle is a sequence of stages that include both Non-Rapid Eye Movement (NREM) and Rapid Eye Movement (REM) sleep. NREM sleep is characterized by four stages, from light sleep to deep sleep, and REM sleep is the stage where dreaming usually occurs.

topk : 100 (Diverse)

The human sleep cycle dances between stages of Non-Rapid Eye Movement (NREM) sleep and Rapid Eye Movement (REM) sleep, creating a rhythm of rest and activity throughout the night. NREM sleep stages carry us from light dozing to deep slumber, while REM sleep bursts forth with rapid eye movements, increased brain activity, and vivid dreams.

Top P

Prompt : Explain the human sleep cycle in two sentences

topp : 0.1 (More focused)

The human sleep cycle is composed of two key phases: Non-Rapid Eye Movement (NREM) sleep and Rapid Eye Movement (REM) sleep. The NREM phase has four stages ranging from light to deep sleep, and the REM phase is where dreaming typically occurs.

topp : 1 (more diverse)

The human sleep cycle is a fascinating rhythm of consciousness, broken down into the slow, deep stages of Non-Rapid Eye Movement (NREM) sleep and the more active, dream-filled stage of Rapid Eye Movement (REM) sleep. As we rest, we dance between these stages, creating a symphony of sleep that helps our bodies recover, our minds process the day's events, and prepare us for a new day ahead.

Other Parameters

- **Response length :** This parameter sets the maximum length of the generated response

- **Stop Sequences :** String that tells the model to stop generating more content. It is another way to control how long your output is. (Eg:- '.')

- Length penalty optimizes the model to be more concise in its output by penalizing longer responses. (0 – No Penalty, < 0 – longer responses, > 0 shorter responses)

- **Repetition penalty (presence penalty)** : Penalizes tokens that have already appeared in the preceding text and scales based on how many times that token has appeared. (1 – No Penalty, >1 decreases repetition)

API operations: `invoke_model()`

PARAMETERS

Parameters	Models			
	Titan Text Large	Jurassic	Claude	Stable Diffusion
Temperature	✅	✅	✅	
TopP	✅	✅	✅	
StopSequences	✅	✅	✅	
MaxTokens	✅	✅	✅	
TopK			✅	
CountPenalty		✅		
PresencePenalty		✅		
FrequencePenalty		✅		
Prompt strength (cfg_scale)				✅
Generation step				✅
Seed				✅

Experiment with models

Playground experience

- You can choose from multiple models and providers
- Easy to use. Just enter text in the Prompt field, then choose Run. In the Response panel, the console displays the response from the model.
- You can view the history of your prompts. You can choose an entry from the history to re-run the prompt.
- You can adjust inference configuration parameters and then re-run your prompt.

© 2023, Amazon Web Services, Inc. or its affiliates.

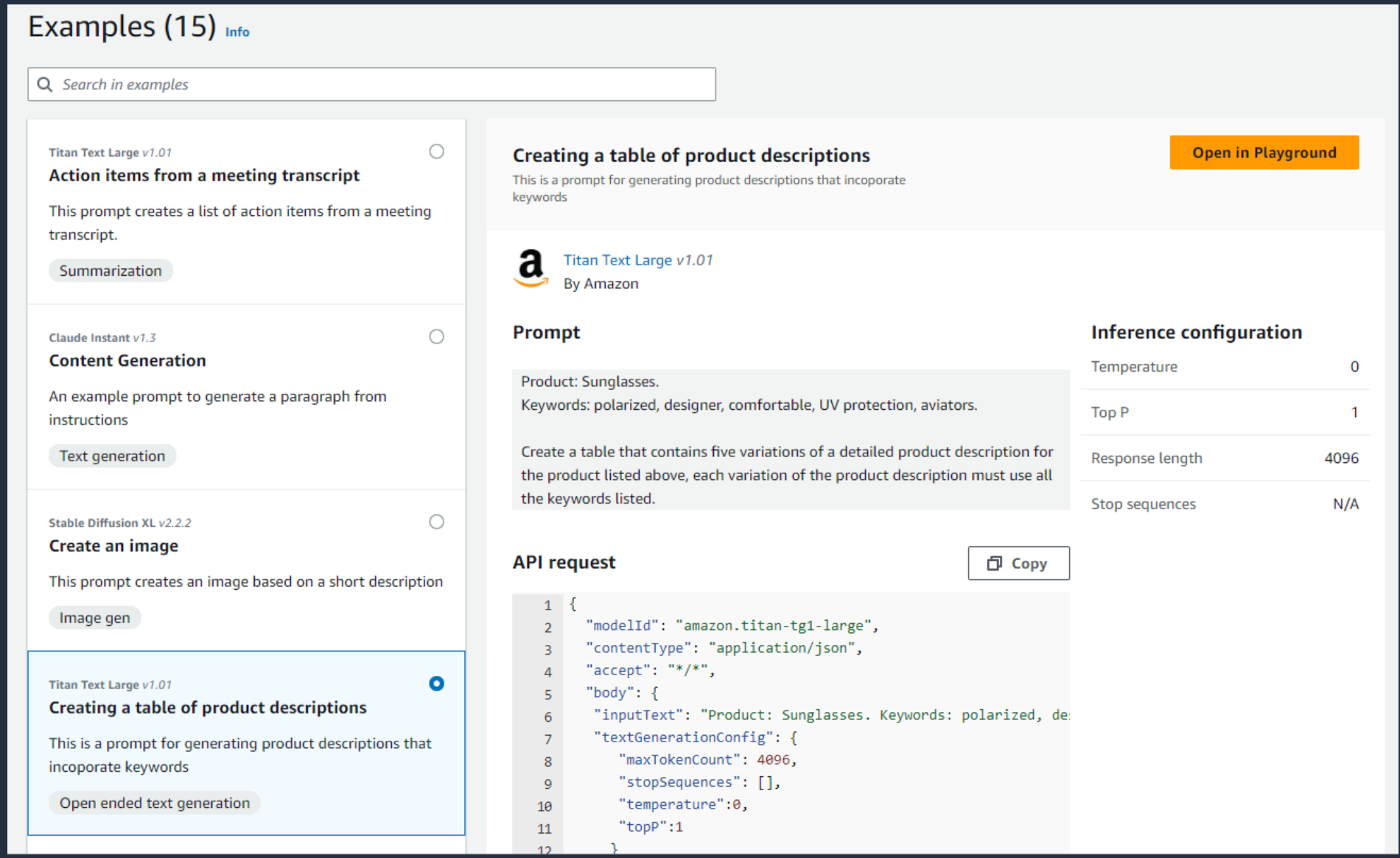
Examples

Try out real-life use cases

- The Amazon Bedrock console displays up to 20 examples for each model provider.
- You can filter the list of examples by one or more of the following attributes: modality, provider name, model name, or example category.
- You can easily run the example by opening it in playground.

© 2023, Amazon Web Services, Inc. or its affiliates.

Experiment with models to try out real-life use cases



Single API to build with generative AI

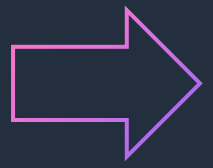
Bedrock core API: InvokeModel

- Pass the model ID, type of content, and body of the request
  - Body includes the prompt and execution parameters
  - Returns model response and metadata
- Handles text-to-text, text-to-image, image-to-image, and more
- Supports current and future Amazon Titan models, third-party models, and even fine-tuned models

© 2023, Amazon Web Services, Inc. or its affiliates.

Bedrock core API: InvokeModel

```
bedrock.invoke_model(
    modelId = model_id,
    contentType = "...",
    accept = "...",
    body = body)
```



Access foundation models

- Amazon Titan models
- Third-party models
- Fine-tuned models

© 2023, Amazon Web Services, Inc. or its affiliates.

Access Bedrock via Boto3: API operations

- `list_foundation_models()`

Use the ListFoundationModels operation to retrieve information about the foundation models.

- `invoke_model()`

Use this call to invoke the desired model. The API parameters and result depend on which model you are invoking.

*You can access the Amazon Bedrock API using the AWS CLI and the AWS SDK for Python (Boto3)

© 2023, Amazon Web Services, Inc. or its affiliates.

API operations: `invoke_model()`

PARAMETERS

Parameters	Models			
	Titan Text Large	Jurassic	Claude	Stable Diffusion
Temperature	✅	✅	✅	
TopP	✅	✅	✅	
StopSequences	✅	✅	✅	
MaxTokens	✅	✅	✅	
TopK			✅	
CountPenalty		✅		
PresencePenalty		✅		
FrequencePenalty		✅		
Prompt strength (cfg_scale)				✅
Generation step				✅
Seed				✅

Integrated with LangChain

```
pip install langchain
```

```
from langchain import Bedrock
from langchain.embeddings
import BedrockEmbeddings

llm = Bedrock()
print(llm("what is generative
AI?"))
```

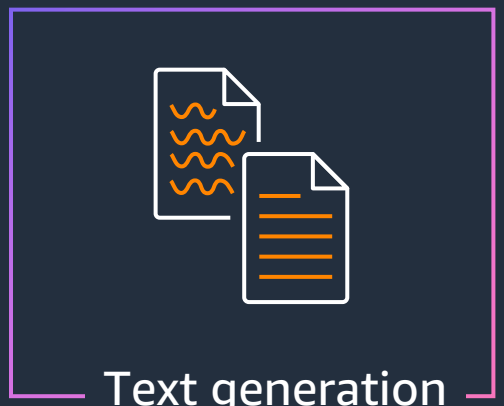
Popular Python framework for developing applications powered by language models

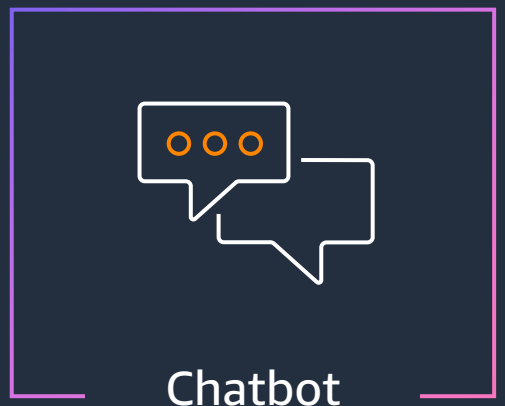
- New LLM and embeddings class for Amazon Bedrock
- Includes code for using the LLM class in a conversation chain
- Includes code for creating an embedding from text

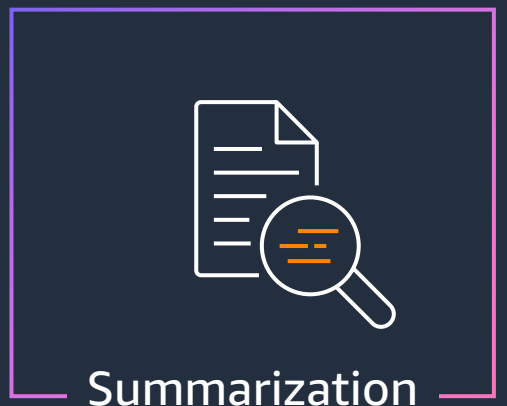
Architecture patterns

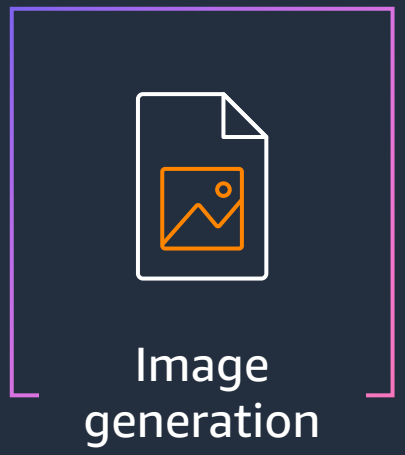
© 2022, Amazon Web Services, Inc. or its affiliates.

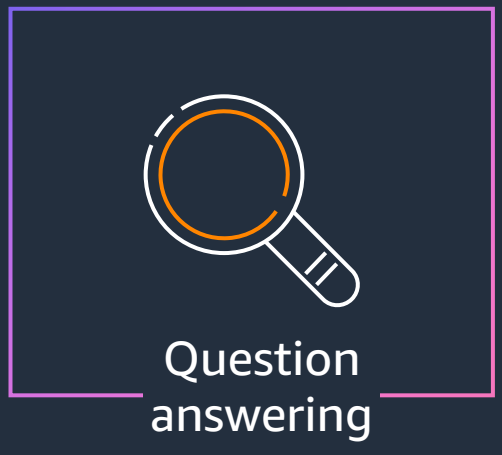
Architecture patterns in:











Text Generation

WITH SIMPLE PROMPT

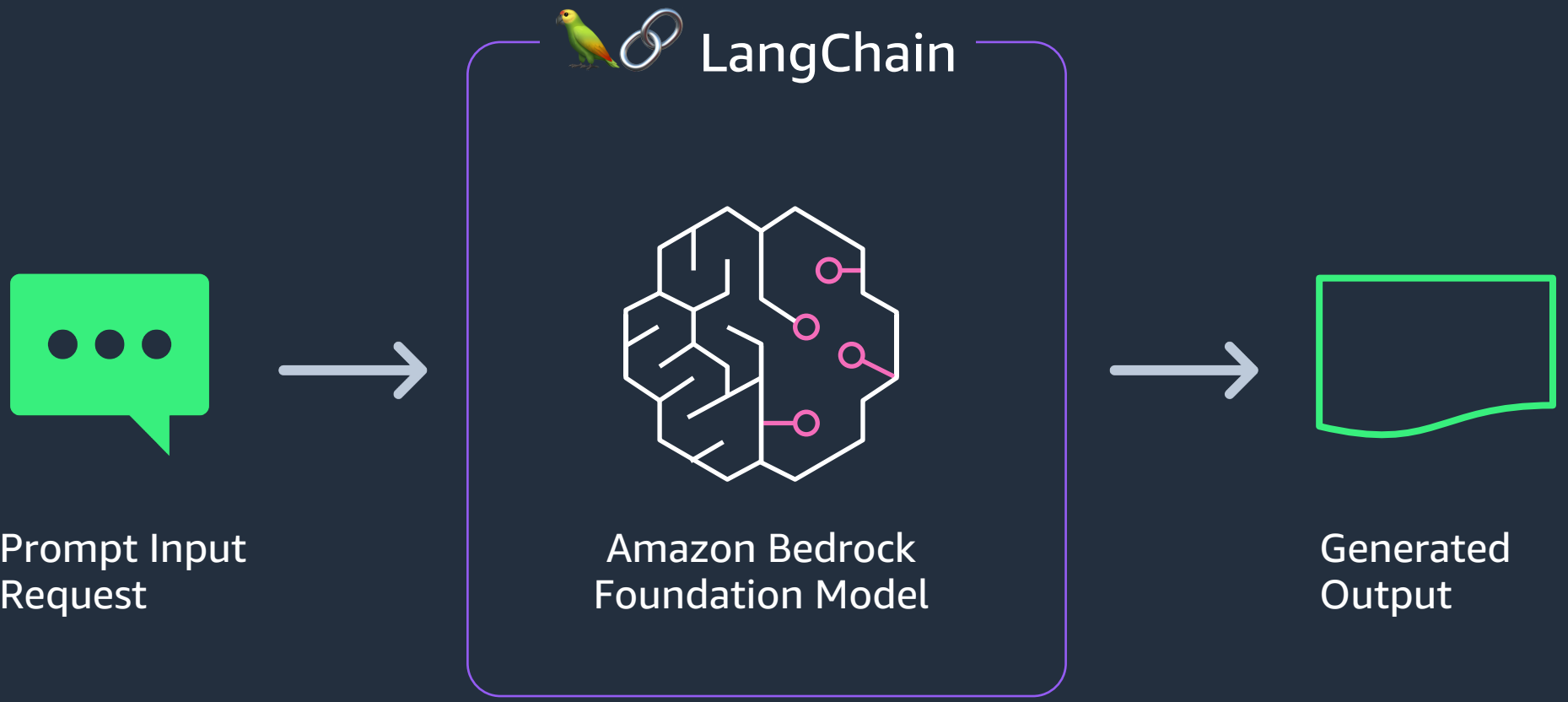
Prompt Input
Request

Amazon Bedrock
Foundation Model

Generated
Output

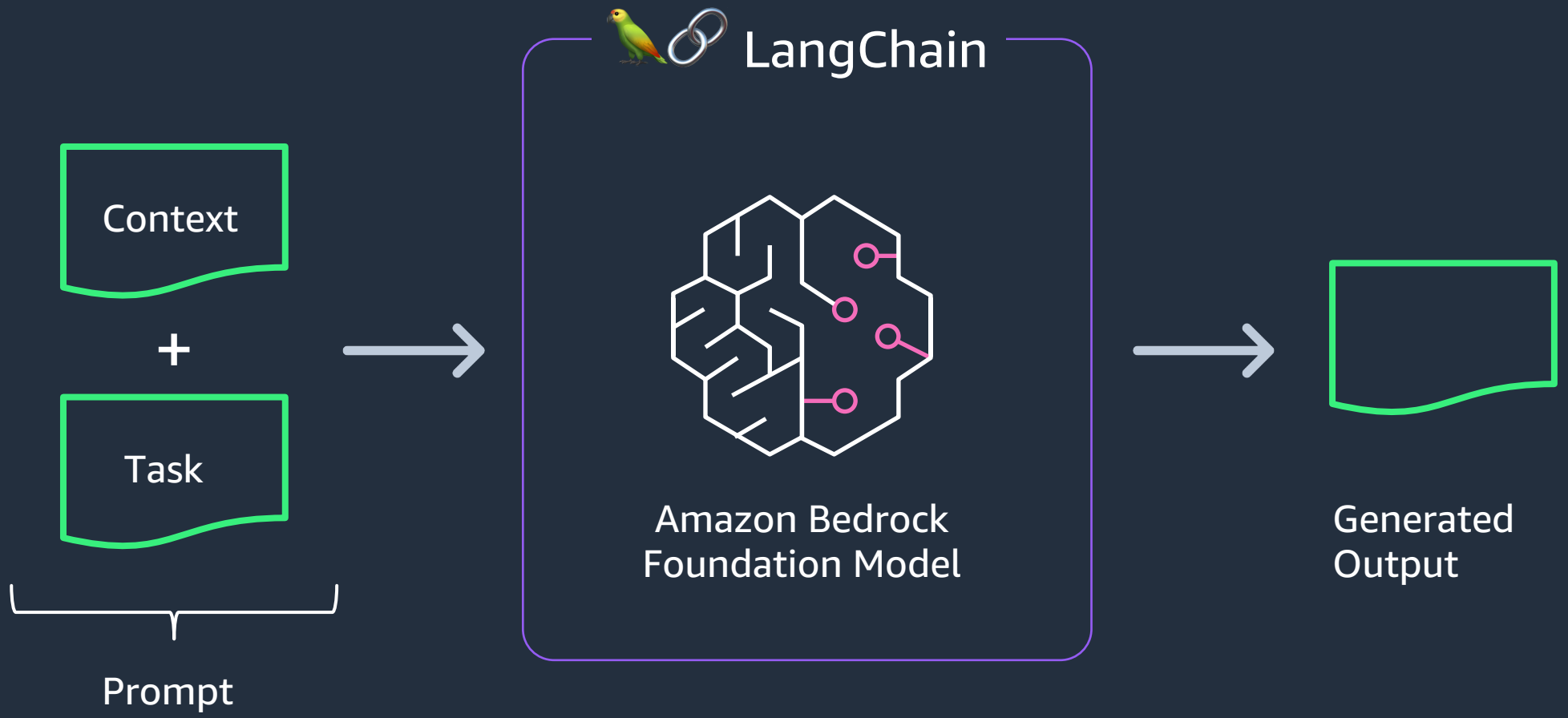
Text Generation

WITH LANGCHAIN



Text Generation

WITH CONTEXT AND LANGCHAIN



Text Summarization

WITH SMALL FILES

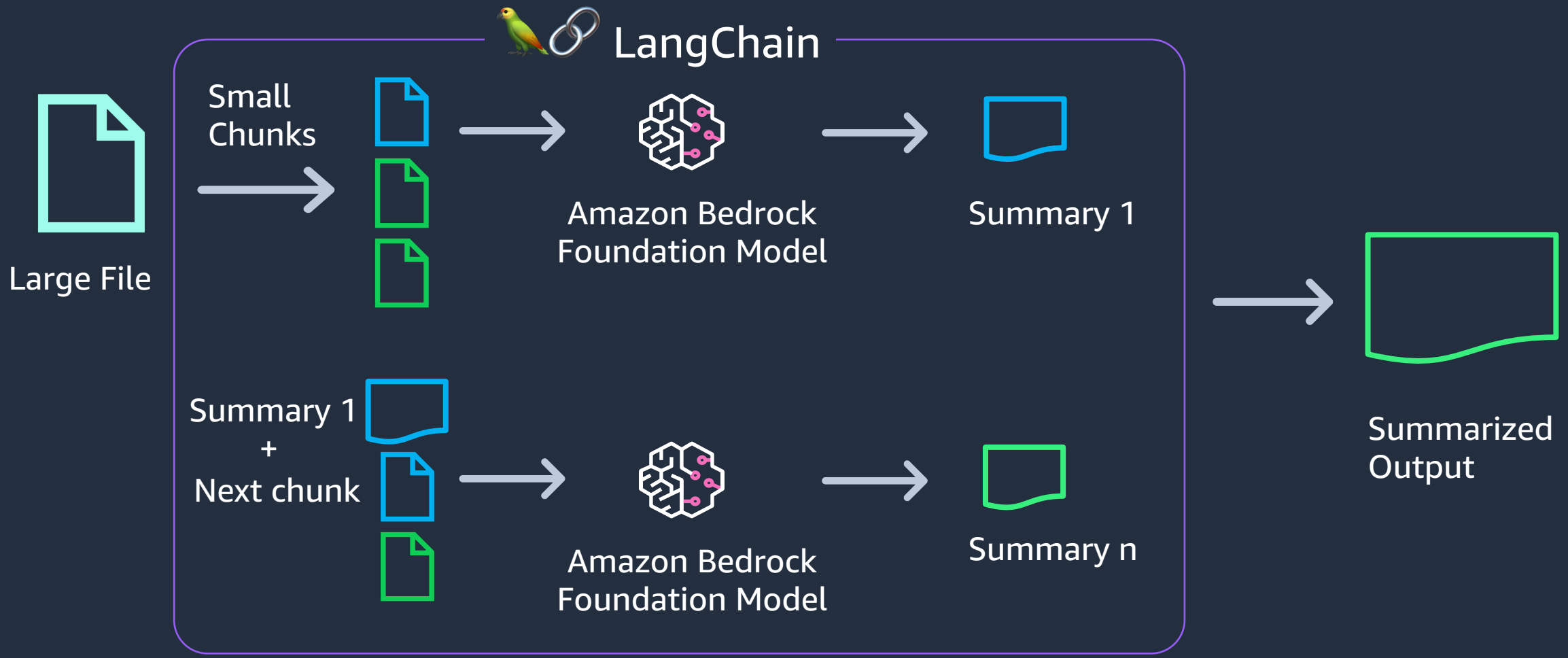
Small file

Amazon Bedrock
Foundation Model

Summarized
Output

Text Summarization

WITH LARGE FILES AND LANGCHAIN



Question Answering

WITH SIMPLE PROMPT

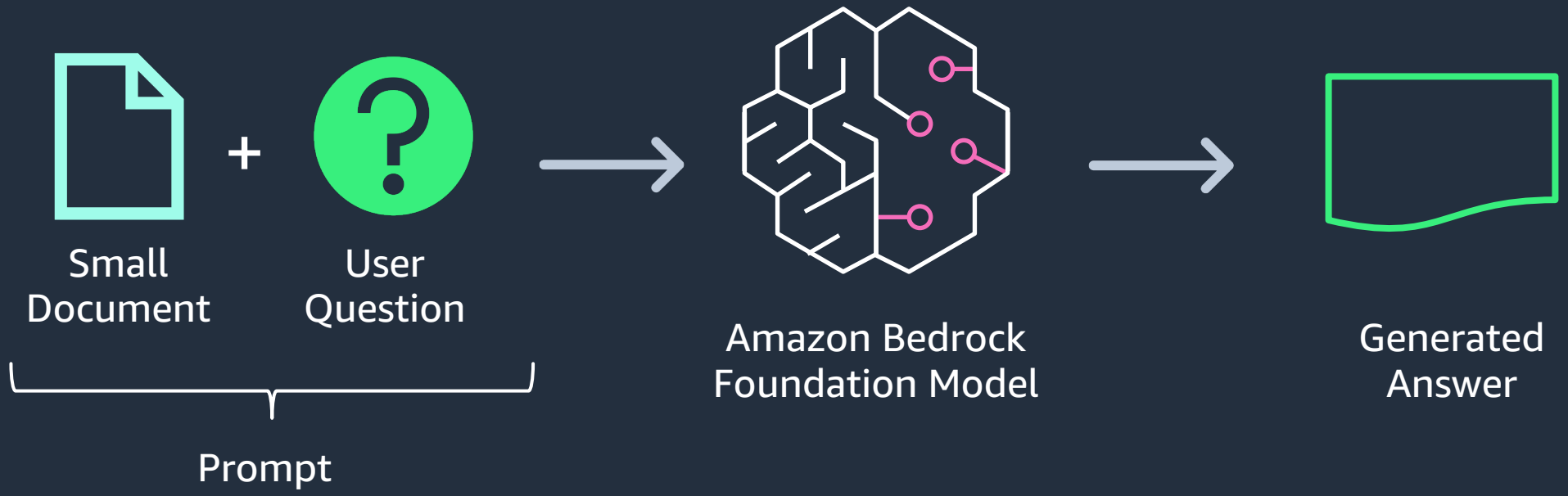
User Question

Amazon Bedrock Foundation Model

Generated Answer

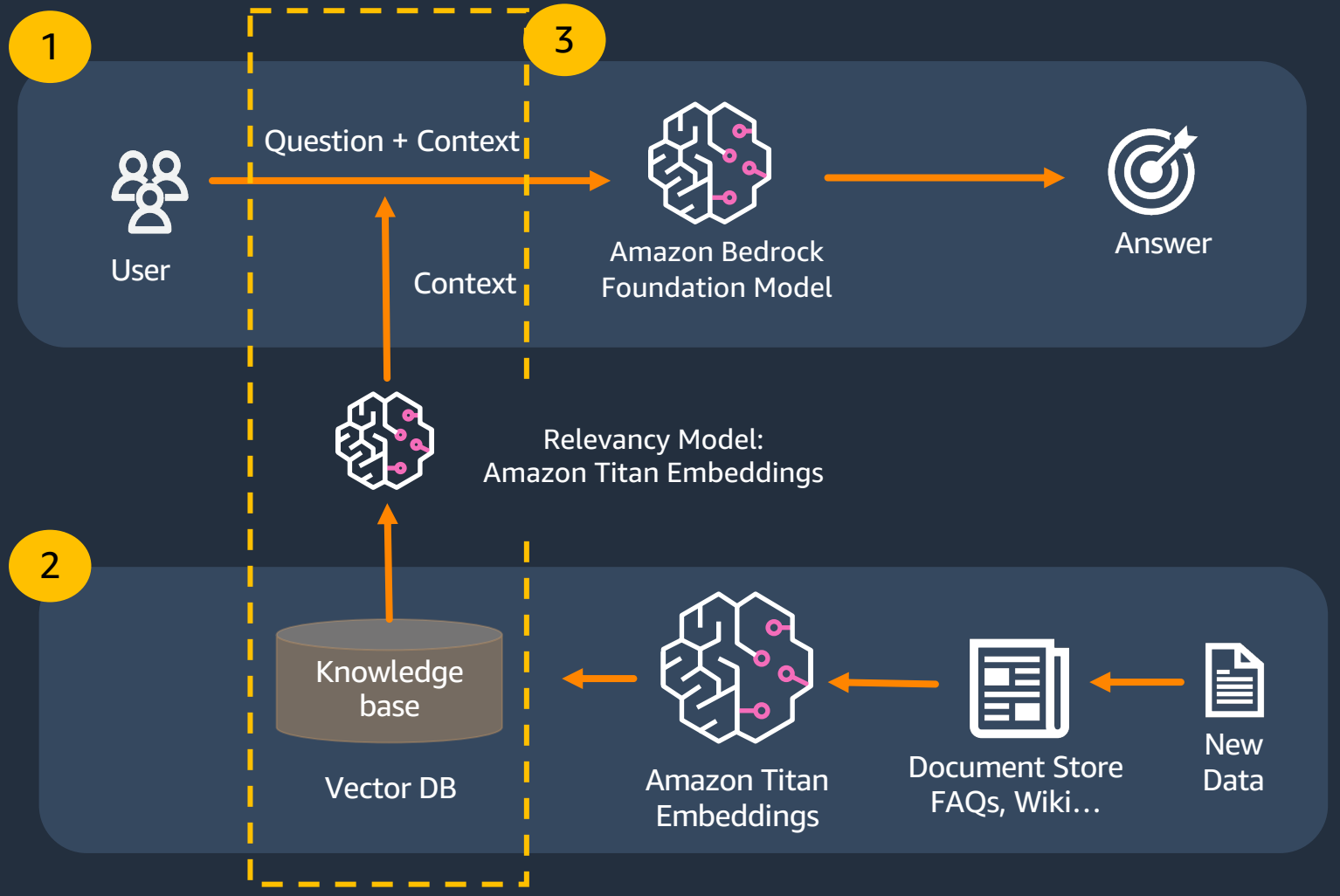
Question Answering

WITH CONTEXT



Question Answering

WITH RETRIEVAL-AUGMENTED GENERATION

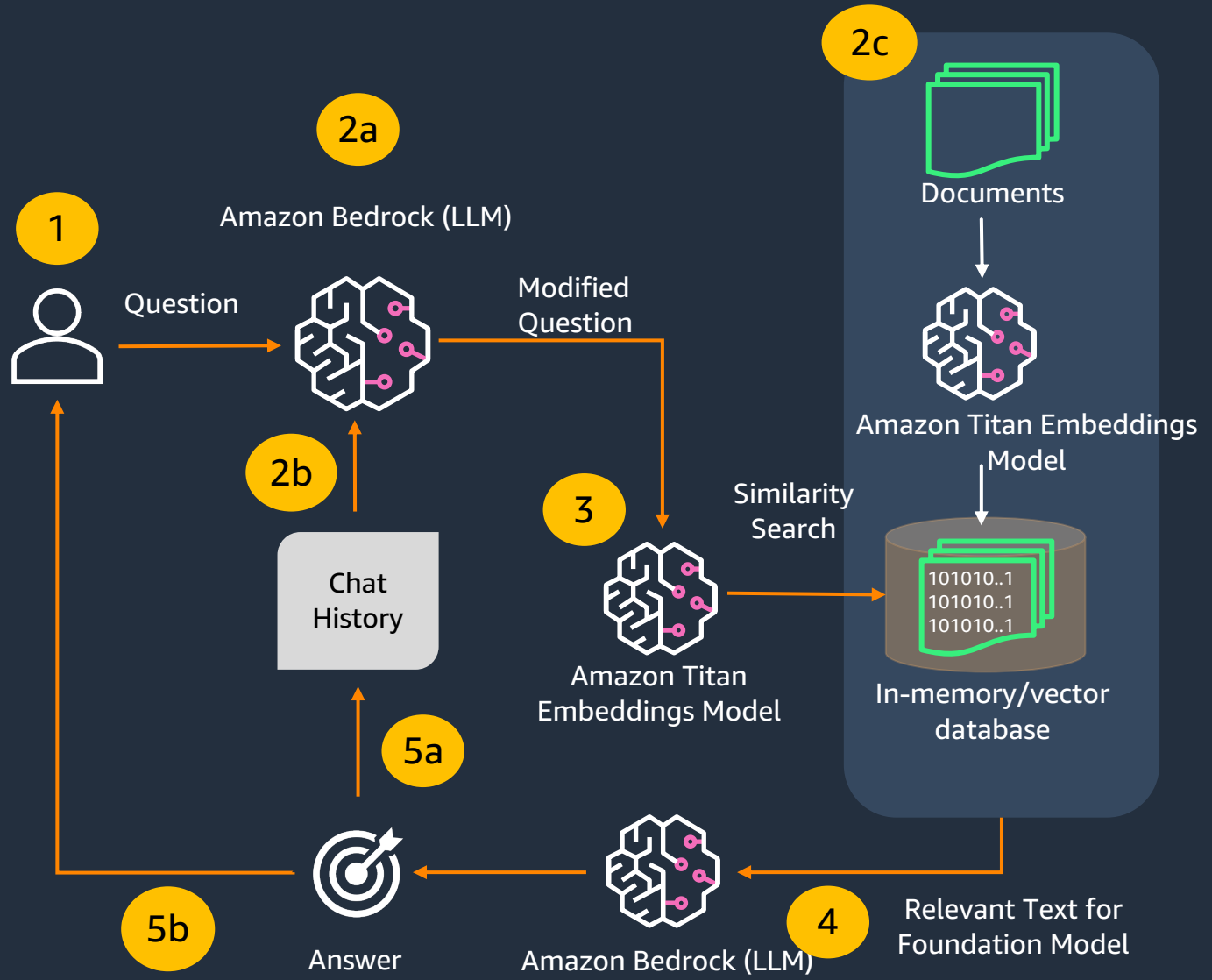


Chatbot

BASIC

Chatbot

WITH CONTEXT



Text to Image

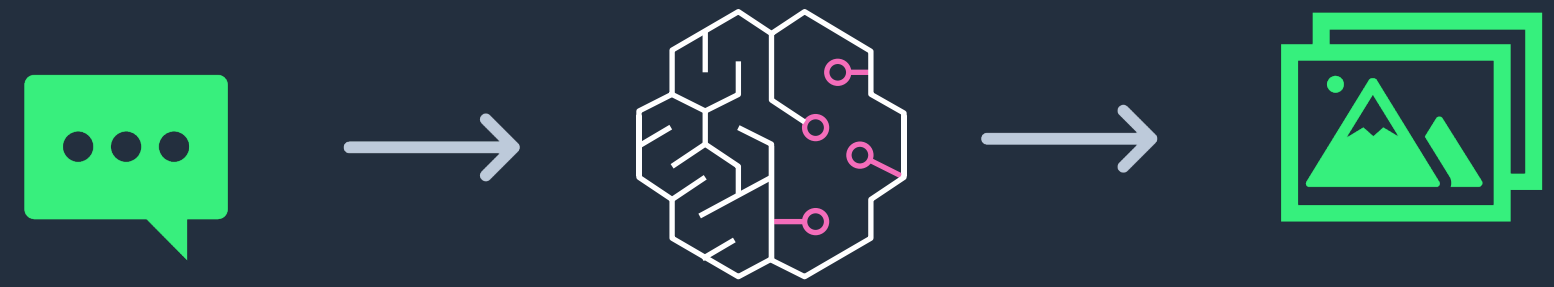
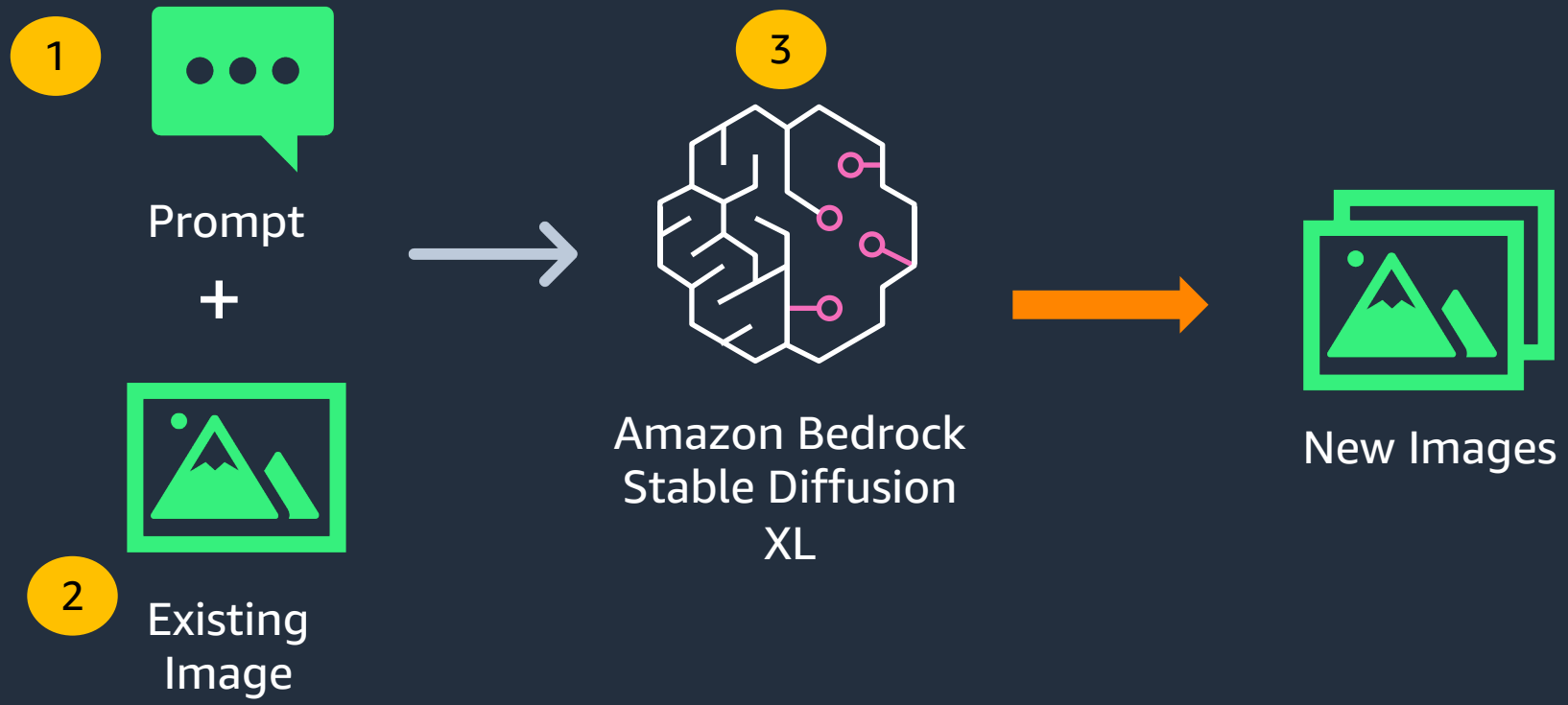


Image to Image (In-painting)



Security

© 2022, Amazon Web Services, Inc. or its affiliates.

Data privacy and localization

You are always in control of your data

- Customer data is not used to improve the Amazon Titan models for other customers and is not shared with other foundation model providers
- Customer data (prompts, responses, fine-tuned models) are isolated per customer and remain in the Region where they were created

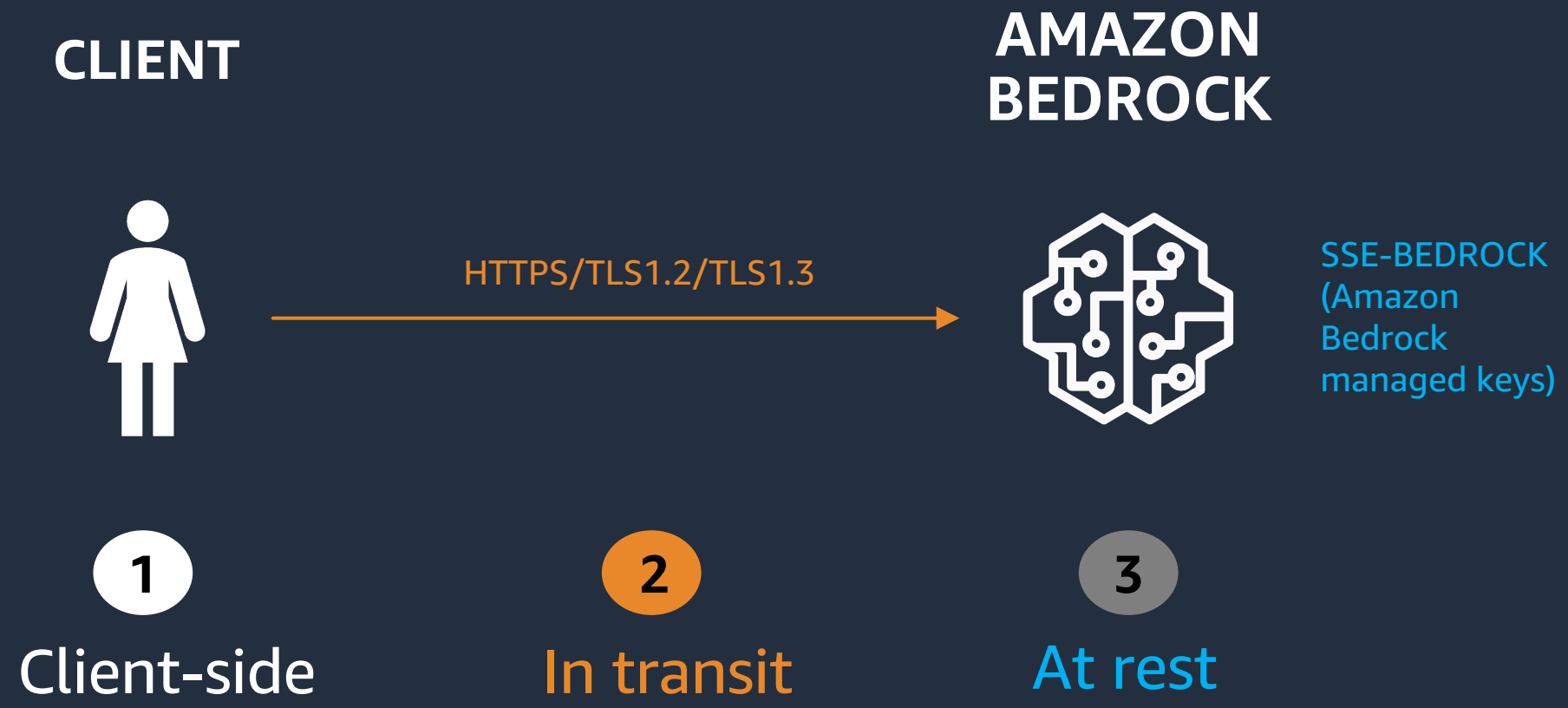
Data security

You are always in control of your data

- Customer data is always encrypted in transit with a minimum of TLS1.2 and AES-256 encrypted at rest using AWS KMS managed data encryption keys
- Integration with AWS Identity and Access Management Service (IAM) to manage inference access, allow/deny access for specific models, enable AWS Management Console access, etc.
- Use AWS CloudTrail to monitor all API activity and troubleshoot issues as you integrate with applications
- Fine-tuned (customized) models are encrypted and stored using customer AWS KMS key. Only you have access to your customized models

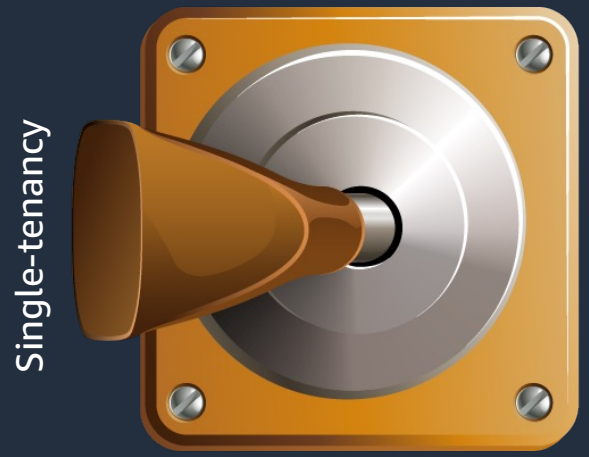
Data protection

ENCRYPTION



Configurable security controls

Data privacy



Model tenancy

AWS KMS data encryption

Model fine tuning

Access management

Model tenancy

Single-tenant endpoint

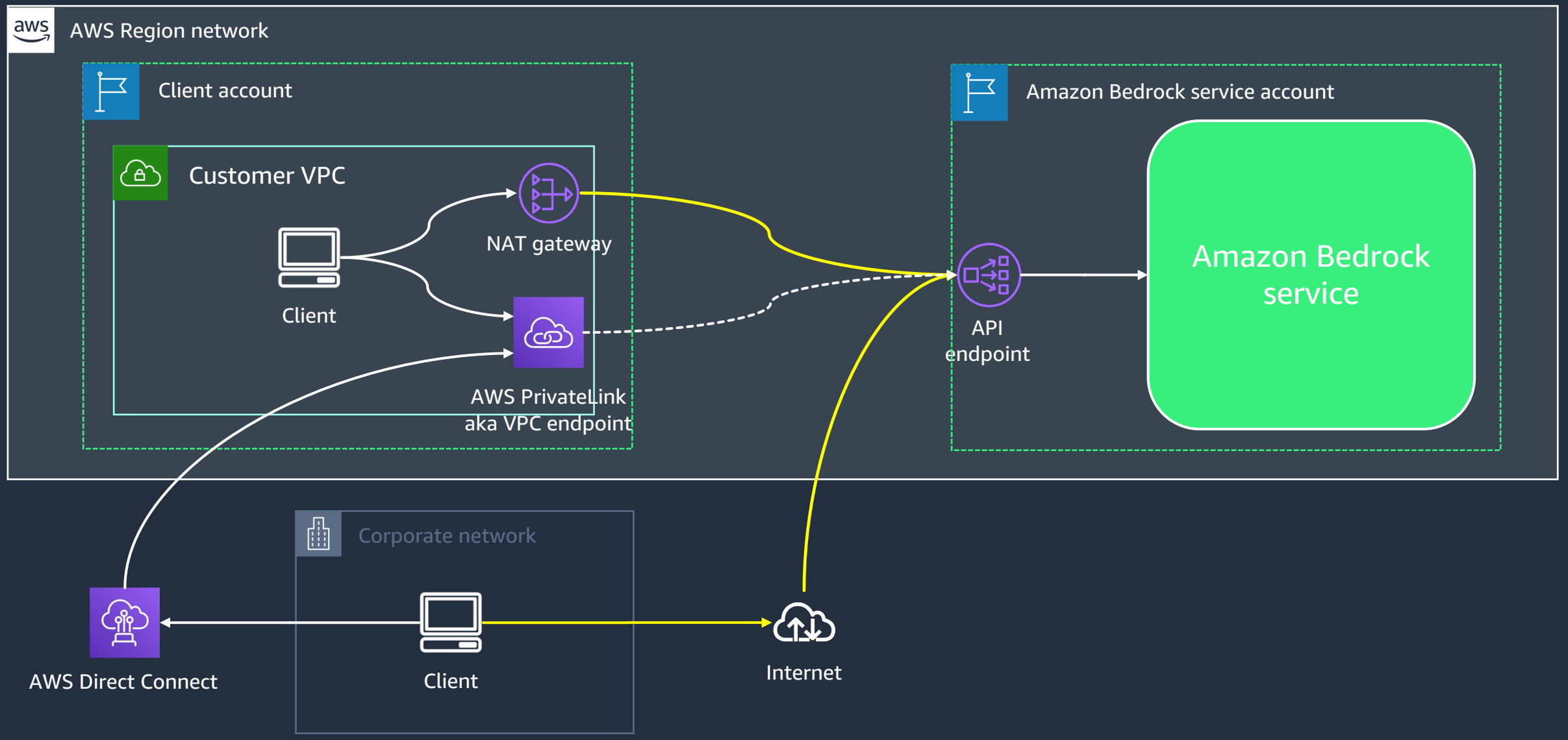
1. Deployment available to a single customer
2. Holds a single version of a baseline 1P/3P model that has been fine-tuned by a customer

Multi-tenant endpoint

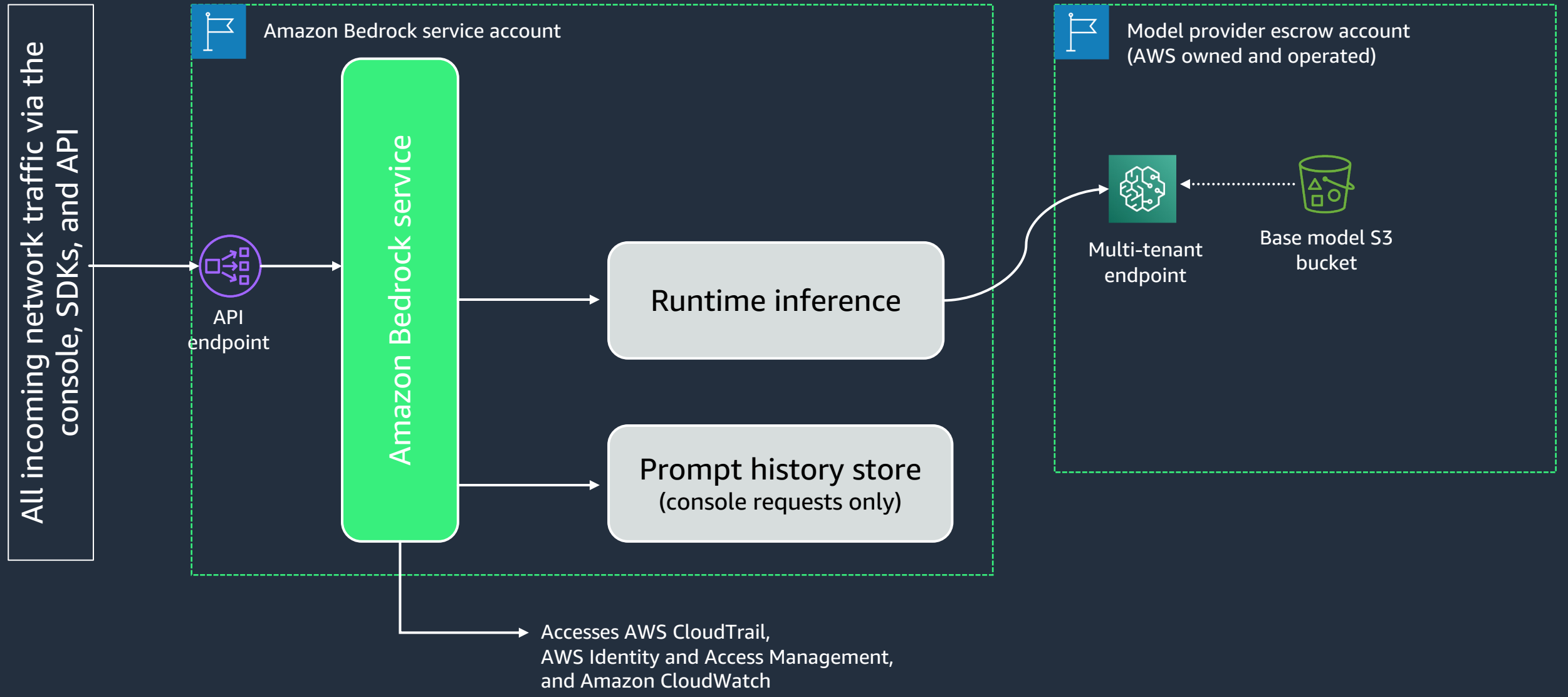
1. Deployment available to all customers
2. Holds a baseline version of each supported 1P/3P model

3. No inference request's input or output text is used to train any model(s) in the deployment
4. Model deployments are inside an AWS account owned and operated by the Bedrock service team
5. Model vendors have no access to any customer data

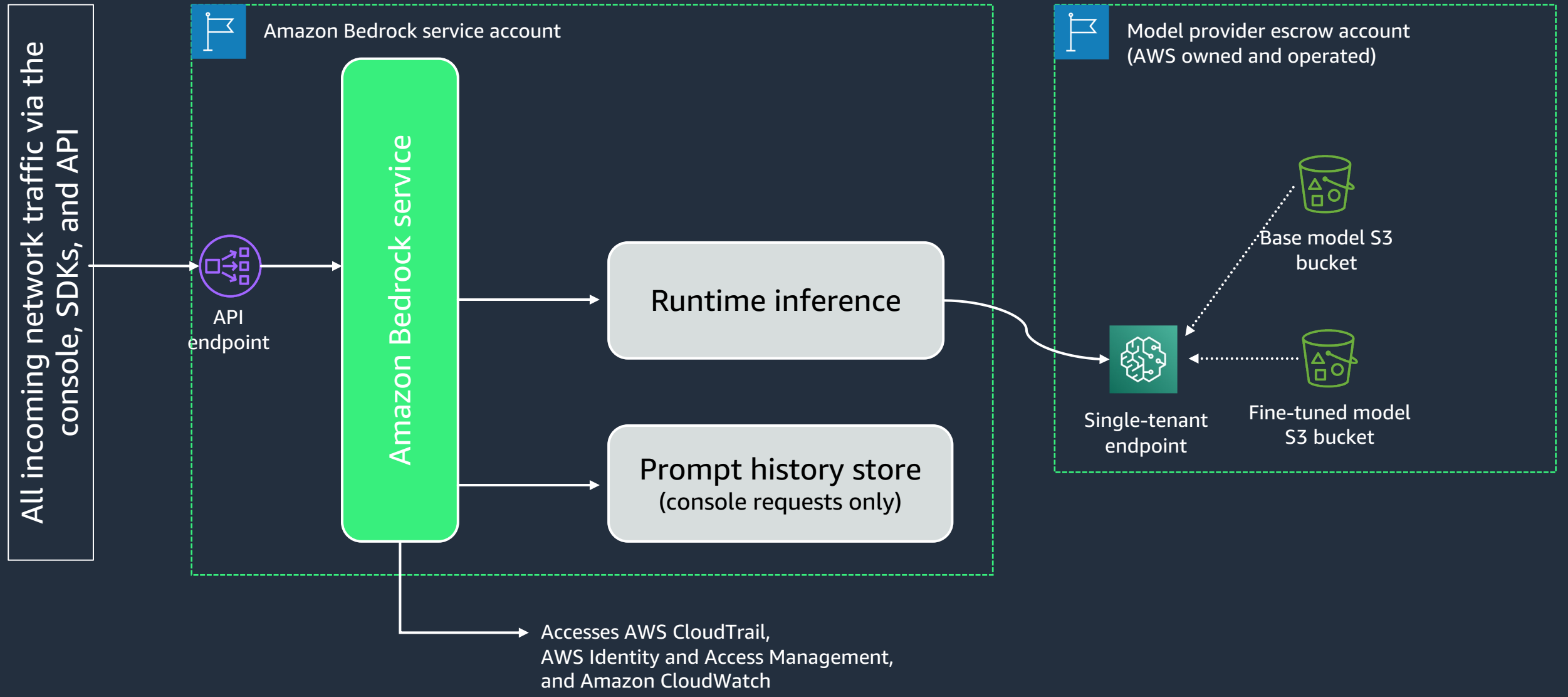
Client connectivity



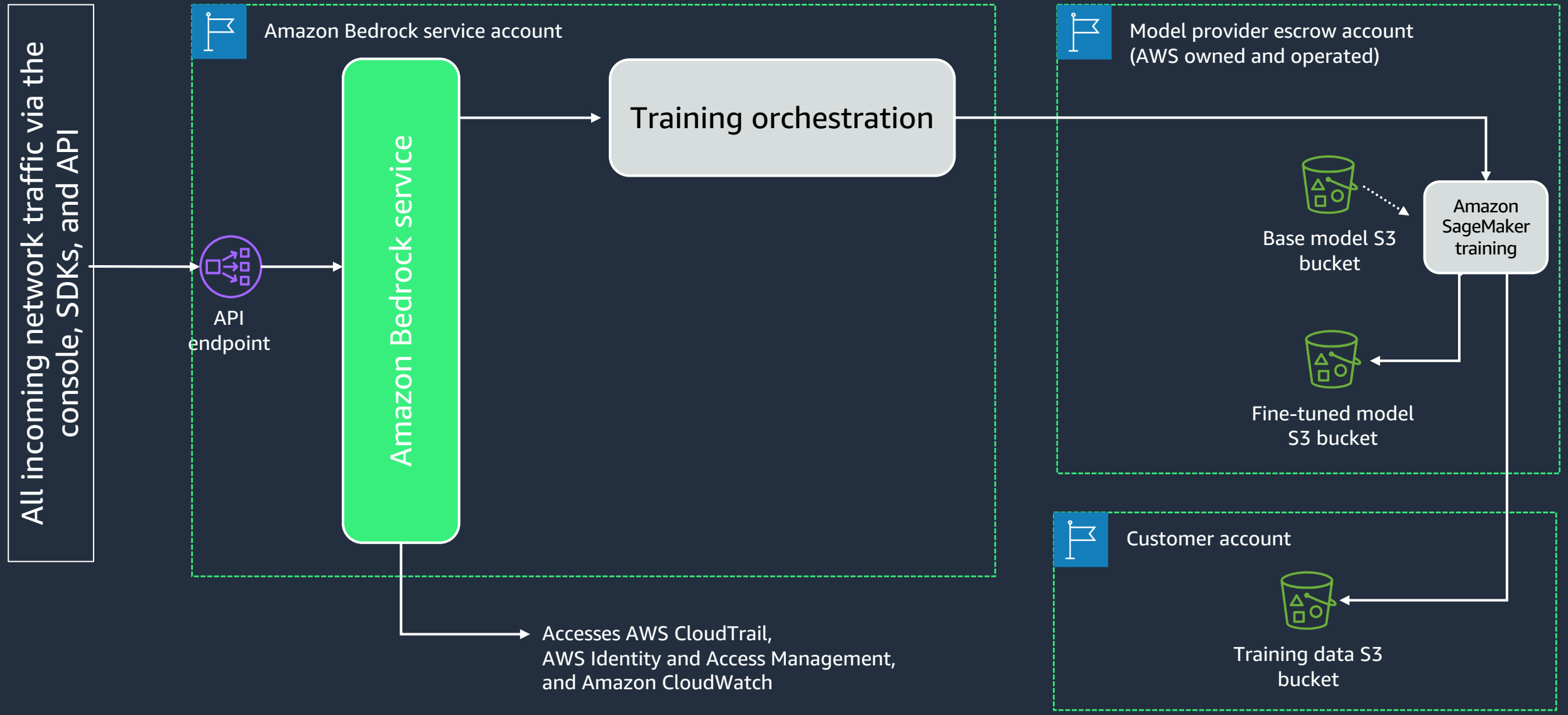
Multi-tenancy inference



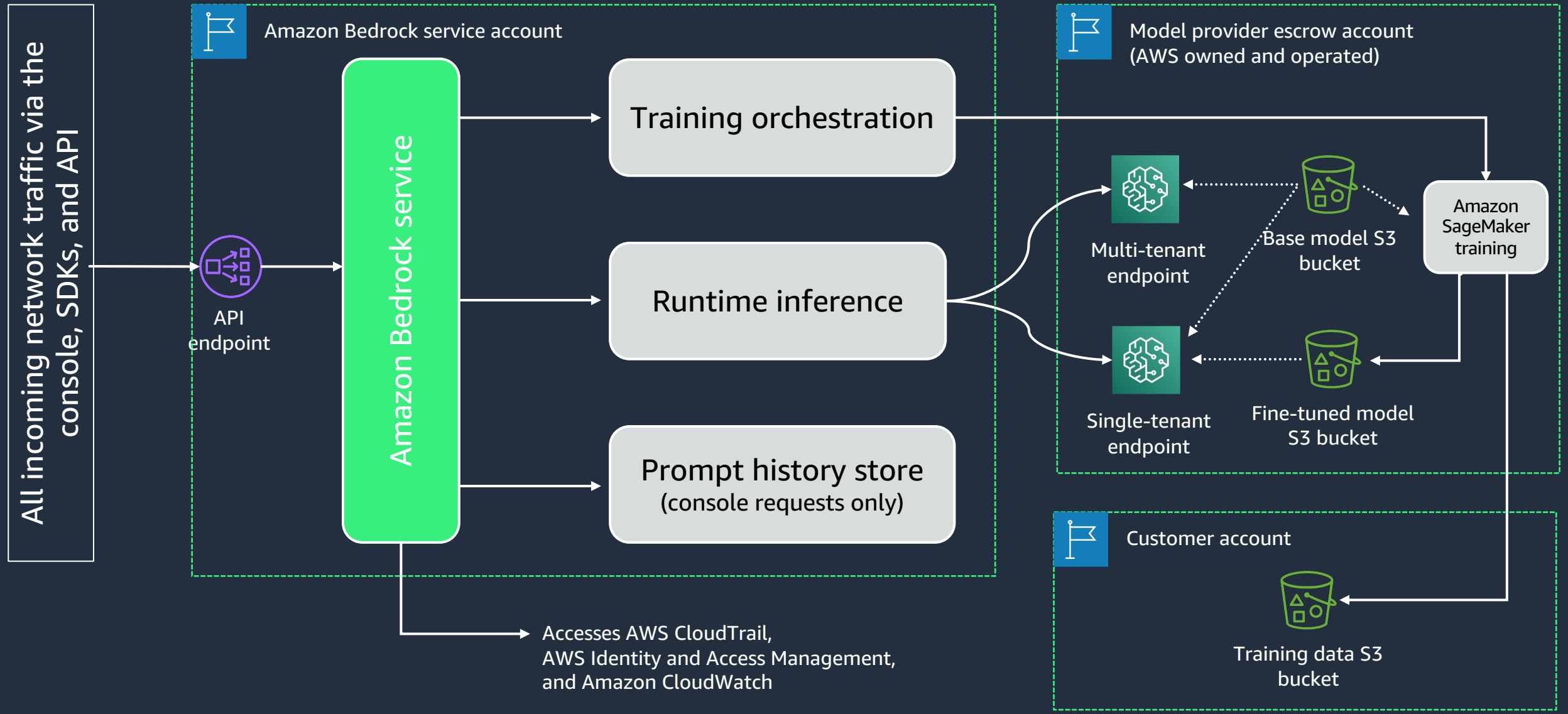
Single-tenancy inference



Model fine tuning



Complete architecture overview



AWS Identity and Access Management

IAM

- Identity-based policies
- Actions
- Resources
- Tags (ABAC)

AWS IAM Fine Grained Access Controls

AMAZON BEDROCK IDENTITY BASED POLICIES

IAM Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BedrockConsole",
            "Effect": "Allow",
            "Action": [
                "bedrock:ListFoundationModels",
                "bedrock:InvokeModel”
            ],
            "Resource": "*"
        }
]
```

Action(s)

Resource(s)

Condition Key(s)

IAM/SCP – Example deny policy

```
{
    "Version": "2012-10-17",
    "Statement":
    {
        "Sid": "DenyInferenceForModelX",
        "Effect": "Deny",
        "Action": "bedrock:InvokeModel",
        "Resource": "arn:aws:bedrock:::foundation-model/<name-of-model>"
    }
}
```

© 2023, Amazon Web Services, Inc. or its affiliates.

AWS IAM Fine Grained Access Controls

IAM SUPPORTED FEATURES WITH AMAZON BEDROCK

Service	Identity Based Policy	RBAC	Policy Actions	Policy Resources	ACL	Temporary Credentials	Service Linked Roles	Service Roles
Amazon Bedrock	Yes	No	Yes	Yes	Yes	No	No	No

Security in the model: challenges and opportunities

Challenges

- ✓ Use of generative AI and FMs for illegitimate or malicious purposes
- ✓ Prompt manipulation to avoid model protections and filters
- ✓ Risks of erroneous or otherwise undesirable outputs

Opportunities

- ✓ Enhanced security tools and UXs based on FMs
- ✓ Domain-focused FMs and model tuning reduce risks
- ✓ Supporting human judgment rather than closed-loop automation

Defining the right level of customization

Prompt engineering	Guiding model to generate useful response by teaching it the “pattern” of desired output using context instructions, examples and output indicators
Retrieval Augmented Generation (RAG)	Text generation based on specified corpus of data, to generate accurate responses with no hallucination
Instruction fine-tuning	Finetuning a language model on a collection of tasks described via instructions—improves the zero-shot performance of language models on unseen tasks.
Finetuning	Finetuning a model using proprietary or domain specific data to improve output quality and domain-relevant results
Retraining the model	Retraining a model using a different dataset, or building a model from scratch

Thank you!

© 2022, Amazon Web Services, Inc. or its affiliates.